Universität Stuttgart

Coupling Schemes and Inexact Newton for Multi-Physics and Coupled Optimization Problems

Von der Fakultät für Informatik, Elektrotechnik und Informationstechnik der Universität Stuttgart sowie dem SRC SimTech zur Erlangung der Würde eines Doktors der Naturwissenschaften (Dr. rer. nat.) genehmigte Abhandlung

Vorgelegt von
Klaudius Scheufele
aus Weilheim an der Teck

Hauptberichter: Prof. Dr. Miriam Mehl

Mitberichter: Prof. Dr. George Biros
Mitprüfer: Prof. Dr. Andrés Bruhn
Prof. Dr. Oliver Röhrle

Tag der mündlichen Prüfung: 16. Januar 2019

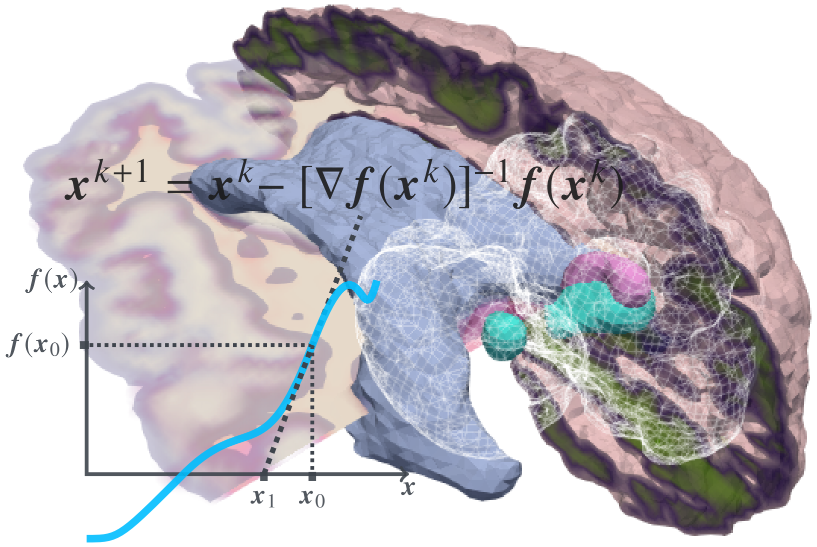
Institut für Parallele und Verteilte Systeme der Universität Stuttgart

2018

Klaudius Scheufele

MULTI-COMPONENT SYSTEMS INVERSE PROBLEMS QUASI-NEWTON

Coupling Schemes and Inexact Newton for Multi-Physics and Coupled Optimization Problems



University of Stuttgart
Germany

SimTech
IPVS
SGS

> *Complexity has nothing to do with intelligence, simplicity does.*
>
> — Larry Bossidy

COVER FIGURE *Volume-rendering of a segmented brain tumor magnetic resonance image.*

Abstract

This work targets mathematical solutions and software for complex numerical simulation and optimization problems. Characteristics are the combination of different models and software modules and the need for massively parallel execution on supercomputers. We consider two different types of multi-component problems in Part I and Part II of the thesis: ***(i) Surface coupled fluid-structure interactions*** and ***(ii) analysis of medical MR imaging data of brain tumor patients***. In (i), we establish highly accurate simulations by combining different aspects such as fluid flow and arterial wall deformation in hemodynamics simulations or fluid flow, heat transfer and mechanical stresses in cooling systems. For (ii), we focus on (a) facilitating the transfer of information such as functional brain regions from a statistical healthy atlas brain to the individual patient brain (which is topologically different due to the tumor), and (b) to allow for patient specific tumor progression simulations based on the estimation of biophysical parameters via inverse tumor growth simulation (given a single snapshot in time, only). Applications and specific characteristics of both problems are very distinct, yet both are hallmarked by strong inter-component relations and result in formidable, very large, coupled systems of partial differential equations.

Part I targets robust and efficient quasi-Newton methods for black-box surface-coupling of partitioned fluid-structure interaction simulations. The partitioned approach allows for great flexibility and exchangeable of sub-components. However, breaking up multi-physics into single components requires advanced coupling strategies to ensure correct inter-component relations and effectively tackle instabilities. Due to the black-box paradigm, solver internals are hidden and information exchange is reduced to input/output relations. We develop advanced quasi-Newton methods that effectively establish the equation coupling of two (or more) solvers based on solving a non-linear fixed-point equation at the interface. Established state of the art methods fall short by either requiring costly tuning of problem dependent parameters, or becoming infeasible for large scale problems. In developing parameter-free, linear-complexity alternatives, we lift the robustness and parallel scalability of quasi-Newton methods for partitioned surface-coupled multi-physics simulations to a new level. The developed methods are implemented in the parallel, general purpose coupling tool preCICE.

Part II targets MR image analysis of glioblastoma multiforme pathologies and patient specific simulation of brain tumor progression. We apply a joint medical image registration and biophysical inversion strategy, targeting at facilitating diagnosis, aiding and supporting surgical planning, and improving the efficacy of brain tumor therapy. We propose two problem formulations and decompose the resulting large-scale, highly non-linear and non-convex PDE-constrained optimization problem into two tightly coupled problems: inverse tumor simulation and medical image registration. We deduce a novel, modular Picard iteration-type solution strategy. We are the first to successfully solve the inverse tumor-growth problem based on a single patient snapshot with a gradient-based approach. We present the joint inversion framework SIBIA, which scales to very high image resolutions and parallel execution on tens of thousands of cores. We apply our methodology to synthetic and actual clinical data sets and achieve excellent normal-to-abnormal registration quality and present a proof of concept for a very promising strategy to obtain clinically relevant biophysical information.

Advanced inexact-Newton methods are an essential tool for both parts. We connect the two parts by pointing out commonalities and differences of variants used in the two communities in unified notation.

Kurzzusammenfassung

Diese Arbeit befasst sich mit der Entwicklung mathematischer Lösungsstrategien und Computersoftware für komplexe numerische Simulationen und Optimierungsprobleme. Charakteristiken hierbei sind die Kombination verschiedener Modelle und Softwaremodule sowie die Notwendigkeit für massiv parallele Berechnung auf Höchstleistungsrechnern. In den Teilen I und II dieser Arbeit betrachten wir zwei verschiedene Arten von Mehrkomponentenproblemen: ***(i) Die Simulation von oberflächengekoppelten Fluid-Struktur Interaktionen*** und ***(ii) die medizinische Bildanalyse von Gehirntumor MRT Bildern***. In (i) erreichen wir Simulationen mit höchster Genauigkeit durch die Kombination verschiedener Effekte wie beispielsweise Fluidströmung und verursachte Deformation von Arterienwänden in Simulationen des Blutkreislaufes oder die Kombination von Fluidströmung, Wärmetransport und mechanischen Belastungen im Zusammenhang mit Kühlungssystemen. Für Problem (ii) legen wir unser Augenmerk auf (a) die Unterstützung des Tranfers bestimmter Informationen, wie etwa die Markierung funktionaler Gehirnregionen, von einem statistischen (gesunden) Atlasgehirn zum jeweiligen Patientengehirn (welches wegen des Tumors topologische Unterschiede aufweist), sowie (b) die Ermöglichung patientenspezifischer Gehirntumorwachstumssimulationen aufgrund kalibrierter biophysikalischer Gehirntumormodelle (mit Parameterschätzung durch inverse Gehirntumorsimulation mit nur einem Datenmesspunkt in der Zeit). Anwendungen und spezielle Charakteristiken beider Probleme sind sehr unterschiedlich, jedoch vereinen beide starke Relationen und Abhängigkeiten zwischen den beteiligten Komponenten und ihre mathematische Formulierung resultiert in beachtlichen, sehr großen, gekoppelten Systemen von partiellen Differentialgleichungen.

Teil I zielt auf die Entwicklung robuster und effizienter quasi-Newton Methoden für die black-box Kopplung partitionierter Fluid-Struktur Interaktion Simulationen ab. Der Partitionierungsansatz ermöglicht hohe Flexibilität und erleichtert die Austauschbarkeit von Subkomponenten. Die Aufteilung von Mehrphysik Problemen in einzelne Bestandteile erfordert jedoch hochentwickelte Kopplungsstrategien um Relationen und Abhängigkeiten zwischen den Komponenten korrekt abzubilden und Instabilitäten wirksam zu eliminieren. Wir entwickeln fortgeschrittene quasi-Newton Methoden, die über die Lösung einer nicht-linearen Fixpunktgleichung am Interface wirksam die Kopplung von zwei (oder mehreren) Komponenten herbeiführen. Allgemeinhin etablierte und hochmoderne Methoden erfordern entweder aufwändiges Einstellen von problemabhängigen Parametern oder sind nicht praktikabel für die Lösung von Systemen mit sehr vielen Unbekannten. Durch die Entwicklung von parameterfreien Alternativen mit linearer Zeit- und Speicherkomplexität heben wir die Robustheit und parallele Skalierbarkeit von quasi-Newton Methoden auf ein neues Niveau. Die entwickelten Methoden sind in dem parallelen Universalkopplungswerkzeug preCICE implementiert.

Teil II behandelt die medizinische Bildanalyse von Glioblastoma Multiforme Pathologien zur Ermöglichung patientenspezifischer Gehirntumorsimulation. Wir kombinieren medizinische Bildregistrierung mit biophysikalischer Parameter Schätzung (Inversion) mit dem Ziel der Erleichterung der Diagnose, der Unterstützung zur Operationsplanung sowie allgemein zur Verbesserung der Wirksamkeit von Therapiemaßnahmen. Wir stellen zwei mathematische Formulierungen des Problems vor und zerlegen das resultierende großskalige, hoch nicht-lineare und nicht-konvexe, PDGL-beschränkte Optimierungsproblem in zwei eng verzahnte Teilprobleme: inverse Gehirntumorsimulation und medizinische Bildregistrierung. Zur Lösung des Problems erschließen wir eine neuartige, modulare Strategie basierend auf einer Picard Iteration. Damit lösen wir erstmalig erfolgreich das Problem der Tumorwachstumsinversion im Gehirn, basierend auf nur einem Datenmesspunkt in der Zeit, mithilfe eines gradientenbasierten Ansatzes. Wir präsentieren SIBIA, ein Framework zur Realisierung der gemeinsamen Invertierung, welches zu extrem hohen Bildauflösungen und massiv paralleler Berechnung auf Zehntausenden Kernen skaliert. Wir wenden unsere Methodik auf synthetische und reale klinische Bilddaten an und verzeichnen hervorragende Ergebnisse für die Registrierung normaler und abnormaler (pathologischer) Gehirne. Desweiteren präsentieren wir Ergebnisse, die einen Machbarkeitsnachweis für eine erfolgsversprechende Methodik zur Extrahierung relevanter biophysikalischer Informationen liefert.

Hochentwickelte inexakte Newton Verfahren sind ein grundlegendes Werkzeug in beiden Teilen. Wir verbinden beide Teile durch Herausarbeitung von Gemeinsamkeiten und Unterschieden der verwendeten Varianten in beiden Forschungsfeldern und Angleichung der verwendeten Notation.

To my wonderful parents
Hanna and Hans-Albrecht,
and to my beloved wife Natasha.

Acknowledgment

> *But thanks be to God! He gives us the victory through our Lord Jesus Christ.*
>
> 1 Corinthians 15:57

First and foremost I would like to give thanks to God, who gave me the opportunity, the talents and the strength to get to this point, and who blessed me beyond belief. Furthermore, I would like to express my gratefulness to those who had an important impact on this thesis.

Over the course of my PhD, I have been blessed with several exceptional people – mentors, co-workers, and friends – from whom I benefited in various ways, subject-specifically and personally. I am deeply grateful to my PhD advisor and mentor Prof. Miriam Mehl. Thank you Miriam for the countless hours and energy you invested in advising me. Thank you for your excellent and insightful advice, your honest comments, and for always having the time to collaborate with me. I learned from your expertise and immense knowledge in many ways. Thank you also for your friendship, for being understanding, and for your motivation and support in tough times. This work would have been not possible without your guidance and support throughout these three and a half years; thank you for being the best PhD advisor I could have wished for.

I was fortunate to have the opportunity to spend nine months abroad as a visiting researcher in the group of Prof. George Biros at the Institute for Computational Engineering and Sciences (ICES), UT Austin. Throughout this time, George became my co-advisor and his experience and expertise were of great benefit to this work. George, thank you for the time and effort you invested in our collaboration, thank you for your hospitality, and for your eminently valuable guidance and teaching. I would like to thank my co-worker and friend Prof. Andreas Mang for his valuable support, his overly helpful advice, and the fruitful discussions. Thank you Andreas for generously sharing your knowledge, for supporting me in any way as a friend, and for an unforgettable time working together. Furthermore I would like to thank Amir Gholami and Shashank Subramanian, whom I collaborated with at UT Austin. Thank you for your hard work and the fruitful collaboration and always pursuing a shared aim. Thank you Shashank for the enjoyable working climate and the interesting discussions. I would like to thank the PADAS group and ICES for their hospitality and amicable incorporation.

I am very thankful for the pleasant and successful collaboration within the preCICE software project and the ExaFSA project. Thank you Benjamin Uekermann, David Blom, and Florian Lindner. In particular, thank you Benjamin for the helpful discussions about quasi-Newton and implementation and thank you David for your generous help with the implementation of the manifold mapping. I would also like to thank all my colleagues at SSE/SGS for the friendly working atmosphere and mutual support. In particular, thank you David for fixing my computer when I was desperate at SC'16 and thank you Julian for providing the template for this thesis. I would also like to thank my two lab-mates Mario and Amin. Thank you for refreshing everydays work and adding fun to it, thank you for our friendship and the mental support.

I would like to thank my committee members Prof. Miriam Mehl, Prof. George Biros, Prof. Andrés Bruhn, Prof. Oliver Röhrle and Prof. Sebastian Padó. I am grateful for the time they spent in reviewing my thesis and to have their endorsement on my PhD.

Furthermore, I thankfully acknowledge the financial support of SimTech and the German Academic Exchange Service (DAAD) as well as the provided computing time on HazelHen at the High Performance Computing Center Stuttgart (HLRS) and on Lonestar5 at the Texas Advanced Computing Center (TACC).

I would like to give thanks to my family and many friends who helped me and supported me throughout my studies. Thanks to my siblings Ruth, Kai, Jul, Jule, Rahel, Ines and Bahaa. Thank you Sabine for your immense support, our good conversations, and your friendship. Thank you Chris, Roman, Dani, Patricia, Sascha and Georg . Thank you Raciel and Glenda for your hospitality and warm and welcoming support. Thank you Eric, Jeff, and Joey for your friendship.

My deepest gratitude goes to my wonderful parents who continuously supported me and gave me guidance. Thank you for your love and caring, thank you for being an anchor in tough times. I would not have gotten to this point without you. Lastly, I would like to thank my dear and beloved future wife Natasha. Thank you Tash for your great love, for supporting me and being by my side. Thank you also for proofreading this thesis.

Stuttgart, December 4, 2018
Klaudius Scheufele

List of Publications

Parts of this thesis have been previously published in:

[Blo15] **Blom**, D. et al.: *A Review on Fast Quasi-Newton and Accelerated Fixed Point Iterations for Partitioned Fluid-Structure Interaction Simulation*. Englisch. Modeling and Simulation in Science, Engineering and Technology. Springer International Publishing, 2015, pp. 1–12.

[Bun16a] **Bungartz**, H.-J. et al.: *Precice – a fully parallel library for multi-physics surface coupling*. Computers and Fluids (2016).

[Bun16b] **Bungartz**, H.-j. et al.: *Partitioned fluid-structure-acoustics interaction on distributed data - coupling via preCICE*. Software for Exascale Computing - SPPEXA 2013-2015. Lecture Notes on Computational Science and Engineering (LNCSE). Springer, Berlin Heidelberg New York, 2016, pp. 239–266.

[Gho17] **Gholami**, A. et al.: *A framework for scalable biophysics-based image analysis*. Proc ACM/IEEE Conference on Supercomputing. 19. 2017, 19:1–19:13.

[Hae15] **Haelterman**, R. et al.: *Improving the performance of the partitioned qn-ils procedure for fluid-structure interaction problems: filtering*. Computers and Structures (2015).

[Lin15] **Lindner**, F. et al.: *A comparison of various quasi-Newton schemes for partitioned fluid-structure interaction*. Proceedings of 6th International Conference on Computational Methods for Coupled Problems in Science and Engineering. Venice, 2015, pp. 1–12.

[Man17] **Mang**, A. et al.: *SIBIA-GlS: Scalable Biophysics-Based Image Analysis for Glioma Segmentation*. Englisch. The multimodal brain tumor image segmentation benchmark (BRATS), MICCAI. -, 2017, pp. 202–209.

[Sch] **Scheufele**, K. et al.: *Coupling brain-tumor biophysical models and diffeomorphic image registration*. Computer Methods in Applied Mechanics and Engineering (), pp. 533–567. issn: 0045-7825. https://doi.org/10.1016/j.cma.2018.12.008.

[Sch17] **Scheufele**, K.; **Mehl**, M.: *Robust multi-secant quasi-newton variants for parallel fluid-structure simulations – and other multi-physics applications*. SIAM SISC Copper Mountain Special Issue (2017).

A publication on the moving-atlas solution scheme in part II of this thesis is work in progress.

Contents

1 Introduction

Interaction of Systems. The concept of ***interaction*** is ever-present and the essence of nature. Communication, for instance—in all its variations—is one form of interaction. In science, a fundamental interaction is the mutual interplay of elementary particles. On a higher level, we speak of the ***interaction of systems***. Scientific computing targets the numerical simulation and assessment of physical, chemical, or biological phenomena. Applications in engineering, medicine, or system behavior forecasting raise the need for ever more complex models, capturing detailed effects and their fine grained interactions to reveal ever deeper insights. Neglecting strong interactions between involved fields, sub-components, or effects would lead to inaccurate or wrong results. With the remarkable advances in technology and rapidly evolving, powerful compute systems, the numerical treatment and analysis of the aforementioned effects of interacting systems have become feasible; ever growing compute resources and massively parallel execution allow for the solution of such large-scale and formidable problems with complicated numerics.

Solving Multi-Component Systems. The primary focus of this work is the numerical treatment and solution of ***strongly coupled multi-component problems*** which are inherently hard to solve as different components require tailored approaches, while the mutual interaction and dependencies amongst the different models need to be fulfilled at all times. In this thesis, we focus on two multi-component problems:

I. ***Multi-physics simulation***, in particular, the coupling of ***partitioned fluid-structure interaction***

II. ***Brain tumor biophysical parameter inversion coupled with medical image registration***

Both problems find very different applications and feature quite distinct characteristics, yet both are hallmarked by strong inter-component relations. Those systems have a natural need for high performance computing, since the smaller modeling error we get when switching from a single-component to a multi-component model is useless if we cannot sustain a high resolution. Furthermore, the time-to-solution is critical. Thus, we opt for the development of coupling schemes and tailored, highly efficient, parallel algorithms for the advanced numerical treatment of each of the challenging problems above. The realization of modular, flexible solution strategies is another design goal for the methodologies developed in this work.

In what follows, we give a short introduction to both problems, present their main challenges, and discuss methodologies and solution approaches pursued in this work. A more detailed introduction for each application problem, alongside with a review of relevant literature is given in §3.1 and §5.2.

1.1 Multi-Physics Simulations (Fluid-Structure Interaction)

Fluid-Structure Interaction. Multi-physics simulations involve multiple physical models constituting the simultaneous treatment of various physical phenomena. Examples include the interaction of electrostatics and magneto-statics, chemical reactions, or the interplay of fluid dynamics and structural mechanics with heat transfer or integration of induced acoustics. They typically involve the solution of coupled systems of partial differential equations. Within this work, we focus on fluid-structure interaction simulations — a surface coupled multi-physics problem. They describe the phenomenon of solid body deformation under fluid exitation, and the reciprocal impact of the deformation on the fluid dynamics. The numerical simulation of this mutual interplay is of great value in various disciplines ranging from different fields of engineering (aerospace engineering, marine engineering, renewable energy) to computational medicine (hemodynamics and cardiovascular system simulations); application examples include the simulation of aircraft structures, the simulation of unsteady deformations of a parachute to improve safety, the simulation of blood flow and heart valves.

Partitioned Solution. While the consideration of interacting effects significantly aids an accurate representation of physical phenomena observed in nature, it also complicates the solution of the problem and increases the overall difficulty and complexity dramatically. The complexity of the corresponding solver algorithms and implementations typically can be tackled with so-called partitioned simulations reusing existing and established software codes for involved components. The ***partitioned approach*** profits from decades of experience and development in terms of models, numerical methods, and parallel scalability of the single components. This strategy contrasts the monolithic approach, where all governing equations (including coupling effects) are implemented in a single, dedicated solver package and solved simultaneously. In this work, we employ the partitioned strategy and opt for a generic, ***black-box coupling*** approach, which offers great ***flexibility*** and reduces the time for setting up multi-physics simulation environments.

Equation Coupling. The separation into sub-components, however, inherently suffers from ***stability problems*** and induced ***oscillations***. To recover a consistent solution in every time step, an outer coupling iteration that enforces the coupling conditions, damps oscillations, and improves stability becomes necessary. Numerically, this is realized via the solution of a non-linear fixed-point equation at the coupling interface. This step requires sophisticated coupling numerics. This is what Part I of this thesis focuses on. Due to hidden solver internals[1], quasi-Newton approaches are a particularly promising numerical approach for the outer coupling of multi-physics simulations. Based on input / output values, they establish a low-rank estimation of the inverse Jacobian.

Advanced Quasi-Newton Methods. In this work, we develop advanced, highly robust and efficient state-of-the-art quasi-Newton methods with excellent convergence properties, applicable in a general purpose setting with minimal tuning of parameters. In particular, we opt for highly scalable, parallel distributed memory algorithms with a runtime and storage complexity that is linear in the number of unknowns at the interface. We discuss this in Chapter 2. A full introduction into the coupling of partitioned multi-physics simulations in given in Chapter 3.

[1] With the black-box paradigm, we assume that only input / output data is available, and, in particular, solver internals like discretization or derivatives are inaccessible.

1.2 Coupling Brain-Tumor Biophysical Models with Medical Image Registration

Data Assimilation in Brain Tumor MR Image Analysis. The second part of this thesis targets the volume coupling of inverse brain tumor progression simulation with medical image registration. We target the analysis of brain tumor MR imaging data of patients diagnosed with glioblastoma, an extremely aggressive grade IV primary brain tumor. The tumor is characterized by a highly infiltrative nature and broad, diffusive invasion into surrounding healthy tissue; its aggressiveness is reflected in a 100% fatality rate within 6-12 months. The integration of biophysical modeling with medical image registration is intended to improve diagnosis and clinical decision making, aid prognosis, and foster the design of new treatment protocols. Applications involve biophysically-augmented image analysis and image-driven biophysical model development. Both subjects are essentially large-scale data-assimilation ***inverse problems*** formulated as ***multi-component PDE-constrained non-linear optimization problems***. As opposed to forward problems, inverse problems are inherently ill-posed and feature strong instabilities. The development of efficient, robust, and accurate solvers for these formidable problems is very challenging. We focus on two main goals:

(a) biophysically aided ***normal-to-abnormal registration***, and

(b) ***biophysical model calibration*** for patient specific simulations and prognosis.

The first is used in automated segmentation of MR imaging data or mapping of structural and functional information from expert-labeled atlas brains to specific patient brains; this predominantly supports surgical planning. Achievement of the second goal fosters quantitative understanding of human physiology and cancer progression, along with enabling predictive medication choices for individuals.

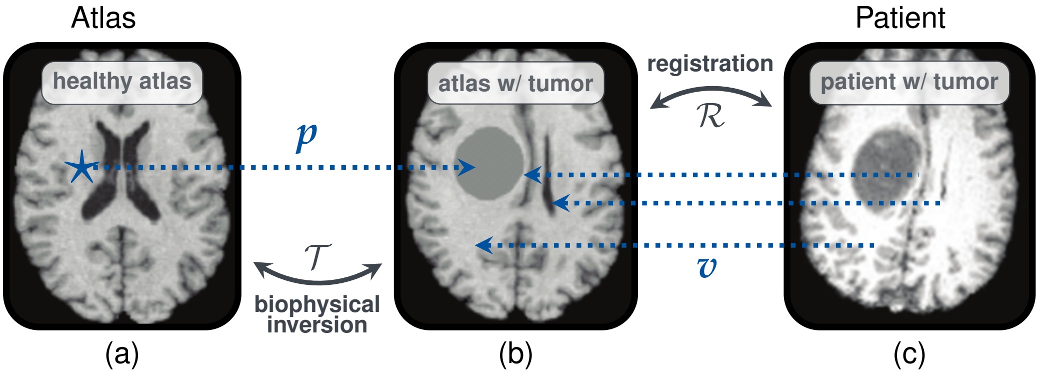


FIGURE 1.1 *Joint biophysical model inversion and medical image registration [Sch; Gho17c]. (a) Normal-to-abnormal registration needs a biophysics enhanced formulation to solve for a (biophysically) plausible map between the topological different spaces. (b) For patient individual tumor growth simulation, we need to calibrate a biophysical model by means of inversion methods. Due to inaccessibility of the healthy patient's brain, simulations need to be carried out in a standard atlas brain (left image). This requires a coupling with medical image registration, mapping the two spaces. Figure modified from [Goo13].*

The Joint Biophysical Inversion and Medical Image Registration Approach. The stated goals both require a tight integration of biophysical brain tumor models with medical image registration. For (a), the ***topological differences*** between the healthy brain and the pathologic patient brain render this problem extremely difficult for the registration, and the inclusion of ***biophysical constraints*** becomes absolutely inevitable in order to aid the registration to result in a (biophysically) ***plausible***

deformation map. Inverse tumor-growth simulation for biophysical model calibration is a time-dependent problem, requiring at least two snap shots in time. In practice, however, a snap shot of the ***healthy patient*** is inaccessible. To achieve (b), we therefore deploy an atlas-matching technique, and use a normal atlas brain to carry out tumor simulations. This step introduces the need for registration between atlas and patient.

The integration of inverse tumor-growth simulation with image registration as explained above is outlined in Fig. 1.1. The underlying idea is most intuitively accessible by explaining an exemplary forward problem[2]. We grow an artificial tumor in the healthy atlas brain using the brain tumor progression model, resulting in a new image of the atlas, comprising both healthy tissue and tumor. The information of the artificially grown tumor is then used to compute a deformation map between patient and atlas, which results in a tumor-plus-deformation warped atlas. Conversely, for the ***inverse*** problem, we seek for a deformation map and biophysical model parameters such that the tumor-plus-deformation warped atlas matches the patient input data.

SIBIA. Physics and imaging need to be tightly integrated both in terms of software and algorithms. The latter need to be reliable and scalable in order to be useful in analyzing clinical data. The need for scalability and distributed memory realization is due to the increasing scanner resolution and advances in medical imaging [Chu13; Kut16; Tom14], which result in extremely large data sets[3]. In this thesis, we develop SIBIA, a framework for Scalable Integrated Biophysics-based Image Analysis, which provides highly scalable and efficient tools and solvers for the realization of the above described integration of biophysics and image registration. Since the biophysical models and registration formulations are always in a state of flux and varying in complexity, we opt for a flexible modular approach and consider the viability of a Picard (or fixed-point) iteration to establish the multi-component coupling. In particular, we are the first to solve this extremely challenging large-scale, highly non-linear, and non-convex multi-component PDE-constrained optimization problem with a gradient based approach. We provide an extensive literature review in §5.2.2.

A thorough introduction into the problem and our developed solver framework is given in Chapter 5.2.

[2] The described forward problem results from the moving-patient formulation of our joint multi-component problem. Later, we introduce another formulation, called moving-atlas.

[3] The resolutions of theses data sets can reach $5\mu\text{m}\times5\mu\text{m}\times5\mu m$ resulting in $\mathcal{O}(4.8TB)$, if stored in half precision.

1.3 Contributions

Condensed, the main contributions of this thesis are:

- We develop and analyze coupling schemes for partitioned fluid-structure interaction simulations. In particular, we develop highly robust and efficient advanced quasi-Newton methods to accelerate the non-linear equation coupling at the coupling interface. We design powerful linear complexity methods that are efficient without tuning of problem dependent parameters. We demonstrate their efficiency and good parallel scalability.
- We give a lucid overview of and compare the methodological components and characteristics of quasi-Newton methods utilized for acceleration of non-linear fixed-point problems and in PDE-constrained optimization.
- We develop the SIBIA framework for integration of biophysical inversion and medical image registration to be applied in brain tumor MR image analysis. It features reliable, highly efficient, and highly scalable solvers to tackle the formidable coupled multi-component optimization problem of joint tumor inversion and medical image registration. We are the first to solve this problem employing a gradient based approach.
- We deduce two formulations for the coupled multi-component problem, tailored for goal (a) and goal (b), respectively, and derive a modular Picard iteration-type solution strategy for each formulation based on the two sub-components inverse tumor-growth simulation and image registration. We show convergence of both schemes by monitoring the reduced gradient of the coupled problem formulations.
- We demonstrate the validity and efficiency of our approach by thorough numerical analysis on synthetic and actual clinical data sets. We compare both schemes with regards to quality and suitability for normal-to-abnormal registration and reconstruction of clinically relevant or diagnostically meaningful information.
- We optimize the parallel, distributed memory solvers and algorithms to achieve excellent parallel scalability and significantly reduce the time-to-solution. Here, we reach a speedup factor of 10-20 for the time-to-solution by employing quasi-Newton methods and a multi-level grid-refinement strategy with gradually improving coarse level solutions for the non-linear optimization.

1.4 Structure of the Thesis

Chapter 2 – Multi-Secant Quasi-Newton Methods for Non-Linear Problems: This chapter is dedicated to quasi-Newton methods used to accelerate non-linear fixed-point problems, or to solve non-linear optimization problems. It gives a lucid overview and compares methodological components of quasi-Newton methods for both applications. We introduce advanced, robust quasi-Newton methods, and present their efficient algorithmic realization on distributed memory architectures.

Part I – Partitioned Coupling of Surface Coupled Multi-Physics Simulation

Chapter 3 – Partitioned Multi-Physics Simulation and Fluid-Structure Interaction: Here, we give an introduction to the black-box coupling of partitioned fluid-structure interaction simulations. We present challenges of partitioned coupling and the realization in the parallel coupling framework preCICE; emphasis is put on the equation coupling.

Chapter 4 – Evaluation of Robust Quasi-Newton Methods for Partitioned FSI Simulations: This chapter investigates the performance of the developed advanced quasi-Newton methods in actual partitioned FSI simulations. It studies convergence properties, characteristics, and parallel efficiency of two highly advanced quasi-Newton methods in 3D applications.

Part II – Coupling of Bio-Physical Brain-Tumor Models with Medical Image Registration

Chapter 5 – Scalable Biophysics-based Image Analysis: This chapter introduces the second application problem: Joint biophysical inversion and medical image registration for analysis and quantification in brain tumor imaging. We present the sub-component solvers for inverse tumor-growth and image registration and give numerical results for their parallel efficiency.

Chapter 6 – Coupling Schemes for the Multi-Component Problem: The formulation of the joint optimization problem and its coupled solution are given in this chapter. Two different Picard-type iteration strategies are derived. This chapter introduces the SIBIA framework developed in this work.

Chapter 7 – Evaluation of Coupling Schemes and Advanced Methods: We provide a thorough numerical analysis of our methodology, the two Picard iteration-type solution strategies and advanced algorithms to speed-up the time-to-solution. We demonstrate the validity and efficiency of our scheme for synthetic and real data.

Chapter 8 – Conclusion: A summary of the main contributions of this work and gained conclusions is provided in this chapter. We also outline limitations of the current approach and recommendations for future directions.

2 Multi-Secant Quasi-Newton Methods for Non-Linear Problems

This chapter is dedicated to quasi-Newton methods – a powerful tool to tackle non-linear problems in regimes where full Newton's method becomes infeasible (such as large-scale simulations, inaccessible or too expensive Jacobians and inexact/erroneous evaluations). We examine methodological components, efficient algorithms and parallel implementations of quasi-Newton variants that are tailored for or predominantly used in (*i*) partitioned black-box multi-physics coupling (here, fluid-structure interaction (FSI), captured in part I of this thesis) in §2.2 and (*ii*) non-linear optimization (here, non-linear PDE constrained optimization, considered in part II of this thesis) in §2.3.

§2.2 contains algorithms and variations for advanced quasi-Newton methods and their efficient implementation on distributed data, that have been shown to be particularly suited to solve the non-linear interface fixed-point equation arising from coupling in partitioned, transient fluid-structure interaction simulations. This part comprises significant own contributions that have been achieved within this work [Sch15; Lin15; Blo15b; Bun16b; Bun16a; Hae15; Sch17] and is therefore emphasized. The respective methods are implemented in the parallel black-box coupling software preCICE[1] (precise Code Interaction Coupling Environment), which has been primarily developed and maintained by Bernhard Gatzhammer, Benjamin Uekermann, Florian Lindner, Miriam Mehl and myself. The presentation within this thesis principally focuses on contributions from [Sch17].

§2.3 on the other hand, mainly contrasts methodological and structural similarities and differences to quasi-Newton methods typically used in the context of non-linear optimization. This section is kept shorter and its primary contribution is the relation of known textbook methods (i.e., LBFGS) to the methods examined in §2.2. A summarizing comparison is given in §2.4.

2.1 Introduction to Multi-Secant Quasi-Newton Methods

A wide range of application problems from all kinds of disciplines boil down to or entail the solution of a non-linear root-finding problem of the form

$$\mathcal{R}(\boldsymbol{x}) = \mathbf{0}, \tag{2.1}$$

with $\mathcal{R} \in \mathbf{R}^{N,N}$, $\boldsymbol{x} \in \mathbf{R}^N$ at their very heart – just as the two application problems considered within this work:

I. The coupling of partitioned fluid-structure (or multi-physics) interaction (see §1.1; detailed in

[1] www.precice.org

part I) results in a fixed-point equation formulation

$$\mathcal{H}(x) = x \quad \text{with residual} \quad \mathcal{R}(x) := \mathcal{H}(x) - x = \mathbf{0}, \tag{2.2}$$

at the multi-physics interface, with an operator $\mathcal{H}\colon \mathbf{R}^N \to \mathbf{R}^N$. Coupling conditions at the interface (such as balance of displacements and forces for fluid-structure interaction coupling) that ensure to recover a solution of the monolithic formulation, are fulfilled at a fixed point $x^\star$ with $\mathcal{H}(x^\star) = x^\star \Leftrightarrow \mathcal{R}(x^\star) = \mathbf{0}$.

II. The joint registration and tumor inversion problem (see §1.2; detailed in part II) is by its nature a multi-component PDE-constraint optimization problem of the form

$$\min_{x} \mathcal{J}[d, u, x] = \mathcal{D}[d, u] + \beta \mathcal{S}[x] \qquad \text{subject to} \quad \mathcal{C}[u, x] = 0, \tag{2.3}$$

with the objective function $\mathcal{J}\colon \mathbf{R}^n \to \mathbf{R}$ quantifying the discrepancy $\mathcal{D}\colon \mathbf{R}^n \to \mathbf{R}$ between the system's predicted state u and the observed data d with the system dynamics encoded in the constraints $\mathcal{C}$. A necessary condition for a solution of (2.3) is stationarity of the first variation of the Lagrangian functional $\mathcal{L}\colon \mathbf{R}^n \to \mathbf{R}$ with respect to x and u, i.e.,

$$\mathcal{R}(x) := g(x) = \delta \mathcal{L}[x] \overset{!}{=} 0. \tag{2.4}$$

The Lagrangian functional $\mathcal{L}$ results from translating (2.3) into an unconstrained optimization problem, following a standard adjoint approach.

The de facto standard tool for non-linear problems of the form (2.1) is Newton's method, i.e.,

$$\textbf{solve} \qquad (\nabla \mathcal{R}(x^k))^T \Delta x^k = -\mathcal{R}(x^k), \tag{2.5a}$$

$$\textbf{set} \qquad x^{k+1} = x^k + \Delta x^k \tag{2.5b}$$

with the Jacobian $(\nabla \mathcal{R}(x^k))^T$ of the non-linear operator $\mathcal{R}$ at the current iterate x^k. We will see, that for multiple reasons, using full Newton is not desirable or even becomes infeasible for a large class of problems of the form (2.1). In particular when going to large-scale applications, Newton becomes infeasible due to unacceptably high computational cost and excessive memory requirements. Less expensive, inexact alternatives, tailored to the specific problem at hand become indispensable. So-called multi-secant quasi-Newton methods that establish an approximation to the system Jacobian based on Taylor's theorem using secant information from previous iterations, are a powerful, efficient and robust alternative with a growing field of application. In [Fan08], Fang and Saad characterize a class of problems of the form (2.1) that pose significant or even prohibitive challenges for full Newton, yet can be tackled using multi-secant quasi-Newton methods. A problem belongs to this class, if

(i) the dimension n of the non-linear operator $\mathcal{R}$ is large,

(ii) exact computation of $(\nabla \mathcal{R}(x))^T$ is either too expensive or its analytic form is inaccessible,

(iii) the computational cost to evaluate $\mathcal{R}(x)$ is very high,

(iv) the evaluation of $\mathcal{R}(x)$ contains errors, i.e., the problem is noisy.

Characteristics *(ii)* and *(iii)* are predominant for the partitioned FSI problem (I), where black-box single-physics solvers are coupled via an interface fixed-point equation (2.2) to recover the monolithic solution in every time step. Closed source codes, black-box solvers, table look-ups or internal stochastics render access to Jacobian information impossible. The joint tumor inversion and medical image registration problem (II) portrays all of the above mentioned characteristics. As a result, various standard methods become impractical or infeasible. For instance, characteristic (i) suggests matrix-free, limited memory algorithms, while (ii) renders standard exact Newton inapplicable. Characteristic (iii) makes line-search algorithms and inner Krylov iterations very costly, and (iv) poses significant convergence challenges on Newton-Krylov methods.

Multi-secant methods take ***inexact*** Newton steps by building an ***approximation*** of the Jacobian matrix $(\nabla\boldsymbol{\mathcal{R}})^T$ required in (2.5a), based on past information. Taylor's theorem suggests that the Jacobian has to satisfy secant equations $(\nabla\boldsymbol{\mathcal{R}})^T\Delta\boldsymbol{x}_i^k \approx \Delta\boldsymbol{r}_i^k$, $i = j, \ldots, k$ for difference vectors $\Delta\boldsymbol{x}_i^k$ and $\Delta\boldsymbol{r}_i^k$ of input and output data of the operator $\boldsymbol{\mathcal{R}}(\boldsymbol{x}^i) = \boldsymbol{r}^i$, for $\boldsymbol{x}^i$ not too far away from the current iterate $\boldsymbol{x}^k$. By exchanging the roles $\Delta\boldsymbol{x}_i^k \leftrightarrow \Delta\boldsymbol{r}_i^k$ and $(\nabla\boldsymbol{\mathcal{R}})^T \leftrightarrow (\nabla\boldsymbol{\mathcal{R}})^{-T}$, this strategy even allows us to directly establish an ***approximation of the inverse Jacobian*** $\boldsymbol{M}^{-1} \approx (\nabla\boldsymbol{\mathcal{R}})^{-T}(\boldsymbol{x}^k)$, rendering the linear system solve in (2.5a) dispensable by computing the Newton step form a simple matrix-vector multiplication as

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k - \boldsymbol{M}^{-1}\boldsymbol{\mathcal{R}}(\boldsymbol{x}^k). \tag{2.6}$$

This is a key point, enhancing multi-secant quasi-Newton methods to be competitive to rapidly converging Newton methods due to significantly reduced computational cost per Newton iteration. We calculate the approximation of the inverse Jacobian matrix, as a solution of the ***multi-secant*** equation

$$\boldsymbol{M}^{-1}\boldsymbol{V}_k^\eta = \boldsymbol{W}_k^\eta, \tag{2.7}$$

with the matrices $\boldsymbol{V}_k^\eta \in \mathbf{R}^{N\times\eta}$ and $\boldsymbol{W}_k^\eta \in \mathbf{R}^{N\times\eta}$ storing the collected input/output difference vectors throughout the performed Newton iterations $k = 0, \ldots,$ i.e.,

$$\boldsymbol{W}_k^\eta = \left[\Delta\boldsymbol{x}_{k-1}^k, \Delta\boldsymbol{x}_{k-2}^k, \ldots, \Delta\boldsymbol{x}_{k-\eta}^k\right], \qquad \text{with } \Delta\boldsymbol{x}_i^k = \boldsymbol{x}^k - \boldsymbol{x}^i\ , \tag{2.8a}$$

$$\boldsymbol{V}_k^\eta = \left[\Delta\boldsymbol{r}_{k-1}^k, \Delta\boldsymbol{r}_{k-2}^k, \ldots, \Delta\boldsymbol{r}_{k-\eta}^k\right], \qquad \text{with } \Delta\boldsymbol{r}_i^k = \boldsymbol{\mathcal{R}}(\boldsymbol{x}^k) - \boldsymbol{\mathcal{R}}(\boldsymbol{x}^i). \tag{2.8b}$$

The amount of past information incorporated into the inverse Jacobian approximation (i.e., the memory of the quasi-Newton method) is controlled by the parameter $\eta, 0 < \eta \leqslant k$[2]. For the sake of readability, we omit the parameter η if it is clear from the context and write short $\boldsymbol{V}_k$ for $\boldsymbol{V}_k^\eta$ and $\boldsymbol{W}_k$ for $\boldsymbol{W}_k^\eta$. The specific choice of η gives raise to various instantiations of the method, featuring different properties. In what follows, we discuss this in more detail.

In practice, it is natural to assume $\eta \ll N$. In this case, (2.7) is highly under-determined and suitable further restrictions in form of a norm-minimization term need to be added to find a unique solution for the N^2 entries of $\boldsymbol{M}^{-1}$. We thus search for a minimizer $\boldsymbol{M}^{-1}$ such that

$$\left\|\boldsymbol{M}^{-1} - \boldsymbol{M}_{prev}^{-1}\right\|_X \to \min \quad \text{subject to} \quad \boldsymbol{M}^{-1}\boldsymbol{V}_k^\eta = \boldsymbol{W}_k^\eta, \tag{2.9}$$

i.e., the new approximation $\boldsymbol{M}^{-1}$ incorporates new multi-secant information while staying as close as

[2] In a transient setting, convergence can be significantly improved if additionally secant information from previous time steps (as opposed to previous information from only the current time step) is retained. In this case, η may be larger than the current number of Newton iterations k, i.e., $0 < \eta \leqslant k + \sum_{q=0}^{n-1} k_q$, depending on the taken time steps n and respective Newton iterations k_q per time step. This is detailed in §2.2.1.

possible to a previous approximation (or initial guess) $\boldsymbol{M}_{prev}^{-1}$ of the inverse Jacobian. Specific choices for $\boldsymbol{M}_{prev}^{-1}$ and the employed norm $\|\cdot\|_X$ result in different variations of this generic multi-secant method with distinct characteristics and properties.

We will present different instantiations (i.e., choices for $\boldsymbol{M}_{prev}^{-1}$, η and $\|\cdot\|_X$) of the generic multi-secant quasi-Newton formulation (2.9) that are applied in the context of non-linear fixed-point problems in §2.2 and others that are predominantly used in the field of non-linear optimization in §2.3.

2.2 Multi-Secant Quasi-Newton for Fixed-Point Problems

A main contribution of this thesis is the design of highly scalable, robust, and efficient black-box quasi-Newton variants and their efficient distributed memory implementation within the parallel coupling library preCICE[3] [Bun16a; Bun16b; Uek16; Gat14], that have been shown to perform very well for the partitioned coupling of fluid-structure interaction simulations [Sch17]. This is detailed in §2.2.1 and §2.2.2. The presentation follows our previous work [Sch17].

We consider non-linear fixed-point equations of the form (2.2) that arise from the coupling of partitioned multi-physics simulations. Typically, the task function $\boldsymbol{\mathcal{H}}$ is evaluated by several software components together. For instance, assume a surface coupled multi-physics problem with two existing solvers for the respective single-physics problems (e.g., compute one time step of the single-physics problem for given boundary values). This induces operators $\boldsymbol{\mathcal{S}}_1$ and $\boldsymbol{\mathcal{S}}_2$ which map input data at the coupling interface in the vector spaces X_1 and X_2 to output values at the coupling interface. Hereby, $\boldsymbol{\mathcal{S}}_2$ requires the output computed by $\boldsymbol{\mathcal{S}}_1$ as input and vice versa, i. e., we have

$$\boldsymbol{\mathcal{S}}_1 \colon X_1 \to X_2 \quad \text{and} \quad \boldsymbol{\mathcal{S}}_2 \colon X_2 \to X_1. \tag{2.10}$$

Implicit interface coupling is then realized via fixed-point equations at the interface. For a sequential execution order, this results in the multiplicative Schwarz method, or Gauß-Seidel-type system (GS)

$$\boldsymbol{\mathcal{H}}(\boldsymbol{x}) := \boldsymbol{\mathcal{S}}_2 \circ \boldsymbol{\mathcal{S}}_1(\boldsymbol{x}) = \boldsymbol{x} \tag{GS}$$

for $\boldsymbol{x} := \boldsymbol{x}_1 \in X_1$. An additive Schwarz or Jacobi-type system (J)

$$\boldsymbol{\mathcal{H}}(\boldsymbol{x}) := \begin{pmatrix} \boldsymbol{0} & \boldsymbol{\mathcal{S}}_2 \\ \boldsymbol{\mathcal{S}}_1 & \boldsymbol{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \end{pmatrix} = \begin{pmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \end{pmatrix} \tag{J}$$

with $\boldsymbol{x} := (\boldsymbol{x}_1, \boldsymbol{x}_2)^T$ allows for parallel execution of the sub-solvers.

The exact Jacobian of the fixed-point operator or the sub-components is typically inaccessible and solver internals are hidden. Multi-secant quasi-Newton methods that establish an approximation to the system's Jacobian have been shown to be particularly suited to solve (i.e., accelerate/stabilize the fixed-point iteration) such types of fixed-point problems. In particular, for partitioned fluid-structure interaction, plain fixed-point iteration for (2.2), i.e.,

$$\boldsymbol{\mathcal{H}}(\boldsymbol{x}^k) = \tilde{\boldsymbol{x}}^k, \quad \boldsymbol{x}^{k+1} = \tilde{\boldsymbol{x}}^k \tag{2.11}$$

often times fails due to a strong interaction of fluid and structure sub-components coupled within

[3] www.precic.org

the operator $\boldsymbol{\mathcal{H}}$ [Van09a; Wal99; Küt08]. Further, the evaluation of $\boldsymbol{\mathcal{H}}$ is very expensive and reducing the number of iterations until convergence is achieved is crucial. Appropriate stabilization of the fixed-point iteration

$$\boldsymbol{x}^k \overset{\text{Fixed-Point}}{\rightsquigarrow} \tilde{\boldsymbol{x}}^k \overset{\text{QN}}{\rightsquigarrow} \boldsymbol{x}^{k+1} \tag{2.12}$$

in the form of (dynamic) under-relaxation or quasi-Newton acceleration (QN) ensues convergence[4]. While under-relaxation often times features slow convergence, quasi-Newton methods provide both stabilization and fast convergence [Deg09; Deg08a; Hae09b; Lin15; Mic04; Sch15]. In what follows, we focus on the latter. We discuss and study concrete instantiations of (2.9) used in literature to solve partitioned fluid-structure interaction problems and propose advanced algorithms developed in this work, which improve the stability and robustness of the methods and allow for large-scale simulations on massively parallel systems.

2.2.1 Methodological Components

In this subsection, we present important methodological components required for fast and robust convergence. Respective efficient algorithmic realization and parallelization on distributed data is presented in §2.2.2.

Preliminary Remarks. Following strategy (2.12), we reformulate the residual operator $\boldsymbol{\mathcal{R}}$ from (2.2) in terms of the inverse fixed-point map $\boldsymbol{\mathcal{H}}^{-1}$

$$\tilde{\boldsymbol{\mathcal{R}}}(\tilde{\boldsymbol{x}}) := \tilde{\boldsymbol{x}} - \boldsymbol{\mathcal{H}}^{-1}(\tilde{\boldsymbol{x}}) \overset{!}{=} \boldsymbol{0} \tag{2.13}$$

to include the latest fixed-point iterate $\tilde{\boldsymbol{x}}^k$ in the quasi-Newton acceleration. Due to (*i*) $\tilde{\boldsymbol{x}}^k = \boldsymbol{x}^k + \boldsymbol{\mathcal{R}}(\boldsymbol{x}^k)$ and (*ii*) $\boldsymbol{\mathcal{R}}^{-1} = \tilde{\boldsymbol{\mathcal{R}}} - \boldsymbol{I}$[5], a Newton step for (2.2) then reads

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k - (\nabla\boldsymbol{\mathcal{R}})^{-T}\boldsymbol{\mathcal{R}}(\boldsymbol{x}^k) \overset{(ii)}{=} \boldsymbol{x}^k - [(\nabla\tilde{\boldsymbol{\mathcal{R}}})^{-T} - \boldsymbol{I}]\boldsymbol{\mathcal{R}}(\boldsymbol{x}^k) \overset{(i)}{=} \tilde{\boldsymbol{x}}^k - (\nabla\tilde{\boldsymbol{\mathcal{R}}})^{-T}\tilde{\boldsymbol{\mathcal{R}}}(\tilde{\boldsymbol{x}}^k). \tag{2.14}$$

Thus, to construct an approximation $\tilde{\boldsymbol{M}} \approx (\nabla\tilde{\boldsymbol{\mathcal{R}}})^{-T}(\tilde{\boldsymbol{x}}^k)$, we collect input/output relations of the modified operator $\tilde{\boldsymbol{\mathcal{R}}}$, i.e.,

$$\tilde{\boldsymbol{W}}_k^{\eta} = \left[\Delta\tilde{\boldsymbol{x}}_{k-1}^k, \Delta\tilde{\boldsymbol{x}}_{k-2}^k, \ldots, \Delta\tilde{\boldsymbol{x}}_{k-\eta}^k\right], \qquad \text{with } \Delta\tilde{\boldsymbol{x}}_i^k = \boldsymbol{\mathcal{H}}(\boldsymbol{x}^k) - \boldsymbol{\mathcal{H}}(\boldsymbol{x}^i) = \tilde{\boldsymbol{x}}^k - \tilde{\boldsymbol{x}}^i\,, \tag{2.15}$$

and $\tilde{\boldsymbol{V}}_k^{\eta} = \boldsymbol{V}_k^{\eta}$ as $\tilde{\boldsymbol{\mathcal{R}}}(\tilde{\boldsymbol{x}}^k) = \boldsymbol{\mathcal{R}}(\boldsymbol{x}^k) = \boldsymbol{r}^k$. Using the fixed-point step $\tilde{\boldsymbol{x}}^k = \boldsymbol{\mathcal{H}}(\boldsymbol{x}^k)$ and formulating the quasi-Newton update from $\tilde{\boldsymbol{x}}^k$ to $\boldsymbol{x}^{k+1}$ is critical. As we seek for the Newton update in the column space of $\boldsymbol{W}_k$, going directly from $\boldsymbol{x}^k$ to $\boldsymbol{x}^{k+1}$ and thus defining $\boldsymbol{W}_k$ in terms $\boldsymbol{x}^k$ and not $\tilde{\boldsymbol{x}}^k$ would lead to linearly dependent updates; the search space would stagnate at the original dimension 2 ($\boldsymbol{x}^0$ and $\boldsymbol{x}^1 - \boldsymbol{x}^0$, where $\boldsymbol{x}^1$ is obtained from under-relaxation). For the sake of readability and comprehension, we abuse our formulation for the remainder of this section and use $\boldsymbol{M}$ instead of $\tilde{\boldsymbol{M}}$, $\boldsymbol{\mathcal{R}}$ as opposed to $\tilde{\boldsymbol{\mathcal{R}}}$, and $\boldsymbol{W}$ for $\tilde{\boldsymbol{W}}$.

Multi-Secant Quasi-Newton Variants. Within the context of solving non-linear fixed-point equations, we consider ***Generalized Broyden*** methods of the form (2.9) that minimize the difference

[4] Compared to applying QN directly in the step $\boldsymbol{x}^k \rightsquigarrow \boldsymbol{x}^{k+1}$, this modified approach allows for the efficient collection of the data for $\boldsymbol{W}_k$ and $\boldsymbol{V}_k$ on the fly throughout Newton iterations. We explain this in more detail below; see also [Deg09].

[5] $\boldsymbol{\mathcal{R}}^{-1} = \tilde{\boldsymbol{\mathcal{R}}} - \boldsymbol{I}$ as $\boldsymbol{\mathcal{R}}^{-1}\boldsymbol{\mathcal{R}}(\boldsymbol{x}) = \boldsymbol{x} = \boldsymbol{\mathcal{H}}(\boldsymbol{x}) - (\boldsymbol{\mathcal{H}}(\boldsymbol{x}) - \boldsymbol{x}) = \tilde{\boldsymbol{\mathcal{R}}}^{-1}\boldsymbol{\mathcal{R}}(\boldsymbol{x}) - \boldsymbol{\mathcal{R}}(\boldsymbol{x})$

to a previous approximation of the inverse Jacobian $\boldsymbol{M}_{prev}^{-1}$ in the ***Frobenius norm***. Using the method of Lagrange multipliers, problem (2.9) then translates to the unconstrained problem

$$\mathcal{L}[\boldsymbol{M}^{-1}, \boldsymbol{\lambda}] := \frac{1}{2}\|\boldsymbol{M}^{-1} - \boldsymbol{M}_{prev}^{-1}\|_F^2 + \boldsymbol{\lambda}(\boldsymbol{M}^{-1}\boldsymbol{V}_k - \boldsymbol{W}_k) \;\rightarrow\; \min \tag{2.16}$$

with the Lagrangian functional $\mathcal{L}$ and Lagrange multiplier $\boldsymbol{\lambda} \in \mathbf{R}^{N,r}$. A solution to (2.16) needs to satisfy the necessary first order conditions $\delta\mathcal{L}$ for optimality, i.e.,

$$\begin{aligned} \delta_{\boldsymbol{M}^{-1}}\mathcal{L} = 0 : \qquad & \boldsymbol{M}^{-1} - \boldsymbol{M}_{prev}^{-1} + \boldsymbol{\lambda}\boldsymbol{V}_k^T \overset{!}{=} 0 \\ \delta_{\boldsymbol{\lambda}}\mathcal{L} = 0 : \qquad & \boldsymbol{M}^{-1}\boldsymbol{V}_k - \boldsymbol{W}_k \overset{!}{=} 0. \end{aligned}$$

Inserting the first equation into the second reads $(\boldsymbol{M}_{prev}^{-1} - \boldsymbol{\lambda}\boldsymbol{V}_k^T)\boldsymbol{V}_k - \boldsymbol{W}_k$ yields $\bar{\boldsymbol{\lambda}} = -(\boldsymbol{W}_k - \boldsymbol{M}_{prev}^{-1}\boldsymbol{V}_k)(\boldsymbol{V}_k^T\boldsymbol{V}_k)^{-1}$ for a minimizer $\bar{\boldsymbol{\lambda}}$. As a result, we arrive at the generic update formula

$$\boldsymbol{M}^{-1} = \boldsymbol{M}_{prev}^{-1} + (\boldsymbol{W}_k - \boldsymbol{M}_{prev}^{-1}\boldsymbol{V}_k)(\boldsymbol{V}_k^T\boldsymbol{V}_k)^{-1}\boldsymbol{V}_k^T = \boldsymbol{M}_{prev}^{-1} + \widetilde{\boldsymbol{W}}_k\boldsymbol{V}_k^\dagger \tag{2.17}$$

for the approximate inverse system Jacobian matrix for ***generalized Broyden*** quasi-Newton methods. Here $\boldsymbol{V}_k^\dagger := (\boldsymbol{V}_k^T\boldsymbol{V}_k)^{-1}\boldsymbol{V}_k^T$ is the pseudo inverse of $\boldsymbol{V}_k$ and $\widetilde{\boldsymbol{W}}_k$ is defined as $\boldsymbol{W}_k - \boldsymbol{M}^{-1}\boldsymbol{V}_k$.

We focus on two classes of methods to be applied in a transient setting, that either use $\boldsymbol{M}_{prev}^{-1} := \boldsymbol{0}$ (Least-Squares) or, minimize the distance to the approximation from the previous time step $n-1$, i.e., $\boldsymbol{M}_{prev}^{-1} := \boldsymbol{M}_{prev}^{-1,(n-1)}$ (Multi-Vector Update). This leads to different strategies for choosing the amount η of secant information, retained from the past.

Least Squares (LS). The ***least-squares approach*** typically re-uses $\eta = k$, i.e., all current time-step information or even $\eta = k + \sum_{q=n-\xi}^{n-1} k_q$ with $\xi \in \{0, \ldots, n\}$ difference vectors from the past to build the matrices $\boldsymbol{V}_k^\eta$ and $\boldsymbol{W}_k^\eta$ in (2.7) and chooses $\boldsymbol{M}_{prev}^{-1} := \boldsymbol{0}$ as initial guess for the approximate inverse Jacobian. This results in the simplified approximation formula

$$\boldsymbol{M}^{-1} = \boldsymbol{W}_k(\boldsymbol{V}_k^T\boldsymbol{V}_k)^{-1}\boldsymbol{V}_k^T = \boldsymbol{W}_k\boldsymbol{V}_k^\dagger \tag{2.18}$$

for the inverse system Jacobian. As $\boldsymbol{M}_{prev}^{-1} = \boldsymbol{0}$, the approximation solely builds upon ***explicitly retained*** input/output vector histories that are represented in the multi-secant equation and incorporated into the Jacobian approximation in an at-once rank-η "update" of the zero matrix. The quality of $\boldsymbol{M}^{-1}$ and therefore the convergence properties of the method are very sensitive with respect to the amount η of retained past information; an optimal choice for η, however, is highly problem dependent and ***tuning of the parameter*** involves a costly trial-and-error processes. For an increasing number of columns in (2.7) (i.e., higher choice for η and ξ, respectively) the conditioning of the multi-secant least-squares problem worsens due to nearly linearly dependent information, and $\boldsymbol{V}_k^\eta$ can even become rank deficient. Elimination of linear-dependent or outdated columns (***filtering***) to maintain acceptable conditioning becomes necessary; different strategies are detailed below.

The advantage of $\boldsymbol{M}_{prev}^{-1} = \boldsymbol{0}$ is that it allows for a ***matrix-free*** realization of the method by providing a Jacobian matvec routine, computing the action $\boldsymbol{W}_k\boldsymbol{V}_k^\dagger(-\boldsymbol{r}^k) = -\boldsymbol{W}_k\boldsymbol{\alpha}$ of $\boldsymbol{M}^{-1}$ on the residual vector $-\boldsymbol{r}^k$. The vector $\boldsymbol{\alpha} = \boldsymbol{V}_k^\dagger\boldsymbol{r}^k$ is computed from solving $\boldsymbol{R}\boldsymbol{\alpha} = -\boldsymbol{Q}^T\boldsymbol{r}^k$ based on a QR-decomposition of $\boldsymbol{V}_k$, $\boldsymbol{V}_k = \boldsymbol{Q}\boldsymbol{R}$, which yields the Jacobian matvec or Newton update on $\boldsymbol{x}^k$ as

$\Delta \boldsymbol{x} = \boldsymbol{W}_k \boldsymbol{\alpha}$. This approach is equivalent to solving the least-squares problem

$$\boldsymbol{\alpha} = \operatorname{argmin}_{\boldsymbol{\beta} \in \mathbf{R}^\eta} \|\boldsymbol{V}_k \boldsymbol{\beta} + \boldsymbol{r}^k\|_2 \quad \rightsquigarrow \quad \Delta \boldsymbol{x}^k = \boldsymbol{W}_k \boldsymbol{\alpha} \tag{2.19}$$

with the reasoning

$$\boldsymbol{M}^{-1}(\boldsymbol{x}^{k+1} - \boldsymbol{x}^k) = \boldsymbol{M}^{-1}\boldsymbol{W}_k \boldsymbol{\alpha} = \boldsymbol{M}^{-1} \sum_{i=1}^{\eta} \boldsymbol{\alpha}_i \Delta \boldsymbol{x}^k_{k-i} \doteq \sum_{i=1}^{\eta} \boldsymbol{\alpha}_i \Delta \boldsymbol{r}^k_{k-i} = \boldsymbol{V}_k \boldsymbol{\alpha} \approx -\boldsymbol{r}^k$$

which is used, e.g., in [Hae09b; Hae09a] to derive the same method. More details can be found in [Sch15; Uek13b]. A high-level pseudo-code implementation of the method is listed in Alg. 2.1.

Multi-Vector Update (MV). The ***multi-vector update approach implicitly retains*** past information through the norm minimization condition (2.9) by searching for an approximation $\boldsymbol{M}^{-1}$ that fulfills the multi-secant equation (2.7) while staying close to the estimation $\boldsymbol{M}^{-1}_{prev} = \boldsymbol{M}^{-1,(n-1)}_{prev}$ from the previous time step. This implicit conservation of information about the estimated operator $\boldsymbol{M}^{-1}$ allows to choose a small number $\eta = k$ of explicitly retained difference vectors in (2.7) for the rank-η update (2.17) for $\boldsymbol{M}^{-1}$. The latter reads

$$\boldsymbol{M}^{-1} = \boldsymbol{M}^{-1,(n-1)}_{prev} + (\boldsymbol{W}_k - \boldsymbol{M}^{-1,(n-1)}_{prev} \boldsymbol{V}_k)(\boldsymbol{V}_k^T \boldsymbol{V}_k)^{-1} \boldsymbol{V}_k^T = \boldsymbol{M}^{-1,(n-1)}_{prev} + \widetilde{\boldsymbol{W}}_k \boldsymbol{V}_k^\dagger \tag{2.20}$$

As a result, the tuning of the parameter η is rendered redundant and filtering of the matrices $\boldsymbol{V}_k^\eta$ and $\boldsymbol{W}_k^\eta$ is no longer strictly required[6].

For stability and robustness reasons, a QR-decomposition of $\boldsymbol{V}_k$ is used to compute the pseudo-inverse $\boldsymbol{V}_k^\dagger$, analogously to the LS method above. Unlike before, the MV method does not allow for a straightforward matrix-free realization as the Jacobian approximation $\boldsymbol{M}^{-1,(n-1)}_{prev}$ from the previous time step needs to be stored explicitly. This can be seen from the update formula (2.20). Explicit computation and storage of $\boldsymbol{M}^{-1}$ (or $\boldsymbol{M}^{-1,(n-1)}_{prev}$) implies $\mathcal{O}(N^2)$ memory requirement and computational complexity, rendering the method impractical or infeasible for large-scale simulations. Alg. 2.1 lists the algorithmical steps of the MV method.

Discussion. Multi-secant quasi-Newton methods [And65] have been widely used in various application fields, predominantly in electronic structure computations [Ni09] and computational quantum mechanics computations [Roh11; Pul80] in the chemistry community. Due to the black-box nature (only input/output information of an operator required) of multi-secant methods and their favorable properties for many different applications, methods within this solver class recently regained increased interest in numerous application fields. Examples include computational statistics [Wal11; Hen13a; Hen13b], groundwater flow problems [Ngu15], coupling of partitioned multi-physics (FSI) simulations [Gan13; Deg09; Deg08b; Deg10; Blo15b; Lin15; Sch15], and dislocation dynamics [Gar15], to only name a few. In particular when applied to problems that do not allow access to an analytical or exact Jacobian, multi-secant methods have been shown to outperform classical inexact Newton methods [Lot12; Wal11]. Over the past 50 years, various communities

[6] Re-using $\eta \leqslant k$ secant vector pairs from previous iterations of the current time step reduces the risk of linearly dependent information in (2.7) to a minimum. Our numerical analysis[Sch15; Hae15; Sch17] (see also Chapter 4.2) shows that in most cases, the MV method yields good convergence rates and maintains good conditioning of $\boldsymbol{V}_k^\eta$ without extensive filtering. Outdated and contradicting information in (2.7) is automatically corrected by minimizing the distance to the previous estimation $\boldsymbol{M}^{-1,(n-1)}_{prev}$ and linearly dependent columns within one time step only occur for near-convergence (or stalling) of the method. Further, the cost for filtering of the matrices $\boldsymbol{V}_k^\eta$ and $\boldsymbol{W}_k^\eta$ scales with η ($\mathcal{O}(N\eta^2) + \mathcal{O}(\eta^4)$) and, thus, remains cheap for small η.

```
func Least-Squares($x^0, \eta$)
  $\tilde{x}^0 = \mathcal{H}(x^0)$, and $r^0 = \mathcal{R}(x^0) = \tilde{x}^0 - x^0$
  $x^1 = x^0 + \omega r^0$
  for $k = 1, 2, \ldots$ do
    $\tilde{x}^k = \mathcal{H}(x^k)$, and $r^k = \mathcal{R}(x^k) = \tilde{x}^k - x^k$
    if converged then
      break
    $W_k^\eta = [\Delta\tilde{x}_{k-\eta}^k, \ldots, \Delta\tilde{x}_{k-1}^k]$, $\Delta\tilde{x}_i^k = \tilde{x}^k - \tilde{x}^i$
    $V_k^\eta = [\Delta r_{k-\eta}^k, \ldots, \Delta r_{k-1}^k]$, $\Delta r_i^k = r^k - r^i$
    compute $V_k^\eta = QR$ and apply filter
    solve $R\alpha = -Q^T r^k$, i.e., $\alpha = -V_k^\dagger r^k$
    $\Delta x^k = W_k^\eta \alpha$
    $x^{k+1} = x^k + \Delta x^k$
```

```
func Multi-Vector($x^0, \eta, M_{prev}^{-1} = M_{prev}^{-1,(n-1)}$)
  $\tilde{x}^0 = \mathcal{H}(x^0)$, and $r^0 = \mathcal{R}(x^0) = \tilde{x}^0 - x^0$
  $x^1 = x^0 + \omega r^0$
  for $k = 1, 2, \ldots$ do
    $\tilde{x}^k = \mathcal{H}(x^k)$, and $r^k = \mathcal{R}(x^k) = \tilde{x}^k - x^k$
    if converged then
      break
    $W_k^\eta = [\Delta\tilde{x}_{k-\eta}^k, \ldots, \Delta\tilde{x}_{k-1}^k]$, $\Delta\tilde{x}_i^k = \tilde{x}^k - \tilde{x}^i$
    $V_k^\eta = [\Delta r_{k-\eta}^k, \ldots, \Delta r_{k-1}^k]$, $\Delta r_i^k = r^k - r^i$
    compute $V_k^\eta = QR$ (and apply filter)
    solve $RV_k^\dagger = Q^T$ for $V_k^\dagger \in \mathbb{R}^{\eta \times N}$
    $\widetilde{W} = (W_k - M_{prev}^{-1} V_k)$
    $M^{-1} = M_{prev}^{-1} + \widetilde{W} V_k^\dagger$
    $\Delta x^k = -M^{-1} r^k$
    $x^{k+1} = x^k + \Delta x^k$
```

ALGORITHM 2.1 *Multi-secant variants Least-Squares (LS) (left) and Multi-Vector Update (MV) (right) in pseudo-code. Input to the methods is an initial guess x^0 and the amount η of retained secant information from the past. The MV method requires $M_{prev}^{-1,(n-1)}$ from the previous time step.*

independently reinvented and further developed variations of multi-secant methods. We give a brief overview of the most important literature, classify existing methods, list names and notations, and integrate the described instantiations LS and MV within this context. An exhaustive literature review can be found in [Uek16; Sch15].

Anderson Acceleration (AA). The pioneering work [And65] of Donald G. Anderson dates back to 1965 and is the first formulation of a multi-secant method, designed and developed to accelerate (potentially slow convergence of) fixed-point iterations or stabilize non-converging fixed-point iterations. This original formulation of the Anderson acceleration method[7] is equivalent[8] to the above presented LS method. Further mathematically equivalent formulations have been derived independently by others in various disciplines for specific applications such as by Carlson and Miller [Car98; Mil05] in method-of-lines applications, by Oosterlee and Washio [Oos00; Was97] for accelerating non-linear multigrid methods and by Degroote [Deg09; Deg10] in the context of implicit coupling of partitioned FSI simulations. Consequently, a multitude of names is used. In the language of the chemistry community, terms like mixing, Anderson mixing [Ni09], Pulay mixing [Pul80], or *direct inversion in the iterative sub-space (DIIS)* [Roh11] are used. The method is also referred to as non-linear GMRES [Was97]. In the FSI context, Anderson acceleration is known as *interface quasi-Newton inverse least-squares* (IQN-ILS) or inverse least-squares (ILS) method [Deg09]; [Hae10]. Based on the idea of an interface-Newton-Krylov or interface-GMRES strategy for the coupling of partitioned FSI by Michler et al. [Mic04; Mic11], Vierendeels et al. [Vie07] formulate a first block-iterative method (IBQN-LS) in the spirit of AA (with approximation of the Jacobian as opposed to its inverse). In 2009, Degroote et al. [Deg09]; [Deg10] proposed the above mentioned (IQN-ILS) which is identical to AA.

Recent theoretical work by Fang and Saad [Fan08], Walker and Ni [Wal11] and Toth and Kelley [Tot15] revived the method's attention and lay the foundation for a rigorous classification and

[7] Anderson referred to the method as extrapolation algorithm.

[8] If the mixing parameters $\beta^k > 0$ in Anderson's formulation are set to 0; this does not limit the method's properties or worsens its approximation quality.

theoretical assessment. In the same vein, Anderson commented on re-invention, recent developments and theoretical aspects of his originally proposed method [And17].

Broyden (B). Broyden's method (Charles G. Broyden; [Bro65]) describes rank-1 updates of an approximation $\boldsymbol{M}_{prev}^{-1}$ of the inverse Jacobian from the previous iteration. In other words, it only uses secant information from a single iteration in the secant equation (2.7), i.e., $\eta = 1$ and $\boldsymbol{M}_{prev}^{-1} = \boldsymbol{M}_{prev}^{-1,(k-1)}$ in (2.9). This results in an iterative improvement of the inverse Jacobian approximation according to the update formula

$$\boldsymbol{M}^{-1,(k)} = \boldsymbol{M}_{prev}^{-1,(k-1)} + \frac{(\boldsymbol{W}_k - \boldsymbol{M}_{prev}^{-1,(k-1)}\boldsymbol{V}_k)\boldsymbol{V}_k^T}{\boldsymbol{V}_k^T\boldsymbol{V}_k}, \tag{2.21}$$

with the initial guess set to zero, i.e., $\boldsymbol{M}^{-1,(0)} = \boldsymbol{0}$. This method is known as ***bad Broyden's method*** as opposed to the original ***good Broyden's method***, where norm minimization and secant equation are formulated in terms of the Jacobian matrix instead of its inverse[9].

As opposed to the Anderson acceleration/extrapolation algorithm, Broyden's method originates from the classical quasi-Newton point of view. Although here we present Broyden's method as a special case of the generalized multi-secant template (2.9), it in fact constitutes pioneering work in the class of methods minimizing the change to a previous estimation of the Jacobian. Historically, the generic formulation (2.9) originates from a generalization of the rank-1 update (2.21) to a rank-η update (2.17), incorporating secant information from η previous iterations and minimizing the change by the update in the Frobenius norm across these η iterations, i.e., it minimizes the distance to an estimation obtained at iteration $k - \eta$. In a transient setting (e.g., fluid-structure interaction simulations), the natural choice for η is the number of iterations $k = k_n$ performed in the current time step n. Hence, the update distance is with respect to the estimation $\boldsymbol{M}_{prev}^{-1,(n-1)}$ from the previous time step. This is identical to what we earlier described as multi-vector update method (MV). The MV strategy has been shown to outperform Broyden's method for systems with slow dynamics and no erratic changes[10], such as non-turbulent flow problems [Bog14; Sch15; Lin15]. In earlier work [Sch15], I studied MV variants based on a multi-secant equation that spans several time steps, as well as one that only spans a fixed, limited amount of iterations, without efficiency benefits. The natural partitioning into sections of iterations per time step outperforms other choices.

Bogaers et al [Bog14] first applied the MV method (here referred to as MVQN) in combination with a block-iterative scheme on the sub-components to couple partitioned FSI simulations. In previous work [Sch15; Lin15; Blo15b], we generalized the method to be used in combination with the more efficient (GS) and (J) system with the potential to exploit massively parallel execution.

Successive Rank-1 Updates (Sr1). Obviously, varying the minimization distance to the previous inverse Jacobian estimate strongly affects the approximation quality. Another interesting question is if an all-at-once rank-η update (as e.g. performed for MV) differs from performing η successively rank-one updates (such as done for LBFGS, see §2.3.1) while keeping the minimization distance fixed. In [Hae09a] Healterman shows that the LS inverse Jacobian update (2.18) can be written as a sequence of rank-one updates

$$\boldsymbol{M}^{-1,(k+1)} - \boldsymbol{M}^{-1,(k)} = \boldsymbol{A}_M\boldsymbol{V}_{k+1}\boldsymbol{V}_{k+1}^{\dagger} - \boldsymbol{A}_M\boldsymbol{V}_k\boldsymbol{V}_k^{\dagger} = \boldsymbol{A}_M\boldsymbol{\ell}_{k+1}\boldsymbol{\ell}_{k+1}^T \tag{2.22}$$

[9] The presented bad method not only exhibits better stability, but also renders the costly linear system solve redundant.

[10] The vast majority of systems modeling physical behavior are characterized by some extent of inertia or slow-acting state changes. Generally speaking, nature does not jump.

using a linear mapping $\boldsymbol{A}_M$ that fulfills $\boldsymbol{A}_M \boldsymbol{V}_{k+1} = \boldsymbol{W}_{k+1}$ and an orthonormalized representation $\boldsymbol{L}_{k+1}$ of the columns of $\boldsymbol{V}_{k+1}$. Here, the initial guess is set to zero, i.e., $\boldsymbol{M}^{-1,(0)} := \boldsymbol{0}$ and $\boldsymbol{\ell}_{k+1}$ indicates the last column of $\boldsymbol{L}_{k+1}$. The vectors $\boldsymbol{\ell}_{k+1}$ and $\boldsymbol{A}_M \boldsymbol{\ell}_{k+1}$ required in the least-squares (LS) rank-one update (2.22) can be computed from

$$\boldsymbol{\ell}_{k+1} \quad = \quad \frac{(\boldsymbol{I} - \boldsymbol{L}_k \boldsymbol{L}_k^T)\Delta \boldsymbol{r}_k^{k+1}}{\|(\boldsymbol{I} - \boldsymbol{L}_k \boldsymbol{L}_k^T)\Delta \boldsymbol{r}_k^{k+1}\|} \quad \text{and} \quad \boldsymbol{A}_M \boldsymbol{\ell}_{k+1} = \frac{(\boldsymbol{M}^{-1,(k)} - \boldsymbol{I})\boldsymbol{r}^{k+1}}{\|(\boldsymbol{I} - \boldsymbol{L}_k \boldsymbol{L}_k^T)\Delta \boldsymbol{r}_k^{k+1}\|}.$$

without knowing the actual mapping $\boldsymbol{A}_M$. Generalizing (2.22) to the multi-vector update (MV) formula yields

$$\boldsymbol{M}^{-1,(k+1)} - \boldsymbol{M}^{-1,(k)} = (\boldsymbol{A}_M - \boldsymbol{M}_{prev}^{-1})\boldsymbol{\ell}_{k+1}\boldsymbol{\ell}_{k+1}^T. \tag{2.23}$$

Here, instead of zero, an initial guess for $\boldsymbol{M}^{-1}$, using $\boldsymbol{M}_{prev}^{-1,(n-1)}$ is employed, i.e., $\boldsymbol{M}^{-1,(0)} := \boldsymbol{M}_{prev}^{-1,(n-1)}$. This exact representation of (2.20) written in terms of successive rank-one updates can be used to compare MV to rank-one update formulations. A prominent example is the limited memory variant of the BFGS update formula (see §2.3.1 for details). An improved least-squares method with implicit old time step reuse, following the same general idea as LBFGS of performing η successive rank-one updates to a Jacobian estimation from iteration $k - \eta$, has been proposed by Haelterman [Hae09a]. In other words, it uses the least squares method in rank-one update formulation (2.22) with $\boldsymbol{M}^{-1,(0)} = \boldsymbol{M}_{prev}^{-1,(n-1)}$. We find that this method is similar, but not identical to (2.23) as it lacks the term $\boldsymbol{M}_{prev}^{-1}\boldsymbol{\ell}_{k+1}\boldsymbol{\ell}_{k+1}^T$ in the update.

Type I/Type II Classification. Fang and Saad [Fan08] view multi-secant methods in a more generalized way in the sense of (2.9) and distinguish between ***type I*** (corresponds to "good" in Broyden's terminology) and ***type II*** (corresponds to "bad" in Broyden's terminology) ***generalized-Broyden*** methods. While type II methods build an approximation of the inverse Jacobian matrix by minimizing and incorporating secant information in terms of the inverse Jacobian as in (2.9), type I methods do so in terms of the Jacobian, resulting in an approximation of the Jacobian and not its inverse. All type I expressions for the approximation of the Jacobian may be explicitly transformed to type II approximations of the inverse Jacobian upon application of the Sherman-Morrison-Woodbury formula [Woo50] and vice versa. For clarity, we refer to the methods as `<name>-<type>`, e.g., LS-II for the above presented LS method. Besides refraining from the linear system solve and reducing the computational effort per Newton iteration, type II methods have furthermore been found to be more stable [Sch15].

The MV-II variant differs from the generalized-Broyden type II variant presented in [Fan08]. The latter considers an update directly from $\boldsymbol{x}^k$ to $\boldsymbol{x}^{k+1}$ and formulates the update in terms of difference vectors $\Delta \boldsymbol{x}_i^k = \boldsymbol{x}^k - \boldsymbol{x}^i$ in $\boldsymbol{W}_k$ as opposed to going from $\tilde{\boldsymbol{x}}^k$ to $\boldsymbol{x}^{k+1}$ and using $\Delta \tilde{\boldsymbol{x}}_i^k = \tilde{\boldsymbol{x}}^k - \tilde{\boldsymbol{x}}^i$ as required in the FSI fixed-point context (cf. preliminary remarks). A lucid overview of the different resulting methods can be found in section 3.2 of [Uek16]. This difference is delicate. Generalized-Broyden methods are sensitive to the initial guess for $\boldsymbol{M}_{prev}^{-1}$. For applications in partitioned FSI, an informed initial guess of the fixed-point operator Jacobian, or the Jacobians of the sub-component solvers are not accessible and $\boldsymbol{M}_{prev}^{-1}$ is set to $\boldsymbol{0}$. Hence, MV-II falls back to LS-II in the first iteration, which, for an update $\boldsymbol{x}^k \rightsquigarrow \boldsymbol{x}^{k+1}$ leads to stagnation of the search space at dimension 2 and stall of convergence (cf. preliminary remarks). The proposed variants in Fang and Saad [Fan08] are only applicable if involved application specific heuristics or estimations for an effective initial guess of the (inverse)

Jacobian are available, such as done by Marks and Luke in [Mar08]. A survey on the performance of different variations of generalized Broyden (type I and type II) methods as used in partitioned FSI simulations can be found in [Hae].

Walker and Ni [Wal11] use the same classification but always fix the previous Jacobian approximation (or initial guess) to zero, such that the herein presented type II method is identical to the above presented LS method or AA-II.

Convergence Considerations. The literature addressing rigorous theoretical analysis and convergence considerations (with proven convergence rates) for multi-secant methods is negligible and its outcome, as of now, is of limited practical value. Various groups showed in independent work equivalence of the LS-II (or AA-II) method to GMRES in the linear case, under the assumption of "unlimited" storage[11], i.e., $\eta \geqslant N$; see, e.g., Walker and Ni [Wal11] and Rohwedder and Schneider [Roh11] in 2011 for LS-II and Haeltermann [Hae10] in 2010 for LS-I. These results are of poor practical relevance, as typically a small number $\eta \ll N$ is used and necessary for efficiency reasons in application. In 2015, Toth and Kelley [Tot15] show general convergence results for the LS-II method as implemented in practice and focus on a proof of acceleration rather than convergence. The results are improved and generalized to inaccurate function evaluations in [Tot17]. Given that the fixed-point map $\boldsymbol{\mathcal{H}}$ is a contraction and the coefficients stay bounded, the method is locally r-linearly convergent[12]. In other words, the authors show that LS-II does no worse than fixed-point iteration in terms of convergence; however, locally (for $\boldsymbol{x}^k$ not too far from the fixed-point $\boldsymbol{x}^\star$) rapid convergence can be shown for LS-II, if the underlying fixed-point iteration converges. The local part and the assumption that the fixed-point iterations converge is substantial and not given for most applications in FSI and computational chemistry. The authors conclude that for a large group of problems, LS-II is much better than plain fixed-point iteration and outperforms Newton-GMRES. However, the performance depends on the physics (i.e., the application problem). Using different norms, i.e., ℓ_1 or ℓ_∞ instead of ℓ_2 in the least-squares problem (2.19) shows no benefit. A comparison of LS and exact Newton can be found in [Lot12; Wal11]. If the Jacobian in Newton's method is obtained from a finite difference approximation, LS outperforms Newton; for an exact Jacobian, Newton's method exhibits much faster convergence, but at higher computational cost per iteration.

Improving Convergence and Robustness. To achieve high efficiency, quasi-Newton typically requires careful parameter choices and additional measures such as filtering. In the following, we accordingly propose techniques that accelerate convergence and increase the robustness and stability of multi-secant methods.

Improving Convergence: Recycle. In our transient, coupled FSI case, the underlying system fixed-point operator, and thus, the approximated inverse Jacobian $\boldsymbol{M}^{-1}$ changes only moderately[13] information from previous time steps can be retained to improve the quality of $\boldsymbol{M}^{-1}$ and, thus, convergence.

We formalize this concept by enhancing the multi-secant equation (2.7) with columns from

[11] In this case, the LS method looks like a Krylov method and iteration histories of GMRES and LS can be constructed from each other.

[12] In other words, $\exists c \in [0,1)$ and $M > 0$ such that if $\boldsymbol{x}^0$ is not too far from the fixed-point $\boldsymbol{x}^\star$, then

$$\|\boldsymbol{x}^k - \boldsymbol{x}^\star\| \leqslant Mc^k \|\boldsymbol{x}^0 - \boldsymbol{x}^\star\|$$

[13] This has been found to hold true for the application problems considered in this thesis. In general, consecutive fixed-point operators of this kind may differ widely, in particular for erratic systems. The strategy has been showed to be very effective for non-turbulent FSI problems [Deg13a; Deg08a; Deg09; Uek16; Sch15; Lin15].

previous time levels, i.e.,

$$\boldsymbol{V}_k^{\eta,n} = [\boldsymbol{V}_k^{k,n},\, \boldsymbol{V}_{k_{n-1}}^{k_{n-1},n-1},\, \dots,\, \boldsymbol{V}_{k_{n-\xi}}^{k_{n-\xi},n-\xi}] \quad \text{and} \tag{2.24a}$$

$$\boldsymbol{W}_k^{\eta,n} = [\boldsymbol{W}_k^{k,n},\, \boldsymbol{W}_{k_{n-1}}^{k_{n-1},n-1},\, \dots,\, \boldsymbol{W}_{k_{n-\xi}}^{k_{n-\xi},n-\xi}], \tag{2.24b}$$

where $\boldsymbol{V}_k^{k,n}$ and $\boldsymbol{W}_k^{k,n}$ are constructed at the current time level n and iteration k, and

$$\boldsymbol{V}_{k_{n-i}}^{k_{n-i},n-i} \quad \text{and} \quad \boldsymbol{W}_{k_{n-i}}^{k_{n-i},n-i}, \quad i = 1, \cdots, \xi$$

are the input and output matrices from the converged time step $n - i$ and $\xi \in \{0, \dots, n\}$ determines the number of explicitly retained time levels. $\eta = \eta(\xi) = k + \sum_{q=n-\xi}^{n-1} k_q$ as used earlier, indicates the total number of columns in $\boldsymbol{V}_k^{\eta,n}$ and $\boldsymbol{W}_k^{\eta,n}$, respectively. We denote the LS method with explicit reuse of ξ time steps as LS(ξ). For the sake of readability, we omit the explicit notation with the superscripts η and n and write $\boldsymbol{V}_k$ and $\boldsymbol{W}_k$, respectively, unless explicit indication of reused time levels is necessary. Later, in the algorithm description, also the matrices at converged time steps $\boldsymbol{V}_{k_{n-i}}^{k_{n-i},n-i}$ and $\boldsymbol{W}_{k_{n-i}}^{k_{n-i},n-i}$ are shortly written as $\boldsymbol{V}_{k_i}$ and $\boldsymbol{W}_{k_i}$. Note, that the recycled data from past converged time steps is inconsistent as the underlying fixed-point operator changes; the employed multi-secant method, therefore, needs to be capable of dealing with this moderate amount of noise.

MV ***implicitly*** reuses past information through norm minimization imposing minimal change of the update with respect to the previous time step estimation of the Jacobian. This may be combined with explicit reuse. Our numerical tests [Sch15], however, showed no benefit from an additional explicit reuse of information beyond the current time step for the MV method such that we do not further consider MV(ξ) methods for $\xi > 0$. The LS method, on the other hand, solely relies upon explicit reuse in (2.7) and the convergence properties can be significantly improved by enlarging the space spanned by the input vectors[14] [Sch15; Lin15; Deg09; Uek16].

Convergence of the LS method is quite sensitive to the amount of recycled information from the past. In practice, therefore, the parameter $\eta(\xi)$ highly depends upon the application problem at hand with its underlying physics and characteristics and needs to be identified in a costly offline tuning phase. A non-optimal choice may cause stagnation or lead to numerical breakdown due to stability issues. Numerical results in §4.2 point this out for different application cases. The next section offers an attempt to remedy this problem through filtering.

Implicit reuse, on the other hand, resolves this problem automatically through the combination of norm minimization and explicit data from the current time step: Already well resolved features retain and linearly dependent or even contradicting information is implicitly overwritten. This renders the tuning parameter $\eta(\xi)$ obsolete, but comes at the expense of requiring an explicit representation of the Jacobian.

Improving Robustness: Filtering. Nearly linearly dependent columns in $\boldsymbol{V}_k$ both render the calculation of $\boldsymbol{V}_k^\dagger = (\boldsymbol{V}_k^T \boldsymbol{V}_k)^{-1} \boldsymbol{V}_k^T$ costly and ill-conditioned, but also lead to low quality of $\boldsymbol{M}^{-1}$ as it typically is an indication for outdated information in $\boldsymbol{V}_k$ and $\boldsymbol{W}_k$. Linear dependencies can be caused by the non-linearity of the mapping $\boldsymbol{\mathcal{H}}$, rounding errors, convergence towards a stationary solution of the transient problem, or simply by too many reused history vectors η, the column vectors in $\boldsymbol{V}_k^\eta$. As a result, we observe stagnation or numerical breakdown; acceptable conditioning of $\boldsymbol{V}_k^\eta$ is

[14] Notice, that generating further input/output pairs within the current time step involves a full time step execution in every participating sub-solver. This is equivalent to the cost of a coupling iteration and prohibitively expensive. Recycling existing input/output pairs from slightly different fixed-point operators (at a different time instance) is therefore a cheap and promising alternative.

absolutely essential for convergence.

In an attempt to alleviate linear dependencies and obliterate contradicting information, we consider so called ***filtering*** techniques that ought to identify and eliminate potentially dangerous columns in $\boldsymbol{V}_k$ and, accordingly, also in $\boldsymbol{W}_k$, and thereby maintain good conditioning of the QR-decomposition and robustness of the quasi-Newton method. We describe two filtering methods ***QR1*** and ***QR2*** based on a modification of the QR-decomposition and a third method that filters bad information via truncation of a proper orthogonal decomposition (***POD***) of $(\boldsymbol{V}_k^\eta)^T\boldsymbol{V}_k^\eta$. The considered methods have been proposed in our earlier work [Hae15], where in a joint effort[15] we extensively studied and compared the filtering techniques applied to LS and MV for different FSI application scenarios and different single-physics solvers. Further thorough numerical analysis for a multitude of different single-physics solvers and FSI test cases in combination with LS and MV can be found in [Uek16]. Within this work, we are primarily interested in analyzing the method's potential in the face of extensive reuse, i.e., for LS(ξ) with very large ξ and forego a revision of an extensive analysis (the interested reader is referred to [Hae15; Uek16]). Results for QR1 and QR2 in combination with advanced variants of LS(ξ) and MV are given in §4.2 and have been earlier published in [Sch17]. Within this work, the stabilizing filter techniques have been integrated into the parallel multi-physics coupling tool preCICE.

We focus on the filters QR1 and QR2, that, based on (a modified) QR-decomposition of $\boldsymbol{V}_k^\eta$, indicate and eliminate (nearly) linearly dependent columns. The QR1 approach is widely used in the FSI community to stabilize LS and has been first introduced in [Deg09]. Given an economy size QR decomposition $\boldsymbol{QR} = \boldsymbol{V}_k^\eta$ of $\boldsymbol{V}_k$[16], QR1 monitors the condition of $\boldsymbol{R}$ (and thereby $\boldsymbol{V}_k$) and eliminates the oldest column i for which

$$\boldsymbol{R}_{ii} < \varepsilon_F \quad \text{or} \quad \boldsymbol{R}_{ii} < \varepsilon_F \|\boldsymbol{R}\|_2 \tag{2.25}$$

from $\boldsymbol{V}_k$ and consistently also in $\boldsymbol{W}_k$. If elimination occurs, the QR-decomposition is discarded and re-computed for the reduced $\boldsymbol{V}_k$. The QR2 technique, on the other hand, compares and judges columns already during the construction of the QR-decomposition with respect to their amount of new information incorporated to the least-squares system. The latter is measured by the norm of the orthogonalized column in $\boldsymbol{Q}$ (before normalization). Alg. 2.2 lists algorithmic details. Whereas QR1 is motivated from a numerical point of view[17], QR2 is purely algebraic. Consequently, QR1 would filter out a (potentially linearly independent) column just because it is small compared to other columns. As a third approach, we mention POD [Bog12; Hae15; Sir87], which eliminates dangerous information based on a truncated proper orthogonal decomposition of $\boldsymbol{V}_k^T\boldsymbol{V}_k$. Algorithmic steps are given in Alg. 2.2. Our analysis in [Hae15] showed no advantage of POD over the QR variants, which alongside with a more involved numerical implementation is the reason why we do not deepen its analysis here.

For the QR2 filter, it is crucial to start the Gram-Schmidt orthogonalization with the most recent information, as it probably reflects the dynamics of the current time step/iterate better than older ones. This in turn requires a costly re-computation of the QR-decomposition in every quasi-Newton iteration. While the implementation used in [Hae15] follows the same strategy for QR1, we decided to employ a very efficient QR-updating scheme based on Givens rotations in combination with QR1. This strategy is prone to filter out most recent information first, which contradicts the idea

[15] R. Haeltermann, A.E.J Bogaers, B. Uekermann, M. Mehl and myself

[16] To enhance readability, we drop the superscript.

[17] The filter threshold ε_F depends on the accuracy of the single-physics solvers and may be determined by analyzing propagation of small input perturbations by the operator map $\boldsymbol{\mathcal{H}}$.

```
func mGS-QR2-Filter($V_k \in \mathbf{R}^{N\times\eta}$, $\varepsilon_F$)
  Output: filtered $\breve{V}_k, \breve{W}_k, \breve{Q} \in \mathbf{R}^{N\times\breve{\eta}}$
               $\breve{R} \in \mathbf{R}^{\breve{\eta}\times\breve{\eta}}$ s.t.  $\breve{V}_k = \breve{Q}\breve{R}$
  set $\breve{R}_{11} = \|V_{k\cdot,1}\|_2$ and $\breve{Q}_{\cdot,1} = V_{k\cdot,1}/R_{11}$
  for $i = 1,2,\dots,\eta$ do
    $\bar{v} = V_{k\cdot,i}$
    for $j = 1,2,\dots,i-1$ do
      $R_{ji} = \breve{Q}^T_{\cdot,j} \cdot \bar{v}$
      $\bar{v} = \bar{v} - \breve{R}_{ji} \cdot \breve{Q}_{\cdot,j}$
    if $\|\bar{v}\|_2 < \varepsilon_F \|V_{k\cdot,i}\|_2$ then
      delete column $i$ from $\breve{V}_k, \breve{W}_k$
      restart procedure
    $\breve{R}_{ii} = \|\bar{v}\|_2$ and $\breve{Q}_{\cdot,i} = \bar{v}/\breve{R}_{ii}$
```

```
func POD-Filter($V_k \in \mathbf{R}^{N\times\eta}$, $\varepsilon_F$)
  Output: filtered $\breve{V}_k, \breve{W}_k \in \mathbf{R}^{N\times\breve{\eta}}$
  compute corr. matrix $\Sigma\frac{1}{r}(V_k)^T V_k$
  compute eigenvalues $\lambda_i$ and -vectors $v_i$ of $\Sigma$
         s.t.  $\Sigma\mathcal{X} = \mathcal{X}\Lambda$ with $\mathcal{X} = (v_i)_{i=1}^{\eta}$
         and $\Lambda =$ diag($\lambda_1, \dots, \lambda_\eta$), $\lambda_1 \geqslant \dots \geqslant \lambda_\eta$
  determine $c$ s.t.  $\frac{\lambda_c}{\lambda_1} \leqslant \varepsilon_F < \frac{\lambda_{c-1}}{\lambda_1}$ or $c = \eta$
  compute $\mathcal{V} = V_k\mathcal{X}$ and $\mathcal{W} = W_k\mathcal{X}$
  truncate  $\breve{V}_k := [\mathcal{V}_{\cdot,1}, \dots, \mathcal{V}_{\cdot,c}]$    and
            $\breve{W}_k := [\mathcal{W}_{\cdot,1}, \dots, \mathcal{W}_{\cdot,c}]$
```

ALGORITHM 2.2 *Pseudo-code for the modified Gram-Schmidt QR2 filer (left) and the proper orthogonal decomposition (POD) filter (right). Both algorithms filter $\mathbf{V}_k$ based on a filter/truncation threshold ε_F and modify $\mathbf{V}_k$, $\mathbf{W}_k \in \mathbf{R}^{N\times\eta}$ to $\breve{\mathbf{V}}_k$, $\breve{\mathbf{W}}_k \in \mathbf{R}^{N\times\breve{\eta}}$.*

of eliminating outdated columns and is therefore potentially less stable. In [Uek16], however, Uekermann showed that the difference is negligible for a well chosen threshold parameter ε_F.

Other similar filter techniques are considered in literature [Fan08; Wal11; Mar08] primarily discarding old columns. We like to emphasize that for transient problems, input/output modes from old time steps might be essential for fast convergence and should not be discarded completely (restart typically implies a loss in robustness). Rather, filtering ought to eliminate particularly dangerous columns (e.g., dropping similar iterates from a "thick" recent time step). We apply QR1 and QR2 to LS(ξ) for large ξ and rely on the filtering to ensure robustness and stability. Although it might improve robustness in some cases, for our application problems filtering for MV was not required. The implicit reuse through norm-minimization can be seen as a filter for past information, and secant-information within one time step is better conditioned.

Improving Robustness: Pre-Scaling for the Jacobi-System. Beyond the above mentioned issues, solving the least-squares system (2.19), i.e., factorizing $\mathbf{V}_k = \mathbf{QR}$ to compute $\boldsymbol{\alpha}^k = \mathbf{R}^{-1}\mathbf{Q}^T\mathbf{r}^k$ or $\mathbf{V}_k^\dagger = \mathbf{R}^{-1}\mathbf{Q}^T$, respectively, in combination with the Jacobi-system (J) is prone to be ill-conditioned as the coupling variables $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2)^T$ with $\mathbf{x}_1 \in \mathbf{X}_1$, and $\mathbf{x}_2 \in \mathbf{X}_2$ are likely to live on very different scales. To stabilize the solution, a ***suitable pre-scaling*** of the least-squares system $\mathbf{V}_k{}' := \mathbf{\Lambda}_k\mathbf{V}_k$ and $\mathbf{r}^{k\prime} := \mathbf{\Lambda}_k\mathbf{r}^k$ with $\mathbf{\Lambda}_k =$ diag$(\lambda_1^k, \dots, \lambda_N^k)$ becomes necessary. Hereby, the weights $\lambda_1^k, \dots, \lambda_N^k$ shall be chosen such that entries in x_1 and x_2 are normalized. In a transient setting, this is typically done with respect to the previous time step. Different choices are conceivable (cf. [Uek16]):

(i) per-entry normalization, i.e., $\lambda_1^k = \lambda_i^k = 1/(x_i^{(n-1)} + \varepsilon_{mach})$, $i = 1, \dots, N$,

(ii) per-sub-vector normalization, i.e., $\bar{\lambda}_1^k = \lambda_i^k = 1/\|x_1^{(n-1)}\|_2$, $\bar{\lambda}_2^k = \lambda_{N/2+i}^k = 1/\|x_2^{(n-1)}\|_2$, $i = 1, \dots, N/2$.

In [Uek16], Uekermann showed by thorough numerical analysis that a per-sub-vector normalization yields more robust behavior than a per-entry normalization. Further, a per-sub-vector normalization based on the weighted sum of the residuals from k previous quasi-Newton iterations, i.e.,

$$\bar{\lambda}_1^k = \lambda_i^k = \left(\sum_{j=1}^{k} \frac{\|\boldsymbol{\mathcal{S}}_2(\boldsymbol{x}_1^j) - \boldsymbol{x}_1^j\|_2}{\|\boldsymbol{\mathcal{H}}(\boldsymbol{x}^j) - \boldsymbol{x}^j\|_2} \right)^{-1} \text{ and } \bar{\lambda}_2^k = \lambda_{N/2+i}^k = \left(\sum_{j=1}^{k} \frac{\|\boldsymbol{\mathcal{S}}_1(\boldsymbol{x}_2^j) - \boldsymbol{x}_2^j\|_2}{\|\boldsymbol{\mathcal{H}}(\boldsymbol{x}^j) - \boldsymbol{x}^j\|_2} \right)^{-1}, \; i = 1, \dots, N/2,$$

has been shown to outperform other approaches[18]. We would like to emphasize that this choice for the pre-scaling operator $\boldsymbol{\Lambda}_k$ results in varying scaling weights in every quasi-Newton iteration. As a result, the QR-decomposition of $\boldsymbol{V}_k$ needs to be re-computed from scratch in every iteration (i.e., the columns updating scheme is no longer applicable). To allow for consistent input/output histories, the pre-scaling needs to be reverted after the solution of the least-squares problem (2.19) (i.e., QR-factorization of $\boldsymbol{V}_k$). For pre-scaled objects $\boldsymbol{V}_k{}' := \boldsymbol{\Lambda}_k \boldsymbol{V}_k$, $\boldsymbol{W}_k{}' := \boldsymbol{\Lambda}_k \boldsymbol{W}_k$, $\boldsymbol{V}_k^{\dagger\prime} := \boldsymbol{V}_k^{\dagger} \boldsymbol{\Lambda}_k^{-1} = \boldsymbol{R}^{-1} \boldsymbol{Q}^T \boldsymbol{\Lambda}_k^{-1}$, $\boldsymbol{M}_{prev}^{-1}{}' := \boldsymbol{\Lambda}_k \boldsymbol{M}_{prev}^{-1} \boldsymbol{\Lambda}_k^{-1}$, $\Delta \boldsymbol{x}^{k\prime} := \boldsymbol{\Lambda}_k \Delta \boldsymbol{x}^k$ and $\boldsymbol{r}^{k\prime} := \boldsymbol{\Lambda}_k \boldsymbol{r}^k$, the computation of the MV quasi-Newton update reads

$$\begin{aligned} \Delta \boldsymbol{x}^{k\prime} &= \boldsymbol{M}_{prev}^{-1}{}' \boldsymbol{r}^{k\prime} - (\boldsymbol{W}_k{}' - \boldsymbol{M}_{prev}^{-1}{}' \boldsymbol{V}_k{}') \boldsymbol{V}_k^{\dagger\prime} \boldsymbol{r}^{k\prime} \\ &= \boldsymbol{\Lambda}_k (\boldsymbol{M}_{prev}^{-1} \boldsymbol{r}^k - \widetilde{\boldsymbol{W}}_k (\boldsymbol{R}^{-1} \boldsymbol{Q}^T \boldsymbol{\Lambda}_k^{-1} \boldsymbol{\Lambda}_k) \boldsymbol{r}^k) \end{aligned}$$

Reverting $\Delta \boldsymbol{x}^k$ via $\boldsymbol{\Lambda}_k^{-1} \Delta \boldsymbol{x}^{k\prime}$ can then be simplified to the update

$$\boldsymbol{x}^{k+1} = \boldsymbol{\mathcal{H}}(\boldsymbol{x}^k) + \boldsymbol{M}_{prev}^{-1} \boldsymbol{r}^k - \widetilde{\boldsymbol{W}}_k [\boldsymbol{V}_k^{\dagger\prime} \boldsymbol{\Lambda}_k] \boldsymbol{r}^k.$$

In particular, one can easily verify that the optimization problem (2.16) is invariant under pre-scaling[19]. For our algorithm, this means, that we only have to (*i*) scale $\boldsymbol{V}_k$ to $\boldsymbol{V}_k{}' = \boldsymbol{\Lambda}_k \boldsymbol{V}_k$, (*ii*) calculate the QR-decomposition of $\boldsymbol{V}_k{}'$ by factorizing $\boldsymbol{V}_k{}' = \boldsymbol{Q}' \boldsymbol{R}' = \boldsymbol{\Lambda}_k \boldsymbol{Q} \boldsymbol{R}$, and (*iii*) calculate $\boldsymbol{V}_k^{\dagger\prime} \boldsymbol{\Lambda}_k = \boldsymbol{R}'^{-1} \boldsymbol{Q}'^T \boldsymbol{\Lambda}_k$. As a result, pre-scaling can be computed at negligible added cost. The residual-sum weights, however, contradict the QR-updating scheme and trigger a re-computation from scratch. If pre-scaling is combined with a filter technique, $\boldsymbol{\Lambda}_k \boldsymbol{V}_k$ is computed prior to application of the filter.

Towards Scalable and Robust Black-Box Solvers. We conclude this section by abstracting the characteristics of the introduced multi-secant quasi-Newton methods LS and MV. In particular, we point out major drawbacks, ultimately rendering each of the methods impractical when going to large-scale and massively parallel execution. Drastically increasing computational cost no longer allows for extensive parameter tuning and $\mathcal{O}(N^2)$ memory consumption to store the Jacobian becomes infeasible when going to large-scale simulations. We introduce enhanced methods designed to remedy each method's limitations. A more detailed description of the methods' efficient algorithmic implementation on distributed data is provided in the next section.

[18] Uekermann considers (*i*) value based normalization per-entry and per-sub-vector, (*ii*) constant weighting per-sub-vector, and residual based per-sub-vector normalization using (*iii*) the residuum from the previous iteration only as well as (*iv*) the sum of k previous residuals. More details and thorough numerical analysis can be found in [Uek16], Sec. 3.6.3.

[19] Solving $\|\boldsymbol{M}_{prev}^{-1}{}' - \boldsymbol{M}^{-1\prime}\| \rightarrow \min$ s. t. $\boldsymbol{M}_{prev}^{-1}{}' \boldsymbol{V}_k{}' = \boldsymbol{W}_k{}'$ results in $\lambda^\star = -\widetilde{\boldsymbol{W}}_k' \boldsymbol{V}_k^{\dagger\prime} = -\boldsymbol{\Lambda}_k \widetilde{\boldsymbol{W}}_k \boldsymbol{V}_k^{\dagger} \boldsymbol{\Lambda}_k^{-1} = -(\boldsymbol{\Lambda}_k \boldsymbol{W}_k - \boldsymbol{\Lambda}_k \boldsymbol{M}_{prev}^{-1} \boldsymbol{\Lambda}_k^{-1} \boldsymbol{\Lambda}_k \boldsymbol{V}_k) \boldsymbol{V}_k^{\dagger} \boldsymbol{\Lambda}_k^{-1}$, and $\|\boldsymbol{M}_{prev}^{-1} - \boldsymbol{M}^{-1}\| \rightarrow \min$ s. t. $\boldsymbol{M}_{prev}^{-1} \boldsymbol{V}_k = \boldsymbol{W}_k$ results in $\lambda^\star = -\widetilde{\boldsymbol{W}}_k \boldsymbol{V}_k^{\dagger} = (\boldsymbol{W}_k - \boldsymbol{M}_{prev}^{-1} \boldsymbol{V}_k) \boldsymbol{V}_k^{\dagger}$.

Abstract: **Towards Scalability and Robustness**

*The **Least-Squares (LS)** method (i) uses $\boldsymbol{M}_{prev}^{-1} := \boldsymbol{0}$ and (ii) solely relies upon explicitly retained multi-secant information in $\boldsymbol{V}$ and $\boldsymbol{W}$. This introduces (iii) the tuning parameter η of the amount of re-used secant information from the (transient) convergence history or requires (iv) sophisticated (and costly) filtering/preconditioning for the multi-secant system (discussed in what follows) to maintain acceptable conditioning of the pseudo-inverse $\boldsymbol{V}^{\dagger}$. (v) The method can however efficiently be realized in a matrix-free manner on distributed data.*

*The **Multi-Vector Update (MV)** method (i) implicitly retains past information by using $\boldsymbol{M}_{prev}^{-1} := \boldsymbol{M}_{prev}^{-1,(n-1)}$ and incorporates an update based on (ii) a small number η of explicitly re-used multi-secant information from the current time step. Omitting linearly dependent or contradicting information by limiting η ensures good conditioning of $\boldsymbol{V}_k^{\dagger}$ while implicit reuse in $\boldsymbol{M}_{prev}^{-1}$ still yields good approximation quality; as a result (iii) filtering and (iv) the tuning of the parameter η become obsolete. (v) This comes at the cost of explicitly storing a representation of the system Jacobian matrix $\boldsymbol{M}^{-1}$.*

*We introduce and compare two variations that remedy each method's major drawbacks—(a) the **LS(∞)** method, rendering the **tuning of η dispensable** by reusing all previous information and ensuring acceptable conditioning of $\boldsymbol{V}$ for stability and robustness by applying a powerful filtering technique, and (b) the **Multi-Vector Restart-SVD (MV-RS-SVD)** method, enhancing the MV concept to a **matrix-free** alternative by on-the-fly evaluations in combination with a limited-memory concept, storing $\widetilde{\boldsymbol{W}}q$ and $\boldsymbol{V}^{\dagger}q$ for a limited number q of time steps; the method's memory is periodically reset. To maintain the good MV convergence properties, Jacobian information is retained across memory-resets via a sophisticated SVD-based sub-space tracking method in $\mathcal{O}(N)$ overall complexity. The methods are detailed in the remainder of this chapter.*

	Least-Squares (LS)	***LS(∞)***	***Multi-Vector Update (MV)***	***MV-RS-SVD***
update	*rank-η, $\eta = k + \sum_{q=n-\xi}^{n-1} k_q$*		*rank-η, $\eta = k_n$*	
reuse	*explicitly in $\boldsymbol{V}_k$, $\boldsymbol{W}_k$*		*implicitly in $\boldsymbol{M}_{prev}^{-1}$*	
tuning	***yes**, ξ (for η)*	***no**, $\xi = n$*	*no*	
filtering	*yes, cost $\mathcal{O}(N\eta^2) + \mathcal{O}(\eta^4)$*		*no*	
matrix-free	*yes*		***no**, cost $\mathcal{O}(N^2\eta)$*	***yes**, restart + sub-space-tracking*

2.2.2 Efficient Algorithms for Calculation and Representation of Jacobians

In the previous section, we proposed advanced quasi-Newton concepts that minimize the need for manual tuning of problem-dependent parameters and described techniques that improve robustness and convergence, but also maintain numerical stability. To be practical for real application scenarios, highly efficient and optimized algorithmic realization is an absolute necessity. In particular, real applications typically feature large problem sizes and growing number of unknowns. Consequently, distributed memory algorithms designed for massively parallel execution on high-performance supercomputers become an imperative. In what follows, we describe the efficient algorithmic realization of the main computational building blocks within the two considered multi-secant variants LS and MV and their implementation on distributed data. We first describe the efficient algorithmic realization for the non-matrix-free MV alternative such as stated in §2.2.1, followed by the description of a more sophisticated, matrix-free variant of this method which remedies the $\mathcal{O}(N^2)$ complexity (memory and computational cost) by exploiting a limited memory strategy combined with a periodically sub-space-tracking restart approach (MV-RS-SVD). The presented content is addressed in our previous work [Bun16a]; [Bun16b] (MV/LS) and [Sch17]. Also [Uek16] describes algorithmic realization of MV and LS within the coupling library preCICE; the latter is, however, genuine work attained within this thesis. We keep the algorithmic description short and omit details to avoid needless repetition and refer to the above work for an elaborate description.

Algorithmic Building Blocks. The primary kernel of both quasi-Newton methods is the computation of the pseudo-inverse $\boldsymbol{V}_k^\dagger = (\boldsymbol{V}_k^T \boldsymbol{V}_k)^{-1} \boldsymbol{V}_k^T$ in (2.18) (LS) and (2.20) (MV). To compute the Newton update $\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \Delta \boldsymbol{x}$, it is sufficient to get the action $\boldsymbol{z} = \boldsymbol{V}_k^\dagger \boldsymbol{y}$ of the pseudo-inverse on a vector $\boldsymbol{y}$. As mentioned above, this is equivalent to solving the unconstrained least-squares optimization problem[20]

$$\boldsymbol{z} = \operatorname{argmin}_{\bar{\boldsymbol{z}} \in \mathbf{R}^\eta} \|\boldsymbol{V}_k \bar{\boldsymbol{z}} - \boldsymbol{y}\|_2, \tag{2.26}$$

which by using the right-hand-side $\boldsymbol{y} := -\boldsymbol{r}^k$ and $\boldsymbol{\alpha} := \boldsymbol{z}$ replaces

$$\begin{array}{lll} \boldsymbol{x}^{k+1} = \boldsymbol{\mathcal{H}}(\boldsymbol{x}^k) - \boldsymbol{W}_k \boldsymbol{V}_k^\dagger \boldsymbol{r}^k & \text{by} & \boldsymbol{x}^{k+1} = \boldsymbol{\mathcal{H}}(\boldsymbol{x}^k) - \boldsymbol{W}_k \boldsymbol{\alpha}^k \qquad \text{and} \\ \boldsymbol{x}^{k+1} = \boldsymbol{\mathcal{H}}(\boldsymbol{x}^k) - (\boldsymbol{M}_{prev}^{-1} + \widetilde{\boldsymbol{W}}_k \boldsymbol{V}_k^\dagger) \boldsymbol{r}^k & \text{by} & \boldsymbol{x}^{k+1} = \boldsymbol{\mathcal{H}}(\boldsymbol{x}^k) - \boldsymbol{M}_{prev}^{-1} \boldsymbol{r}^k - \widetilde{\boldsymbol{W}}_k \boldsymbol{\alpha}^k. \end{array}$$

For the ***LS method***, the computation of $\boldsymbol{\alpha}$ via QR-decomposition (***backS***) is followed by a matrix-vector product $\boldsymbol{W}_k \boldsymbol{\alpha}$ to calculate the quasi-Newton update in every iteration.

For the ***MV method***, a similar product $\widetilde{\boldsymbol{W}}_k \boldsymbol{\alpha}$ is required, together with an additional matrix-vector product $\boldsymbol{M}_{prev}^{-1} \boldsymbol{r}^k$ including the representation of the inverse Jacobian estimation from the previous time step for the calculation of Δx^k for the current iteration. The matrix $\widetilde{\boldsymbol{W}}_k = (\boldsymbol{W}_k - \boldsymbol{M}_{prev}^{-1} \boldsymbol{V}_k)$, however, changes in every iteration and requires an additional matrix-vector multiplication $\boldsymbol{M}_{prev}^{-1} \boldsymbol{V}_k$, denoted by (***MV***) for construction. Since MV requires an explicit representation of $\boldsymbol{M}_{prev}^{-1}$, the Jacobian needs to be built and stored after convergence of the quasi-Newton iterations at the end of every time step. This involves large and costly matrix-matrix multiplications. Instead of only computing the vector $\boldsymbol{\alpha} = \boldsymbol{V}_k^\dagger \boldsymbol{r}^k$, the full matrix $\boldsymbol{V}_k^\dagger$ for the update of the Jacobian inverse according to $\boldsymbol{M}^{-1} = \boldsymbol{M}_{prev}^{-1} + \widetilde{\boldsymbol{W}}_k \boldsymbol{V}_k^\dagger$ is needed and computed from solving (2.26) for all unit vectors $\boldsymbol{y} := \boldsymbol{e}_i \in \mathbf{R}^N$, $i = 1, \ldots, N$. We refer to this step as (***iV***). After calculating $\boldsymbol{V}_k^\dagger$, the Jacobian update is computed from (*i*) calculating $\widetilde{\boldsymbol{W}}_k = (\boldsymbol{W}_k - \boldsymbol{M}_{prev}^{-1} \boldsymbol{V}_k)$ involving an $\mathbf{R}^{N \times N} \times \mathbf{R}^{N \times \eta}$ matrix-matrix multiplication (***Wtil***), followed by (*ii*) multiplying $\widetilde{\boldsymbol{W}}_k \boldsymbol{V}_k^\dagger$ with dimensions $\mathbf{R}^{N \times \eta} \times \mathbf{R}^{\eta \times N}$, denoted by (***WV***), and finally (*iii*) adding $\boldsymbol{M}_{prev}^{-1}$. In what follows, we present the algorithmic design of the involved building blocks and describe their efficient realization on distributed memory.

An Efficient QR-Updating Scheme. Solving (2.26) by factorizing $\boldsymbol{V}_k = \widehat{\boldsymbol{Q}} \widehat{\boldsymbol{R}}$ into an economy-sized[21] orthogonal part $\widehat{\boldsymbol{Q}} \in \mathbf{R}^{N \times \eta}$ and an upper triangular part $\widehat{\boldsymbol{R}} \in \mathbf{R}^{\eta \times \eta}$ and computing $\boldsymbol{z}$ from

$$\widehat{\boldsymbol{R}} \boldsymbol{z} = \widehat{\boldsymbol{Q}}^T \boldsymbol{y} \tag{2.27}$$

via ***backward-substitution*** is not only better conditioned and, thus, more stable as computing

[20] For the (J) system, pre-scaling the least-squares problem is required. The modified problem reads $\boldsymbol{z} = \operatorname{argmin}_{\bar{\boldsymbol{z}} \in \mathbf{R}^\eta} \|\boldsymbol{\Lambda}_k \boldsymbol{V}_k \bar{\boldsymbol{z}} - \boldsymbol{\Lambda}_k \boldsymbol{y}\|_2$ for a suitable weighting matrix $\boldsymbol{\Lambda}_k$ (see sub-section *Improving Robustness: Pre-Scaling for the Jacobi-System*). For changing weights in $\boldsymbol{\Lambda}_k$, the QR-decomposition needs to be computed from scratch, i.e., we compute $\mathbf{qr}(\boldsymbol{\Lambda}_k \boldsymbol{V}_k) = \boldsymbol{\Lambda}_k \widehat{\boldsymbol{Q}}_k^T \widehat{\boldsymbol{R}}_k$ and $\boldsymbol{\alpha} = \widehat{\boldsymbol{R}}^{-1} \widehat{\boldsymbol{Q}}^T \boldsymbol{\Lambda}_k^{-1} (\boldsymbol{\Lambda}_k \boldsymbol{r}^k)$ in every iteration. After an initialization phase, the weighting factors can be held constant and re-computation is no longer required. Consequently, the below described updating strategy can be exploited.

[21] Here $\widehat{\boldsymbol{Q}} \in \mathbf{R}^{N \times \eta}$ and $\widehat{\boldsymbol{R}} \in \mathbf{R}^{\eta \times \eta}$ is the economy-size QR-decomposition for "tall and skinny" matrices, computing only the first η columns of the orthogonal matrix $\boldsymbol{Q} \in \mathbf{R}^{N \times N}$ and the upper $\eta \times \eta$-block in $\boldsymbol{R} \in \mathbf{R}^{N \times \eta}$

$\boldsymbol{z} = (\boldsymbol{V}_k^T \boldsymbol{V}_k)^{-1} \boldsymbol{V}_k \boldsymbol{y}$, but also particularly efficient. In every quasi-Newton iteration, one column is added to the left of the matrix $\boldsymbol{V}_k$ (and possibly one column is deleted from the right), i.e., $\boldsymbol{V}_k = [\Delta \boldsymbol{r}_{k-1}^k, \boldsymbol{V}_{k-1}(:, \eta - 1)]$ which leads to a sequence of similarly successive least-squares problems[22]. Instead of recurrent re-computation of the QR-decomposition, we employ an efficient, stable updating scheme of an existing decomposition based on Givens rotations [Bun16b; Dan76]. The update is realized by means of ***column insertion*** and ***column deletion*** operations. Roughly speaking, inserting a column $\boldsymbol{v}$ to an existing decomposition $\widehat{\boldsymbol{Q}}\widehat{\boldsymbol{R}}$ first requires orthogonalization of $\boldsymbol{v}$ against the columns of $\widehat{\boldsymbol{Q}}$ followed by application of a series of Given rotations to eliminate previously introduced non-zero sub-diagonal entries in $\widehat{\boldsymbol{R}}$. This involves $(\eta/2)(\eta - 1)$ Givens rotations. That means, given $\widehat{\boldsymbol{Q}}$ and $\boldsymbol{V}_k = [\boldsymbol{v}, \boldsymbol{V}_{k-1}]$, we search for the new orthogonal column $\boldsymbol{q} \in \mathbf{R}^N$, the respective orthogonalization weight vector $\boldsymbol{s} \in \mathbf{R}^\eta$ and the normalization factor $\rho \in \mathbf{R}$ such that $\boldsymbol{v} = \widehat{\boldsymbol{Q}}\boldsymbol{s} + \rho \boldsymbol{q}$, i.e.,

$$\left(\boldsymbol{v}\ \widehat{\boldsymbol{Q}}\right) = \left(\boldsymbol{q}\ \widehat{\boldsymbol{Q}}\right) \begin{pmatrix} \rho & 0 \\ \boldsymbol{s} & \boldsymbol{I} \end{pmatrix} \quad \text{with} \quad \widehat{\boldsymbol{Q}}^T \boldsymbol{q} = 0 \ \text{ and } \ \|\boldsymbol{q}\|_2 = 1$$

Then $\boldsymbol{s}$ is calculated from $\boldsymbol{s} = \widehat{\boldsymbol{Q}}^T \boldsymbol{v}$. The orthogonal part $\rho \boldsymbol{q} =: \bar{\boldsymbol{v}}$ of $\boldsymbol{v}$ is computed via Gram-Schmidt orthogonalization $\bar{\boldsymbol{v}} = \boldsymbol{v} - \widehat{\boldsymbol{Q}}\boldsymbol{s} = (\boldsymbol{I} - \widehat{\boldsymbol{Q}}\widehat{\boldsymbol{Q}}^T)\boldsymbol{v}$ and normalized, i.e., $\|\boldsymbol{q}\|_2 \overset{!}{=} 1$, hence $\rho = \|\bar{\boldsymbol{v}}\|_2$ and $\boldsymbol{q} := \bar{\boldsymbol{v}}/\rho$; compare Alg. 2.2. This yields the new decomposition

$$(\boldsymbol{v}, \boldsymbol{V}_{k-1}) = (\widehat{\boldsymbol{Q}}\boldsymbol{s} + \rho \boldsymbol{q}, \widehat{\boldsymbol{Q}}\widehat{\boldsymbol{R}}) = (\boldsymbol{q}, \widehat{\boldsymbol{Q}}) \begin{pmatrix} \rho & \widehat{\boldsymbol{R}} \\ \boldsymbol{s} & \boldsymbol{0} \end{pmatrix}.$$

The column $(\rho, \boldsymbol{s})^T$ added to the front of $\widehat{\boldsymbol{R}}$ destroys the upper triangular structure and requires the application of suitable Givens rotations.

Analogously in case of deletion of a column by filtering, the column is dropped from the decomposition and $\widehat{\boldsymbol{R}}$ needs to be re-structured by means of Givens rotations to maintain upper triangular structure. Note, that deletion of the oldest column on the right does not involve any computations. For later reference, we refer to a column insertion or deletion operation with (***iQR***) and to an update step consisting of ι column insertions/deletions[23] with (***updQR***). A re-computation from scratch[24] is referred to as (***QR***). For a detailed and robust algorithmic description of the updating scheme, we refer to [Dan76; Bun16b].

Re-Orthogonalization. Due to limited numerical accuracy and round off errors, the above column updating algorithm may lead to non-orthogonal columns in $\widehat{\boldsymbol{Q}}$ and re-orthogonalization becomes necessary to avoid emphasized errors in subsequent computations (e.g., computation of $\boldsymbol{\alpha}$). We shortly sketch the implemented re-orthogonalization strategy and detection of the necessity thereof. Assume the orthogonal part $\rho = \|\bar{\boldsymbol{v}}\|_2$ of $\boldsymbol{v}$ with respect to $\widehat{\boldsymbol{Q}}$ is zero, it follows that the new column $\boldsymbol{v} \in \operatorname{span}(\widehat{\boldsymbol{Q}})$ and the column is discarded. Inexact floating point numerics and round off errors, however, rather yield $0 < \rho \ll 1$ and $\|\widehat{\boldsymbol{Q}}^T \bar{\boldsymbol{v}}\|_2$ almost zero, but not quite. That means $\|\widehat{\boldsymbol{Q}}^T \bar{\boldsymbol{v}}\|_2 = \epsilon \|\boldsymbol{v}\|$

[22] We slightly abuse notation here and in the following, and disregard the potential elimination of columns (filtering) when referring to the object dimensions. In other words, we denote the number of columns used in the multi-secant equation (whether filtered or unfiltered) for the current time step by η (instead of $\breve{\eta}$), i.e., $\boldsymbol{V}_k$, $\boldsymbol{W}_k$, $(\boldsymbol{V}_k^\dagger)^T$, $\widetilde{\boldsymbol{W}}_k$, $\widehat{\boldsymbol{Q}} \in \mathbf{R}^{N \times \eta}$ and $\widehat{\boldsymbol{R}} \in \mathbf{R}^{\eta \times \eta}$. In the same spirit, we relate to the dimensionality of matrices from previous time steps, e.g., $\boldsymbol{V}_{k_{n-i}}^{k_{n-i}, n-i}$ (or short $\boldsymbol{V}_{k_i}$) with η_i, i.e., $\boldsymbol{V}_{k_i} \in \mathbf{R}^{N \times \eta_i}$.

[23] If no filtering is considered, $1 \leqslant \iota \leqslant 2$, i.e., one column is inserted at the front and possibly one column dropped at the end, if $k > \eta$, or $\sum_{q=n-\xi}^{n} k_q > \eta$ (sliding-window approach). For enabled filtering, a larger number of columns may be deleted.

[24] Re-computation of the QR-decomposition from scratch is done by performing η consecutive column insertion operations.

for a small positive number ϵ. As a result, $\|\widehat{\boldsymbol{Q}}^T \bar{\boldsymbol{q}}\|_2 = \epsilon\|v\|_2/\rho$ and orthogonality of $[\widehat{\boldsymbol{Q}}\ \boldsymbol{q}]$ is no longer given as the factor $0 < \epsilon/\rho$ can be large for ρ sufficiently small. To detect and remedy this problem, we use a criterion similar to the QR2 filter and re-orthogonalize $\bar{\boldsymbol{v}}$ if $\rho = \|\bar{\boldsymbol{v}}\|_2 < \sqrt{2}\|\boldsymbol{v}\|_2$ and recursively re-apply the Gram-Schmidt process until $\bar{\boldsymbol{v}}$ is sufficiently orthogonal to $\widehat{\boldsymbol{Q}}$, or the column is discarded after 4 trials.

Updated QR-Decomposition on Distributed Data. Distributed memory algorithms and massively parallel execution are inevitable for a growing problem size[25]. The here considered data distribution is natural for partitioned surface-coupled multi-physics problems (such as FSI), but is also applicable in a more generic setting. Assuming parallel execution on p MPI ranks, the vector of unknowns is distributed into sub-vectors of length N/p, assuming ideal load-balancing[26]. This translates into a row-block-wise distribution with blocks of size $N/p \times \eta$ of the matrices $\boldsymbol{V}, \boldsymbol{W}, \widehat{\boldsymbol{Q}}$ and $\widetilde{\boldsymbol{W}}$, and, analogously to a block-column-wise decomposition with blocks $\eta \times N/p$ of the pseudo inverse matrix $\boldsymbol{V}^\dagger$. The quadratic $\eta \times \eta$ matrix $\widehat{\boldsymbol{R}}$ is locally replicated on each MPI rank as the number of columns η is assumably small; cf. Fig. 2.1.

Inserting a column using the above QR-updating strategy requires communication between MPI ranks only incurred by the inner products and ℓ_2-norm vector operations $\widehat{\boldsymbol{Q}}^T \boldsymbol{v}$ and $\|\boldsymbol{v}\|_2$. Applying a series of Givens rotations can be done fully locally without communication as $\widehat{\boldsymbol{R}}$ is locally replicated and only row elements in $\widehat{\boldsymbol{Q}}$ are required. For (***iQR***), this translates into a computational complexity of $\mathcal{O}(\eta N/p) + \mathcal{O}(\eta \log p)$ for η inner products of length N in the orthogonalization process and a $\mathcal{O}(\eta^2 N/p) + \mathcal{O}(\eta^3)$ complexity for the $\eta^2/2$ local Givens rotations. This results in the parallel runtime complexity

$$\textbf{(\textit{iQR})} \qquad \mathcal{O}(\eta^2 N/p) + \mathcal{O}(\eta \log p) + \mathcal{O}(\eta^3)$$

for updating an existing decomposition (insertion/deletion of a column). Re-computing the decomposition from scratch (***QR***), leads to a parallel complexity of $\mathcal{O}(\eta^3 N/p) + \mathcal{O}(\eta^2 \log p) + \mathcal{O}(\eta^4)$.

Solving the quadratic system (2.27) via ***backward-substitution*** on distributed data requires the parallel computation of $\widehat{\boldsymbol{Q}}^T \boldsymbol{y}$ involving an allreduce operation, which translates to a runtime complexity of $\mathcal{O}(\eta N/p) + \mathcal{O}(\eta \log p)$, followed by the fully locally back-substitution step. Summarized, the computation of the vector $\boldsymbol{\alpha}$, as used in the LS method has a parallel runtime complexity of

$$\textbf{(\textit{backS})} \qquad \mathcal{O}(\eta N/p) + \mathcal{O}(\eta \log p) + \mathcal{O}(\eta^2).$$

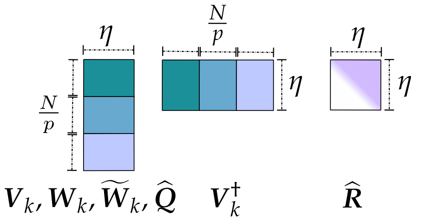
Computation of the pseudo inverse matrix $\boldsymbol{V}^\dagger$, as required for the MV method, involves solvingf the back-substitution for all unit vectors $\boldsymbol{e}_i, i = 1, \ldots, N$. Computing $\widehat{\boldsymbol{Q}}^T \boldsymbol{e}_i$, however, neither involves computational cost nor communication effort and the overall parallel runtime complexity to form the pseudo-inverse (***iV***) is given by $\mathcal{O}(\eta^2 N/p)$, assuming a block-column-wise distribution of $\boldsymbol{V}^\dagger$.

An efficient, distributed memory implementation of the updating scheme with sophisticated re-orthogonalization criteria including the above described QR filters has been added to the coupling

[25] Note, that for, e.g., partitioned FSI, the number of unknowns N at the coupling interface (i.e., a 2D surface) is small compared to the number of unknowns within the three dimensional domains for the single-physics solvers. Domain decomposition data distribution is thus typically induced and dictated by the sub-component solvers; the decomposition at the interface then automatically defines the load-balancing and data distribution for the considered algorithms. As a result, parallel execution and distributed memory algorithms might become necessary long before the number of unknowns N at the interface grow too large. For volume coupled problems, however, the situation is different.

[26] For partitioned multi-physics simulations, the load-balancing and therefore, the domain decomposition is optimized with respect to the unknowns in the volumetric domain of the single-physics solver. In general, this results in a non-optimally balanced load at the interface.

FIGURE 2.1
Decomposition and storage distribution of the matrices $\widehat{\boldsymbol{Q}} \in \mathbf{R}^{N\times\eta}$, $\boldsymbol{V}_k^\dagger \in \mathbf{R}^{\eta\times N}$ and $\widehat{\boldsymbol{R}} \in \mathbf{R}^{\eta\times\eta}$. The matrices $\boldsymbol{V}_k, \boldsymbol{W}_k$ and $\widetilde{\boldsymbol{W}}_k$ are distributed analogously to $\widehat{\boldsymbol{Q}}$. The matrix decomposition on distributed data is indicated with different colors, representing local sub-blocks on different MPI ranks (here $p = 3$ ranks); the purple color indicates a low-dimensional matrix that is locally replicated on all processors (such as $\widehat{\boldsymbol{R}}$). η indicates the number of stored columns and is typically small, e.g., $\eta < k$ for MV.


η
$\frac{N}{p}$
$\frac{N}{p}$
η
η
η
$\boldsymbol{V}_k, \boldsymbol{W}_k, \widetilde{\boldsymbol{W}}_k, \widehat{\boldsymbol{Q}}$
$\boldsymbol{V}_k^\dagger$
$\widehat{\boldsymbol{R}}$

library preCICE in the course of this thesis. A similar implementation for the massively parallel realization of the LS method is presented in [Lof15]. Numerical results demonstrating the robustness of the implementation and, in particular, the effect of different filtering techniques to ensure stability can be found in [Hae15; Uek16]. Results demonstrating the computational efficiency and parallel scalability are published in [Bun16b; Sch17].

Least-Squares Method on Distributed Data. For the sake of completeness, we summarize the computational steps required per iteration for the LS method. Updating of the QR-decomposition and subsequent filtering, comprises ι insertion/deletion operations (***iQR***), or even re-computation from scratch (***QR***), followed by calculation of $\boldsymbol{\alpha}$ via (***backS***) and compute $\Delta\boldsymbol{x}$ by locally multiplying $\boldsymbol{W}\boldsymbol{\alpha}$ on each rank. The parallel runtime complexity per iteration is dominated by

$$(\boldsymbol{LS\,/\,it}) \qquad \mathcal{O}(\eta^2 N/2) + \mathcal{O}(\eta^4) + \mathcal{O}(\eta^2 \log p).$$

Multi-Vector Update Method on Distributed Data. The original MV method [Bog14] as presented above requires explicit storage of the Jacobian and involves three large and costly dense matrix-matrix products, namely (*i*) $\widetilde{\boldsymbol{W}}_k = (\boldsymbol{W}_k - \boldsymbol{M}_{prev}^{-1}\boldsymbol{V}_k)$, (*ii*) $\widetilde{\boldsymbol{W}}_k\boldsymbol{V}_k^\dagger$ and (*iii*) $\boldsymbol{M}_{prev}^{-1}\boldsymbol{r}^k$. This results in $\mathcal{O}(N^2)$ complexity both in terms of computational effort and storage requirement. We first present the realization of the plain-vanilla MV method on distributed data. Thereafter, an advanced limited-memory variation with efficient storage management of the Jacobian which reduces the overall cost to linear $\mathcal{O}(N)$ complexity, is introduced.

Dense Multiplication $\boldsymbol{M}_{prev}^{-1}\boldsymbol{V}$ on Distributed Data. Considering parallel execution on p MPI ranks, the Jacobian is distributed into column-blocks of size $N \times N/p$ (i.e., equaling the global size of the matrices $\boldsymbol{V}, \boldsymbol{W}$); cf. Fig. 2.2. To compute $\boldsymbol{M}_{prev}^{-1}\boldsymbol{V}_k$, each processor j first computes its local contribution $\boldsymbol{M}_{prev_{\#j}}^{-1}\boldsymbol{V}_{k\#j}$ of size $N \times N/p$, which is then summed up via a reduce operation. The result needs to be scattered to the sub-processes; compare Fig. 2.2 (left). The parallel runtime complexity for $\boldsymbol{M}_{prev}^{-1}\boldsymbol{V}$ sums up to

$$(\boldsymbol{MV}) \qquad \mathcal{O}(\eta N^2/p) + \mathcal{O}(\eta N \log p).$$

Alternatively, a distributed inner product per entry of the resulting matrix $\widetilde{\boldsymbol{W}}$ can be computed, omitting the allocation of extra $N \times N/p$ matrix blocks per MPI rank. The communication overhead in turn increases. Both alternatives are available within the distributed memory implementation of the plain-vanilla MV method in preCICE. This product is formed per iteration of the MV method.

Dense Multiplication $\widetilde{\boldsymbol{W}}_k\boldsymbol{V}_k^\dagger$ on Distributed Data. This matrix product is more involved as it requires an all-to-all communication of information, i.e., each processor j requires corresponding sub-blocks $\widetilde{\boldsymbol{W}}_{\#i}$ of $\widetilde{\boldsymbol{W}}$ from all other processors $i \neq j$. A schematic illustration is given in Fig. 2.2 (right). This is realized employing a cyclic communication principle, similar to Cannon's algorithm for distributed

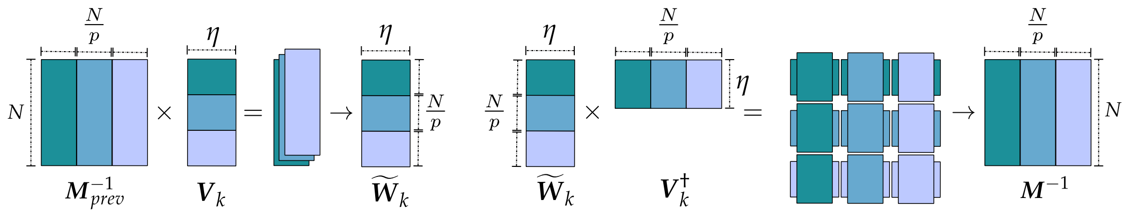


FIGURE 2.2 *Schematic view of the distributed memory matrix-multiplications involved in the MV method. Block partitioning and parallel multiplication of $M^{-1}_{prev}V_k$ (left) and of $\widetilde{W}_kV_k^\dagger$ (right). Figure modified from [Bun16b].*

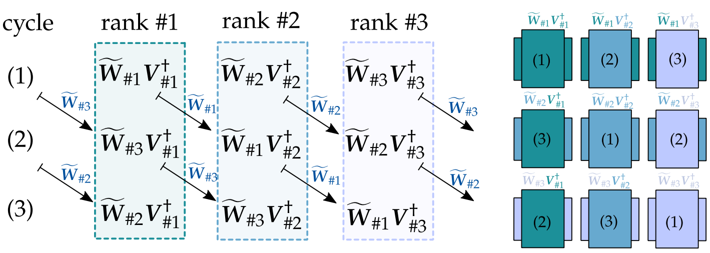

cycle
rank #1
rank #2
rank #3
(1)
(2)
(3)
$\widetilde{W}_{\#1}V^\dagger_{\#1}$
$\widetilde{W}_{\#2}V^\dagger_{\#2}$
$\widetilde{W}_{\#3}V^\dagger_{\#3}$
$\widetilde{W}_{\#3}V^\dagger_{\#1}$
$\widetilde{W}_{\#1}V^\dagger_{\#2}$
$\widetilde{W}_{\#2}V^\dagger_{\#3}$
$\widetilde{W}_{\#2}V^\dagger_{\#1}$
$\widetilde{W}_{\#3}V^\dagger_{\#2}$
$\widetilde{W}_{\#1}V^\dagger_{\#3}$
$\widetilde{W}_{\#3}$
$\widetilde{W}_{\#1}$
$\widetilde{W}_{\#2}$

FIGURE 2.3 *Schematic view of cyclic scheme for the dense mat-mult $M^{-1} = \widetilde{W}_kV_k^\dagger$. The arrows indicate cyclic inter-processor communication of sub-blocks $\widetilde{W}_{\#i}$ with minimal communication and memory overhead. After p multiplications and $(p-1)p$ send-receive operations of sub-matrices $\widetilde{W}_{\#i}$ (left), the multiplication is completed and readily available on distributed ranks (right); (cf. [Bun16b]).*

matrix-multiplication. Each processor locally computes the product $\widetilde{W}_{\#i}V^\dagger_{\#i}$ of the currently available blocks of $\widetilde{W}_k$ and $V_k^\dagger$ and hands over his local block $\widetilde{W}_{\#i}$ to the neighboring processor "on the right" and consequently also receives a new block "from the left". The communication pattern and cyclic strategy is given in Fig. 2.3. Multiplying the new block of $\widetilde{W}_k$ with the local block of $V_k^\dagger$ yields the next contribution. All contributions are computed after p cycles of send-receive operations and the resulting matrix M^{-1} is readily available on distributed MPI ranks; no further communication is required. This algorithm allows for an efficient communication-computation overlay: Before computing the local matrix products, asynchronous send and receive operations for the next block of $\widetilde{W}$ are triggered. The local matrix products sum up to a parallel runtime complexity of $\mathcal{O}(\eta N/p)$. The communication cost is $\mathcal{O}(\eta N/p) = \mathcal{O}(\eta N)$. Combined, the parallel cost sums up to

$$(\boldsymbol{WV}) \qquad \mathcal{O}(\eta N^2/p).$$

The computation of this expensive matrix-product is only required to explicitly form the Jacobian update at the end of a (converged) time step.

Compute Update $\Delta\tilde{x} = -M^{-1}r^k$ on Distributed Data. Computation of the matrix-vector product $M^{-1}r^k$ is analogous to the above described computation of $M^{-1}_{prev}V_k$. Each processor locally computes its additive contribution to the resulting Newton update vector. In a subsequent reduce step, all contributions are summed up and row blocks scattered to the responsible processors. This accumulates to a parallel runtime of

$$(\boldsymbol{Mr}) \qquad \mathcal{O}(N^2/p) + \mathcal{O}(N\log p).$$

This product is formed per MV iteration. Summarizing, the MV method has a parallel runtime complexity of

$$(\boldsymbol{MV/it}) \qquad \mathcal{O}(\eta N^2/p) + \mathcal{O}(\eta N\log p)$$

per iteration.

Conclusion. Summarizing, we note that the cost for LS grows linearly in N and scales down with the number of parallel MPI ranks p. The QR-decomposition dominates the computational cost for LS. Its cubic cost in the number of columns requires η to be small in order for LS to be computationally efficient. On the other hand, a small η may hamper convergence. In comparison, the MV method entails large, dense matrix multiplications that dominate the overall runtime and result in quadratic cost; an additional linear term occurs which grows with $\log p$ instead of scaling down with the number of parallel tasks. Furthermore, MV requires explicit storage of $\boldsymbol{M}_{prev}^{-1}$ resulting in an $\mathcal{O}(N^2/p)$ memory footprint per rank. This quadratic memory consumption rules out the method for most practical applications featuring a large number of unknowns. With an $\mathcal{O}(\eta N/p)$ memory requirement per rank, the LS method is well suited for large scale applications.

An Efficient and Scalable Multi-Vector Restart Alternative. We now present an enhanced variant of MV which reduces both computational cost and memory requirement from quadratic complexity to linear complexity by exploiting an efficient storage management and truncation of the Jacobian, following a limited-memory approach with periodic restart strategy. We published this method earlier in [Sch17]. The description herein follows our previous work. The fundamental idea is to store a limited number $m \leqslant n$ of sub-matrices $\widetilde{\boldsymbol{W}}_{k_q}$ and $\boldsymbol{V}_{k_q}^{\dagger}$ for $q = n, \dots, n-m$ over several steps and compute the Newton update from matrix-vector operations on-the-fly instead of explicitly building and storing the entire Jacobian matrix. Furthermore, we optimize the computation of $\widetilde{\boldsymbol{W}}_k$ and perform column updates rather than re-computations of the entire matrix.

Efficient Representation of the Jacobian. Unrolling the recursive MV Jacobian update formula (2.20), the matrix at time step n decomposes into the explicit sum of products $\widetilde{\boldsymbol{W}}_k \boldsymbol{V}_k^{\dagger}$ from all previous time steps, i.e.,

$$\boldsymbol{M}^{-1} = \widetilde{\boldsymbol{W}}_{k_0} \boldsymbol{V}_{k_0}^{\dagger} + \widetilde{\boldsymbol{W}}_{k_1} \boldsymbol{V}_{k_1}^{\dagger} + \cdots + \widetilde{\boldsymbol{W}}_{k_n} \boldsymbol{V}_{k_n}^{\dagger}. \tag{2.28}$$

Without building the Jacobian, the Newton update can then be computed from on-the-fly matrix-vector computations

$$\Delta \boldsymbol{x}^{k+1} = \boldsymbol{x}^k - \sum_{q=0}^{n} \widetilde{\boldsymbol{W}}_{k_q} (\boldsymbol{V}_{k_q}^{\dagger} \boldsymbol{r}^k). \tag{2.29}$$

Obviously, this approach is efficient only if the number of stored matrices is substantially smaller than N. In a transient setting, however, n may grow arbitrarily large. Thus, we introduce a limited depth m of the method's memory and apply a periodic reset of storage, or ***restart of the method***, i.e., either re-setting the Jacobian estimate to zero (clear all) or transforming the sum representation into a compact reduced representation. Consequently, we divide the simulation time in chunks of m time steps for which each partition represents one era for the Jacobian estimation. At restart, we free all memory for the sum representation and reset $\boldsymbol{M}^{-1}$ to $\widetilde{\boldsymbol{W}}_0 \boldsymbol{V}_0^{\dagger}$ with suitable matrices $\widetilde{\boldsymbol{W}}_0$ and $\boldsymbol{V}_0^{\dagger}$. Thus, with restart, $\boldsymbol{M}^{-1}$ is represented by

$$\boldsymbol{M}^{-1} = \widetilde{\boldsymbol{W}}_0 \boldsymbol{V}_0^{\dagger} + \sum_{q=n-m}^{n} \widetilde{\boldsymbol{W}}_{k_q} \boldsymbol{V}_{k_q}^{\dagger}$$

For transient problems, clear-all restart typically goes hand in hand with a severe degradation of convergence speed due to the loss of Jacobian information. Other restart strategies allow retaining of information from previous chunks to tackle this issue. Let m be an upper bound on the number of time steps held in memory and $\bar{\eta} = \max(k_{n-m}, k_{n-m+1}, \ldots, k_n)$ be an upper bound on the number of columns/iterations per time step. We present and investigate three different restart alternatives:

(***RS-0***) Clear all. No information is retained and the Jacobian approximation is started from scratch. The Newton update $\boldsymbol{M}^{-1}\boldsymbol{r}^k$ is computed from $2m$ matrix-vector products, i.e.,

$$\boldsymbol{y} = \boldsymbol{V}_{k_q}^{\dagger}\boldsymbol{r}^k \quad \text{and} \quad \boldsymbol{W}_{k_q}\boldsymbol{y}, \quad \text{for } q = n-m, \ldots, n$$

This results in a (sequential) runtime complexity and memory consumption of $\mathcal{O}(m\bar{\eta}N)$.

(***RS-LS***) Clear the sum representation of $\boldsymbol{M}^{-1}$, i.e., drop stored matrices $\widetilde{\boldsymbol{W}}_{k_q}, \boldsymbol{V}_{k_q}^{\dagger}$, but ***explicitly retain secant information*** from $\xi_{rs} < m$ previous time steps within the current chunk. In other words, restart MV with

$$\boldsymbol{M}_{prev}^{-1} := \widetilde{\boldsymbol{W}}_0\boldsymbol{V}_0^{\dagger}, \ \ \widetilde{\boldsymbol{W}}_0 := \boldsymbol{W}_k^{\eta_{rs},n} \ \text{ and } \ \boldsymbol{V}_0^{\dagger} := \widehat{\boldsymbol{R}}^{-1}\widehat{\boldsymbol{Q}}^T \ \text{ with } \ \boldsymbol{V}_{rs}^{\eta_{rs}} = \widehat{\boldsymbol{Q}}\widehat{\boldsymbol{R}}$$

where

$$\boldsymbol{V}_{rs}^{\eta_{rs}} = [\boldsymbol{V}_k^{k,n}, \ \ldots, \ \boldsymbol{V}_{k_{n-\xi_{rs}}}^{k_{n-\xi_{rs}}, n-\xi_{rs}}] \ \text{ and } \ \boldsymbol{W}_{rs}^{\eta_{rs}} = [\boldsymbol{W}_k^{k,n}, \ \ldots, \ \boldsymbol{W}_{k_{n-\xi_{rs}}}^{k_{n-\xi_{rs}}, n-\xi_{rs}}].$$

The associated cost for this variant are $\mathcal{O}(\eta_{rs}N) + \mathcal{O}(m\bar{\eta}N)$ if η_{rs} previous columns are reused.

(***RS-SVD***) Clear the sum representation of $\boldsymbol{M}^{-1}$, but ***retain most dominant modes*** by employing a ***truncated singular value decomposition (SVD)*** of the Jacobian estimation as restart, i.e., use

$$\boldsymbol{M}_{prev}^{-1} := \widetilde{\boldsymbol{W}}_0\boldsymbol{V}_0^{\dagger}, \ \ \widetilde{\boldsymbol{W}}_0 := \overline{\boldsymbol{\Psi}} \ \text{ and } \ \boldsymbol{V}_0^{\dagger} := \overline{\boldsymbol{\Sigma}}\overline{\boldsymbol{\Phi}}^T$$

where

$$\boldsymbol{M}^{-1} = \underbrace{\left(\boldsymbol{\Psi}_{\cdot,j}\right)_{j=1,\ldots,\bar{\kappa}}}_{=:\,\overline{\boldsymbol{\Psi}}} \underbrace{\begin{pmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_{\bar{\kappa}} \end{pmatrix}}_{=:\,\overline{\boldsymbol{\Sigma}}} \underbrace{\left(\boldsymbol{\Phi}_{\cdot,j}\right)_{j=1,\ldots,\bar{\kappa}}^T}_{=:\,\overline{\boldsymbol{\Phi}}^T} \tag{2.30}$$

with $\overline{\boldsymbol{\Psi}}, \overline{\boldsymbol{\Phi}} \in \mathbf{R}^{N\times\bar{\kappa}}$ and $\overline{\boldsymbol{\Sigma}} \in \mathbf{R}^{\bar{\kappa}\times\bar{\kappa}}$ is the truncated version of the singular value decomposition

$$\boldsymbol{M}^{-1} = \boldsymbol{\Psi\Sigma\Phi}^T \ \text{ with } \ \boldsymbol{\Sigma} = \ \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_N) \in \mathbf{R}^{N\times N}$$

of the Jacobian with $\sigma_1 \geqslant \sigma_2 \geqslant \ldots \geqslant \sigma_N \geqslant 0$. The truncated SVD is obtained after cutting off all singular values below a given threshold. We use an efficient rank-k updating strategy for the computation of the SVD to track the estimated Jacobian's sub-space. The storage requirement and computational complexity for this approach accumulate to $\mathcal{O}(\bar{\kappa}^2)$ $+\mathcal{O}(\bar{\kappa}N)$ $+\mathcal{O}(m\bar{\eta}N)$, disregarding the computational effort for the restart numerics and assuming the SVD is truncated to the $\bar{\kappa}$ most dominant singular values. For the sake of clarity, the RS-SVD restart alternative for the MV method is outlined in Alg. 2.3.

While offering fast and robust convergence, the practicality of RS-SVD critically depends on the efficient algorithmic realization of the singular value decomposition. SVD algorithms typically feature cubic runtime complexity. We employ an efficient truncated SVD updating scheme with computational cost growing cubically in the number of significant modes only, i.e., as a function of the truncated rank $\bar{\kappa}$ of the SVD representation of the Jacobian. To obtain overall acceptable cost

```
func MV-RS-SVD($x^0, m, \varepsilon_{svd}$)
  $\widetilde{W}, V^\dagger = \{\ \}$,  $\overline{\Psi}, \overline{\Sigma}, \overline{\Phi} = \mathbf{0}$,  $s = 0$
  for $n = 1, 2, \ldots$ do
    $x^0 =$ extrapolate($x^*_{n-1}, x^*_{n-2}, x^*_{n-3}$)
    $\tilde{x}^0 = \mathcal{H}(x^0)$, and $r^0 = \mathcal{R}(x^0) = \tilde{x}^0 - x^0$
    $x^1 = x^0 + \omega r^0$
    for $k = 1, 2, \ldots$ do
      $\tilde{x}^k = \mathcal{H}(x^k)$, and $r^k = \mathcal{R}(x^k) = \tilde{x}^k - x^k$
      if $converged$ then
        break
      $W_k^\eta = [\Delta\tilde{x}^k_{k-\eta}, \ldots, \Delta\tilde{x}^k_{k-1}]$,  $\Delta\tilde{x}^k_i = \tilde{x}^k - \tilde{x}^i$
      $V_k^\eta\ = [\Delta r^k_{k-\eta}, \ldots, \Delta r^k_{k-1}]$,   $\Delta r^k_i = r^k - r^i$
      update QR-dec.  $V_k^\eta = QR$ and apply filter
      solve $RV_k^\dagger = Q^T$ for $V_k^\dagger$ (2.27)
      update $\widetilde{W}_k = W_k - \sum_{q=n-s}^{n} \widetilde{W}_{k_q}(V^\dagger_{k_q} V_{k_q})$ via (2.36)
      $\Delta x^{k+1} = -\overline{\Psi}\left(\overline{\Sigma\Phi}^T r^k\right) - \sum_{q=n-s}^{n} \widetilde{W}_{k_q}\left(V^\dagger_{k_q} r^k\right)$ (2.35)
      $x^{k+1} = x^k + \Delta x^k$
    store $\widetilde{W} = \{\widetilde{W}_{k_n}, \ldots, \widetilde{W}_{k_{n-s}}\}, V^\dagger = \{V^\dagger_{k_n}, \cdots, V^\dagger_{k_{n-s}}\}$
    $s = s + 1$,  $t = t + \Delta t$
    if $s \geqslant m$ then
      $[\overline{\Psi}, \overline{\Sigma}, \overline{\Phi}] =$ update-SVD($\widetilde{W}, V^\dagger, \varepsilon_{svd}$)
      clear $\widetilde{W}, V^\dagger = \{\ \}$, $s = 0$
```

ALGORITHM 2.3 *Pseudo code for the MV restart method RS-SVD. Restart is triggered after completion of m time steps. The truncated singular value decomposition representation of the inverse Jacobian approximation is efficiently updated by means of rank-k updates. For a schematic overview of the SVD update, see Fig. 2.4; more details on distributed memory implementation are given below.*

(in particular we aim for linear complexity), the rank $\overline{\kappa}$ needs to be significantly smaller than the number of unknowns N, i.e., $\overline{\kappa} < \sqrt[3]{N}$. Consequently, the efficiency, robustness and applicability of RS-SVD strongly depends on the assumption that the ***inverse Jacobian of the system matrix is of low rank***, or, that a good low-rank approximation exists[27]. Note that the low-rank approximability requirement already is a prerequisite for quasi-Newton to work reliably. For algorithmic scalability, we additionally require the rank of the inverse Jacobian to be invariant with respect to the number of unknowns (i.e., mesh size and discretization), or at least to be bounded.

An Efficient SVD Updating Scheme. As before, we rely on a rank-k updating strategy to maintain an existing, thresholded singular value decomposition, rather than re-compute from scratch. The latter is not only much more expensive, but also would require the explicit computation of M^{-1}. Assuming a truncated SVD representation of M^{-1}_{prev} as given in (2.30), the inverse Jacobian approximation upon completion of the current chunk reads

$$\overline{\Psi}\,\overline{\Sigma}\,\overline{\Phi}^T + \sum_{q=1}^{m} \widetilde{W}_{k_q} V^\dagger_{k_q}. \tag{2.31}$$

[27] Note, that in case the inverse Jacobian cannot be approximated sufficiently accurate by a low-rank matrix, the cost of the RS-SVD approach can be kept acceptable by selecting a sharper threshold; the robustness of the method, however, worsens.

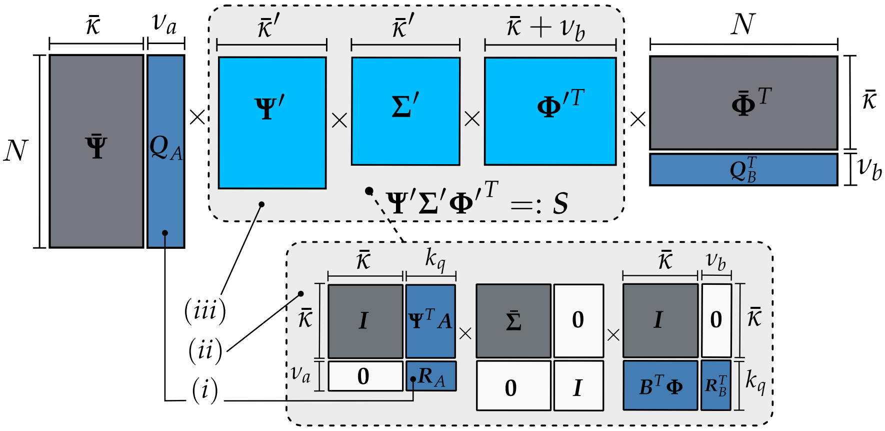


FIGURE 2.4 *Schematic view of the SVD rank-k update of the form* $\overline{\boldsymbol{\Psi}}\,\overline{\boldsymbol{\Sigma}}\,\overline{\boldsymbol{\Phi}}^T + \boldsymbol{A}\boldsymbol{B}^T$. *The three major steps are indicated:* (i) *computation of the orthogonal parts* $\boldsymbol{A}^\perp := (\boldsymbol{I} - \overline{\boldsymbol{\Psi}}\,\overline{\boldsymbol{\Psi}}^T)\boldsymbol{A}$ *and* $\boldsymbol{B}^\perp := (\boldsymbol{I} - \overline{\boldsymbol{\Phi}}\,\overline{\boldsymbol{\Phi}}^T)\boldsymbol{B}$ *and QR-decomposition thereof, resulting in* $\boldsymbol{A}^\perp = \boldsymbol{Q}_A\boldsymbol{R}_A$ *and* $\boldsymbol{B}^\perp = \boldsymbol{Q}_B\boldsymbol{R}_B$; (ii) *composition of the matrix* $\boldsymbol{S} \in \mathbf{R}^{\bar{\kappa}+\nu_a \times \bar{\kappa}+\nu_b}$ *with* ν_a *and* ν_b *dimensionalities of the column space of* $\boldsymbol{A}^\perp$ *and* $\boldsymbol{B}^\perp$; *and* (iii) *diagonalization of* $\boldsymbol{S}$ *via singular value decomposition and subsequent rotation of left- and right-singular vector sub-spaces.*

Clearing memory and re-setting MV requires the computation of a new truncated SVD, representing (2.31), which is done by means of m rank-k_q updates of the form

$$\overline{\boldsymbol{\Psi}}\,\overline{\boldsymbol{\Sigma}}\,\overline{\boldsymbol{\Phi}}^T + \boldsymbol{A}\boldsymbol{B}^T = \begin{bmatrix} \overline{\boldsymbol{\Psi}} & \boldsymbol{A} \end{bmatrix} \begin{bmatrix} \overline{\boldsymbol{\Sigma}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \overline{\boldsymbol{\Phi}} & \boldsymbol{B} \end{bmatrix}^T \tag{2.32}$$

with $\boldsymbol{A} := \widetilde{\boldsymbol{W}}_{k_q} \in \mathbf{R}^{N \times k_q}, \boldsymbol{B}^T := \boldsymbol{V}_{k_q}^\dagger \in \mathbf{R}^{k_q \times N}$, $q = n-m, \ldots, n$. For an efficient realization of these low-rank updates, we follow the algorithm proposed in [Bra06] and compute the orthogonal components of $\boldsymbol{A}$ and $\boldsymbol{B}$ with respect to the left- and right-singular vectors $\overline{\boldsymbol{\Psi}}$ and $\overline{\boldsymbol{\Phi}}$, respectively. Via QR-decomposition, we find matrices $\boldsymbol{Q}_A$ and $\boldsymbol{Q}_B$, representing an orthonormal basis of the column space of $\boldsymbol{A}^\perp := (\boldsymbol{I} - \overline{\boldsymbol{\Psi}}\,\overline{\boldsymbol{\Psi}}^T)\boldsymbol{A}$ and $\boldsymbol{B}^\perp := (\boldsymbol{I} - \overline{\boldsymbol{\Phi}}\,\overline{\boldsymbol{\Phi}}^T)\boldsymbol{B}$ with the corresponding upper triangular matrices

$$\boldsymbol{R}_A := \boldsymbol{Q}_A^T(\boldsymbol{I} - \overline{\boldsymbol{\Psi}}\,\overline{\boldsymbol{\Psi}}^T)\boldsymbol{A} \quad \text{and} \quad \boldsymbol{R}_B := \boldsymbol{Q}_B^T(\boldsymbol{I} - \overline{\boldsymbol{\Phi}}\,\overline{\boldsymbol{\Phi}}^T)\boldsymbol{B}.$$

We can now restate (2.32) as the update of the orthogonal components $\boldsymbol{A}^\perp$ and $\boldsymbol{B}^\perp$, only, in terms of the respective QR-factors, i.e.,

$$\overline{\boldsymbol{\Psi}}\,\overline{\boldsymbol{\Sigma}}\,\overline{\boldsymbol{\Phi}}^T + \boldsymbol{A}\boldsymbol{B}^T = [\overline{\boldsymbol{\Psi}}\ \boldsymbol{Q}_A]\ \boldsymbol{S}\ [\overline{\boldsymbol{\Phi}}\ \boldsymbol{Q}_B]^T$$

with the matrix $\boldsymbol{S} \in \mathbf{R}^{\bar{\kappa}+\nu_a \times \bar{\kappa}+\nu_b}$ composed of

$$\boldsymbol{S} = \begin{bmatrix} \boldsymbol{I} & \overline{\boldsymbol{\Psi}}^T\boldsymbol{A} \\ \boldsymbol{0} & \boldsymbol{R}_A \end{bmatrix} \begin{bmatrix} \overline{\boldsymbol{\Sigma}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{I} & \overline{\boldsymbol{\Phi}}^T\boldsymbol{B} \\ \boldsymbol{0} & \boldsymbol{R}_B \end{bmatrix}^T = \begin{bmatrix} \overline{\boldsymbol{\Sigma}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} + \begin{bmatrix} \overline{\boldsymbol{\Psi}}^T\boldsymbol{A} \\ \boldsymbol{R}_A \end{bmatrix} \begin{bmatrix} \overline{\boldsymbol{\Phi}}^T\boldsymbol{B} \\ \boldsymbol{R}_B \end{bmatrix}^T, \tag{2.33}$$

compare Fig. 2.4. Here, ν_a and ν_b denote the dimensionality of the column space of $\boldsymbol{A}^\perp$ and $\boldsymbol{B}^\perp$, respectively. To re-establish singular value character, $\boldsymbol{S}$ needs to be diagonalized as $\boldsymbol{\Psi}'^T\boldsymbol{S}\boldsymbol{\Phi}' = \boldsymbol{\Sigma}'$, which finally leads to the rank-k update

$$\overline{\boldsymbol{\Psi}}\,\overline{\boldsymbol{\Sigma}}\,\overline{\boldsymbol{\Phi}}^T + \boldsymbol{A}\boldsymbol{B}^T = ([\overline{\boldsymbol{\Psi}}\ \boldsymbol{Q}_A]\ \boldsymbol{\Psi}')\ \boldsymbol{\Sigma}'\ ([\overline{\boldsymbol{\Phi}}\ \boldsymbol{Q}_B]\ \boldsymbol{\Phi}')^T \tag{2.34}$$

of the truncated singular value representation of the inverse Jacobian. After each update, the SVD is truncated with respect to a threshold ε_{svd} to maintain a small, but accurate representation. A schematic illustration for the computation of the SVD update is given in Fig. 2.4.

Before we give a more detailed algorithmic description of the SVD updating step, and, in particular, its realization on distributed data, we discuss the efficient on-the-fly evaluation of the Newton update as well as an improved column updating scheme for $\widetilde{\boldsymbol{W}}_k$ on distributed data.

Efficient Limited-Memory Computation of the Newton Update. Unlike for the plain-vanilla MV method, the quasi-Newton update for the RS-SVD restart variant is computed as on-the-fly evaluation of a series of cheap matrix-vector operations

$$\Delta \boldsymbol{x}^{k+1} = -\overline{\boldsymbol{\Psi}}\left(\overline{\boldsymbol{\Sigma}}(\overline{\boldsymbol{\Phi}}^T\boldsymbol{r}^k)\right) - \sum_{q=n-s}^{n} \widetilde{\boldsymbol{W}}_{k_q}\left(\boldsymbol{V}_{k_q}^{\dagger}\boldsymbol{r}^k\right), \tag{2.35}$$

where $s \leqslant m$ is the number of completed time steps since the last restart. In particular, this reduces the quadratic $\mathcal{O}(N^2/p) + \mathcal{O}(N\log p)$ cost of plain-vanilla MV to the linear[28] parallel runtime complexity of

$$\mathcal{O}(\overline{\kappa}N/p) + \mathcal{O}(m\overline{\eta}N/p) + \mathcal{O}(m\overline{\eta}\log p) \tag{\textit{\textbf{lmDx}}}$$

for the computation of the quasi-Newton update. The runtime results from m matrix-vector products $\gamma = \boldsymbol{V}_{k_q}^{\dagger}\boldsymbol{r}^k$, computed in $\mathcal{O}(\overline{\eta}N/p) + \mathcal{O}(N\log p)$ time each (local matrix-vector products and allreduce operation), and m fully local matrix-vector multiplications $\widetilde{\boldsymbol{W}}_{k_q}\gamma$ with runtime complexity $\mathcal{O}(\overline{\eta}N/p)$ each. The intermediate result $\gamma \in \mathbf{R}^{k_q}$ is stored locally on each rank. The matrices $\overline{\boldsymbol{\Psi}}$ and $\overline{\boldsymbol{\Phi}}$ are distributed in row-blocks of size $N/p \times \overline{\kappa}$ and the computation of the products on distributed data is analogous to the above described matrix-vector multiplications involving $\widetilde{\boldsymbol{W}}_{k_q}$ and $\boldsymbol{V}_{k_q}^{\dagger}$.

Efficient Column-Updating of $\widetilde{\boldsymbol{W}}$. The matrix $\widetilde{\boldsymbol{W}}_k$ depends on changes that propagate from the matrices $\boldsymbol{W}_k$ and $\boldsymbol{V}_k$. The latter change from iteration to iteration by adding a column to the front and possibly deleting a column at the end or in the middle, due to filtering. Since $\boldsymbol{M}_{prev}^{-1}$ remains constant within a time step, these events can be translated to efficient column updating operations, avoiding re-computation of the entire matrix. In the standard case, we update $\widetilde{\boldsymbol{W}}_k := \boldsymbol{W}_k - \boldsymbol{M}_{prev}^{-1}\boldsymbol{V}_k$ by adding the new column $\tilde{\boldsymbol{w}} := \Delta\tilde{\boldsymbol{x}}_{k-1}^k - \boldsymbol{M}_{prev}^{-1}\Delta\boldsymbol{r}_{k-1}^k$, i. e.,

$$(\widetilde{\boldsymbol{W}}_{k_n})_{i,0} = \tilde{\boldsymbol{w}} = (\boldsymbol{W}_{k_n})_{i,0} - \sum_{q=n-s}^{n} \widetilde{\boldsymbol{W}}_{k_q}\left(\boldsymbol{V}_{k_q}^{\dagger}\cdot(\boldsymbol{V}_{k_n})_{i,0}\right) \tag{2.36}$$

The computation of the new column on distributed data is analogous to the computation of the Newton update above. The updating scheme reduces the quadratic runtime $\mathcal{O}(\eta N/p) + \mathcal{O}(\eta N\log p)$ for (***Wtil***) to

$$\mathcal{O}(m\overline{\eta}N/p) + \mathcal{O}(m\overline{\eta}\log p). \tag{\textit{\textbf{upWtil}}}$$

Deletion of a column can be done without additional computations in $\mathcal{O}(1)$.

Efficient Computation of Truncated SVD Update. In the sequel, we give more details on the

[28] The rank $\overline{\kappa}$ of the truncated singular value representation of the inverse Jacobian approximation is assumed to be significantly smaller than $\sqrt[3]{N}$ and the upper bound $\overline{\eta} = \max(k_{n-m}, k_{n-m+1}, \ldots, k_n)$ equals the number of quasi-Newton iteration per time step and is typically small for MV.

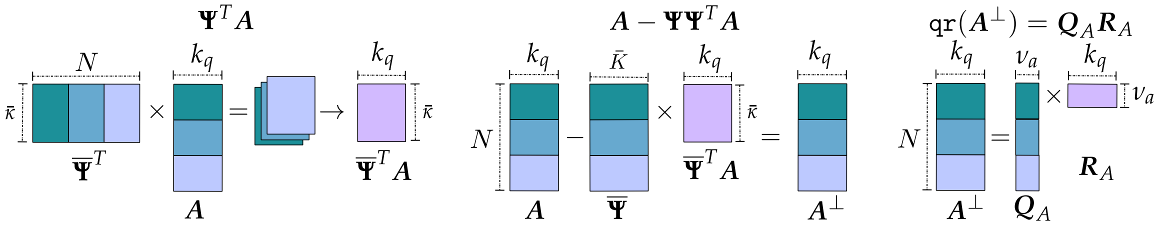


FIGURE 2.5 *Schematic view of block partitioning of matrices and the distributed memory realization of the matrix products, as required in step (i) of the SVD update. Shown are the involved steps to compute the orthogonal components of* $\boldsymbol{A}$ *and* $\boldsymbol{B}$ *with respect to* $\overline{\boldsymbol{\Psi}}$ *and* $\overline{\boldsymbol{\Phi}}$*, respectively.* $\bar{\kappa}$ *denotes the rank of the existing, truncated SVD representation,* k_q *the number of columns to be considered in the update, and* ν_a *and* ν_b *the dimensionalities of the column space of the orthogonal components* $\boldsymbol{A}^\perp$ *and* $\boldsymbol{B}^\perp$*.*

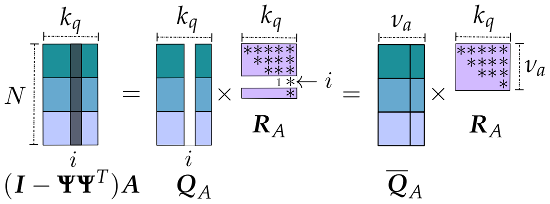


FIGURE 2.6 *Schematic view of eliminating a linearly dependent column from the QR-decomposition for* $(\boldsymbol{I} - \overline{\boldsymbol{\Psi}}\,\overline{\boldsymbol{\Psi}}^T)\boldsymbol{A}$*.*

algorithmic realization of the SVD update strategy, presented above and discuss runtime complexities for parallel execution on distributed data.

Algorithmic Steps on Distributed Data. We partition the matrices $\overline{\boldsymbol{\Psi}}$ and $\overline{\boldsymbol{\Phi}} \in \mathbf{R}^{N\times\bar{\kappa}}$ into row-blocks of size $N/p \times \bar{\kappa}$ and distribute the inverse Jacobian representation on p ranks. The rank $\bar{\kappa}$ of the truncated SVD is assumably small and the diagonal of $\overline{\boldsymbol{\Sigma}}$ is held locally on every rank. When restart is triggered, m consecutive rank-k_q updates (2.32) are performed. This involves the following computational steps (compare Fig. 2.4):

(i) Compute the orthogonal components of the matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ that form the update $\boldsymbol{A}\boldsymbol{B}^T$, with respect to the left- and right-singular vectors $\overline{\boldsymbol{\Psi}}$ and $\overline{\boldsymbol{\Phi}}$ of the existing SVD representation (cf. (2.32)), and factorize the new orthogonal information by means of QR-decomposition:

$$\text{(2.37a)} \qquad \boldsymbol{A}^\perp := (\boldsymbol{I} - \overline{\boldsymbol{\Psi}}\,\overline{\boldsymbol{\Psi}}^T)\boldsymbol{A}, \qquad \mathbf{qr}(\boldsymbol{A}^\perp) = \boldsymbol{Q}_A\boldsymbol{R}_A$$

$$\text{(2.37b)} \qquad \boldsymbol{B}^\perp := (\boldsymbol{I} - \overline{\boldsymbol{\Phi}}\,\overline{\boldsymbol{\Phi}}^T)\boldsymbol{B}, \qquad \mathbf{qr}(\boldsymbol{B}^\perp) = \boldsymbol{Q}_B\boldsymbol{R}_B,$$

with $\boldsymbol{Q}_A \in \mathbf{R}^{N\times\nu_a}$, $\boldsymbol{R}_A \in \mathbf{R}^{\nu_a\times k_q}$, $\boldsymbol{Q}_B \in \mathbf{R}^{N\times\nu_b}$, and $\boldsymbol{R}_B \in \mathbf{R}^{\nu_b\times k_q}$. This decomposes into the following sub-steps:

(a) Compute the products $\widetilde{\boldsymbol{A}} := \overline{\boldsymbol{\Psi}}^T\boldsymbol{A}$ and $\overline{\boldsymbol{\Psi}}\widetilde{\boldsymbol{A}}$, and compute $\boldsymbol{A}^\perp = \boldsymbol{A} - \overline{\boldsymbol{\Psi}}\widetilde{\boldsymbol{A}}$; analogously for $\boldsymbol{B}$. A schematic illustration of the parallel computations is given in Fig. 2.5. For $\widetilde{\boldsymbol{A}} = \overline{\boldsymbol{\Psi}}^T\boldsymbol{A}$, every rank computes its local contribution, followed by an allreduce step. The result is replicated locally on every rank, as the rank of the truncated SVD $\bar{\kappa}$ and the number of columns k_q are assumably small. The parallel complexity for this step is given by $\mathcal{O}(\bar{\eta}\bar{\kappa}N/p) + \mathcal{O}(\bar{\eta}\bar{\kappa}\log p)$. Computing $\boldsymbol{A}^\perp$ as $\boldsymbol{A} - \overline{\boldsymbol{\Psi}}\widetilde{\boldsymbol{A}}$ is fully local and requires no communication. The computational cost are $\mathcal{O}(\bar{\eta}\bar{\kappa}N/p)$.

(b) Compute the QR-decomposition $\boldsymbol{A}^\perp = \boldsymbol{Q}_A\boldsymbol{R}_A$ and $\boldsymbol{B}^\perp = \boldsymbol{Q}_B\boldsymbol{R}_B$, respectively. This is done using the updating scheme described earlier, yet we insert the columns on the right as there is no need to prioritize columns. As a result, no additional Givens rotations are required.

The accumulated parallel runtime complexity for steps (a) and (b) is therefore given by

$$\mathcal{O}(\overline{\eta}\,\overline{\kappa}\,{}^{N}\!/_{p}) + \mathcal{O}(\overline{\eta}\,\overline{\kappa}\log p).$$

The numbers ν_a and ν_b of new orthogonal modes added to the SVD become smaller as the simulation proceeds, since a lot of information is already encoded within the truncated SVD representation. In particular, we have $\nu_a, \nu_b \ll k_q$. Due to numerical errors, however, $\boldsymbol{A}^{\perp}$ is likely to have the same number of non-zero columns as $\boldsymbol{A}$. Normalization of the columns $\boldsymbol{a}_i \in \operatorname{img}(\overline{\boldsymbol{\Psi}})$ within the QR-decomposition creates problems when dividing by (almost) zero. The respective columns need to be deleted by dropping them from $\boldsymbol{Q}_A$ and eliminating the corresponding row from $\boldsymbol{R}_A$, but not the column[29] (compare Fig. 2.6). Note that, if columns are deleted, as shown in Fig. 2.6, the additional application of Givens rotations is required to maintain QR characteristics. In our case, this is rendered moot as we only eliminate rows at the end.

(ii) Construct the matrix $\boldsymbol{S} \in \mathbf{R}^{(\overline{\kappa}+\nu_a)\times(\overline{\kappa}+\nu_b)}$ according to (2.33) locally on every rank. After step (i), the components $\overline{\boldsymbol{\Psi}}^T\boldsymbol{A}$, $\overline{\boldsymbol{\Phi}}^T\boldsymbol{B}$, $\boldsymbol{R}_A$ and $\boldsymbol{R}_B$ are readily available. All multiplications are executed fully locally; the computational runtime complexity accumulates to $\mathcal{O}(\overline{\eta}(\overline{\kappa}+\overline{\nu}))$, if $\overline{\nu} := \max(\nu_a, \nu_b)$.

(iii) Diagonalize $\boldsymbol{S}$ as $\boldsymbol{\Psi}'^T\boldsymbol{S}\boldsymbol{\Phi}' = \boldsymbol{\Sigma}'$ and re-establish the SVD character for the truncated SVD representation. This step decomposes into:

(a) Compute the SVD $\boldsymbol{S} = \boldsymbol{\Psi}'\boldsymbol{\Sigma}'\boldsymbol{\Phi}'^T$. This step is sequential and yields cubic runtime complexity $\mathcal{O}((\overline{\kappa}+\overline{\nu})^3)$ in terms of the dimensionality of $\boldsymbol{S}$.

(b) Rotate left- and right-singular vector sub-spaces, i.e., compute $\widetilde{\boldsymbol{\Psi}}$ and $\widetilde{\boldsymbol{\Phi}}$ as

$$\underbrace{\left([\overline{\boldsymbol{\Psi}}\ \boldsymbol{Q}_A]\ \overline{\boldsymbol{\Psi}}'\right)}_{\widetilde{\boldsymbol{\Psi}}}\overline{\boldsymbol{\Sigma}}'\underbrace{\left([\overline{\boldsymbol{\Phi}}\ \boldsymbol{Q}_B]\ \overline{\boldsymbol{\Phi}}'\right)^T}_{\widetilde{\boldsymbol{\Phi}}^T}.$$

Fig. 2.7 shows the rotations schematically. As $\overline{\boldsymbol{\Psi}}'$ and $\overline{\boldsymbol{\Phi}}'$ are replicated on each rank, this step is embarrassingly parallel without need of communication. The runtime complexity is given by $\mathcal{O}((\overline{\kappa}+\overline{\nu})^2\,{}^{N}\!/_{p})$.

(c) Truncate the SVD representation of $\boldsymbol{M}^{-1}$ to the most dominant modes. That is, find $\overline{\kappa}' \in \{2, \ldots, \overline{\kappa} + \min(\nu_a, \nu_b)\}$ for which ${}^{\sigma_{\overline{\kappa}'}}\!/_{\sigma_1} \leqslant \varepsilon_{svd} < {}^{\sigma_{\overline{\kappa}'-1}}\!/_{\sigma_1}$ and truncate

$$\overline{\boldsymbol{\Psi}}\,\overline{\boldsymbol{\Sigma}}\,\overline{\boldsymbol{\Phi}}^T \leftarrow \widetilde{\boldsymbol{\Psi}}\,\widetilde{\boldsymbol{\Sigma}}\,\widetilde{\boldsymbol{\Phi}}^T \quad \text{as}$$
$$\overline{\boldsymbol{\Psi}} = [\widetilde{\boldsymbol{\Psi}}_{\cdot,1}, \ldots, \widetilde{\boldsymbol{\Psi}}_{\cdot,\overline{\kappa}'}], \quad \overline{\boldsymbol{\Phi}} = [\widetilde{\boldsymbol{\Phi}}_{\cdot,1}, \ldots, \widetilde{\boldsymbol{\Phi}}_{\cdot,\overline{\kappa}'}], \quad \text{and} \quad \overline{\boldsymbol{\Sigma}} = \operatorname{diag}(\sigma_1, \ldots, \sigma_{\overline{\kappa}'}).$$

Summarizing, the overall parallel runtime complexity to perform a rank-$\overline{\eta}$ update on the truncated SVD representation of the form (2.32) with $\boldsymbol{A}\boldsymbol{B}^T = \widetilde{\boldsymbol{W}}_{k_q}\boldsymbol{V}_{k_q}^{\dagger}$, accumulates to

$$\begin{aligned} &\mathcal{O}(\overline{\eta}(\overline{\kappa}+2)\,{}^{N}\!/_{p}) + \mathcal{O}((\overline{\kappa}+\overline{\nu})^2({}^{N}\!/_{p}+\overline{\eta})) + \mathcal{O}((\overline{\kappa}+\overline{\nu})^3) + \mathcal{O}(\overline{\eta}(\overline{\kappa}+1)\log p) \\ (\boldsymbol{updSVD}) \qquad &\in \mathcal{O}(\overline{\kappa}({}^{N}\!/_{p}+\overline{\eta})) + \mathcal{O}(\overline{\kappa}^3) \quad \text{for } \overline{\kappa} > \overline{\eta}. \end{aligned}$$

[29] The elimination of linearly dependent columns for the QR-decomposition as used for the SVD update differs from the earlier described scheme. For the QR decomposition of $\boldsymbol{V}_k$, we delete the corresponding row and column in $\boldsymbol{R}$ in case of linear dependence, which corresponds to deletion of the column from the matrix $\boldsymbol{V}_k$. In this context, however, a column $\boldsymbol{A} \ni \boldsymbol{a}_i \in \operatorname{img}(\overline{\boldsymbol{\Psi}})$ is not related to $\overline{\boldsymbol{\Psi}}^T\boldsymbol{A}$. Thus, we cannot delete the corresponding row in $\overline{\boldsymbol{\Psi}}^T\boldsymbol{A}$ as before.

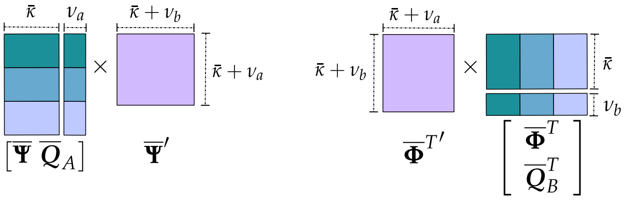

$\bar{\kappa}$
ν_a
$\bar{\kappa}+\nu_b$
$\times$
$\bar{\kappa}+\nu_a$
$[\overline{\boldsymbol{\Psi}}\ \overline{\boldsymbol{Q}}_A]$
$\overline{\boldsymbol{\Psi}}'$
$\bar{\kappa}+\nu_a$
$\bar{\kappa}+\nu_b$
$\times$
$\bar{\kappa}$
ν_b
$\overline{\boldsymbol{\Phi}}^{T\prime}$
$\begin{bmatrix}\overline{\boldsymbol{\Phi}}^T\\ \overline{\boldsymbol{Q}}_B^T\end{bmatrix}$

FIGURE 2.7
Schematic view of the parallel implementation of the sub-space rotations.

For a complete update of the inverse Jacobian sub-space, such rank-k_q updates have to be performed m times. Restart occurs periodically after m time steps, hence, (***updSVD***) can be seen as cost per time step. If amortized over the number of iterations, the cost can be bounded by a parallel runtime complexity of

$$(\textbf{\textit{updSVD / it}}) \qquad \mathcal{O}(\bar{\kappa}^2 N/p) + \mathcal{O}(\bar{\kappa}/\bar{\eta})$$

per iteration. With this, the amortized cost for the parallel runtime of the scalable and efficient RS-SVD MV alternative can be stated as

$$(\textbf{\textit{RS-SVD / it}}) \quad \mathcal{O}(\eta^2 N/p) + \mathcal{O}(\eta \log p) + \mathcal{O}(\eta^3) + \mathcal{O}(m\bar{\eta}N/p) + \mathcal{O}(m\bar{\eta}\log p) + \mathcal{O}(\bar{\kappa}^2 N/p) + \mathcal{O}(\bar{\kappa}/\bar{\eta})$$

per quasi-Newton iteration. For a small rank of the truncated SVD representation of the inverse Jacobian, the method exhibits linear runtime and memory complexity in the number of unknowns N.

A Remark on Pre-Scaling for RS-SVD. We mentioned earlier that in cases where the underlying fixed-point operator of the system we would like to solve stems from a Jacobi-type equation coupling, pre-scaling of variables to ensure equal weighting, and good conditioning might become necessary. We explained this concept in §2.2.1. Here, we outline modifications for the pre-scaling that are required when applied to the RS-SVD algorithm. When computing the SVD update, QR-factorizations of the matrices $\widetilde{\boldsymbol{W}}_{k_q}^{\perp} = (\boldsymbol{I} - \overline{\boldsymbol{\Psi}}\,\overline{\boldsymbol{\Psi}}^T)\widetilde{\boldsymbol{W}}_{k_q}$ and $\boldsymbol{V}_{k_q}^{\dagger\,\perp} = (\boldsymbol{I} - \overline{\boldsymbol{\Phi}}\,\overline{\boldsymbol{\Phi}}^T)\boldsymbol{V}_{k_q}^{\dagger}$ need to be computed, cf. (2.37). Therefore, the SVD update has to be computed from pre-scaled matrices $\widetilde{\boldsymbol{W}}_{k_q}' := \boldsymbol{\Lambda}_{\bar{k}}\widetilde{\boldsymbol{W}}_{k_q}$ and $\boldsymbol{V}_{k_q}^{\dagger\,\prime} := \boldsymbol{V}_{k_q}^{\dagger}\boldsymbol{\Lambda}_{\bar{k}}^{-1}$ for $q = n-m, \dots, n$, i.e., at restart we compute

$$\overline{\boldsymbol{\Psi}}\,\overline{\boldsymbol{\Sigma}}\,\overline{\boldsymbol{\Phi}}^T + \sum_{q=n-m}^{n} \boldsymbol{\Lambda}_{\bar{k}}\widetilde{\boldsymbol{W}}_{k_q}\boldsymbol{V}_{k_q}^{\dagger}\boldsymbol{\Lambda}_{\bar{k}}^{-1}, \tag{2.38}$$

with $\bar{k} = k_m$. In other words, we maintain an SVD representation of the pre-scaled inverse Jacobian matrix. In (2.38), it is important to keep the pre-scaling weights fixed to a certain value after the initial SVD construction; all subsequent updates have to be scaled with the same weights $\boldsymbol{\Lambda}_{\bar{k}}$. Otherwise, the matrices $\overline{\boldsymbol{\Psi}}$ and $\overline{\boldsymbol{\Phi}}$ are not unitary matrices any more and the singular value decomposition properties fall apart. Our numerical analysis showed that the weights for, e.g., the residual-sum pre-scaling adjust to small oscillations around a fixed value after an initial phase. Therefore, keeping the weights fixed after an initialization phase of m time steps has not shown unfavorable behavior. In particular, we aim for chunk sizes $m \in \{8, 16, 32\}$.

Complexity Overview. We wrap up this section with an overview of the parallel runtime complexities of the presented quasi-Newton variants in Tab. 2.1.

TABLE 2.1 *Overview of the parallel runtime complexities of the algorithmic building blocks for the LS, MV and RS-SVD quasi-Newton multi-secant variants. Parallel runtime complexities are given with respect to the number of unknowns N (at the coupling-interface), the number of columns stored in the matrices $\boldsymbol{W}$ and $\boldsymbol{V}$ (denoted by η for the LS method and by an upper bound $\bar{\eta} = max(k_{n-m}, \dots, k_n)$ for the MV-RS-SVD method), the rank $\overline{\kappa}$ of the truncated SVD representation of $\boldsymbol{M}^{-1}$, the chunk size m defining restart periods, and the number p of parallel MPI ranks. The number of columns η within the least-squares system is typically much higher than for the MV method due to a high number ξ of reused time steps (for the MV, we use $\xi = 0$), i,.e., it holds $\bar{\eta} \ll \eta$. For most cases also $\eta > \overline{\kappa}$ holds true. The effect for the runtime complexity can be seen from the scaling results in §4.3.*

Stage	LS	MV	MV-RS-SVD
QR-comp.	$(\boldsymbol{QR})$ $\mathcal{O}(\eta^3 N/p) + \mathcal{O}(\eta^2 \log p) + \mathcal{O}(\eta^4)$	$(\boldsymbol{updQR})$ $\mathcal{O}(\eta^2 N/p) + \mathcal{O}(\eta \log p) + \mathcal{O}(\eta^3)$	
pseudo inverse	$(\boldsymbol{backS})$ $\widehat{\boldsymbol{R}}\boldsymbol{\alpha} = -\widehat{\boldsymbol{Q}}^T \boldsymbol{r}$ $\mathcal{O}(\eta N/p) + \mathcal{O}(\eta \log p) + \mathcal{O}(\eta^2)$	$(\boldsymbol{iV})$ $\widehat{\boldsymbol{R}}\boldsymbol{V}^\dagger = \widehat{\boldsymbol{Q}}^T$ $\mathcal{O}(\eta^2 N/p)$	$(\boldsymbol{iV})$ $\widehat{\boldsymbol{R}}\boldsymbol{V}^\dagger = \widehat{\boldsymbol{Q}}^T$ $\mathcal{O}(\eta^2 N/p)$
Newton update	$(\boldsymbol{Wa})$ $\Delta \boldsymbol{x} = \boldsymbol{W}_k \boldsymbol{\alpha}$ $\mathcal{O}(N/p)$	$(\boldsymbol{Mr})$ $\Delta \boldsymbol{x} = \boldsymbol{M}^{-1} \boldsymbol{r}$ $\mathcal{O}(N/p) + \mathcal{O}(N \log p)$	$(\boldsymbol{lmDx})$ $\Delta \boldsymbol{x} = -\overline{\boldsymbol{\Psi}}(\overline{\boldsymbol{\Sigma}\boldsymbol{\Phi}}^T \boldsymbol{r}) - \sum_{q=n-m}^{n} \widetilde{\boldsymbol{W}}_{k_q} (\boldsymbol{V}_{k_q}^\dagger \boldsymbol{r})$ $\mathcal{O}(\overline{\kappa} N/p) + \mathcal{O}(m\bar{\eta} N/p) + \mathcal{O}(m\bar{\eta} \log p)$
comp. $\widetilde{\boldsymbol{W}}$	—	$(\boldsymbol{Wtil})$ $\widetilde{\boldsymbol{W}} = (\boldsymbol{W} - \boldsymbol{M}_{prev}^{-1} \boldsymbol{V})$ $\mathcal{O}(\eta N^2/p) + \mathcal{O}(\eta N \log p)$	$(\boldsymbol{upWtil})$ $(\widetilde{\boldsymbol{W}})_{i,0} = (\boldsymbol{W})_{i,0} - \sum_{q=0}^{m} \widetilde{\boldsymbol{W}}_{k_q} (\boldsymbol{V}_{k_q}^\dagger (\boldsymbol{V})_{i,0})$ $\mathcal{O}(m\bar{\eta} N/p) + \mathcal{O}(m\bar{\eta} \log p)$
restart/ build $\boldsymbol{M}^{-1}$	—	$(\boldsymbol{WV})$ $\widetilde{\boldsymbol{W}}_k \boldsymbol{V}_k^\dagger$ $\mathcal{O}(\eta N/p)$	$(\boldsymbol{updSVD\,/\,it})$ $\overline{\boldsymbol{\Psi}}\,\overline{\boldsymbol{\Sigma}}\,\overline{\boldsymbol{\Phi}}^T + \sum_{q=n-m}^{n} \widetilde{\boldsymbol{W}}_{k_q} \boldsymbol{V}_{k_q}^\dagger$ $\mathcal{O}(\overline{\kappa}^2 N/p) + \mathcal{O}(\overline{\kappa}/\bar{\eta})$
Total / Iteration	$\mathcal{O}(\eta^2 N/2) + \mathcal{O}(\eta^4) + \mathcal{O}(\eta^2 \log p)$	$\mathcal{O}(\eta N^2/p) + \mathcal{O}(\eta N \log p)$	$\mathcal{O}(\eta^2 N/p) + \mathcal{O}(\eta \log p) + \mathcal{O}(\eta^3) + \mathcal{O}(m\bar{\eta} N/p) + \mathcal{O}(m\bar{\eta} \log p) + \mathcal{O}(\overline{\kappa}^2 N/p) + \mathcal{O}(\overline{\kappa}/\bar{\eta})$

2.3 Quasi-Newton for Non-Linear Optimization Problems (BFGS)

This section contrasts methodological and structural similarities and differences for quasi-Newton methods used in the classic context of non-linear optimization to the above presented fixed-point iteration context. In particular, we relate LBFGS, the most common quasi-Newton method in non-linear optimization, to our previously studied variants. This section is kept shorter and makes no attempt to be exhaustive; it serves as a preface and introduction to the discussion of the application of inexact Newton methods, and specifically the LBFGS quasi-Newton method, for PDE-constrained optimization problems in §6.4.

Non-linear optimization refers to the general problem of finding a candidate $\boldsymbol{x}^\star \in \mathbf{R}^N$ such that

$$f(\boldsymbol{x}^\star) \rightarrow \min \tag{2.39}$$

for a non-linear, possibly non-convex, ***objective function*** $f\colon \mathbf{R}^N \rightarrow \mathbf{R}$ with sufficient regularity. Within this work, we consider minimization of Lagrangian functions arising from PDE-constrained optimization problems of the form (2.3); the discussed concepts are, however, fully generic. Finding a minimum translates to vanishing first order derivatives, i.e., solving for a root $\boldsymbol{x}^\star$ of the gradient $\boldsymbol{g} := \nabla f\colon \mathbf{R}^N \rightarrow \mathbf{R}^N$ of the objective function, i.e., $\boldsymbol{g}(\boldsymbol{x}^\star) = \boldsymbol{0}$. The gradient $\boldsymbol{g}(\boldsymbol{x})$ relates to the operator $\boldsymbol{\mathcal{R}}(\boldsymbol{x})$ introduced in the opening words §2.1 as given in (2.4). A vanishing gradient of the objective function is a necessary, though not sufficient condition for a local minimum: The objective function's Hessian matrix $\boldsymbol{H} := \nabla\nabla^T f\colon \mathbf{R}^N \rightarrow \mathbf{R}^{N \times N}$ additionally needs to be symmetric positive definite for an admissible solution $\boldsymbol{x}^\star$.

Efficient optimization algorithms require an evaluation of the gradient $\boldsymbol{g}$ of the objective function and employ an ***iterative*** approach $\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \alpha \Delta \boldsymbol{x}^k$ with step-size α and step-direction $\Delta \boldsymbol{x}$. A straightforward choice is ***gradient-descent***, i.e., use the negative gradient $\Delta \boldsymbol{x}^k = -\boldsymbol{g}(\boldsymbol{x}^k)$ as step-direction. This method is, however, only linearly convergent and not transformation invariant, meaning that scaling the descent direction by a local measure can substantially improve convergence.

One possible (or the "correct") scaling is to use local curvature information, i.e., scale with the inverse Hessian matrix for the current iterate as $\Delta \boldsymbol{x}^k = -[\boldsymbol{H}(\boldsymbol{x}^k)]^{-1}\boldsymbol{g}(\boldsymbol{x}^k)$, giving raise to the locally quadratically convergent Newton method as outlined in (2.5a). Its basic idea is to locally form a quadratic approximation

$$m_k(\Delta \boldsymbol{x}^k) = f(\boldsymbol{x}^k) + \boldsymbol{g}(\boldsymbol{x}^k)^T \Delta \boldsymbol{x}^k + 1/2 (\Delta \boldsymbol{x}^k)^T \boldsymbol{H}(\boldsymbol{x}^k) \Delta \boldsymbol{x}^k$$

of f at $\boldsymbol{x}^k + \Delta \boldsymbol{x}^k$ with the analytic minimizer $\Delta \boldsymbol{x}^k = [\boldsymbol{H}(\boldsymbol{x}^k)]^{-1}\boldsymbol{g}(\boldsymbol{x}^k)$ obtained as a solution of the respective linear system. Newton's method is ***efficient*** in the sense that (*i*) for a quadratic objective function, this algorithm would solve the minimization problem in one step, and that (*ii*) in a neighborhood of the solution, the method features rapid quadratic convergence [Den77]. The methods' main drawback is, however, as discussed above in §2.1, its significant computational cost ($\mathcal{O}(N^2)$ for constructing the Hessian, $\mathcal{O}(N^3)$ for Hessian inversion), rendering the method impractical for high dimensional or large-scale problems. Various forms of inexact-Newton methods that try to counterbalance computational cost and convergence speed are used in practice; we give a discussion on different approaches in §6.4.3. A common solver in PDE-constrained optimization is the Gauß-Newton-Krylov method (see §6.4.4), combining inexact Krylov solves for the Hessian inversion with a cheaper symmetric positive definite Hessian approximation, omitting second order terms; yet the cost for a Hessian matrix-vector operation are typically still high.

Powerful and widely used alternatives are ***quasi-Newton methods***, which we will discuss here. As we have seen above, these methods combine possibly fast convergence with low computational cost by maintaining a low-rank approximation of the (inverse) Hessian matrix based on only gradient information. In this sense, quasi-Newton methods are essentially learning methods. The optimization and fixed-point communities have developed different manifestations due to different requirements in the respective field. In the context of optimization and function minimization, it is crucial for the quasi-Newton update to maintain an approximation of a symmetric positive definite matrix, since it is sought for the approximation of a Hessian matrix rather then a Jacobian. The most widely known variants in optimization are ***SR1*** [Bro67], ***DFP*** [Fle63; Den68], and ***BFGS*** [Fle63; Noc06; Noc80; Bro69; Sha70; Fle70; Gol70; Den68]. Recently, Hennig et al. presented variations capable to cope with noisy evaluations for probabilistic optimization [Hen13a] and reinterpreted quasi-Newton multi-secant methods as approximate Bayesian linear regression [Hen13b]. Many intermediate forms, mixtures and advanced alternatives exist, and the literature is extensive. An excellent overview is given in the text book [Noc06] and the older, yet ever relevant, insightful, and extensive review paper [Den77] including theoretical discussion.

The ***BFGS method*** and, in particular, its limited memory alternative LBFGS is commonly accepted to be the most efficient and powerful quasi-Newton method for non-linear optimization. Its update formula maintains an symmetric positive definite (spd) matrix and, as opposed to DFP, it directly approximates the inverse Hessian, rendering the linear system solve dispensable. As a result, the method often features super-linear to near-quadratic convergence rates, while featuring overall ***linear runtime complexity***. In the following, we limit our discussions to the BFGS and LBFGS quasi-Newton method and work out conceptual similarities and differences to earlier discussed multi-secant methods.

2.3.1 Methodological Components

Following the general idea of quasi-Newton as introduced in §2.1 in a more generic notation, we estimate the objective function's (or Lagrangian's) Hessian with iterate- and gradient-differences throughout the iterations, i.e.,

$$\text{(2.40a)}\qquad \boldsymbol{W}_k^{\eta} = \left[\Delta \boldsymbol{x}_{k-1}^{k}, \Delta \boldsymbol{x}_{k-2}^{k}, \ldots, \Delta \boldsymbol{x}_{k-\eta}^{k}\right], \qquad \text{with } \Delta \boldsymbol{x}_i^k = \boldsymbol{x}^k - \boldsymbol{x}^i\,,$$

$$\text{(2.40b)}\qquad \boldsymbol{V}_k^{\eta} = \left[\Delta \boldsymbol{g}_{k-1}^{k}, \Delta \boldsymbol{g}_{k-2}^{k}, \ldots, \Delta \boldsymbol{g}_{k-\eta}^{k}\right], \qquad \text{with } \Delta \boldsymbol{g}_i^k = \boldsymbol{g}(\boldsymbol{x}^k) - \boldsymbol{g}(\boldsymbol{x}^i).$$

which, with the reasoning

$$\Delta \boldsymbol{g}(\boldsymbol{x}_{k-1}^k) = \boldsymbol{g}(\boldsymbol{x}^k) - \boldsymbol{g}(\boldsymbol{x}^{k-1}) \approx \boldsymbol{H}(\boldsymbol{x}^k)\Delta \boldsymbol{x}_{k-1}^k$$

and assuming a sufficiently good representation of the objective function by a local quadratic model, motivates the (multi-)secant equation (2.7) as the core ingredient for quasi-Newton methods. Defining $\boldsymbol{\mathcal{R}} := \boldsymbol{g}$ and $\boldsymbol{M} := \boldsymbol{H}$ (or $\boldsymbol{M}^{-1} := \boldsymbol{H}^{-1}$, respectively), the generic quasi-Newton formulation (2.9) is inherited. To facilitate the comparison, we adopt the notation from §2.1 and refer to the inverse Hessian approximation as $\boldsymbol{M}^{-1}$ from now on.

Symmetric, Positive Definite Rank-1 Updates. As opposed to the application in fixed-point problems, we need the quasi-Newton update to respect two additional characteristics when used for function minimization, that is, to maintain (*i*) ***symmetry*** and (*ii*) ***positive definiteness*** of the estimator. To ensure these properties, the norm minimization in the generic formulation (2.9) is imposed under a ***weighted Frobenius norm*** with the symmetric matrix $\boldsymbol{Z}$ and the additional symmetry constraint $\boldsymbol{M}^{-T} = \boldsymbol{M}^{-1}$ is added, i.e., we solve

$$\text{(2.41)}\qquad \left\| \boldsymbol{M}^{-1} - \boldsymbol{M}_{prev}^{-1} \right\|_{\boldsymbol{Z}} \to \min \quad \text{subject to} \quad \boldsymbol{M}^{-1}\boldsymbol{V}_k^{\eta} = \boldsymbol{W}_k^{\eta} \;\text{ and }\; \boldsymbol{M}^{-T} = \boldsymbol{M}^{-1},$$

with $\|\cdot\|_{\boldsymbol{Z}} := \|\boldsymbol{Z}^{1/2} \cdot \boldsymbol{Z}^{1/2}\|_F$ using any symmetric matrix $\boldsymbol{Z}$[30] fulfilling the (multi-)secant equation $\boldsymbol{Z}\boldsymbol{W}_k = \boldsymbol{V}_k$.

Broyden-Fletcher-Goldfarb-Shanno (BFGS). Solving problem (2.41) for $\boldsymbol{M}^{-1}$ with $\boldsymbol{W}_k = \Delta \boldsymbol{x}_{k-1}^k$ and $\boldsymbol{V}_k = \Delta \boldsymbol{g}_{k-1}^k$ (i.e., $\eta = 1$) yields the ***symmetric BFGS rank-1 update***

$$\text{(2.42)}\quad \boldsymbol{M}^{-1,(k+1)} = \left(\boldsymbol{I} - \boldsymbol{W}_k(\boldsymbol{V}_k^T\boldsymbol{W}_k)^{-1}\boldsymbol{V}_k^T\right)\boldsymbol{M}_{prev}^{-1,(k)}\left(\boldsymbol{I} - \boldsymbol{V}_k(\boldsymbol{V}_k^T\boldsymbol{W}_k)^{-1}\boldsymbol{W}_k^T\right) + \boldsymbol{W}_k(\boldsymbol{V}_k^T\boldsymbol{W}_k)^{-1}\boldsymbol{W}_k^T,$$

which, using $\rho_k := (\boldsymbol{V}_k^T\boldsymbol{W}_k)^{-1}$ and $\boldsymbol{U} := (\boldsymbol{I} - \rho_k\boldsymbol{V}_k\boldsymbol{W}_k^T)$ can be re-written (in standard literature presentation such as, e.g., in Nocedal & Wright [Noc06]) as

$$\text{(2.43)}\qquad \boldsymbol{M}^{-1,(k+1)} = \boldsymbol{U}_k^T\,\boldsymbol{M}_{prev}^{-1,(k)}\,\boldsymbol{U}_k + \rho_k\boldsymbol{W}_k\boldsymbol{W}_k^T.$$

Limited-Memory BFGS (LBFGS). With a view to large-scale simulations, storing the inverse Hessian estimator $\boldsymbol{M}_{prev}^{-1,(k)}$ from the previous iteration is prohibitive. Similar to the MV restart alternative in §2.2.2, a limited number μ of vectors is kept in storage; a sliding-window approach[31] is employed

[30] For concreteness, say, we exemplarily use the ***average Hessian*** $\overline{\boldsymbol{H}}$, as weighting matrix, i.e., $\boldsymbol{Z} := \overline{\boldsymbol{H}} = \left[\int_0^1 \nabla^2 f(\boldsymbol{x}^k + \tau\Delta\boldsymbol{x}^k)\,\mathrm{d}\tau\right]$. Note that $\boldsymbol{Z}$ does not need to be given explicitly in the algorithm, we only need to state its theoretical existence.

[31] With sliding-window, we mean always keeping a window of μ vectors in storage, i.e., dropping the oldest vector when

rather than restart [Den77; Gil89]. With the additional matrices $\boldsymbol{L}_k$ taken to be the strictly lower triangular part of the matrix $\boldsymbol{V}_k^T \boldsymbol{W}_k$ and $\boldsymbol{D}_k$ as the diagonal of $\boldsymbol{V}_k^T \boldsymbol{W}_k$, the ***compact matrix*** form of the LBFGS updates [Noc06] is given by

$$\boldsymbol{M}^{-1,(k+1)} = \boldsymbol{M}_{prev}^{-1,0} + \boldsymbol{\Gamma} \boldsymbol{B} \boldsymbol{\Gamma}^T \tag{2.44}$$

where

$$\boldsymbol{\Gamma} = \left[\boldsymbol{M}_{prev}^{-1,0} \boldsymbol{W}_k,\ \boldsymbol{V}_k \right] \quad \text{and} \quad \boldsymbol{B} = \begin{bmatrix} \boldsymbol{W}_k^T \boldsymbol{M}_{prev}^{-1,0} \boldsymbol{W}_k & \boldsymbol{L}_K \\ \boldsymbol{L}_k^T & -\boldsymbol{D}_k \end{bmatrix}^{-1}.$$

It can be seen that the LBFGS update can be expressed as an outer product of two rectangular matrices $\boldsymbol{\Gamma}$ and $\boldsymbol{\Gamma}^T$ with an intervening multiplication by a small $2\mu \times 2\mu$ matrix. The update computed in this form has (serial) runtime complexity of $\mathcal{O}(2\mu N) + \mathcal{O}(\mu^3)$.

Two important points to notice are (*i*) the ***independent*** and arbitrary ***choice of the initial inverse Hessian approximation*** $\boldsymbol{M}_{prev}^{-1,0}$ in every iteration, due to the limited memory character of the update, and (*ii*) the ***successive rank-*1 *character*** of the limited memory update. The latter can be seen from the efficient two way recursion for the LBFGS update in (2.51) and Alg. 2.4, see §2.3.2. More explicitly, the LBFGS method solves μ successive optimization problems

$$\left\| \boldsymbol{M}^{-1,i} - \boldsymbol{M}_{prev}^{-1,i-1} \right\|_{\boldsymbol{Z}} \to \min \quad \text{subject to} \quad \boldsymbol{M}^{-1,i} \boldsymbol{V}_{k-\mu+i}^1 = \boldsymbol{W}_{k-\mu+i}^1 \text{ and } \boldsymbol{M}^{-T,i} = \boldsymbol{M}^{-1,i},$$

for $i = 1, \ldots, \mu$, respectively, leading to μ rank-1 updates of the inverse Hessian estimator. The new estimator in iteration k is then given by $\boldsymbol{M}^{-1} = \boldsymbol{M}^{-1,(\mu)}$. Due to this fact, ***filtering*** such as used in the context of solving non-linear fixed-point problems in §2.2.1 is not required for LBFGS.

The ***rank-η counterpart of the limited memory BFGS update*** (2.44) with $\eta > 1$, alternatively, is given by equation (2.42) assuming $\eta > 1$ columns in the matrices $\boldsymbol{W}_k$ and $\boldsymbol{V}_k$. Such an at-once rank-η LBFGS method can be seen as analogous to the earlier presented multi-vector (MV) method, with the additional property to maintain a symmetric and positive definite initial estimator $\boldsymbol{M}_{prev}^{-1,0}$. To the best of my knowledge, such a method has not yet been formulated, or is at least rarely used in non-linear optimization. In §2.2.1, we showed that η successive rank-1 update and at-once rank-η updates are not equivalent. Whether or not rank-η updates improve convergence (such as is the case for coupling of partitioned FSI) is unclear, and numerical investigation of this variant in practical application may be worthwhile; but this is beyond the scope of this work. Many concepts developed in §2.2.1 may be beneficial, in particular, filtering can no longer be omitted.

Notice also, that the BFGS update formula such as stated in equation (2.41) is not standard; the common display is in terms of vectors, rather than matrices. The presented formulation results from relating the MV method to the BFGS update. This is what we discuss next.

Relation of BFGS to the Multi-Vector Method. In an attempt to align the notation of the multi-vector (MV) method, discussed in §2.2.1, and standard text book BFGS, we would like to work out conceptual differences and similarities. This fosters a better understanding of the respective methodological characteristics and potentially allows a link to be forged between quasi-Newton manifestations for different application fields.

Minimization in a Weighted Matrix Norm. Rather than using the canonical Frobenius norm to obtain a unique solution of problem (2.9), classical quasi-Newton methods as used in optimization employ a ***weighted*** matrix norm $\| \cdot \|_{\boldsymbol{Z}} = \| \boldsymbol{Z}^{1/2} \cdot \boldsymbol{Z}^{1/2} \|_F$ with a symmetric matrix $\boldsymbol{Z}$ fulfilling the multi-secant

collecting the most recent, if the window is saturated.

equation $\boldsymbol{Z}\boldsymbol{W}_k = \boldsymbol{V}_k$, or, transformed $\boldsymbol{W}_k = \boldsymbol{Z}^{-1}\boldsymbol{V}_k$ and solve the optimization problem

$$\left\| \boldsymbol{M}^{-1} - \boldsymbol{M}_{prev}^{-1} \right\|_{\boldsymbol{Z}}^2 \to \min \qquad \text{subject to } \ \boldsymbol{M}^{-1}\boldsymbol{V}_k = \boldsymbol{W}_k,$$

to derive an expression for the Hessian estimator. This is equivalent to solving

$$\left\| \boldsymbol{Z}^{1/2}\boldsymbol{M}^{-1}\boldsymbol{Z}^{1/2} - \boldsymbol{Z}^{1/2}\boldsymbol{M}_{prev}^{-1}\boldsymbol{Z}^{1/2} \right\|_F^2 \to \min \quad \text{s.t. } \ \boldsymbol{Z}^{1/2}(\boldsymbol{M}^{-1} - \boldsymbol{M}_{prev}^{-1})\boldsymbol{Z}^{1/2}\boldsymbol{Z}^{-1/2}\boldsymbol{V}_k = \boldsymbol{Z}^{1/2}(\boldsymbol{W}_k - \boldsymbol{M}_{prev}^{-1}).$$

We know the solution to this problem, cf. equation (2.17), and use $\boldsymbol{Z}^{-1}\boldsymbol{V}_k = \boldsymbol{W}_k$ and the symmetry of $\boldsymbol{Z}^{-1/2}$ to simplify the formula:

$$\begin{aligned}
\boldsymbol{M}^{-1} - \boldsymbol{M}_{prev}^{-1} &= \boldsymbol{Z}^{1/2}(\boldsymbol{Z}^{1/2}(\boldsymbol{M}^{-1} - \boldsymbol{M}_{prev}^{-1})\boldsymbol{Z}^{1/2})\boldsymbol{Z}^{1/2} \\
&= \boldsymbol{Z}^{1/2}\left(\boldsymbol{Z}^{1/2}(\boldsymbol{W}_k - \boldsymbol{M}_{prev}^{-1}\boldsymbol{V}_k)\left((\boldsymbol{Z}^{1/2}\boldsymbol{V}_k)^T(\boldsymbol{Z}^{1/2}\boldsymbol{V}_k)\right)^{-1}(\boldsymbol{Z}^{1/2}\boldsymbol{V}_k)^T\right)\boldsymbol{Z}^{1/2} \\
&= (\boldsymbol{W}_k - \boldsymbol{M}_{prev}^{-1}\boldsymbol{V}_k)\underbrace{\left(\boldsymbol{V}_k^T\boldsymbol{Z}^{-1}\boldsymbol{V}_k\right)}_{=\boldsymbol{W}_k^T\boldsymbol{V}_k=\boldsymbol{V}_k^T\boldsymbol{W}_k}{}^{-1}\boldsymbol{V}_k^T\boldsymbol{Z}^{-1} \\
&= (\boldsymbol{W}_k - \boldsymbol{M}_{prev}^{-1}\boldsymbol{V}_k)\left(\boldsymbol{V}_k^T\boldsymbol{W}_k\right)^{-1}\boldsymbol{W}_k^T. \qquad (2.45)
\end{aligned}$$

As a result, we get

$$\begin{aligned}
\boldsymbol{M}^{-1} &= \boldsymbol{M}_{prev}^{-1} + (\boldsymbol{W}_k - \boldsymbol{M}_{prev}^{-1}\boldsymbol{V}_k)\left(\boldsymbol{V}_k^T\boldsymbol{W}_k\right)^{-1}\boldsymbol{W}_k^T \\
&= \boldsymbol{W}_k\left(\boldsymbol{V}_k^T\boldsymbol{W}_k\right)^{-1}\boldsymbol{W}_k^T + \boldsymbol{M}_{prev}^{-1}(\boldsymbol{I} - \boldsymbol{V}_k\left(\boldsymbol{V}_k^T\boldsymbol{W}_k\right)^{-1}\boldsymbol{W}_k^T). \qquad (2.46)
\end{aligned}$$

As can easily be seen, equation (2.46) is not symmetric for $\boldsymbol{M}_{prev}^{-1} \neq \boldsymbol{0}$[32]. However, it shows some resemblance with the BFGS formula (2.41) in the sense that the latter can be recognized as the symmetrized version equation (2.46):

In order to symmetrize $\boldsymbol{M}^{-1}$, we modify the estimator derived in (2.46) to

$$\widetilde{\boldsymbol{M}}^{-1} = \boldsymbol{M}^{-1} + \boldsymbol{X} \ \text{ such that } \ \widetilde{\boldsymbol{M}}^{-1} = \widetilde{\boldsymbol{M}}^{-T} \ \text{ with } \ \boldsymbol{X}\boldsymbol{V}_k = \boldsymbol{0} \ \text{ and } \ \|\boldsymbol{X}\|_{\boldsymbol{Z}} \to \min.$$

The above is equivalent to $\boldsymbol{X} - \boldsymbol{X}^T = \boldsymbol{M}^{-T} - \boldsymbol{M}^{-1}$. This implies

$$\boldsymbol{X}^T\boldsymbol{V}_k = (\boldsymbol{M}^{-1} - \boldsymbol{M}^{-T})\boldsymbol{V}_k \ \text{ with } \ \|\boldsymbol{X}\|_{\boldsymbol{Z}} \to \min$$

and multiplying the pseudo-inverse $(\boldsymbol{V}_k^T\boldsymbol{W}_k)^{-1}\boldsymbol{W}_k^T$ corresponding to the $\boldsymbol{Z}$-norm from the right side yields (analogous to the derivation of (2.46)):

$$\boldsymbol{X}^T = (\boldsymbol{M}^{-1} - \boldsymbol{M}^{-T})\boldsymbol{V}_k\left(\boldsymbol{V}_k^T\boldsymbol{W}_k\right)^{-1}\boldsymbol{W}_k^T,$$

[32] By setting $\boldsymbol{M}_{prev}^{-1} = \boldsymbol{0}$ in equation (2.46), we arrive at a symmetric Hessian estimator with the reasoning

$$\boldsymbol{M}^{-T} = \left(\boldsymbol{W}_k(\boldsymbol{V}_k^T\boldsymbol{W}_k)^{-1}\boldsymbol{W}_k^T\right)^T = (\boldsymbol{W}_k^T)^T\left((\boldsymbol{V}_k^T\boldsymbol{W}_k)^{-1}\right)^T\boldsymbol{W}_k^T = \boldsymbol{W}_k\left((\boldsymbol{V}_k^T\boldsymbol{W}_k)^T\right)^{-1}\boldsymbol{W}_k^T = \boldsymbol{W}_k(\boldsymbol{W}_k^T\boldsymbol{V}_k)^{-1}\boldsymbol{W}_k^T = \boldsymbol{M}^{-1}.$$

This is equivalent to the BFGS update formula (2.41) with likewise using $\boldsymbol{M}_{prev}^{-1} = \boldsymbol{0}$. This can be seen as symmetric rank-η variant of the least-squares (LS) method, presented in §2.3.1

which, after transposition and inserting of the definition of $\boldsymbol{M}^{-1}$ in (2.46) and some term transformation[33] results in

$$\boldsymbol{X} = -\boldsymbol{W}_k \left(\boldsymbol{V}_k^T\boldsymbol{W}_k\right)^{-1} \boldsymbol{V}_k^T \boldsymbol{M}_{\text{prev}}^{-1}(\boldsymbol{I} - \boldsymbol{V}_k \left(\boldsymbol{V}_k^T\boldsymbol{W}_k\right)^{-1} \boldsymbol{W}_k^T) \tag{2.47}$$

Now, the term (2.47) exactly explains the difference resulting if subtracting equation (2.46) from the BFGS estimator (2.41), i.e., the missing term for symmetry. With this, we conclude, that a symmetric rank-η BFGS update formula as given in equation (2.42) for $\eta \geqslant 1$ results from solving the optimization problem (2.41), i.e., is derived in almost the same way as the quasi-Newton inverse Jacobians in §2.2 with the only differences that a different norm is used and symmetry is enforced as an additional condition.

Improving Convergence and Robustness: Dominant Factors. To achieve high efficiency and rapid convergence rates, we need to understand and identify the method's most dominant factors influencing convergence, robustness, and computational complexity. Two of the most critical choices are (*i*) the number μ of collected vectors used for the limited memory update, and (*ii*) the selection of the initial guess $\boldsymbol{M}_{prev}^{-1,0}$ for the inverse Hessian approximation. This is what we discuss in the following. We also look at the role and importance of line-search for quasi-Newton methods in the optimization context. In particular, we draw connections to the specific PDE-constrained optimization application problem of biophysical tumor inversion simulation, which is considered in part II of this thesis.

Improving Convergence: Recycle. In analogy to the methods considered in §2.2, we consider recycling previous information, which is reflected by keeping μ previous gradient- and iterate-observations in storage (the limited memory sliding-window), cf. (2.40). This additional information is then used in the LBFGS update formula. We have seen that the depth μ of the vector-storage sensitively affects the method's convergence and runtime complexity: generally speaking, more vectors improve convergence but increase computational complexity. However, there is a turning-point due to outdated, wrong, or contradicting information that potentially slows down convergence drastically. An optimal choice is therefore highly problem dependent. With respect to computational effort, the typically employed efficient two-loop recursion in Alg. 2.4 features a runtime complexity of $\mathcal{O}(4\mu N)$ (disregarding the matrix-vector multiplication for the initial guess in line 6 of Alg. 2.4) and a memory footprint of $\mathcal{O}(2\mu N)$.

As opposed to the rank-η update methods considered in §2.2 within the fixed-point context, linearly dependent vectors are not an issue due to the successive rank-1 update nature of the classical LBFGS update formula. On the downside, however, contradicting or wrong information cannot be identified or ruled out anymore; this wrong information stays in the inverse Hessian approximation and only gets corrected slowly.

In part II of this thesis, we consider a joint biophysical inversion and medical image registration problem, which results in repeatedly solving non-linear PDE-constrained optimization problems with continually modified observation data. An interesting question within this context is, if reuse of

[33]

$$\Rightarrow \boldsymbol{X}^T = (\boldsymbol{M}^{-1} - \boldsymbol{M}^{-T})\boldsymbol{V}_k \left(\boldsymbol{V}_k^T\boldsymbol{W}_k\right)^{-1} \boldsymbol{W}_k^T, \quad \Rightarrow \boldsymbol{X} = \boldsymbol{W}_k \left(\boldsymbol{V}_k^T\boldsymbol{W}_k\right)^{-1} \boldsymbol{V}_k^T(\boldsymbol{M}^{-T} - \boldsymbol{M}^{-1})$$
$$= -\boldsymbol{W}_k \left(\boldsymbol{V}_k^T\boldsymbol{W}_k\right)^{-1} \boldsymbol{V}_k^T\boldsymbol{W}_k \left(\boldsymbol{V}_k^T\boldsymbol{W}_k\right)^{-1} \boldsymbol{V}_k^T\boldsymbol{M}_{\text{prev}}^{-1} + \boldsymbol{W}_k \left(\boldsymbol{V}_k^T\boldsymbol{W}_k\right)^{-1} \boldsymbol{V}_k^T\boldsymbol{M}_{\text{prev}}^{-1}\boldsymbol{V}_k \left(\boldsymbol{V}_k^T\boldsymbol{W}_k\right)^{-1} \boldsymbol{W}_k^T$$

vectors obtained from previous non-linear solves (for moderately changed operators due to modified observation data) is beneficial. Various approaches to select the recycled vectors are conceivable; we present two that might be worthwhile for investigation: recycling of $\mu - k$ vectors can be done either

(i) ***sequentially*** in analogy to (2.24), but with n denoting the number of recurrent solves[34] of the (slightly changing) non-linear problem, rather than the number of completed time steps, and $\xi \in \{0, \dots, n\}$ as before the number of retained levels with the storage depth $\mu = \mu(\xi) = k + \sum_{q=n-\xi}^{n-1} k_q$; or we can

(ii) ***selectively*** retain information. By this we mean to collect only the first[35] $\overline{\mu} = {}^{\mu-k}/_{\xi}$, say, e.g., $\overline{\mu} = 5$ difference vectors from the $\xi \leqslant n-1$ non-linear problem solutions.

Initial Guess for Inverse Hessian Approximation. The choice of the initial guess for the inverse Hessian approximation $\boldsymbol{M}_{prev}^{-1}$ significantly affects the quality of the obtained step direction and is the most dominant factor affecting the quasi-Newton LBFGS convergence. A good choice for $\boldsymbol{M}_{prev}^{-1}$ is non-trivial; for the quasi-Newton method to be competitive, we need (*i*) fast convergence, i.e., a good initial guess, but coincidentally require it to be obtained with (*ii*) low computational complexity to limit the cost of a Newton-step. The latter becomes more pressing when approaching large-scale; in any case, finding a good trade-off is crucial. In contrast to the multi-vector update (MV) method we discussed in §2.2 for fixed-point problems, the here considered LBFGS method allows us to choose a ***different initial approximation*** $\boldsymbol{M}_{prev}^{-1} := \boldsymbol{M}_{prev}^{-1,(k)}$ to the inverse Hessian in every iteration.

We discuss three conceptually different approaches and sketch the idea of several choices for the initial guess. We refer to [Gil89; Noc06] for a discussion. Some of the presented alternatives are also implemented in PETSc/TAO's Limited Memory Variable Metric (LMVM) quasi-Newton method (resembling the LBFGS method), which we use for our studies in §7.4.5; details can be found in [Ben03]. We revisit some of these approaches in §6.4.5 and §7.4.5, where we investigate quasi-Newton LBFGS for the tumor inversion sub-component of our joint biophysical inversion and image registration application problem, discussed in part II, and compare its performance against a Gauß-Newton-Krylov method.

A. *Generic but uninformed approaches.*

A.1 ***Use the (scaled) identity for the Hessian*** $\boldsymbol{M}_{prev}^{(k)} = \gamma \boldsymbol{I}$ and "solve" $\gamma \boldsymbol{I} \Delta \boldsymbol{x}^k = \boldsymbol{r}$ in line 6 of Alg. 2.4.

A.2 ***Use the (scaled) identity for the inverse Hessian*** $\boldsymbol{M}_{prev}^{-1,(k)} = \gamma \boldsymbol{I}$, with $\gamma \in \mathbf{R}$. The canonical choice is $\gamma = 1$, which, however, typically yields poor convergence. No additional cost occurs to provide the initial guess.

B. *Tailored approaches that incorporate complimentary a priori knowledge about the problem.*

B.1 ***Use an informed approximation of the Hessian*** $\boldsymbol{M}_{prev}^{(k)} = \boldsymbol{Z}$ and solve $\boldsymbol{M}_{prev}^{(k)} \Delta \boldsymbol{x}^k = \boldsymbol{r}$ for $\Delta \boldsymbol{x}^k$ in line 6 of Alg. 2.4. If anything, (partially) a priori information is available for the Hessian, not for its inverse. With this approach, a linear system needs to be solved in every quasi-Newton iteration. Although the initial guess $\boldsymbol{M}_{prev}^{(k)} \approx \boldsymbol{H}$ probably has a better condition number than the true Hessian (or Gauß-Newton approximation), this approach destroys the complexity benefits of quasi-Newton. If for all recurrent non-linear solves $i = 1, \dots, n$ the same initial guess $\boldsymbol{M}_{prev}^{(k)} = \boldsymbol{H}^0$

[34] When presenting our solution algorithm for this coupled problem in Chapter 6, this will be the number of Picard iterations executed.

[35] The first quasi-Newton iterations correct the most significant singular values in the inverse Hessian.

is used, it could be factorized in an offline phase. In §6.4.5 we give some specific suggestions to realize such an informed Hessian initial guess for the considered PDE-constrained optimization application problem of biophysical inversion for brain tumor MRI.

C. *Generic approaches that incorporate knowledge about the problem in an automated or implicit way.*

C.1 ***Use a scaled identity*** $\boldsymbol{M}_{prev}^{-1,(k)} = \boldsymbol{\gamma I}$, where $\boldsymbol{\gamma} = \mathrm{diag}(\gamma_1, \ldots, \gamma_N) \in \mathbf{R}^{N \times N}$ is a diagonal scaling matrix with weights $\gamma_i \in \mathbf{R}$. Here, various alternatives inspired by [Gil89] with different computational cost are considered. These versions are also implemented in the PETSc/TAO toolbox [Ben03]).

(i) ***Uniform scaling***: with $\boldsymbol{\gamma} = \mathrm{diag}(\gamma, \ldots, \gamma)$ and $\gamma \in \mathbf{R}$ chosen by solving the one-dimensional optimization problem

$$\gamma = \min_{\sigma} \| \sigma^{\alpha} \boldsymbol{V}_k^s - \sigma^{\alpha-1} \boldsymbol{W}_k^s \|_F^2 \tag{2.48}$$

where $\alpha \in [0,1]$ is given and $\boldsymbol{V}_k^s$ and $\boldsymbol{W}_k^s$ are matrices storing iterate and gradient difference vectors from $s \leqslant \mu$ previous iterations. In other words, γ is chosen to fulfill the multi-secant equation $\gamma \boldsymbol{V}_k^s = \boldsymbol{W}_k^s$ in a least-squares sense. Another popular choice proven to be efficient in practice is $\gamma = w_{k-1}^T v_{k-1} / v_{k-1}^T v_{k-1}$.

(ii) ***Broyden scaling***: with $\boldsymbol{\gamma} = \mathrm{diag}(\gamma_{11}^B, \ldots, \gamma_{n_q n_q}^B)$ and $\gamma_{ii}^B \in \mathbf{R}, i = 1, \ldots, n_q$ the diagonal entries of a Broyden approximation of the inverse Hessian. This alternative requires comparatively high computational costs to compute the Broyden factors for the scaling matrix. However, it typically results in a significantly better convergence rate.

An additional ***re-scaling*** of the matrix $\boldsymbol{\gamma}$, i.e., using $\boldsymbol{M}_{prev}^{-1,(k)} = \omega \boldsymbol{\gamma I}$ is a further option to improve the convergence properties. Here, the scalar $\omega \in \mathbf{R}$ is chosen by solving the one-dimensional optimization problem

$$\omega = \min_{\sigma} \| \sigma^{\alpha} \gamma^{\beta} \boldsymbol{V}_k^r - \sigma^{\alpha-1} \gamma^{\beta-1} \boldsymbol{W}_k^r \|_F^2 \tag{2.49}$$

where $\alpha, \beta \in [0,1]$ are given and $\boldsymbol{V}_k^r$ and $\boldsymbol{W}_k^r$ are the matrices storing past iterate and gradient information from $r \leqslant \mu$ previous iterations. This has been shown to significantly reduce the number of required quasi-Newton iterations for certain problems [Gil89], but it also increases the computational effort for the LBFGS update and introduces further problem dependent parameters that need costly and careful tuning.

C.2 ***Use a subspace tracking technique*** and approximate the inverse Hessian by an updated and truncated singular value decomposition, i.e.,

$$\overline{\Psi}_n, \overline{\Sigma}_n, \overline{\Phi}_n^T = \texttt{\textbf{update-SVD}} \left(\overline{\Psi}_{n-1}, \overline{\Sigma}_{n-1}, \overline{\Phi}_{n-1}^T, \boldsymbol{M}_{prev}^{-1,k_{n-1}}, \varepsilon_{svd} \right),$$

where the updating step **update-SVD** maintains an existing singular value factorization by incorporating the latest inverse Hessian approximation $\boldsymbol{M}_{prev}^{-1,k_{n-1}}$ (from the previous non-linear solve). Note that this information is given in implicit form in the matrices $\boldsymbol{V}_k$ and $\boldsymbol{W}_k$ via the LBFGS update formula (2.51) to compute the matvec. The initial guess $\boldsymbol{M}_{prev}^{-1,(k)}$ required in the LBFGS update for the current iteration k is then encoded by appending the matrices $\boldsymbol{V}_k' = [\boldsymbol{V}_k \; \overline{\boldsymbol{\Sigma \Phi}}^T]$ and $\boldsymbol{W}_k' = [\boldsymbol{W}_k \; \overline{\boldsymbol{\Psi}}]$. We have given details on the realization of such an approach in §2.2.2.

Line-Search. Lastly, the employed line-search affects the convergence properties. The history of taken quasi-Newton steps is remembered within the LBFGS update for the Hessian. Thus, the past affects the quality of current step directions and sub-optimal or wrong steps stemming from inexact or insufficient line-search conditions should be prevented. For quasi-Newton methods in general, the line-search routine must enforce the ***Wolfe conditions***

$$f(\boldsymbol{x}^k + \alpha \Delta \boldsymbol{x}^k) \leqslant f(\boldsymbol{x}^k) + c_1 \alpha (\boldsymbol{g}(\boldsymbol{x}^k))^T \Delta \boldsymbol{x}^k, \quad \text{and} \tag{2.50a}$$

$$(\boldsymbol{g}(\boldsymbol{x}^k + \alpha \Delta \boldsymbol{x}^k))^T \Delta \boldsymbol{x}^k \geqslant c_2 \, (\boldsymbol{g}(\boldsymbol{x}^k))^T \Delta \boldsymbol{x}^k, \tag{2.50b}$$

for some constants $0 < c_1 < c_2 < 1$ to prevent possible degradation of the convergence rate [Noc06]. The ***sufficient decrease*** condition (2.50a) requires that the decrease in the objective function for a trial step $\alpha \Delta \boldsymbol{x}^k$ is proportional to both, the step length α and the directional derivative $(\boldsymbol{g}(\boldsymbol{x}^k))^T \Delta \boldsymbol{x}^k$. This is also known as the ***Armijo condition***. To prevent unacceptably small or sub-optimal steps, a second ***curvature condition*** (2.50b) is required. For an illustration of the Wolfe conditions and further details, we refer to [Noc06], p. 33f.

While for other inexact-Newton methods such as, e.g., Gauß-Newton-Krylov, a line-search that only satisfies the sufficient decrease condition (such as, e.g., Armijo-backtracking) is sufficient, this does not apply for most quasi-Newton methods due to the lower quality of the computed step-direction and the above mentioned issues. Furthermore, without satisfied Wolfe conditions, we can (*i*) neither rely on the self-correcting properties of the BFGS update, (*ii*) nor do we have a guarantee that the required curvature condition $\boldsymbol{v}_k^T \boldsymbol{w}_k > 0$ for the well-posedness of the BFGS update will be satisfied by the chosen step (step length $\alpha > 1$ may be required to satisfy this condition), (*iii*) nor will the BFGS update formula maintain a positive definiteness of an initially positive definite approximation of the Hessian.

For all Newton-type methods, a trial step length of $\alpha_0 = 1$ should always be used first to allow for the rapid rate-of-convergence properties of these methods to take effect. Inaccurate line searches with $c_1 = 1\text{E}{-4}$ and $c_2 = 0.9$ are commonly used and yield a good trade-off between computational expense and accuracy of the step length, compare [Noc06], p. 143.

2.3.2 Efficient Algorithms for Calculation and Representation of Jacobians

An efficient, distributed memory realization of the LBFGS algorithm in conjunction with the Moré-Thuente line-search method, ensuring fulfillment of the Wolfe conditions, is implemented in PETSc's optimization toolbox TAO, and known under the term Limited Memory Variable Metric Method (LMVM). It comes with a wide variety of choices for the initial inverse Hessian approximation (compare listing above). For more information, we refer to [Ben03].

Rather than implementing the compact matrix representation (2.44), the LBFGS update usually is obtained from a recursive formulation. The latter is found by repeatedly applying (2.43), i.e.,

```
1  func $\Delta x^k$ = matvecLBFGS($g^k$, $V_k = (v_i)$, $W_k = (w_i)$, $M_{prev}^{-1,(k)}$)
2      $q = g^k$
3      for $i = k-1, \dots, k-\eta$ do
4          $\alpha_i = \rho_i w_i^T q$
5          $q = q - \alpha_i v_i$
6      $r = M_{prev}^{-1,(k)} q$
7      for $i = k-\eta, \dots, k-1$ do
8          $\beta = \rho_i v_i^T \hat{q}_k$
9          $r = r + w_i(\alpha_i - \beta)$
10     return $M^{-1} g^k = \Delta x^k = r$
```

ALGORITHM 2.4 *Computation of LBFGS matvec based on stored vectors $v_i = V_i^1$ and $w_i = W_i^1$, cf [Noc06], p. 178*

compute

$$
\begin{aligned}
M^{-1,(k+1)} = & \left(U_{k-1}^T \dots U_{k-\eta}^T\right) M_{prev}^{-1,(k)} \left(U_{k-\eta} \dots U_{k-1}\right) \\
& + \rho_{k-\eta} \left(U_k^T \dots U_{k-\eta+1}^T\right) w_{k-\eta} w_{k-\eta}^T \left(U_{k-\eta+1} \dots U_{k-1}\right) \\
& + \rho_{k-\eta+1} \left(U_{k-1}^T \dots U_{k-\eta+2}^T\right) w_{k-\eta+1} w_{k-\eta+1}^T \left(U_{k-\eta+2} \dots U_{k-1}\right) \\
& + \dots \\
& + \rho_{k-1} w_{k-1} w_{k-1}^T,
\end{aligned}
\tag{2.51}
$$

with $\rho_j := (v_j^T w_j)^{-1}$ and $U_j := (I - \rho_j v_j w_j^T)$ for $j = k, \dots, k-\mu$, where we use $v_j = V_j^1 and w_j = W_j^1$ for clarity. From this expression, we can directly verify the earlier given statement that the LBFGS update results from μ successive rank-1 updates. We can further derive a recursive procedure from (2.51) to compute the LBFGS Hessian matrix-vector multiplication $\Delta x^k := M^{-1} g(x^k)$, i.e., the quasi-Newton update. An exemplary pseudo-code implementation is given in Alg. 2.4. Disregarding the matrix-vector multiplication for the initial guess in line 6, this two-loop recursion algorithm has a runtime complexity of $\mathcal{O}(4\mu N)$. This algorithm exclusively relies on inner products of vectors and matrix-vector products—two standard linear algebra operations for which highly efficient distributed memory implementations exist. Thus, parallelization of LBFGS is straightforward; the resulting algorithm is highly efficient.

2.4 Summary of Commonalities and Differences

LBFGS (optimization)	**MV/LS** (fixed-point)
$\min_{\boldsymbol{x}\in\mathbb{R}^N} f(\boldsymbol{x})$	solve $\boldsymbol{\mathcal{H}}(\boldsymbol{x}) - \boldsymbol{x} = \boldsymbol{0}$, $\boldsymbol{x} \in \mathbb{R}^N$
quadratic approx. of f at $\boldsymbol{x}^k + \Delta\boldsymbol{x}^k$: $m_k(\Delta\boldsymbol{x}^k) = f_k + \boldsymbol{g}_k^T\Delta\boldsymbol{x}^k + \frac{1}{2}(\Delta\boldsymbol{x}^k)^T\boldsymbol{M}^{-1}\Delta\boldsymbol{x}^k$	
corr. to the linear approx. of $\boldsymbol{g}(\boldsymbol{x}^k + \Delta\boldsymbol{x}^k) := \nabla f(\boldsymbol{x}^k + \Delta\boldsymbol{x}^k)$: $\boldsymbol{r}^k(\Delta\boldsymbol{x}^k) = \boldsymbol{g}_k + \boldsymbol{M}_k^{-1}\Delta\boldsymbol{x}^k$	linear approximation of $\boldsymbol{\mathcal{R}} = \boldsymbol{\mathcal{H}} - \boldsymbol{I}$ at $\boldsymbol{x}^k + \Delta\boldsymbol{x}^k$: $\boldsymbol{r}^k(\Delta\boldsymbol{x}^k) = \boldsymbol{\mathcal{H}}(\boldsymbol{x}^k) - \boldsymbol{x}^k + \boldsymbol{M}\Delta\boldsymbol{x}^k$
$\boldsymbol{M}^{-1}$ symmetric positive definite approx. of the inverse Hessian of f, i.e., $\boldsymbol{M}_k^{-1} \approx (\nabla^2 f(\boldsymbol{x}^k))^{-1}$	$\boldsymbol{M}^{-1}$ approx. of the inverse Jacobian of $\boldsymbol{\mathcal{R}}$, i.e., $\boldsymbol{M}_k^{-1} \approx (\nabla\boldsymbol{\mathcal{R}}(\boldsymbol{x}^k))^{-1}$
$\min_{\Delta\boldsymbol{x}^k} m_k(\Delta\boldsymbol{x}^k)$ corresponds to solve $\boldsymbol{r}^k(\Delta\boldsymbol{x}^k) = \boldsymbol{0}$ $\Rightarrow \Delta\boldsymbol{x}^k = -\boldsymbol{M}_k^{-1}\nabla f_k$	solve $\boldsymbol{r}^k(\Delta\boldsymbol{x}^k) = \boldsymbol{0}$ $\Rightarrow \Delta\boldsymbol{x}^k = -\boldsymbol{M}_k^{-1}\underbrace{(\boldsymbol{\mathcal{H}}(\boldsymbol{x}^k) - \boldsymbol{x}^k)}_{= \boldsymbol{\mathcal{R}}(\boldsymbol{x}^k)}$ $= -(\widetilde{\boldsymbol{M}}^{-1} - \boldsymbol{I})\boldsymbol{\mathcal{R}}(\boldsymbol{x}^k)$ with the Jacobian $\widetilde{\boldsymbol{M}}^{-1}$ of $\tilde{\boldsymbol{\mathcal{R}}} : \boldsymbol{x} \mapsto \boldsymbol{x} - \boldsymbol{\mathcal{H}}^{-1}(\boldsymbol{x})$.
secant equation for update of $\boldsymbol{M}_k^{-1}$ to $\boldsymbol{M}_{k+1}^{-1}$	secant equation for update of $\widetilde{\boldsymbol{M}}_{prev}^{-1,n-1}$ to $\widetilde{\boldsymbol{M}}^{-1,n}$
$\lVert\boldsymbol{M}^{-1} - \boldsymbol{M}_{prev}^{-1,(n-1)}\rVert_F \rightarrow \min$ s.t. $\boldsymbol{M}^{-1}\boldsymbol{V}_k = \boldsymbol{W}_k$	$\lVert\boldsymbol{M}^{-1} - \boldsymbol{M}_{prev}^{-1,(k-1)}\rVert_Z \rightarrow \min$ s.t. $\boldsymbol{M}^{-1}\boldsymbol{V}_k = \boldsymbol{W}_k$ and $\boldsymbol{M}^{-T} = \boldsymbol{M}^{-1}$
estimator: $\boldsymbol{M}^{-1} = (\boldsymbol{I} - \boldsymbol{W}_k(\boldsymbol{V}_k^T\boldsymbol{W}_k)^{-1}\boldsymbol{V}_k^T)\boldsymbol{M}_{prev}^{-1,(k-1)}$ $(\boldsymbol{I} - \boldsymbol{V}_k(\boldsymbol{V}_k^T\boldsymbol{W}_k)^{-1}\boldsymbol{W}_k^T) + \boldsymbol{W}_k(\boldsymbol{V}_k^T\boldsymbol{W}_k)^{-1}\boldsymbol{W}_k^T$ initial guess: $\boldsymbol{M}_{prev}^{-1,0} \neq \boldsymbol{0}$ (e.g. Broyden scaling)	estimator: $\boldsymbol{M}^{-1} =$ $\boldsymbol{M}_{prev}^{-1} + (\boldsymbol{W}_k - \boldsymbol{M}_{prev}^{-1,(n-1)}\boldsymbol{V}_k)(\boldsymbol{V}_k^T\boldsymbol{V}_k)^{-1}\boldsymbol{V}_k^T$ initial guess: $\boldsymbol{M}_{prev}^{-1,0} = \boldsymbol{0}$
μ successive rank-1 updates no filtering required update symmetric, positive definite	at-once rank-η update filtering required

Part I

Partitioned Coupling of Surface Coupled Multi-Physics Simulation

3 Partitioned Multi-Physics Simulation and Fluid-Structure Interaction

The first part of my thesis targets multi-physics simulations, in particular the simulation of fluid-structure interaction (FSI). FSI describes the phenomenon of solid body deformation (or movement) under fluid excitation; conversely, the deformation of the solid also influences the fluid phase, which leads to a bi-directional coupling. Modeling and simulating this mutual influence is of great importance for many practically relevant problems in various fields such as, e.g., quantification in computational medicine, classical engineering, marine engineering and aerospace engineering.

This chapter summarizes the most important aspects of the coupling of partitioned fluid-structure interaction simulations as required for a good understanding of the contributions within this thesis. §3.1 introduces the coupled problem. We give a practical view on FSI simulations and exemplary application problems in §3.1.1, followed by a brief description of the mathematical models for fluid dynamics and structural mechanics in §3.1.2. The partitioned solution approach is abstracted in §3.1.3 and discussed against the monolithic approach. Methodological components of partitioned FSI are covered in §3.2, where we discuss the required components for employing a partitioned solution strategy in §3.2.1. In §3.2.2 we go into more detail on how to establish the external coupling via fixed-point equations at the interface. The section concludes with an overview of the parallel general purpose coupling library preCICE (precise Code Interaction Coupling Environment) for partitioned multi-physics simulations in §3.2.3. All methods developed in this work (and described in Chapter 2) are implemented in preCICE.

3.1 A Coupled Multi-Physics Problem: Fluid-Structure Interaction

Going from single-physics to multi-physics modeling allows for a highly accurate representation of physical phenomena and helps to understand the effects of physical system interactions. Resolving the mutual interaction dynamics, however, increases the computational complexity and overall hardness of the problem dramatically. In addition to solving two (or more) possibly non-linear sub-problems, strategies for the realization of the actual coupling of physical components become a necessity and complicate the solution. High overall accuracy is desirable for the additional effort of multi-physics modeling to pay off. This implies the careful solution of the sub-components, high resolution and accurate models. Summarizing, coupled multi-physics simulations are a computationally extremely challenging problem—numerically and algorithmically, but also in terms of time-to-solution and computational complexity. Massively parallel execution and distributed data are an imperative.

In the emerging field of multi-physics simulations, sub-component models might change or be enhanced by incorporation of further effects or new connections between physical fields need to be established. It is desirable for the simulation environment to be adaptive to such modifications and provide high flexibility. In particular, when new approaches are to be tested and fast prototyping becomes important. We therefore consider a highly flexible partitioned coupling approach. Thereby, an emerging challenge is to combine flexibility with parallel scalability and hardware efficiency.

3.1.1 Applications

With the rapid advances in computing power and the development of large parallel compute systems, FSI problems have become feasible for numerical assessment and first FSI simulations date back to the mid 90's [Bat95; Ceb97; Mam95; Pip95; Ste97; Wal99; Wol96]. Due to the wide application field of FSI in engineering and computational medicine, their importance has grown ever since. For engineering, one of the classical applications is the assessment of the stability of an elastic body exposed to a fluid flow, called aeroelasticity. In aeronautics, this is critical to estimate durability and stability of an aircraft, based on the simulation of flow induced vibrations and flutter [Far06]; [Cav07], in connection with the weight minimization [Far03]. Similar techniques are used for engineering of wind turbines [Baz11]. Another prominent example for FSI is marine engineering, where the interaction of foils [Lot13], or interactions of wind and sails [Lom13] are fields of research. In cargo shipping, multi-phase fluid simulations become important to simulate the behavior of partially filled tanks [Gra08]. Other important applications of FSI simulation in engineering are, e.g., the simulation of opening and descending parachutes [Ste97; Ste05; Sat07], the simulation of other lightweight structures such as tents [Glü01], and inflatable structures such as airbags [Wüc06].

When combined with the simulation of conjugate thermal flow, also referred to as thermal FSI [Bir13], such simulations play an important role for the numerical simulation of combustion engines [Mir16] in automotive engineering. Another effect often combined with FSI is the simulation of induced acoustics for noise prediction [Sch10].

With the emerging importance of quantification in computational medicine in recent years, FSI simulations have gained a growing significance in hemodynamics and simulations of the human cardiovascular system. In particular, this includes, the simulation of artery flow [Le 05; Rie98; Vie07; Cau05], the assessment of calcification and prediction of aneurysms [Tez07; Bal16b] as well as the simulation of the human heart [Kam14; Qua15; Dum07; Din06; Loo06].

3.1.2 Fluid Dynamics and Structural Mechanics

When modeling the interplay of fluid dynamics and structural mechanics, three sets of equations come into play. For the description of fluid dynamics, we herein use the ***Navier-Stokes equations*** for incompressible flow and employ a standard structural mechanics model in conjunction with a ***Saint-Venant-Kirchhoff*** material model. A set of coupling conditions ensures physical correctness and consistency and implements the reciprocal interactions. We summarize the mathematical formulation in the following; a more detailed description can be found, e.g., in [Gat14; Deg10] or standard text books such as [Fle12] (fluid dynamics) and [Bre12] (structural dynamics).

Fluid Dynamics. The governing equations for modeling fluid dynamics are characterized by the type of flow in the considered application problem. Important categories are compressible or incompressible flow, viscous or inviscid flow, and laminar or turbulent flow. Within this work, we focus on ***incompressible flow***, characterized by a nearly constant density of the considered fluid and very high speed of sound. These characteristics typically lead to stability problems when coupled with structural mechanics, and thus, constitute the most challenging configuration for partitioned FSI. We furthermore merely focus on laminar flow as opposed to turbulent flow. The latter occurs for high Reynold numbers and features highly irregular flow patterns in space and time.

The Navier-Stokes-Equations. The flow of a fluid in the space-time domain $\Omega_F \times [0, T] \in \mathbf{R}^d \times \mathbf{R}$, $d \in \{2, 3\}$, is characterized by the ***Navier-Stokes equations***

$$\rho \left(\frac{\partial \boldsymbol{v}}{\partial t} + (\boldsymbol{v} \cdot \nabla)\boldsymbol{v} \right) = -\nabla p + \mu \Delta \boldsymbol{v} + \rho \boldsymbol{f} \qquad \text{in } \Omega_F \tag{3.1a}$$

$$\nabla \cdot \boldsymbol{v} = 0 \qquad \text{in } \Omega_F \tag{3.1b}$$

where $\boldsymbol{v} \colon \Omega_F \times [0, T] \to \mathbf{R}^d$ denotes the ***velocity field*** of the flow, $p \colon \Omega_F \times [0, T] \to \mathbf{R}$ the spatial ***pressure*** distribution, and $\rho \colon \Omega_F \times [0; T] \to \mathbf{R}$ the fuild's ***density***. For incompressible flow, we assume $\rho(x, t) = \bar{\rho}$ as constant. μ is the shear viscosity of the fluid and $\boldsymbol{f}$ denotes volume forces such as, e.g., gravity. The set of partial differential equations (3.1) is derived from the fundamental conservation laws for mass (yields equation (3.1b)) and momentum (yields equation (3.1a)). A derivation can be found, e.g., in [Gri97].

Boundary & Initial Conditions. To enhance equations (3.1) to a well-posed initial value problem (IVP), proper boundary conditions and initial values need to be defined. Assuming a generic partitioning of the boundary $\Gamma = \partial\Omega_F$ into a Dirichlet part Γ_D and a Neumann part Γ_N with $\Gamma = \Gamma_D \cup \Gamma_N$ and $\Gamma_D \cap \Gamma_N = \emptyset$, we impose

$$\boldsymbol{v} = \boldsymbol{v}_D \qquad \text{on } \Gamma_D, \tag{3.2a}$$

$$\boldsymbol{\sigma} \cdot \boldsymbol{n} = \boldsymbol{f}_N \qquad \text{on } \Gamma_N, \tag{3.2b}$$

where the velocity is fixed to $\boldsymbol{v}_D$ on the Dirichlet boundary Γ_D, while the Neumann boundary Γ_N represents a free surface and the velocity values are set dynamically. $\boldsymbol{\sigma} = -p\boldsymbol{I} + \boldsymbol{\tau}$ is the so called ***Cauchy stress tensor***, composed of a unidirectional pressure p and a trace-less viscous (or deviatoric) stress tensor $\boldsymbol{\tau}$, modeling the shear stresses. For a Newtonian fluid as considered in (3.1), the relation $\nabla \cdot \boldsymbol{\tau} = \mu \Delta \boldsymbol{v}$ holds, i.e., the divergence of the stress appears in the momentum equation (3.1a)[1]. $\boldsymbol{n}$ denotes the unit normal of the boundary. The initial condition $\boldsymbol{v}_0$ for the velocity

$$\boldsymbol{v}(t_0, \boldsymbol{x}) = \boldsymbol{v}_0(\boldsymbol{x}) \qquad \text{in } \Omega_F, \tag{3.2c}$$

completes the set of boundary conditions.

Structural Dynamics. In structural mechanics, the governing kinematic equations are formulated in terms of a constitutive model that describes the simulated material. The latter is often derived in terms of tensor notation, i.e., the definition of a stress and strain tensor, and their mutual dependency describe the material's deformation properties. The variety of different models is broad and ranges from elastic, inelastic, and viscoelastic to plastic and hyperplastic deformation models. When considering fluid-structure interaction, the resulting material deformations can be quite large and the commonly used linear-elasticity model is not sufficient. We thus employ a non-linear material model known as the ***Saint-Venant-Kirchhoff*** material model, which furthermore assumes the solid to be homogeneous and isotropic. In the following, we summarize the governing equations; a derivation alongside with variations and discussion of different models can be found in [Bre12].

[1] It can be written in a more generalized form as $\rho \frac{d\boldsymbol{v}}{dt} = -\nabla p + \nabla \cdot \boldsymbol{\tau} + \boldsymbol{f}$.

Mathematical Model. The dynamics of a solid structure $\Omega_S \in \mathbf{R}^d$ are described by its ***deformations*** over time $[0, T] \subset \mathbf{R}$ under the influence of an external force. These deformations are represented by a ***displacement field*** $\boldsymbol{u} \colon \Omega_S \times [0, T] \to \mathbf{R}^d$. Based on Newton's second law and an equilibrium of forces, the ***equation of motion***

$$\rho \left(\frac{\partial^2 \boldsymbol{u}}{\partial t^2} \right) = \nabla \cdot \boldsymbol{\sigma}_S + \rho \boldsymbol{f} \qquad \text{in } \Omega_S \tag{3.3a}$$

is derived, with the density ρ and the distributed volume force $\boldsymbol{f}$. The stress tensor $\boldsymbol{\sigma}_S$ encodes the constitutive material model; we employ the second Piola-Kirchhoff stress tensor given as

$$\boldsymbol{\sigma}_S = \lambda \cdot \operatorname{tr}(\boldsymbol{e}) \boldsymbol{I} + 2\mu \boldsymbol{e} \quad \text{with } \boldsymbol{e} = \frac{1}{2} \left(\boldsymbol{F} + \boldsymbol{F}^T + \boldsymbol{F}^T \boldsymbol{F} \right). \tag{3.3b}$$

It relates the stress $\boldsymbol{\sigma}_S$ to the strain $\boldsymbol{F} = \nabla \boldsymbol{u}$, and models non-linear deformations for a material with linear-elasticity like behavior. $\boldsymbol{e}$ is the Lagrangian Green strain tensor that models the kinematics in terms of non-linear deformation. The parameters λ and μ are also known as the Lamé constants and are directly related to material parameters such as Young's modulus E and Poisson's ratio ν, via

$$E = \frac{\mu(3\lambda + 2\mu)}{\lambda + \mu} \quad \text{and} \quad \nu = \frac{\lambda}{2(\lambda + \mu)}. \tag{3.4}$$

Boundary & Initial Conditions. As before, appropriate boundary and initial conditions are required to enhance equation (3.3a) to a well posed initial value problem. Analogously as above, we impose the following Dirichlet and Neumann boundary conditions

$$\boldsymbol{u} = \boldsymbol{u}_D \qquad \text{on } \Gamma_D, \tag{3.5a}$$

$$\boldsymbol{\sigma}_S \cdot \boldsymbol{n} = \boldsymbol{f}_N \qquad \text{on } \Gamma_N \tag{3.5b}$$

for a fixed displacement $\boldsymbol{u}_D$ and a prescribed vector force $\boldsymbol{f}_N$. Defining initial values $\boldsymbol{u}_0$ and $\boldsymbol{v}_0$ for the displacement and deformation velocity

$$\boldsymbol{u}(t_0, \cdot) = \boldsymbol{u}_0 \qquad \text{in } \Omega_S, \tag{3.5c}$$

$$\frac{\partial \boldsymbol{u}(t_0, \cdot)}{\partial t} = \boldsymbol{v}_0 \qquad \text{in } \Omega_S \tag{3.5d}$$

completes the initial value problem statement.

Surface Coupling of Fluid and Structure. To establish a consistent interaction of fluid dynamics and structural mechanics, certain ***kinematic*** and ***dynamic coupling conditions*** have to be satisfied at the fluid-structure interface $\Gamma_{FS} = \Gamma_F \cap \Gamma_S$, also called ***wet-surface***. In continuum mechanics, interacting materials can neither overlap, nor give raise to gaps between material domains. A direct consequence of this is the demand for ***equality of displacements*** and ***velocities*** normal to the wet-surface. Based on the concept of molecular attraction forces, it is furthermore assumed that fluid molecules in the vicinity of the fluid-structure interface stick to the solid boundary. This motivates to further enforce the ***equality of forces*** and ***tangential velocities*** on Γ_{FS}. Summarized, the ***kinematic interface condition*** reads

$$\boldsymbol{v}_F = \frac{\partial \boldsymbol{u}_S}{\partial t} \qquad \text{on } \Gamma_{FS} \tag{3.6a}$$

whereas the ***dynamic interface condition*** is written as

$$\sigma_F \cdot \boldsymbol{n}_F = -\sigma_S \cdot \boldsymbol{n}_S \qquad \text{on } \Gamma_{FS}. \tag{3.6b}$$

Together, they form the set of ***coupling conditions***. Equation (3.6b) describes a point-wise balance of forces at the interface, represented in terms of the surface stresses; $\boldsymbol{n}_F = -\boldsymbol{n}_S$ are the interface normal vectors.

Discretization. Equations (3.1) and (3.3) are partial differential equations in space and time. For their numerical solution, the continuous representations need to be discretized into a finite number of unknowns. The thereby introduced error should be kept small to obtain an accurate approximation of the continuous representation. At the same time, the computational complexity needs to be balanced. Within this work, we consider black-box coupling, that means we abstract from the actual solver numerics and focus on generic coupling concepts. Discretization details and single-physics numerics are specific for the respective solvers and play a minor role in this work. We therefore only summarize the most common concepts.

Spatial Discretization. Numerous approaches to discretize partial differential equations are used in practice; amongst them the ***finite difference*** method (FDM), the ***finite volume*** method (FVM), and the ***finite element*** method (FEM) are the most popular forms. FEM and FVM work with a partitioning of the computational domain into elements or cells and are typically employed in fluid dynamics. For structural mechanics, the FEM method has been shown to be most effective and is predominantly used. For details of these standard methods, we refer to the vast literature; the text books [Bab01] and [LeV02] are a good starting point. A fluid dynamics specific discussion can be found in [Gri97; Hai96; Wan91; Gre98b]. For discourses on FEM in structural mechanics, we refer to [Bra07; Zie77].

The numerical treatment of fluids and structures raises different needs, which, are also reflected in different discretization approach hes and different underlying grids. Integrating the numerical treatment of fluid and structure is, therefore, not straightforward. This is closely related to the the fact that, for fluid dynamics, commonly an Eulerian or Arbitrary Lagrangian Eulerian (ALE) point of view is used to describe the system dynamics, while for structural mechanics, a Lagrangian viewpoint is dominant. Deformations, caused by the interaction additionally may raise the need for mesh-deformation, mesh-movement or re-meshing. In any case, data need to be mutually exchanged between non-conforming or non-matching meshes for the fluid- and solid-domain, respectively. Commonly used approaches to integrate moving structures in flow numerics are immersed boundary (IB) methods [Mit05], fictitious domain (FD) methods [Yu05], cut-cell methods [Qui94], marker-and-cell methods [Tom94], or dynamic mesh-movement methods combined with, e.g., radial-basis function interpolation [Lyn80; Sar01; De 07].

Temporal Discretization. To solve the system of ordinary differential equations which arises after spatial discretization in time, sequential, explicit or implicit time integration methods are used. As opposed to explicit methods, implicit time integrators come at higher computational cost as they additionally require the solution of a non-linear system of equations, but typically feature better stability. For more details, we refer to standard text books, e.g., [Hai93; Wan91], and to [Dok89; Fle12; Fle91] for an application specific discussion.

3.1.3 Monolithic and Partitioned Coupling

Solving multi-component problems featuring strong interactions such as, e.g., fluid-structure interaction, can be approached in two fundamentally different ways. The ***monolithic approach*** solves the set of all involved equations at once, that is, it typically employs the same discretization and numerics for all sub-components and composes all governing equations into one global system of equations, including the coupling terms. The monolithic problem can be stated as

$$\boldsymbol{\mathcal{A}}(x_F, x_S) = 0 \tag{3.7}$$

with the (usually) non-linear operator $\boldsymbol{\mathcal{A}}$ and the discrete flow and structural unknown variables x_F and x_S. Typically, (inexact-) Newton methods are employed for (3.7) resulting in the repeated solution of a linear system for the Newton update. Examples for FSI simulations are, e.g., [Tur11; Tez07]. Due to the direct integration and enforcement of the coupling conditions, the monolithic approach yields very high robustness and highly accurate solutions for sophisticated problems.

When a monolithic solution is intended, though, usually a new tailored solver has to be implemented which meets the special needs of the application problem, typically characterized by a large condition number. The flexibility and adaptability of the monolithic solver to changing demands or problem setups is, thus, quite limited. The implementation is tailored to a specific application; changing the application, adding features, or exchanging sub-components involves a significant amount of software development and maintenance.

By breaking up multi-component simulations into clearly defined sub-components and their coupling, we can exploit full flexibility and benefit from decades of experience in developing the sub-component solvers such as, e.g., computational fluid dynamics (CFD) or computational structural mechanics (CSM) simulation software. This so called ***partitioned approach*** is what we consider within this thesis and discuss further in what follows.

Towards a More Generic Solution: The Partitioned Approach. The partitioned approach (also called loose coupling, segregated coupling, or staggered approach) originates from domain decomposition theory [Tos06] and can be seen as ***Dirichlet-Neumann method*** for the iterative solution of a decomposed problem with non-overlapping sub-domains. This induces operators $\boldsymbol{\mathcal{F}}$ and $\boldsymbol{\mathcal{S}}$ that represent the fluid[2] and the solid sub-component solver with the following input-output assignment at the interface (compare (2.10) for a more generic setting)

$$\boldsymbol{\mathcal{F}}(x_S^\Gamma) = x_F^\Gamma \ \text{ and } \ \boldsymbol{\mathcal{S}}(x_F^\Gamma) = x_S^\Gamma, \tag{3.8}$$

that means, the fluid operator $\boldsymbol{\mathcal{F}}$ maps kinematic input values (displacements) x_S^Γ at the interface Γ to dynamic values (forces/velocities) $x_F^\Gamma = \boldsymbol{\sigma} \cdot \boldsymbol{n}$, while the structure operator $\boldsymbol{\mathcal{S}}$ conversely takes dynamic values x_F^Γ as input and computes the kinematic output values $x_S^\Gamma = \partial_t \boldsymbol{u}$. From now on, we only consider unknown variables at the interface Γ; thus, to simplify notation, we omit the superscript $^\Gamma$ in the following. The partitioned approach then allows to separately solve the single-physics sub-component equations with respect to the imposed boundary conditions. It ensures the coupling of the components externally, i.e., consistency of the kinematic and dynamic values at the coupling interface as given in (3.6) is forced in an outer loop. For problems with strong interactions, this outer coupling is formulated in terms of a non-linear fixed-point equation, compare (2.2). Besides the two extremes, many hybrid and intermediate forms are employed in literature; [Uek16] gives an extensive literature review.

Opportunities of the Partitioned Approach. The separation into well-defined sub-components with a clear interface entails many benefits. The governing equations in fluid dynamics and structural

[2] We assume the mesh-movement or re-meshing entity to be part of the fluid solver.

mechanics feature very different characteristics. As opposed to an all-at-once monolithic solution, the partitioned approach allows for both sub-components to independently employ tailored discretization schemes and numerical methods that precisely meet special needs. It is also noteworthy, that the monolithic system (3.7) typically is substantially harder to solve than the sub-components alone. The partitioned concept furthermore allows to effectively couple existing and closed source solvers, where no information about solver internals is known, in a minimally-invasive way. With this, decades of experience and sophisticated solvers for single-physics problems can be reused. In particular, sub-component solvers are seen as modular black-boxes and can be easily exchanged in an almost plug-and-play manner; exchanging one sub-component solver does not affect the remaining overall coupling framework. Furthermore, often times FSI simulations are enriched by further physical effects such as heat-transfer, acoustics, or the consideration of multiple fluid fields. Thus, there is a need for FSI simulations to be flexible, which should be transparent to, and assisted by the solver framework. This said, we can conclude that the time-to-solution seen as a process from problem statement to implementation to numerical simulation is significantly reduced for a partitioned coupling strategy as compared to the monolithic approach.

Challenges of the Partitioned Approach. On the downside, the partitioned approach inherently suffers from stability problems and induced oscillations. Each sub-component takes the boundary values as a result from the respective other component and assumes them to be fixed during its next time step solution. The magnitude of the induced oscillations depends on the characteristics of the problem and the nature of the coupling interaction. For strongly coupled problems, oscillations are typically severe. Advanced ***equation coupling schemes*** that define rules on how to combine solutions of the sub-components need to be employed to re-establish a consistent solution, i.e., the monolithic solution. This additional equation coupling is accompanied with a loss in efficiency in terms of runtime complexity due to an increased computational effort: Strongly coupled problems may require a high number of outer coupling iterations to re-establish a consistent solution per time step; for those scenarios, monolithic solvers are more efficient with respect to runtime complexity. Recent advances in employing powerful and sophisticated acceleration methods such as those presented in §2.2 to stabilize and speed up the equation coupling have been shown to make the partitioned approach competitive, even for problems with very strong coupling interactions [Uek13a; Deg09; Gat14; Bog14; Blo15b; Blo14a].

In the following, we briefly summarize challenges, specifically for FSI, which in turn motivate the necessity of the main ingredients for efficient partitioned coupling of fluid-structure interaction simulations. We discuss all remedies mentioned in this summary in §3.2.

- Due to the partitioning of equations and alternating solution of sub-components with fixed boundary values throughout the solution process, ***stability problems and oscillations*** occur. These effects become particularly emphasized if the density of the structure is similar or less then the density of the fluid, the structural stiffness decreases, or a smaller time step size is chosen. In partitioned FSI, this phenomenon is also referred to as ***added mass effect*** [Cau05; Van09b]. Implicit fixed-point ***equation coupling*** tackles this issue.
- Due to different discretizations and numerics within the solvers, inconsistent surface representations, non-matching or non-conforming grids occur. An inter-solver translation, or ***data mapping*** becomes necessary.
- Due to different time scales of sub-component solvers, ***sub-cycling*** might become necessary. This actually expresses an advantage of the partitioned approach, as time-scales can be easily

decoupled if, e.g., the fluid solver needs significantly smaller time steps than the structure solver.

- Due to required (implicit) equation coupling, the ***computational cost*** increases. This raises the need for good ***parallel scalability***.
- Due to the immense cost asymmetry between the fluid and the structure solver, the partitioned approach as commonly used, however, has ***limited parallel scalability***. Intra-solver parallelism[3] is improved by an efficient, decentralized ***peer-to-peer data communication*** approach, and inter-solver parallelism[4] is fostered by a parallel, ***Jacobi-type equation coupling***, combined with highly scalable quasi-Newton acceleration methods.

3.2 Components for Partitioned Coupling of Fluid-Structure Interaction

Within this thesis, we employ the ***partitioned coupling approach*** for the numerical treatment of fluid-structure interaction, which only requires minimal information at the coupling interface and is independent of sub-component solver internals. With its ***high flexibility*** fostering the minimally invasive and easy combination of independent sub-component solvers and allowing for the interchange of modular solvers in an almost plug-and-play manner, it perfectly meets the fluid-structure interaction demands with a steady progression in physics modeling and solver development.

Using a partitioned coupling of interacting components, e.g., fluid and structure, poses a number of challenges that need to be tackled to arrive at an efficient and accurate competitor for the monolithic solution. We have summarized these challenges above. They define ***core functionalities*** required for a realization using the partitioned approach; see Fig. 3.1 for an illustration. In §3.2.1, we shortly present and discuss solution approaches and realization for these core functionalities. Within this work, the focus lies on the development of advanced quasi-Newton schemes to accelerate the fixed-point equation coupling. Highly efficient, robust and scalable acceleration methods have been presented in §2.2 of Chapter 2. In §3.2.2 we re-visit the equation coupling and fixed-point formulation to enforce the coupling conditions at the common interface and associate the earlier discussed quasi-Newton methods with this context.

The entire coupling process can be realized in a separated coupling tool. In §3.2.3, we give a reference to the general purpose parallel coupling library preCICE, which realizes all of the required ingredients of the partitioned approach in a generic and modular way, allowing for execution on massively parallel systems. All quasi-Newton methods developed in this work are provided in preCICE.

3.2.1 Ingredients of the Partitioned Coupling Approach

In this section, we discuss the three main methodological concepts required to establish a partitioned black-box coupling of sub-solver components that share a common interface where mutual interaction occurs (coupling surface): (*i*) a translational data mapping between solver internal representations and grids, (*ii*) a data transportation component, establishing peer-to-peer communication of sub-solver ranks in a massively parallel setup, and (*iii*) a fixed-point equation coupling realizing the external component coupling in an implicit way.

[3] degree of parallelism within a sub-component solver, i.e., parallel scalability of sub-components

[4] parallelism between solvers, i.e., possibility of simultaneous solver execution

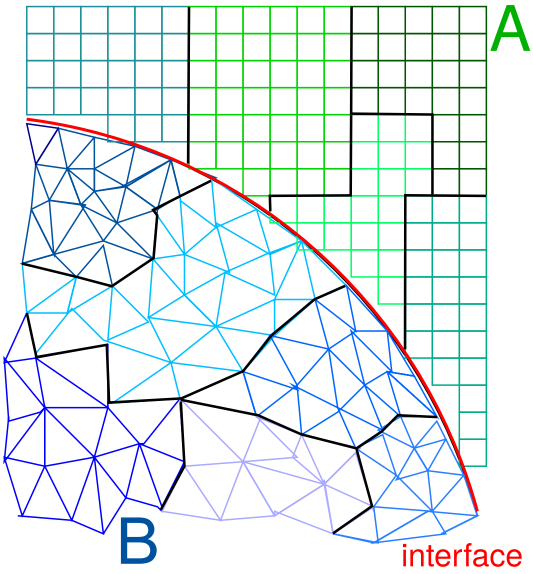


FIGURE 3.1
Schematic view of partitioned coupling. We exemplarily consider a two-dimensional surface coupled problem with two sub-components (solver A and solver B), sharing a coupling surface. Both solvers are executed in parallel and employ a domain decomposition and data distribution as indicated by the coloring; only a subset of ranks is involved in the coupling. The sub-components use different discretizations which results in non-matching grids at the shared interface. (i) A data mapping module is required to translate between the different spatial representations while preserving physical conservation laws. (ii) An inter-solver communication mechanism is required to determine interacting processes of the opposite solver, and effectively establish peer-to-peer communication tunnels between sub-solver processes, avoiding synchronization via a central instance. (iii) To enforce the coupling conditions at the shared interface, implicit fixed-point equation coupling is used. It retains the monolithic solution and eliminates instabilities and oscillations. Image modified from [Bun16b].

Translation: Data Mapping Between Solvers. The black-box nature and the modularity of the partitioned approach render non-matching or non-conforming grids and coupling interface representations to be a typical use case. Even gaps or overlaps may occur at the coupling interface. Defining a mapping to translate data representation between solvers is an essential ingredient for partitioned coupling. The mapping technique has to be chosen carefully to be capable of dealing with challenging grid-configurations at the interface and to minimize introduced errors, while preserving important physical conservation laws. A particular choice is application dependent, which is why we aim for a modular approach with a well defined interface to be able to easily exchange specific mapping implementations. Within this thesis, the mapping is subordinate: Thus, we merely abstract three different approaches. Let $\boldsymbol{x}_A \in \mathbf{R}^{N_A}$ and $\boldsymbol{x}_B \in \mathbf{R}^{N_B}$ denote the vectors of unknowns located at the nodes of the coupling interface representations Γ_A and Γ_B of the two participating solvers A and B. (cf. Fig. 3.1). We consider interpolation-based methods. An interpolation from Γ_A to Γ_B is given by a linear mapping Π_A^B, i.e.,

$$\boldsymbol{x}_B = \Pi_A^B \boldsymbol{x}^A$$

with $\Pi_A^B \in \mathbf{R}^{N_B \times N_A}$. Constant functions are mapped exactly if a mapping is ***consistent***, that means all rows in Π_A^B sum up to one. Consistent interpolations are required if values such as, e.g., positions, displacements, fluxes or densities are to be mapped. The transposed operator $\Pi_B^A = (\Pi_A^B)^T$ induces a ***conservative*** mapping in the reverse direction; such mappings are applied for mapping of integral values, e.g., forces. They conserve the sum of the interface values.

(a) ***Nearest-Neighbor Mapping.*** This projection-based mapping simply identifies the closest neighbor $n_{\Gamma_B}^k \in \Gamma_B$ of a node $n_{\Gamma_A}^i \in \Gamma_A$ on the surface representation of solver A among the vertices on the surface representation of solver B. The respective values are then simply copied. An illustration is given in Fig. 3.2. This first order method is mainly useful for matching grids, or as a first step in coupling new models or software.

(b) ***Nearest-Projection Mapping.*** For every node $n_{\Gamma_A}^i \in \Gamma_A$ on the surface representation of solver A, the projection-based mapping identifies the closest neighboring element $e_{\Gamma_B}^k$ (triangle or quad element) among the elements in $e_{\Gamma_B}^j \subset \Gamma_B$. An orthogonal projection of the node $n_{\Gamma_A}^i$ onto the element $e_{\Gamma_B}^k$ is then followed by a (bi-)linear interpolation of the values at the element's adjacent

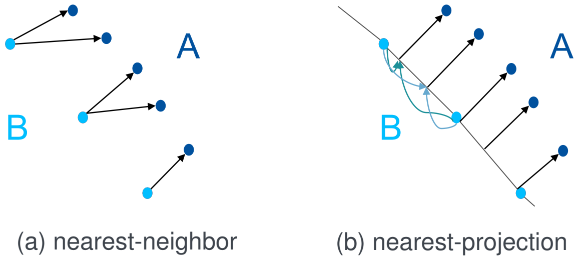


FIGURE 3.2
Schematic sketch of inter-solver mapping strategies. The nearest-neighbor mapping (left) simply copies values according to an ℓ_2-distance-based nearest-neighbor relation between solver A and B; the nearest-projection mapping (right) performs (bi-)linear interpolation based on an orthogonal projection. Image modified from [Bun16b].

vertices to the projection point which is then assigned to $n^i_{\Gamma_A}$. A simple two-dimensional example is given in Fig. 3.2. This results in a second order accurate method if there are no gaps or overlaps between the grids of the solvers at the coupling interface.

(c) ***Radial Basis Function Mapping.*** This sophisticated mapping method builds up a global, continuous interpolant of the discrete data points consisting of the mapped value at nodes $n^j_{\Gamma_B} \in \Gamma_B$. This interpolant is then evaluated for all nodes $n^i_{\Gamma_A}$ on Γ_A. As a basis, radially symmetric basis functions, centered at $n^j_{\Gamma_B}$, $j = 0, \ldots, N_B$, are used for the interpolation.

For more information and the realization on distributed data, we refer, e.g., to [Gat14; Uek16; Bun16b].

Transportation: Parallel Communication Between Solvers. Parallel execution on distributed data is an imperative for demanding fluid-structure interaction simulations. As mentioned above, using a partitioned approach allows to benefit from highly scalable sub-component solvers for fluid dynamics and structural mechanics. Using independent solvers, however, results, in general, in different partitioning of the coupling interface, due to independently chosen domain decompositions of the sub-component solvers; compare Fig. 3.1. It is desirable to omit gathering and scattering of send-receive data through a central instance, and to establish direct peer-to-peer communication channels between MPI ranks of both solvers that correspond to sub-domains at the interface. Thus, for every rank of solver A that "touches" the interface, we need to partition its local part of the coupling interface grid representation according to the domain decomposition of solver B. In other words, we need to identify destination rank numbers of solver B for every node $n^i_{\Gamma_A} \in \Gamma_A$. This send/receive rank relation is induced by the mapping. We do not consider re-meshing or re-partitioning[5] here. The computation of a corresponding destination rank for each node on Γ_A and Γ_B requires the communication of the coupling interface grid representations. To avoid deadlocks, asynchronous communication is used. To allow for an efficient and scalable overall simulation, sophisticated algorithms have to be employed for the grid re-partitioning and the computation of the communication maps[6]. Furthermore, setting up the point-to-point communication channels between ranks of different communicators in an efficient way is critical. A detailed discussion on the efficient realization of the peer-to-peer communication between the sub-components can be found in [Shu15; Uek16; Gat14; Bun16b].

Equation Coupling. Employing the partitioned solution approach for strongly interacting sub-components inherently causes stability issues and oscillations. In particular, for incompressible FSI, the induced instabilities are severe due to the so called added mass effect [Cau05; Van09b]. To control the induced instabilities, a solution of the same quality as the monolithic solution has to be recovered

[5] To maintain an optimal load-balancing, dynamic re-partitioning throughout the simulation might become necessary. This is, however, typically only the case for solvers using dynamical grid adaptivity or Eulerian grids.

[6] relations for send-/receive-rank for each node on the coupling interface grid-representation

in every time step. To this end, a sub-iteration process which repeatedly exchanges boundary values and re-computes the current time step for the sub-component solvers becomes necessary. The sub-component invocations are assumably very costly and the number of additional sub-iterations (also called ***coupling iterations***) critically affects the overall performance of the simulation.

Explicit schemes (cf. [Gat14], p. 33 ff and [Far00]) perform a fixed number of sub-iterations, and thus, the introduced error is not directly controlled but depends on the severity of the present instabilities; such schemes are sufficient and yield high efficiency for compressible FSI with moderate interactions and slight oscillations (e.g., aerodynamics simulations), but fail to converge in the face of severe oscillations and interaction with incompressible flow [Pip95; Pip01; Les98; Fel01].

Implicit schemes (cf. [Gat14], p. 44 ff), on the other hand, continue the aforementioned sub-iteration until convergence is achieved, i.e., for FSI simulations until the kinematic (equality of forces, cf. (3.6a)) and dynamic (equality of velocities, cf. (3.6b)) coupling conditions are fulfilled sufficiently accurate. Numerically, implicit interface coupling is realized as the iterative solution of a non-linear fixed-point equation $\boldsymbol{\mathcal{H}}\colon \mathbf{R}^N \to \mathbf{R}^N$, $\boldsymbol{\mathcal{H}}(\boldsymbol{x}) = \boldsymbol{x}$ which, for an admissible solution, enforces the Dirichlet-Neumann coupling conditions at the coupling interface. In what follows, we briefly recapitulate and discuss the herein employed fixed-point formulations and their respective solution to facilitate the relation with the concepts developed in §2.2.

3.2.2 Fixed-Point Formulations and Coupling Schemes

The implicit coupling iteration can be realized in various ways that primarily differ

(1) in terms of the ***execution order*** of the sub-components (i.e., fluid and structure solver), and

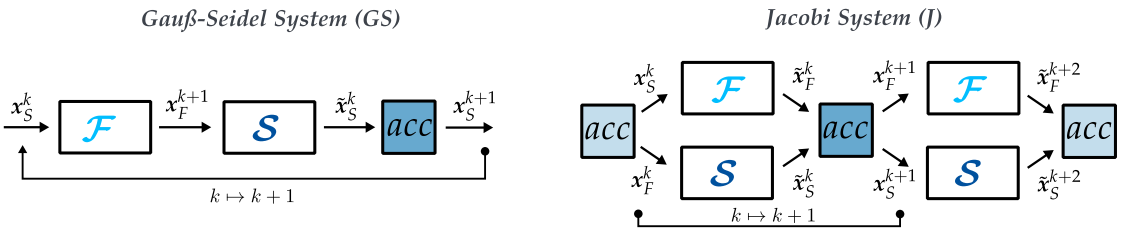
(2) in terms of employed methods used to ***accelerate and stabilize*** the coupling process.

The first item defines the fixed-point operator $\boldsymbol{\mathcal{H}}$. The second item critically affects the number of coupling iterations, and, thus, the method's efficiency in terms of runtime. A main contribution of this thesis lies in the development of advanced, highly robust and scalable methods for (2), which we presented in §2.2.

Execution Orders and Fixed-Point Equations. Various coupling strategies with diverse execution orders for the sub-components of partitioned FSI have been investigated in literature. They result in distinct fixed-point operators [Far00]; an overview is given in [Gat14], p. 33 ff. Within this work, we only consider the ***Gauß-Seidel-type (GS)*** coupling system and the ***Jacobi-type (J)*** coupling system, introduced in §2.2 in a more general setting. We shortly recapitulate both strategies. A detailed discourse can be found in [Gat14].

The Serial Gauß-Seidel-type (GS) System. This scheme executes both solvers in a staggered way and is the de facto standard approach in partitioned FSI simulations [Far00; Fel01; Deg09]. After the execution of the flow solver sub-component, the resulting stresses and forces which are exerted on the structure by the surrounding fluid are taken as input values by the structure solver sub-component to compute the resulting displacements, and wet-surface velocities of the solid. In a converged state, the displacements and velocities, i.e., structure deformations do not change anymore. This can be written as a fixed-point equation, which is given in Fig. 3.3 (left) along with a flow chart for the corresponding solver execution order.

The computational complexity of FSI simulations demands for massively parallel, distributed memory execution on large systems. The inherently serial execution order of the (GS) system entails some substantial drawbacks with respect to the efficient parallelization and parallel scalability. This


Gauß-Seidel System (GS)

Jacobi System (J)

$$(\mathcal{S} \circ \mathcal{F})\,(x_S) = x_S \qquad \begin{pmatrix} 0 & \mathcal{S} \\ \mathcal{F} & 0 \end{pmatrix} \begin{pmatrix} x_S \\ x_F \end{pmatrix} = \begin{pmatrix} x_S \\ x_F \end{pmatrix}$$

FIGURE 3.3 *Schematic illustration of the Gauß-Seidel-type (GS) and the Jacobi-type (J) fixed-point formulations to establish the equation coupling of fluid and structure solver $\mathcal{F}$ and $\mathcal{S}$ (compare §2.2). The fixed-point iterates $\tilde{x}_S^k$ and $\tilde{x}_F^k$ are modified to the iterates x_S^{k+1} and x_F^{k+1} of the implicit coupling iteration by an acceleration method acc, following the two-step strategy $x^k \overset{FP}{\rightsquigarrow} \tilde{x}^k \overset{acc}{\rightsquigarrow} x^{k+1}$ explained below. We employ quasi-Newton such as outlined in §2.2 to accelerate the fixed-point iteration.*

is caused by a usually large imbalance of work load between the fluid solver and the structure solver: While the fluid solver can be efficiently scaled on a high number of processors $2^{10} < p_F$, the computationally less expensive structure solver scales only to a fraction $p_S \ll p_F$ of the number of fluid processors. As a result, the remaining processes are idle during the execution of the solid solver, which leads to a worst case parallel efficiency of only 50%.

The Parallel Jacobi-type (J) System. The Jacobi-type system allows for a concurrent execution of the sub-components fluid and structure solver, and, thereby, remedies the above mentioned drawbacks of the (GS) system. It uses the original input/output relation for both solvers, but exchanges boundary values after each parallel execution of the sub-solvers. This leads to the vectorial fixed-point equation, outlined in Fig. 3.3 (right); compare also the flow chart for better understanding. If solved by plain-vanilla fixed-point iterations, the parallel (J) system results in two independent instances of the serial (GS) system, requiring twice the number of iterations as the (GS) system. To make the (J) system competitive, the two computation strings need to be combined by a suitable acceleration method (indicated by *acc* in Fig. 3.3) for the fixed-point iteration. Quasi-Newton methods as presented in §3.3 are powerful enough to stabilize and accelerate the fixed-point iteration sufficiently, such that one iteration of the parallel (J) system becomes comparable to one iteration of the serial (GS) system (cf. [Uek13b; Sch15]).

Other sub-component combination schemes are, e.g., the so called Steklov-Poincaré (P) scheme [Dep06] or a block-iterative variant (B) of the (GS) system. The latter has been used widely due to its improved stability over the (GS) system, but still entails strictly sequential solver execution. Applications can be found, e.g., in [Bog14; Vie07].

Adopting the notation from §2.2, we define the fixed-point operator $\mathcal{H} \colon \mathbf{R}^N \to \mathbf{R}^N$ in matrix-like notation as

$$\mathcal{H} := \begin{cases} \mathcal{S} \circ \mathcal{F} \\ \begin{pmatrix} 0 & \mathcal{F} \\ \mathcal{S} & 0 \end{pmatrix} \end{cases} \text{ and } x := \begin{cases} x_S & \text{for the (GS) system ,} \\ \begin{pmatrix} x_F \\ x_S \end{pmatrix} & \text{for the (J) system .[7]} \end{cases}$$

and solve

$$\boldsymbol{\mathcal{H}}(\boldsymbol{x}) = \boldsymbol{x} \Leftrightarrow \boldsymbol{\mathcal{R}}(\boldsymbol{x}) := \boldsymbol{\mathcal{H}}(\boldsymbol{x}) - \boldsymbol{x} \overset{!}{=} \mathbf{0}\,. \tag{3.9}$$

Convergence Criteria. The set of convergence criteria affects the performance judgment of each method. Simply monitoring the (relative) norm of the residual $\boldsymbol{\mathcal{R}}$ of the fixed-point equation yields an unfair comparison between the different equation coupling schemes: For the (GS) system, it results in an displacement-check only, while for the (J) system, displacements and forces are required to be small. As, typically, the displacement residual decays faster, both criteria need to be fulfilled for either schemes. This is furthermore the physically most reasonable argument, as both metrics depend on each other and a small displacement residual also implies a small residual for the forces. Forces and displacements might, however, live on fairly different scales. We therefore rely on relative convergence measures and use the convergence criteria

$$\|\boldsymbol{x}_F^{k+1} - \boldsymbol{x}_F^k\|_2 < \varepsilon_r \|\boldsymbol{x}_F^{k+1}\|_2 \quad \text{and} \quad \|\tilde{\boldsymbol{x}}_S^k - \boldsymbol{x}_S^k\|_2 < \varepsilon_r \|\tilde{\boldsymbol{x}}_S^k\|_2 \quad \text{for (GS)}\,, \tag{3.10a}$$

$$\|\tilde{\boldsymbol{x}}_F^k - \boldsymbol{x}_F^k\|_2 < \varepsilon_r \|\tilde{\boldsymbol{x}}_F^k\|_2 \quad \text{and} \quad \|\tilde{\boldsymbol{x}}_S^k - \boldsymbol{x}_S^k\|_2 < \varepsilon_r \|\tilde{\boldsymbol{x}}_S^k\|_2 \quad \text{for (J)}\,. \tag{3.10b}$$

Note that the convergence criterion for the sub-component solvers has to be sufficiently tight (i.e., smaller than the tolerance ε for the implicit coupling iteration).

Fixed-Point Equation Solvers and Coupling Schemes. The canonical and most simple way to solve (3.9) is plain vanilla fixed-point iteration (cf. eq. (2.11)), resulting in a multiplicative Schwarz procedure for (GS), and an additive Schwarz procedure for (J), respectively. This iteration, however, only converges if $\boldsymbol{\mathcal{H}}$ is a contraction. For most strongly coupled problems, severe instabilities occur and this is not the case. Therefore, the basic Schwarz procedures need to be enhanced with an ***acceleration method*** which stabilizes the fixed-point iteration and improves its convergence rate.

Accelerating the Fixed-Point Iteration. As introduced in §2.2, we follow a two-step outline: In every coupling iteration, a fixed-point iteration $\tilde{\boldsymbol{x}}^k = \boldsymbol{\mathcal{H}}(\boldsymbol{x}^k)$ is followed by an acceleration method "*acc*", i.e., a new iterate is found by performing

$$\boldsymbol{x}^k \underset{\boldsymbol{\mathcal{H}}}{\overset{\text{FP}}{\rightsquigarrow}} \tilde{\boldsymbol{x}}^k \overset{\text{acc}}{\rightsquigarrow} \boldsymbol{x}^{k+1}$$

with the residual $\boldsymbol{r}^k = \tilde{\boldsymbol{x}}^k - \boldsymbol{x}^k$. The simplest method for stabilization is to perform underrelaxation. Here, the new iterate is found by computing

$$\boldsymbol{x}^{k+1} = \omega^k \tilde{\boldsymbol{x}}^k + (1-\omega)\boldsymbol{x}^k = \boldsymbol{x}^k + \omega \boldsymbol{r}^k$$

with a relaxation factor $\omega \in (0, 1]$. The optimal relaxation factor is problem dependent and might also change throughout the solution process. A dynamic Aitken relaxation with

$$\omega^k = -\omega^{k-1} \frac{(\boldsymbol{r}^{k-1})^T (\boldsymbol{r}^k - \boldsymbol{r}^{k-1})}{\|\boldsymbol{r}^k - \boldsymbol{r}^{k-1}\|_2^2}$$

has been widely used for (weakly coupled) FSI problems. However, for most challenging scenarios with severe instabilities, the simple underrelaxation fails to stabilize the fixed-point iteration, or features very slow convergence. [Bog14; Uek13b] provide a comparison of various acceleration

[7] Note that $\boldsymbol{\mathcal{F}}$ and $\boldsymbol{\mathcal{S}}$ are non-linear operators in general.

methods. In particular, they compare Aitken's dynamic relaxation with quasi-Newton. Aitken, in some sense, belongs to the class of multi-secant methods: For the one dimensional case, it is equivalent to the LS($\eta = 1$) method, discussed in §2.2.

Quasi-Newton multi-secant methods have been shown to clearly outperform the above approaches and their excellent stabilization and convergence acceleration properties have been demonstrated for a large number of FSI scenarios [Deg09; Uek13a; Blo14b; Deg13b; Bog14]. Furthermore, only quasi-Newton methods are powerful enough to effectively combine the two simulation strings for the parallel (J) system (cf. Fig. 3.3) [Uek13b]. Within this work, we focus on quasi-Newton acceleration only. We discussed different methods and presented new contributions in §2.2.

Coupling Schemes. As ***coupling scheme*** we denote the combination of the execution order of the sub-component solvers, i.e., the (GS) or (J) system (or also the (P) and (B) system), and an acceleration method. §2.2 provides a thorough discussion, literature review and classification on the most commonly used multi-secant acceleration methods. [Sch15] and [Uek16] provide a listing of valid combinations along with a numerical analysis to judge and compare their performance for a one-dimensional flexible tube problem with traveling pressure pulse (see §4.1.1). A review and historically consideration of the development of these schemes within the FSI community can be found in [Uek16], p. 36 ff. The most powerful, and, thus, practically relevant combinations are the methods that are emphasized and improved within this work: The (J) system in combination with the LS type II (J-LS-II) and MV type II (J-MV-II/J-RS-SVD) outperform other combinations (cf. [Uek16], p, 36 ff and [Sch15], p. 60 ff).

3.2.3 preCICE: A Generic Library for Black-Box Component Coupling on Parallel Systems

The general purpose coupling library preCICE[8] has been designed as a tool to provide all functionality required (compare §3.2.1) to carry out partitioned multi-physics simulations based on the coupling of existing, sophisticated single-physics solvers. preCICE aims to meet two major design goals: (*i*) minimize the effort of preparing an existing solver for partitioned coupling, and (*ii*) maximize the flexibility that comes with the partitioned approach. In particular, this allows for a minimally-invasive coupling of existing black-box solvers in an almost plug-and-play manner, facilitates the exchange of solvers, and fosters quick and effortless coupling and fast prototyping. preCICE is designed for parallel execution and inter-solver parallelism. Parallel communication techniques help to exploit the massively parallel capabilities of current and future HPC machines. As mentioned above, preCICE provides efficient, parallel implementations of the three main components detailed in §3.2.1 that are required for partitioned multi-physics simulations. It implements an (*i*) efficient, decentralized peer-to-peer inter-component communication via MPI or TCP/IP, (*ii*) various data-mapping methods for translation between non-matching grids at the coupling interface, and (*iii*) efficient and robust (explicit and implicit, serial (GS) and parallel (J)) iterative methods to solve the interface fixed-point equation to enforce the coupling conditions and establish a consistent solution. Figure Fig. 3.4 gives a schematic overview of the functionality included in preCICE.

preCICE is generic in the sense that it is not limited to FSI but can be used for any surface-coupled multi-physics problem. Its primary focus is on bi-directional surface-coupled problems, but also volume coupling can be tackled with preCICE; the communication cost, however, increase dramatically for volume-coupling.

[8] http://www.precice.org and https://github.com/precice

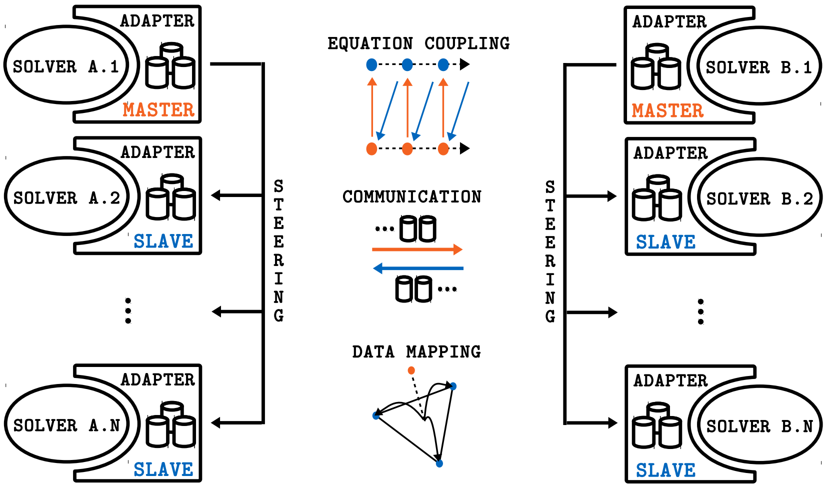


FIGURE 3.4 *Schematic view on the* preCICE *functionality.* preCICE *provides highly efficient and parallel realizations for the main ingredients of partitioned multi-physics simulations, i.e., (i) the equation coupling, enforcing the coupling conditions in an implicit sub-iteration, (ii) the inter-solver communication, realized in a decentralized way with a peer-to-peer concept allowing to inherit parallel scalability of sub-solvers, and (iii) the data mapping routines to translate information between non-conforming grid representations at the interface. Image taken from [Uek16].*

preCICE is open-source and not developed for use with dedicated solvers, it provides efficient parallel realizations of all components required for partitioned multi-physics coupling, it comes with a clean high-level API, is runtime-configurable, and, thus, allows for high flexibility. Therewith, it features short time-to-solution, fosters prototyping and allows non-coupling experts to achieve stable solutions with minimal effort.

An excellent introduction to preCICE from a user's perspective can be found in [Uek16]. Gatzhammer originally presented preCICE in [Gat14] and gave an elaborate tract on all aspects of partitioned multi-physics simulations with an emphasis on FSI. Uekermann extends this work to a fully parallel, decentralized coupling tool that can be applied on massively parallel compute systems. preCICE has been primarily developed by Bernhard Gatzhammer, Miriam Mehl, Benjamin Uekermann, Florian Lindner and myself. I have been a member of the developer team since 2015 focusing particularly on improving robustness, performance and parallel scalability of the iterative solution of the interface coupling based on quasi-Newton methods. The equation coupling methods and their efficient parallel realization on distributed data have been developed in large parts within this thesis.

Coupled Solvers. preCICE has been successfully coupled with a variety of solvers for compressible, and incompressible flow, acoustics, and structural mechanics. This includes fluid-structure-acoustic (FSA) coupling, fluid-structure (FS) coupling, fluid-fluid (FF) coupling and multi-phase coupling. Coupled codes include, e.g., OpenFoam[9], Ateles, Alya[10], Carat[11], SU2[12], FEAP[13], FASTEST, Ansys Fluent and COMSOL. A full listing of coupled solvers and more information about the respective preCICE adapters can be found in [Uek16]. Within this work, we exclusively use fluid and structure solvers based on the OpenFoam toolbox.

[9] http://www.openfoam.org
[10] http://www.bsc.es/es/computer-applications/alya-system
[11] http://www.st.bgu.tum.de/en/lehre0/research/carat
[12] http://github.com/su2code/SU2
[13] http://www.ce.berkeley.edu/projects/feap

OpenFOAM. The open source numerical simulation toolbox OpenFOAM[14] (Open Source Filed Operation and Manipulation) provides solvers and functionality for the numerical simulation of problems in continuum mechanics. It emphasizes on incompressible fluid mechanics, but also compressible flow and non-linear elasticity is supported. The preCICE adapter and the actual fluid and structure solver have been developed by David Blom from the Technical University Delft and are freely available[15]. They build upon the foam-extend-3.1 fork[16].

The ***fluid solver*** uses a second-order finite volume discretization of the incompressible Navier-Stokes equations and employs a second-order implicit time integrator combined with a fully implicit pressure-velocity solver [Dar09], rather than following the standard pressure implicit with splitting of operator (PISO) algorithm. The governing equations are formulated in an arbitrary-Lagrangian-Eulerian point of view. Grid deformation and grid-movement are realized based on radial-basis function interpolation [Boe10].

The ***structure solver*** uses a Lagrangian formulation and a Saint-Venant-Kirchhoff material model with FEM discretization. The time integrator is a second-order implicit scheme.

[14] http://www.openfoam.org
[15] http://github.com/davidsblom/FOAM-FSI
[16] http://www.extend-project.de

4 Evaluation of Robust Quasi-Newton Methods for Partitioned FSI Simulations

This chapter contains numerical results for the performance of the different multi-secant quasi-Newton methods discussed in §2.2. We apply these advanced quasi-Newton methods to accelerate the convergence of the implicit equation coupling at the wet-surface, to reduce instabilities, and to establish a consistent solution for partitioned FSI simulations in every time step. In particular, we present numerical results, that

- indicate the ***sensitivity*** of the convergence for the LS(ξ) method to the parameter $\eta = \eta(\xi)$ of ***reused histories*** in the multi-secant equation, and show the ***improved robustness*** of the ***MV method*** (see §4.2.2),
- compare the different ***restart alternatives*** for the MV method with respect to convergence speed and efficiency (see §4.2.3),
- compare the newly developed, (nearly) "parameter-free" and robust quasi-Newton methods (cf. §2.2) — (*i*) the ***Least-Squares infinity-reuse*** method (LS(∞)) and (*ii*) the ***Multi-Vector Update SVD-restart*** method (MV RS-SVD) — with respect to their ***convergence***, to their overall ***efficiency***, to their dependency and sensitivity to an efficient ***filtering*** method (see §4.2.4), as well as to their ***parallel scalability*** and ***runtime*** (see §4.3).

The numerical analysis presented in this chapter suggests, that the Multi-Vector Update method with SVD-restart approach is a highly robust and efficient acceleration method, which features linear runtime complexity, offers very good parallel scalability and renders most problem dependent tuning parameters redundant. Although various other aspects such as the performance of different coupling schemes, the impact of the number of reused histories, the convergence improvement through filtering, and pre-scaling have been investigated within this work, we limit the presentation herein to the most relevant results towards "parameter-free" and robust quasi-Newton methods for partitioned FSI as presented in §2.2. In §4.2.1, we shortly summarize some of the findings of the omitted aspects and give references. Large parts of this chapter have been already published in [Sch17].

We evaluate the efficiency of quasi-Newton acceleration methods based on the (averaged) number of coupling iterations required to re-establish the monolithic solution with sufficient accuracy in every time step. Since, for most practical cases, the cost for a call to the fluid or solid solver (to solve for the current time step) outweighs the cost of the coupling numerics by orders of magnitude, a low number of coupling iterations is critical. Furthermore, the employed algorithms for the coupling numerics need to offer sufficient parallel scalability to not deteriorate the overall parallel efficiency of the coupled partitioned FSI simulation.

This chapter is organized as follows: An overview of the simulation setting with a detailed description of the three considered test case scenarios is given in §4.1. §4.2 motivates the necessity for more advanced and robust quasi-Newton acceleration methods, and, in particular provides a thorough numerical analysis of the MV restart alternatives and the LS(∞) method, described in §2.2. In §4.3, we present results that show and compare the parallel scalability and runtime of the MV RS-SVD and the LS(∞) method.

4.1 General Setup of Fluid-Structure Interaction Test Cases

For our numerical experiments, we use the three different test cases: a simple 1D elastic tube for extensive numerical tests, a 3D elastic tube and the FSI3 benchmark [Tur06] that, in contrast to the tubes features an elastic structure immersed in the fluid.

4.1.1 1D Elastic Tube with Internal Flow

We consider the simulation of a traveling fluid pressure pulse through an elastic tube. The mathematical model is given by means of a simplified 1D fluid model. Despite its simplified nature and lack of practical relevance, this test case scenario captures all relevant FSI characteristics and features the added-mass instability. Due to its simplicity, it allows for fast prototyping and testing of new methods. The test case has been originally formulated in [Deg10; Deg08a]. The description follows [Gat14].

Mathematical Model. We model a radially-symmetric flexible tube with internal incompressible, inviscid flow. Averaging across the radial coordinate yields a 1D model, as outlined in Fig. 4.1. The conservation of mass and momentum simplifies to

$$\partial_t(av) + \partial_x(av^2) + \frac{1}{\rho}a\partial_x p = 0 \tag{4.1a}$$

$$\partial_t a + \partial_x(av) = 0 \tag{4.1b}$$

with the independent variable x indicating the axial direction of the tube (Fig. 4.1), the inflow velocity v, the kinematic pressure p, the cross sectional area of the tube a and the fluid density ρ. At the inlet, a time-varying sinusoidal inflow velocity is imposed and non-reflecting boundary conditions are applied:

$$v_{in} = v_0 - \frac{v_0}{100}\sin^2(\pi\frac{t}{T}) \quad \text{and} \quad \partial_t v = \frac{1}{c}p. \tag{4.1c}$$

Here, c with $c^2 = c_{mk}^2 - p/2$ is the wave speed with the characteristic Moens-Korteweg wave speed c_{mk} defined as $c_{mk} := \sqrt{Eh/2\rho r_0}$ (with the tube's Young's modulus E, the radius r_0 of the reference configuration, the thickness h of the tube wall and the density ρ). For the structure model, the inertia of the tube wall is neglected, and the flexible wall is modeled by a Hookean constitutive law with linear circumferential stress

$$\sigma_{\varphi\varphi} = E\frac{r - r_0}{r_0} + \sigma_0. \tag{4.2}$$

This configuration leads to significant added-mass effect. The fluid exerts stresses only in circumferential direction. Therefore, the displacement of the tube wall is restricted to radial direction. The

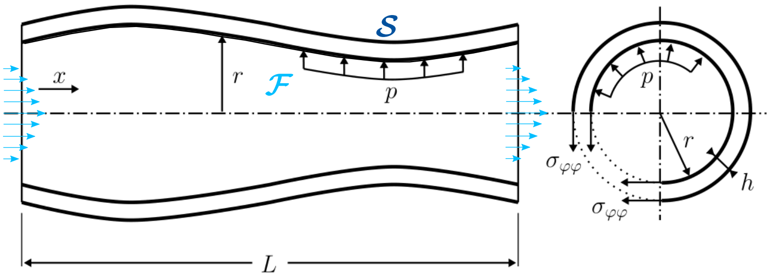


FIGURE 4.1 *Schematic illustration of 1D elastic tube scenario. The scenario models a pressure pulse with flow in x-direction traveling through an elastic tube with radius r, wall thickness h and total length L. The internal flow causes pressure which acts on the inner tube walls in radial direction. Circumferential stresses $\sigma_{\varphi\varphi}(x)$ result in a deformation of the elastic tube wall in radial direction. The test case is based on [Deg08a]. Image modified from [Gat14].*

dynamic coupling condition at the interface is given by $pr = \sigma_{\varphi\varphi}h$. Combining (4.2) and the dynamic coupling condition, the cross sectional area a can be written as an explicit function of the pressure

$$a(p) = a_0 \left(\frac{p_0 - 2c_{mk}^2}{p - 2c_{mk}^2} \right)^2 \tag{4.3}$$

with a_0 and p_0 denoting the cross sectional area and pressure of the initial configuration. Note, that the cross sectional area a and the pressure p are associated with the variables $\boldsymbol{x}_S$ and $\boldsymbol{x}_F$, respectively, used in Chapter 3. (4.1) constitutes the fluid model with input a and output v and p, (4.3) the structure model with input p and output a.

Discretization. We discretize the spatial domain $[0, L]$ using an equidistant, regular grid with cell size $h_x = {}^{L}/{}_{N_S}$, and N_S denoting the number of unknowns in space. The fluid and the solid solver use a matching grid at the coupling interface; the unknowns a_i, p_i and v_i are defined at the cell centers. We employ a second-order accurate finite volume discretization approach for the fluid equations (4.1) with a first-order upwind scheme for the convective part. Further, a pressure stabilization is used. At the inflow and outflow boundary, values from the cell centers are linearly extrapolated to the boundary. For discretization in time, a first-order backward Euler scheme is used with time step $\Delta t = {}^{T}/{}_{N_T}$, subdividing the time horizon $(0, T])$ into N_T time steps. The discretized system can, e.g., be found in [Uek16], p. 39.

Time step size and structural stiffness of the flexible tube control the magnitude of the induced instabilities and overall hardness of the problem. Similar to [Deg08a], we define the dimensionless structural stiffness parameter κ and the dimensionless time step size τ as

$$\kappa := \frac{\sqrt{\frac{Eh}{2\rho r_0} - \frac{p_0}{2\rho}}}{v_0}, \quad \text{and} \quad \tau := \frac{v_0 \Delta t}{L}$$

to control the physical properties of the scenario. The stability analysis in [Deg08a] shows increasing instabilities due to a pronounced added-mass effect for decreasing κ, decreasing τ, and increasing spatial resolution. We fix the spatial resolution to $N_S = 100$ and choose $N_T = 100$ time steps to discretize a full period of the sinusoidal inlet velocity profile.

Numerical Solution. The above described solution algorithm for the fluid sub-component and the quasi-static structure solver are implemented as MatLab modules. An open source C++

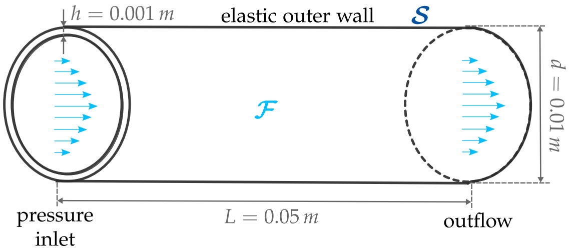


FIGURE 4.2
Schematic illustration of the 3D elastic tube scenario. The tube has a length of $L = 5\text{E}{-}2\,\text{m}$ and a wall thickness of $h = 1\text{E}{-}3\,\text{m}$; both ends are fixed. The internal fluid domain has a diameter of $d = 1\text{E}{-}2\,\text{m}$ in the reference configuration.

implementation is freely available[1] within the preCICE package. The sub-solvers are coupled via the coupling library preCICE, introduced in §3.2.3.

4.1.2 3D Elastic Tube with Internal Flow

Scenario Description. This 3D FSI scenario models the propagation of a flow wave through an elastic tube. It has been widely used in literature as a benchmark application for FSI, e.g., in [Fer05; Bat07; Bog14; Deg09] and was first described in [For01]. The scenario is inspired by the type of flow problems encountered in hemodynamics and blood vessel simulations. The interaction of the internal incompressible flow and the elastic tube wall constitutes a very strongly coupled FSI problem with severe instabilities due to the added-mass effect and a density ratio of almost one between fluid and solid. The geometry of the scenario is outlined in Fig. 4.2. The governing equations for the fluid and the solid are given as described in §3.1.2.

A propagating wave is induced by a pressure driven internal flow. We consider two variations. (*a*) For the first standard configuration, the pressure inlet is initially activated for a duration of $3\text{E}{-}3\,\text{s}$. To avoid spurious pressure waves, the pressure-profile is ramped up linearly with a peak value of $1333.20\,\text{N}\,\text{m}^{-2}$. For $t > 3\text{E}{-}3\,\text{s}$, the pressure is set to zero at the inlet boundary. (*b*) The second case is a modification featuring ***multiple pressure pulses***: the inlet is opened to initiate a pressure pulse with peak value $1333.20\,\text{N}\,\text{m}^{-2}$ at $t = 0\,\text{s}$, $t = 6.00\text{E}{-}3\,\text{s}$ and $t = 1.00\text{E}{-}2\,\text{s}$ with a duration of $3.00\text{E}{-}3\,\text{s}$ each. The outflow boundary uses homogeneous Dirichlet conditions for the pressure. The tube wall is modeled using a hyper-elastic material model. The physical parameters are listed in Tab. 4.1.

TABLE 4.1 *Physical parameters for the FSI3 benchmark and the 3D tube scenario. The table lists the density ρ_f and ρ_s for fluid and solid, respectively, as well as the dynamic fluid viscosity ν_f, the Reynolds number Re, the Young's modulus E, and the Poisson's ratio ν_s.*

FSI3 cylinder flap		**3D flexible tube**	
Fluid	Solid	Fluid	Solid
$\rho_f = 1\text{E}3\,\text{kg}\,\text{m}^{-3}$	$\rho_s = 1\text{E}3\,\text{kg}\,\text{m}^{-3}$	$\rho_f = 1\text{E}3\,\text{kg}\,\text{m}^{-3}$	$\rho_s = 1.20\text{E}3\,\text{kg}\,\text{m}^{-3}$
$\nu_f = 1.0 \times 10^{-3}\,\text{Pa}\cdot\text{s}$	$E = 1.40\text{E}6\,\text{N}\,\text{m}^{-2}$	$\nu_f = 3.0 \times 10^{-3}\,\text{Pa}\cdot\text{s}$	$E = 3.00\text{E}5\,\text{N}\,\text{m}^{-2}$
$Re = 200$	$\nu_s = 0.4$	$Re = 200$	$\nu_s = 0.3$
$\Delta t = 1\text{E}{-}3\,\text{s}$	$\Delta t = 1\text{E}{-}3\,\text{s}$	$\Delta t = 1\text{E}{-}4\,\text{s}$	$\Delta t = 1\text{E}{-}4\,\text{s}$

Numerical Solution. The scenario is included in the OpenFOAM FSI toolkit[2] which has been developed by David Blom for the partitioned coupling of OpenFOAM solvers via preCICE. For the solution of the FSI scenario, we follow the partitioned approach with Dirichlet-Neumann coupling and employ the modified fluid and solid solvers described in §3.2.3. The simulation is carried out in

[1] https://github.com/precice/elastictube1d
[2] https://www.github.com/davidsblom/FOAM-FSI

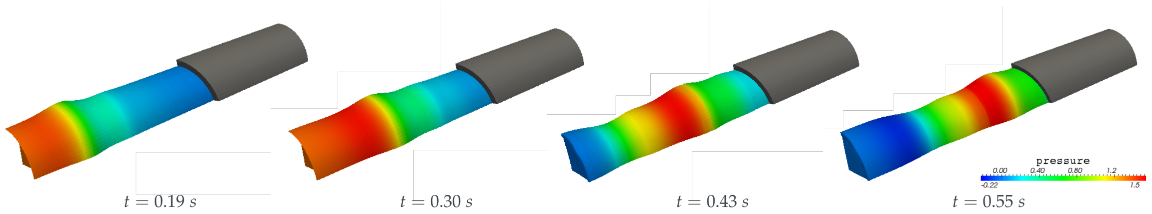


FIGURE 4.3 *Pressure profile and flow induced structural deformations for the 3D elastic tube scenario.*

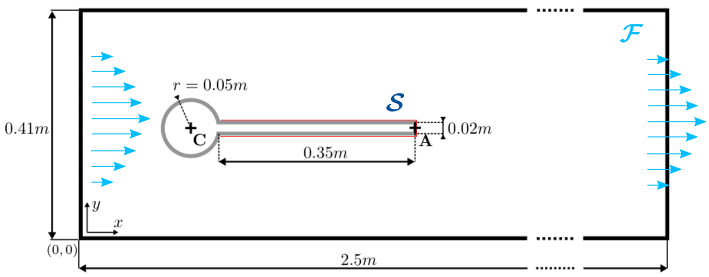


FIGURE 4.4 *Schematic illustration of the FSI3 benchmark scenario. A fixed cylinder with attached elastic cantilever is placed a little off-centric at point $C = (0.20\,\mathrm{m}, 0.20\,\mathrm{m})$. The Displacement of the cantilever is monitored at point $A = (0.60\,\mathrm{m}, 0.20\,\mathrm{m})$. A parabolic inflow velocity profile is imposed at the left boundary. Free outflow is assumed for the right boundary. The test case was originally proposed in [Tur06]. Image modified from [Gat14].*

parallel using preCICE. The respective preCICE adapters for the fluid and the solid solver are also included in the mentioned FSI toolkit. To ensure a consistent solution, we use a relative convergence criterion of $\varepsilon_r = 1\mathrm{E}{-5}$ in (3.10) for the termination of the implicit coupling per time step. Due to symmetry, the simulation can be reduced to only a quarter of the tube. We use unstructured grids for both components with 16,000 hexahedral cells for the fluid and 6,400 hexahedral cells for the solid solver. At the coupling interface, both grids match, and share 9,600 unknowns. We simulate a total time of $T = 1\mathrm{E}{-2}\,\mathrm{s}$, subdivided into 100 time steps of size $\Delta t = 1\mathrm{E}{-4}\,\mathrm{s}$. This corresponds to a full simulation period, i.e., a full propagation of the pressure wave through the tube. Fig. 4.3 shows the traveling pressure pulse and the induced deformations for four different time instances.

4.1.3 FSI3 Benchmark (Cylinder Flap)

Scenario Description. The FSI3 scenario was proposed by Turek and Hron [Tur06] as a standard FSI benchmark problem. It models a 2D flow around a fixed cylinder with attached flexible cantilever. To foster oscillations, the cylinder is placed a little offcentric in vertical direction in the two dimensional flow canal. Fig. 4.2 shows an illustration of the scenario's geometry. The mathematical models as presented in §3.1.2 are used. The incompressible flow is driven by a parabolic inflow velocity profile from the left with a mean inflow velocity of $\bar{v} = 0.20\,\mathrm{m\,s^{-1}}$. Free outflow is considered at the right boundary. No-slip boundary conditions are applied for the top and bottom boundary as well as on the cylinder and flap surface. To model the elastic cantilever, which is deflected by the flow around the cylinder, we employ a Saint-Venant-Kirchhoff material model. The physical parameters for both, the fluid and the structure, are listed in Tab. 4.1. Turek and Hron originally propose three different variations. Here, we limit ourselves to the most challenging third case. After an initialization phase, the unstationary flow induces regular oscillations of the attached cantilever. The magnitude of these oscillations can be monitored and compared to a reference solution.

Numerical Solution. As before, we employ partitioned coupling and use OpenFOAM in conjunction with preCICE for the numerical treatment of the test case. The fluid and the solid solver are identical to the above 3D tube scenario. We bypass an up-ramping of the velocity magnitude by using a

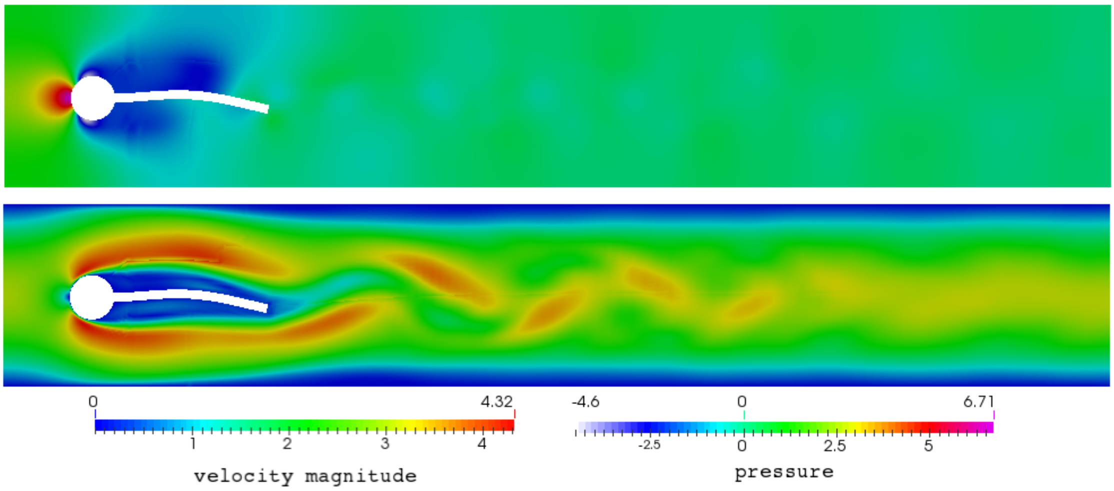


FIGURE 4.5 *Pressure and velocity profile for FSI3 benchmark cylinder flap scenario. The pressure values (top) are given in* Pa *and the velocity magnitude (bottom) is given in* $\mathrm{m\,s^{-1}}$.

pre-computed velocity field for the canal flow as initial value. We use a rather coarse grid with 23, 924 cells for the flow solver and 328 cells for the solid solver, respectively. The grids are conforming and match at the interface. This setup results in 672 unknowns at the interface for the (J) system. After an initialization phase of 10 s, we monitor results for the simulation of a time frame of 0.18 s, which corresponds to 10 oscillation periods of the cantilever. A time step of $\Delta t = 1\mathrm{E}{-3}\,\mathrm{s}$ is used. As before, we impose a relative criterion with $\varepsilon_r = 1\mathrm{E}{-5}$ in (3.10) for termination of the implicit coupling iteration.

Within this work, we focus on the performance of the implicit coupling iteration using different quasi-Newton acceleration schemes. Quantitative simulation results to validate the correctness of the numerical solution are presented in [Bun16a] for the given configuration and solvers. These results are in very good agreement to the reference solution given in [Tur06], independent of the convergence acceleration method of the implicit coupling iterations.

4.2 Convergence-Analysis of Quasi-Newton Variants

All of the above described FSI test case scenarios are characterized by strong physical interactions and thus, depict challenging benchmarks for partitioned FSI simulations. Due to the strong oscillations and instabilities, powerful quasi-Newton methods are a necessity for convergence. We report numerical results for, and analyze the performance of the quasi-Newton methods developed within this work (presented in §2.2). At first, we motivate the necessity of stabilizing features such as filtering or implicit reuse in §4.2.2. A numerical analysis of the different restart alternatives is given in §4.2.3. The restart strategy represents the central component that allows us to reduce the MV cost to linear complexity and make MV feasible for large scale simulations. Aiming for robust efficient and parameter-free quasi-Newton acceleration methods, we thoroughly analyze the herein developed MV RS-SVD method and compare it with the LS(∞) method in §4.2.4. The latter is equipped with a powerful filtering method to ensure stability.

4.2.1 Summary of Results Beyond the Scope of this Chapter

As mentioned above, the numerical analysis carried out within this chapter puts an emphasis on the newly developed, efficient, robust and scalable MV restart alternatives from §2.2. The focus is clearly towards "parameter-free" and efficient advanced quasi-Newton variants. In the following, we briefly summarize other important findings that have been obtained within this work[3], and give references to respective publications.

Serial (GS) versus Parallel (J) Equation Coupling Scheme. Apart from Tab. 4.2, we only present results for the parallel Jacobi-type (J) system. When combined with powerful quasi-Newton acceleration methods, both equation coupling systems, the (GS) and the (J) system, show very similar results in terms of required coupling iterations [Sch15; Uek16; Lin15]. Depending on the application problem, (J) might even outperform (GS) in terms of coupling iterations. Even for cases where the (J) system leads to a slightly higher iteration count, it's overall efficiency always outperforms the (S) system due to a better parallelization and simultaneous sub-component execution. We therefore only consider the (J) system.

Performance of Different Equation Coupling Schemes. As coupling scheme, we refer to the combination of equation coupling strategy with a suitable acceleration method. In [Sch15] we examined a large variety of such combinations, including underrelaxation and sophisticated quasi-Newton acceleration methods. [Uek16] extends this study by additional variants. For all tested scenarios, the Jacobi-type equation coupling with either the LS(ξ) method or the MV method, i.e., (J)-LS(ξ) and (J)-MV, respectively yield the best performance.

Improving Robustness via Filtering and Pre-Scaling. When reusing information from previous time steps, the application of a powerful filtering technique becomes necessary to maintain stability and robustness of the method. A suitable filtering technique in combination with reuse of past information can speed up convergence drastically (see, e.g. Tab. 4.3, and Fig. 4.6). Within this work, three filter techniques, the QR1, QR2, and POD filter (see §2.2.1) have been implemented in the black-box coupling library preCICE. A thorough numerical analysis of the performance of the different filtering methods can be found in [Hae15] for a number of test cases and different solvers. Large parts of the results therein have been obtained within this work. The numerical results reported in [Hae15] suggest that the newly introduced QR2 filter, which judges columns by their amount of new information, outperforms the well established QR1 filter, and also the POD filter variant. In Fig. 4.6, we show an exemplary study considering the QR1 and QR2 filter used in combination with the LS(ξ) and MV method for the 1D flexible tube scenario. We can conclude, that, for this setting, (*i*) filtering becomes increasingly important for an increasing number of time steps, that (*ii*) the QR1 filter fails to stabilize, and that (*iii*) only the QR2 filter allows to stabilize the LS(ξ) method sufficiently to converge and reach the MV performance.

In [Uek16], Uekermann extends this analysis and considers different sub-component solvers; namely, Alya Nastin for the fluid and Alya Solidz for the structure solver. He concludes that the performance of the filtering techniques is problem and solver dependent: For the Alya Nastin–Alya Solidz coupling, the QR1 filter yields best results. As opposed to the sophisticated radial-basis function interpolation-based grid-movement of the OpenFOAM solvers, the approach employed within the Alya coupling is very simple. Uekermann mentions this as reason for the discrepancy.

As different coupling variables may live on fairly different scales, the parallel (J) system requires a pre-scaling to counterbalance the respective weight of the variables in the computation of the

[3] as a results of joint work with Benjamin Uekermann, Rob Haelterman, Miriam Mehl and myself

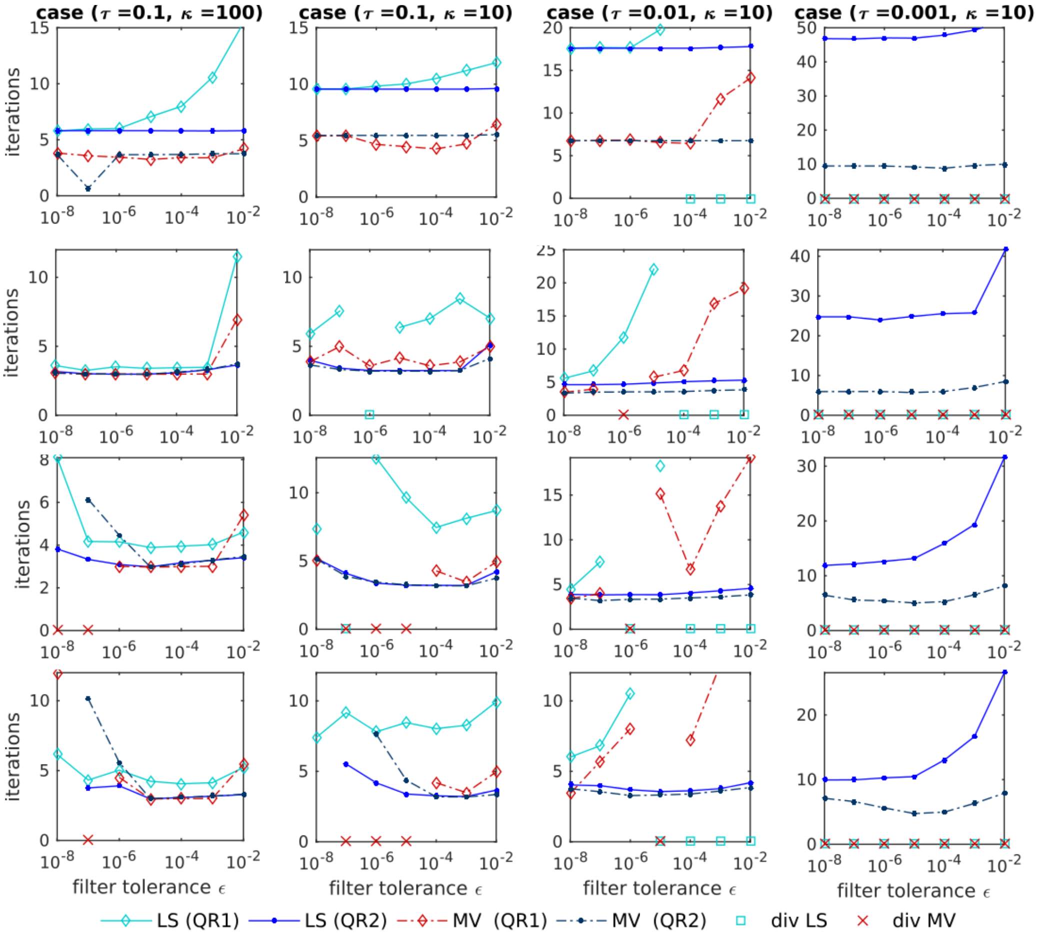


FIGURE 4.6 *Filtering techniques for LS and MV method, applied to the one-dimensional flexible tube scenario. We report average iteration counts for the implicit coupling per time step of the one-dimensional flexible tube scenario (see §4.1.1) using the LS(ξ) and the MV method. In particular, we study the stabilization properties of the QR1 and QR2 filter, described in §2.2.1 for different amount of reused information from the past. The first row shows $\xi = 0$, followed by reuse of information from $\xi = 4$, $\xi = 8$, and $\xi = 16$ previous time steps. We consider four different parameter settings: The induced instabilities and overall hardness of the problem increases for decreasing structural stiffness κ and decreasing time step size τ.*

quasi-Newton update. Uekermann proposes different strategies for an automatic choice of the correct pre-scaling weight. An analysis is given in [Uek16].

Combining Manifold Mapping with Quasi-Newton Methods. In joint work with David Blom, we investigated Manifold Mapping as acceleration method for partitioned fluid-structure interaction simulations. Manifold Mapping [Ech05] is a defect-correction multi-level approach to accelerate the solution of a high-fidelity model by means of repeatedly solving a surrogate low-fidelity or coarse model. An introduction to Manifold Mapping and its application in the context of partitioned fluid-structure interaction can be found in [Blo14a; Blo15a]. In [Blo15b] we applied Manifold Mapping combined with different quasi-Newton methods to accelerate the fixed-point equation coupling for partitioned FSI. By using quasi-Newton methods such as LS and MV as solver for the coarse model, we could show a reduction of the number of high-fidelity (coupling) iterations of about 50%. Within this work a prototype for the Manifold Mapping algorithm for the (GS) and (J) system has been implemented into the coupling library preCICE. The implemented quasi-Newton solvers LS, MV and MV RS-SVD can be used as coarse model solvers and be exchanged in a modular way.

4.2.2 Basic Methods: LS(ξ) versus Plain-Vanilla MV

Purpose. We investigate the convergence properties of the LS(ξ) method with explicit reuse of information from ξ previous time steps, as opposed to the convergence properties for the MV method with implicit reuse of past information. We showed in [Sch15; Lin15] that the number of reused vectors critically affects the convergence properties of the LS(ξ) method. An optimal choice for ξ is problem dependent and involves a costly trial and error tuning. The MV method, on the other hand, has been shown to yield best results if only vectors from the current time step are reused [Sch15]; [Lin15].

Setup. We consider all three test scenarios described in §4.1. The numerical treatment of the cases is as described above, using preCICE to couple the OpenFOAM or simple 1D sub-component solvers, respectively. For the 1D tube scenario, we vary the structural stiffness κ and the dimensionless time step size τ, resulting in nine different cases with increasing degree of instability for decreasing κ and τ. For the OpenFOAM scenarios, we consider the (GS) and the (J) system and report results for MV without explicit reuse compared to LS with explicit reuse of $\xi \in \{0, 2, 4, 6, 8, 16, 32\}$ previous time steps. A relative convergence criterion with $\varepsilon_r = 1\text{E}{-5}$ is used for all scenarios. The fluid and the structure sub-components are solved with a criterion that is two orders of magnitude tighter.

Results. We report average iteration counts for MV and LS(ξ), $\xi \in \{0, 8\}$ for the 1D flexible tube scenario in Fig. 4.7, and for $\xi \in \{0, 2, 4, 6, 8, 12, 16, 32\}$ for the OpenFOAM 3D tube and 2D FSI3 benchmark scenario in Tab. 4.2.

TABLE 4.2 *Comparison of the convergence properties of the LS(ξ) versus the MV method. We report the average number of coupling iterations for the LS(η) method with explicit reuse, and the MV method, using implicit reuse of past information. The amount of explicitly retained previous information $\eta = \eta(\xi)$ is varied for the LS method. Results are reported for the (GS) and the (J) equation coupling system, respectively, for the 2D FSI3 benchmark scenario ([Tur06]; §4.1.3; Fig. 4.5) and the 3D flexible tube scenario ([For01]; §4.1.2; Fig. 4.3). The relative convergence criterion is set to* $\varepsilon_r = 1\text{E}{-5}$.

	3d Flexible Tube					**FSI3 Cylinder Flap Benchmark**						
reuse $\eta(\xi)$	0	4	8	12	16	0	2	4	6	8	16	32
(GS)-LS	15.5	9.3	**8.8**	9.2	9.4	11.5	6.2	**5.3**	5.5	5.7	6.6	7.8
(GS)-MV	**8.5**					**5.5**						
(J)-LS	28.9	14.6	13.4	**13.2**	13.3	20.0	8.0	6.2	5.6	**5.3**	6.2	11.9
(J)-MV	**11.6**					**6.2**						

Observations. For all scenarios, the LS convergence properties highly depend on the number ξ of reused time steps. Without reuse of previous information, the LS method is not competitive, the optimal amount of reuse, however, remains problem dependent and is different for every scenario. In Fig. 4.7 for example, LS shows excellent performance for the more challenging cases using $\xi = 8$. For the easier cases, however, this configuration fails to converge. Typically, for the (J) system a higher amount of reuse is beneficial than for the (GS) system (see also [Uek16]). For an optimal choice of explicit reuse, LS shows fairly similar performance as MV. This is also supported by results in [Uek16; Sch15].

Conclusion. The LS method benefits from explicit reuse. The optimal amount of reused information is, however, highly problem dependent and involves a costly tuning process. The MV method outperforms LS and renders the tuning parameter redundant.

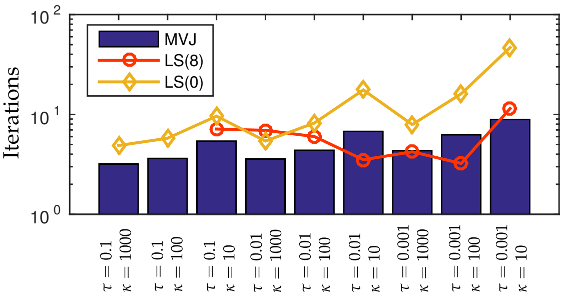


FIGURE 4.7
1D flexible tube scenario. Comparison of the average number of coupling iterations required using LS(8), LS(0), and MV, respectively. We vary the structural stiffness κ and the time step size τ. The scenario becomes more challenging for decreasing κ and decreasing τ. Image modified from [Sch17].

4.2.3 Advanced MV Methods: Restart Alternatives

Purpose. We have seen the excellent convergence properties of the MV method, which by relying on implicit reuse, renders the costly tuning of ξ for an acceptable LS convergence dispensable. Implicate reuse comes, however, at the price of having to keep an explicit representation of the Jacobian. The resulting quadratic storage and runtime complexity (cf. Tab. 2.1) makes the method impractical for large scale simulations. We proposed three different MV restart approaches (cf. §2.2.2), that, with an efficient limited memory Jacobian representation in conjunction with periodic restart, reduce the method's complexity to a linear one. Due to the superior robustness of the MV method, we restrict ourselves to only compare the MV restart variants among each other and with the plain-vanilla MV method, but not to LS(ξ), in terms of convergence properties The latter are variable as proposed restart strategies differ in their ability to recover Jacobian information across restarts, and, thus, influence the quality of the Jacobian estimation.

Setup. We benchmark and compare the restart variants for the one- and three-dimensional flexible tube scenario described in §4.1.1 and §4.1.2, respectively. The numerical solution and setup follows the description of §4.1.1 and the previous experiment in §4.2.2. For the 1D tube scenario, the degree of the added mass effect and the amount of induced instabilities is controlled by specific choices for the structural stiffness parameter κ and the time step length τ. We consider four different cases $(\tau, \kappa) \in \{(0.1, 100), (0.1, 10), (0.01, 10), (0.001, 10)\}$ with increasing difficulty. We apply three approaches, the "clear all" RS-0, the "explicit reuse at restart" RS-LS, and the "subspace tracking" RS-SVD approach, and study effects of different choices for the restart period m (also referred to as "chunk size"), the truncation threshold ε for the subspace-tracking alternative, and the number of explicitly reused time steps ξ_{rs} at restart for the RS-LS method. For the 3D flexible tube scenario, we use $\xi_{rs} = m$ and collect the first five iterations from each of the last ξ_{rs} time steps for explicit reuse at restart. We additionally employ a QR2 filter with $\varepsilon_F = 1\text{E}{-2}$ to ensure stability. For all variants, we use the (J) system only, together with a residual-sum pre-scaling. For all experiments, a relative convergence criterion of $\varepsilon_r = 1\text{E}{-4}$ as given in (3.10) is used.

Results. In the course of prototyping and preliminary method testing, we analyze all three restart alternatives for the 1D flexible tube scenario in Fig. 4.8. Despite its simplistic character the scenario features different common characteristics of partitioned FSI simulations and allows to easily vary the degree of induced instability and the hardness of the problem. We study the influence of different choices for the ***restart period*** m, reflecting the number of time steps that fit in the method's limited memory. For the SVD-restart method, we investigate the trend of the rank of the SVD (after truncation) and its reflection in the average iteration count for different truncation thresholds. For the RS-LS method, we study the method's performance in relation to the amount of retained information

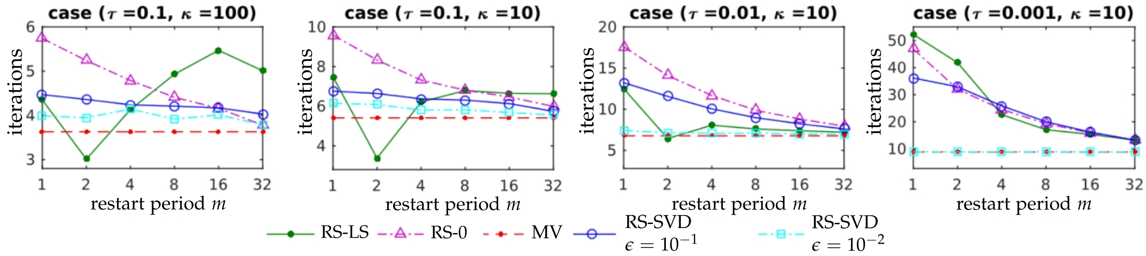


FIGURE 4.8 *MV restart variants for the 1D flexible tube scenario. Comparison of the average number of coupling iterations required using different MV restart strategies, outlined in §2.2. The chunk size m, i.e., the restart period is varied in $m \in \{1,2,4,8,16,32\}$. We report iteration counts for the RS-0 (magenta), the RS-LS (green), and the RS-SVD (blue) strategy. For the latter, a truncation threshold of $\varepsilon = 1\text{E}{-1}$ and $\varepsilon = 1\text{E}{-2}$ is used. Iteration counts for MV (red) are given as a reference. Instabilities and the hardness of the problem increases for decreasing structural stiffness κ and decreasing time step size τ. Image modified from [Sch17].*

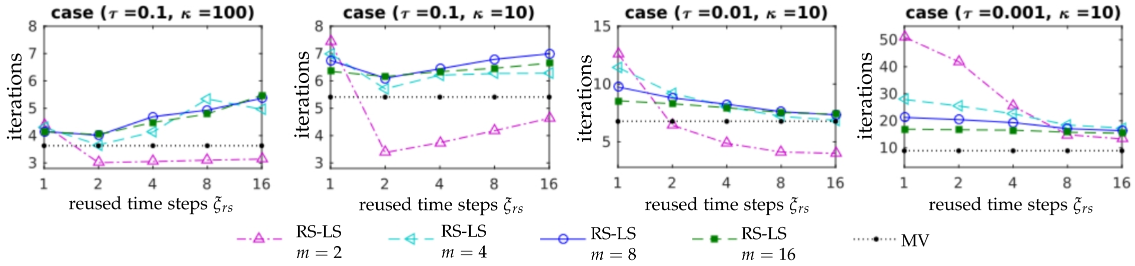


FIGURE 4.9 *RS-LS restart strategy for the 1D flexible tube scenario. Average number of coupling iteration for the RS-LS restart strategy with different number $\xi_{rs} \in \{1,2,4,8,16\}$ of retained previous time steps at restart. We report results for different restart periods $m \in \{2,4,8,16\}$. Iteration counts for MV (black) are given as a reference. Instabilities and the hardness of the problem increases for decreasing structural stiffness κ and decreasing time step size τ. Image modified from [Sch17].*

across restart, i.e., the number ξ_{rs} of explicitly reused time steps. Results for the latter are given in Fig. 4.9. For a setting with more realistic problem dimensions and characteristics, we compare all three variants in Fig. 4.10 in a similar fashion for the 3D flexible tube scenario.

Observations. The most important observation is, that the RS-SVD restart approach shows excellent results for all considered experiments and preserves the original MV convergence properties. In particular, this is achieved at a fraction of the original cost due to the small rank of the truncated SVD, indicated by red lines in Fig. 4.10 (we investigate this in more detail in §4.2.4). The performance slightly depends on a good choice for the truncation parameter ε.

Considering the 1D example, we observe that the RS-0 strategy (which discards all accumulated information after m time steps) yields the worst performance. This method represents a mixture between the LS(0) and the MV method, and, therefore, inherits the bad LS(0) convergence properties. The implicit reuse within the MV part and its stabilizing properties become more pronounced for larger restart periods m, and the number of coupling iterations decreases for increasing restart periods. If, in contrast, explicit reuse of multi-secant information at every restart is considered, such as done for the RS-LS method, good convergence is achieved for very small restart periods $m \in \{1,2\}$. Only for the hardest case with $(\tau = 0.001, \kappa = 10)$ the amount of retained information across restart borders is not sufficient and a larger m is required. The good convergence of RS-LS$(\xi_{rs}, m = 2)$ for cases 1–3 raises hope for a very cheap and efficient alternative for scenarios with a moderate degree of instability. On the downside, employing explicit reuse at restart re-introduces the dependency on the parameter $\xi = \xi_{rs}$, but in a less sensitive way. This can be seen from Fig. 4.9, where the amount

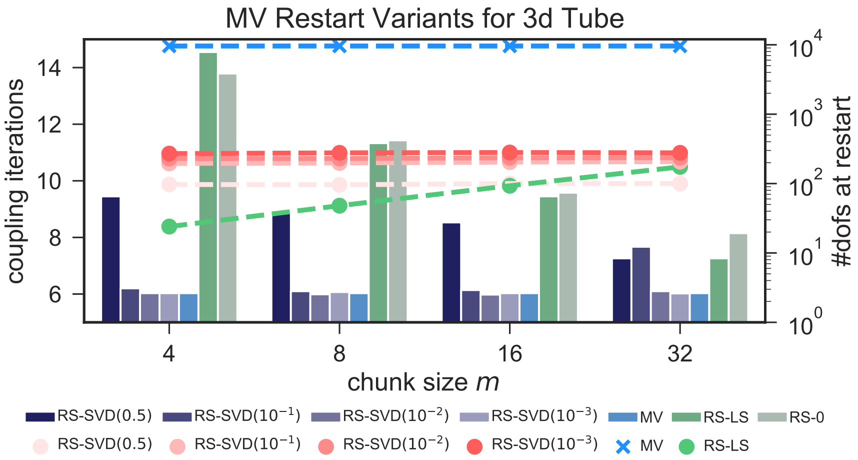


FIGURE 4.10 *MV restart variants for the 3D flexible tube scenario. Comparison of the average number of coupling iterations required using different MV restart strategies, outlined in §2.2. The restart period m is varied in $m \in \{4, 8, 16, 32\}$. We report iteration counts (left axis) for the RS-0 (gray), the RS-LS (green), and the RS-SVD (shades of navy-blue) strategy. For the latter, we monitor the rank of the truncated SVD representation (right axis; red dashed lines) for varying truncation thresholds $\varepsilon \in \{0.5, 1\mathrm{E}{-1}, 1\mathrm{E}{-2}, 1\mathrm{E}{-3}\}$ and outline the sensitivity of the iteration count to ε. For the RS-LS strategy, the number of reused columns at restart is given (green dashed line). The dimensionality N of the Jacobian (as required for the plain-vanilla MV method) is given as a reference (blue dashed line), along with the average MV iteration counts (blue bars). We use a QR2 filter with $\varepsilon_F = 1\mathrm{E}{-2}$ and a relative convergence criterion with $\varepsilon_r = 1\mathrm{E}{-4}$. Image modified from [Sch17].*

of explicitly reused information at restart (i.e., the number of time steps ξ_{rs}) is varied for the RS-LS method. As before, a small restart period of $m = 2$ outperforms other choices of m, regardless of the number of reused time steps (except for the last case). The optimum is reached for $\xi_{rs} = 2$ for cases 1 and 2, and for $\xi_{rs} = 8$ for case 3. This coincides with the optimal parameter ξ for LS(ξ) for the respective cases (cf., e.g., [Uek16], p. 42). For this scenario, RS-LS($M = 2, \xi_{rs}$) even outperforms MV as it benefits from both reuse approaches.

Considering the practically more relevant 3D flexible tube test case, Fig. 4.10 exhibits poor convergence for both the RS-0 and the RS-LS strategy. To some extend, this contradicts the results for the RS-LS method, for the one-dimensional example. Fig. 4.10 furthermore compares the number of retained columns (dashed green line) for the RS-LS method, and the average rank of the truncated SVD representation of the inverse Jacobian (dashed red lines) for the RS-SVD method, respectively. The total number of unknowns at the interface (dashed blue line) is given as a reference. For both, methods RS-LS and RS-SVD, the degrees of freedom are reduced to a fraction of the original system.

Conclusion. We conclude, that the SVD-based restart approach (RS-SVD) outperforms all other variants in terms of convergence speed and robustness. In particular, it achieves identical performance as the original MV method. A slight, yet clearly identifiable dependency on the truncation parameter ε of the SVD representation, remains. The cost is reduced dramatically, due to a very low rank of the truncated SVD representation of the Jacobian, compared to the number of unknowns at the interface. The RS-LS method may, in some cases, be a cheap and efficient alternative, however, its convergence properties are highly problem dependent.

4.2.4 Towards Parameter-Free Methods: MV SVD-Restart and LS(∞)

In the following, we benchmark and compare the advanced, "parameter-free" and robust quasi-Newton variants RS-SVD and LS(∞), both featuring linear runtime and storage complexity. We

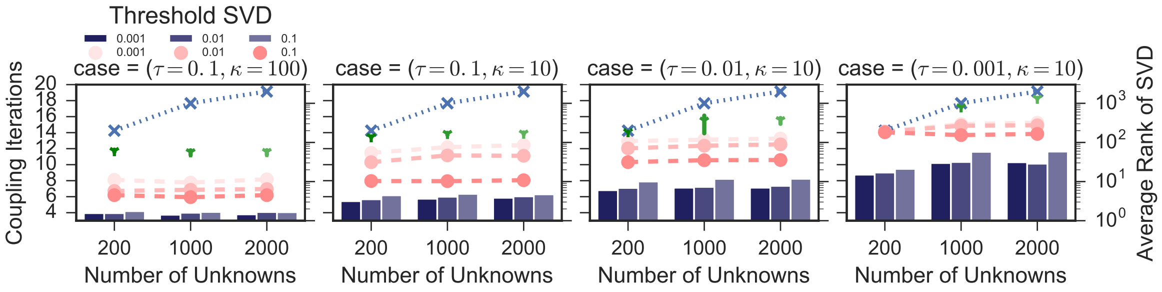


FIGURE 4.11 *SVD-restart strategy for the 1D flexible tube scenario. We report the method's performance in terms of average number of required coupling iterations (left axis; shades of navy blue) for varying truncation thresholds $\varepsilon \in \{1E{-}1, 1E{-}2, 1E{-}3\}$ for the subspace tracking strategy at restart. The average rank of the truncated SVD after restart is given as red dashed lines (right axis), dependent on the truncation threshold. In particular, the sensitivity of the truncated rank of the SVD with respect to an increasing number of unknowns $N \in \{200, 1000, 2000\}$ is analyzed. As reference, the number of unknowns N (blue dashed line) and the actual rank of the Jacobian as estimated by the plain-vanilla MV update formula (green markers) are outlined. The restart period m is fixed to $m = 8$. We consider different cases with increasing instabilities for decreasing structural stiffness κ and decreasing time step size τ. Image modified from [Sch17].*

provide thorough numerical analysis of convergence properties and influencing parameters: (*i*) the threshold ε for the truncation of the SVD, (*ii*) the length of the restart period m, and (*iii*) the employed filtering method for LS(∞).

Low-Rank Approximability in MV SVD-Restart.

Purpose. The rank $\bar{\kappa}$ of the SVD representation of the inverse Jacobian $\boldsymbol{M}^{-1}$ critically affects the method's runtime complexity (cf. Tab. 2.1). An overall linear runtime complexity (in the number of unknowns N at the interface) can only be achieved if the rank of the internal representation of $\boldsymbol{M}^{-1}$ is substantially smaller than, and independent on N. Second, the actual inverse Jacobian $(\nabla \boldsymbol{\mathcal{R}})^{-T}$ needs to be low-rank approximable. Further, the rank needs to be independent of the number of unknowns N at the interface. We investigate the trend of the actual rank of $\boldsymbol{M}^{-1}$ and the rank of the truncated SVD representation for a growing number of unknowns N.

Setup. As before, we consider the one- and three-dimensional flexible tube scenario, described in §4.1.1 and §4.1.2, respectively. Apart from what follows, we use an identical setup as in §4.2.3. For the 1D flexible tube scenario, we consider the earlier viewed characteristic cases and run experiments with a growing number of unknowns at the interface, i.e., $N \in \{100, 200, 1000\}$. For the 3D tube scenario, we consider five levels of refined grids, that result in a series where the number of unknowns at the interface doubles at each refinement step.

Results. We report the trend of the average rank $\bar{\kappa}$ (red dashed lines) of the truncated SVD representation of $\boldsymbol{M}^{-1}$ and the required number of coupling iterations (bars in shades of navy blue) for a growing number of unknowns N at the interface in Fig. 4.11. Three different truncation thresholds $\varepsilon \in \{0.10, 0.01, 0.00\}$ are considered. The actual rank of the estimator $\boldsymbol{M}^{-1}$ as obtained from the original MV method is indicated with green markers. A similar study is carried out for the 3D tube scenario (going from $N = 9{,}600$ to $N = 153{,}600$) and presented in Fig. 4.12. The number of unknowns at the interface is given as a reference (blue dotted line). In Fig. 4.12, we furthermore report the required number of coupling iterations for the original MV method on a grid with $N = 9{,}600$ unknowns at the interface. Beyond this point, the MV method exceeded the system's memory resources.

Observations. The most important observation is, that the rank $\bar{\kappa}$ (indicated by dashed red lines) of

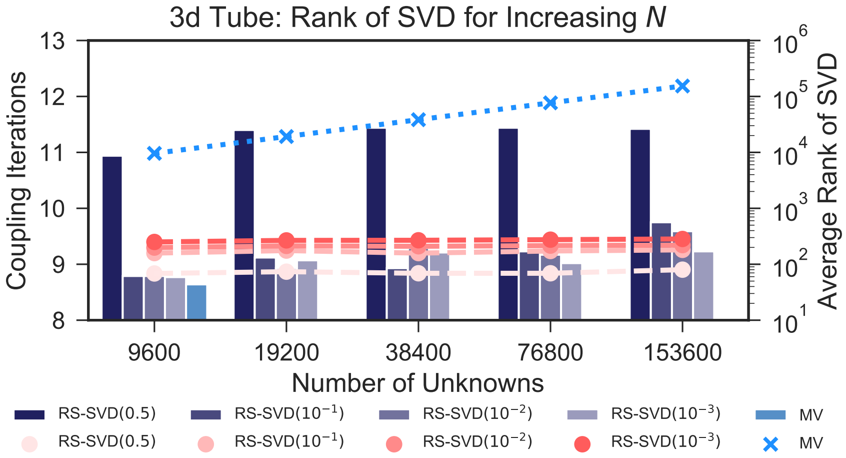


FIGURE 4.12 *SVD-restart strategy for the 3D flexible tube scenario. We report average number of required coupling iterations (left axis; shades of navy blue) for varying truncation thresholds* $\varepsilon \in \{0.5, 1\text{E}{-1}, 1\text{E}{-2}, 1\text{E}{-3}\}$ *applied to the subspace tracking strategy at restart. The average rank of the truncated SVD after restart is given as red dashed lines (right axis), dependent on the truncation threshold. In particular, we analyze the sensitivity of the truncated rank of the SVD with respect to an increasing number of unknowns N at the interface. The average iteration counts for the plain-vanilla MV method and the dimensionality of the corresponding Jacobian (blue dashed line) are outlined as a reference. A restart period of* $m = 8$ *and a relative convergence criterion of* $\varepsilon_r = 1\text{E}{-4}$ *are used. For stabilization, a QR2 filter with threshold* $\varepsilon_F = 1\text{E}{-2}$ *is applied. Image modified from [Sch17].*

the truncated SVD remains constant for both scenarios (cf. Fig. 4.11 and Fig. 4.12) for an increasing number of unknowns N, provided that the discretization has become sufficiently fine (N sufficiently large) to capture all physically relevant interface modes. At the same time, the average iteration count (depicted as bars in shades of navy blue) increases slightly, which is a little more pronounced for the 3D tube in Fig. 4.12. We explain this by the effect of the employed grid-coarsening strategy[4] within OpenFOAM, which results in a loss of accuracy and consequently higher iteration numbers to establish the equation coupling[5]. Nonetheless, the iteration counts almost reach the numbers for the original MV method for a suitable choice of the truncation threshold. In particular, the original MV becomes infeasible beyond $N = 9{,}600$ unknowns at the interface, while the RS-SVD approach remains efficient and stable far beyond this point.

Conclusion. We conclude that, for the considered scenarios, the average rank $\overline{\kappa}$ of the truncated SVD representation of $\boldsymbol{M}^{-1}$ is very small and remains (nearly) constant for ever finer resolutions and a growing number of unknowns N at the interface. This is a very important characteristic and renders the SVD-restart approach to be highly efficient.

Sensitivity to Restart Period m and Truncation Parameter ε.

Purpose. Since we claim the MV RS-SVD method to be "parameter-free", we now investigate the sensitivity of the method's performance with respect to its two remaining parameters: the length of the restart period m (in number of time steps) and the truncation threshold ε for the subspace tracking.

Setup. We consider the FSI3 benchmark scenario (see §4.1.3) and the 3D tube scenario with a single pressure pulse as used above, and, additionally, a variation with multiple pressure pulses

[4] Due to the high dimensionality of the problem, a grid-coarsening is applied which selects only a small subset of the grid points for the grid-movement computations. This allows us to go to higher levels of grid refinement.

[5] The effect is emphasized for finer resolutions, as the coarse grid used for grid-movement calculation remains unchanged and, thus, the error increases.

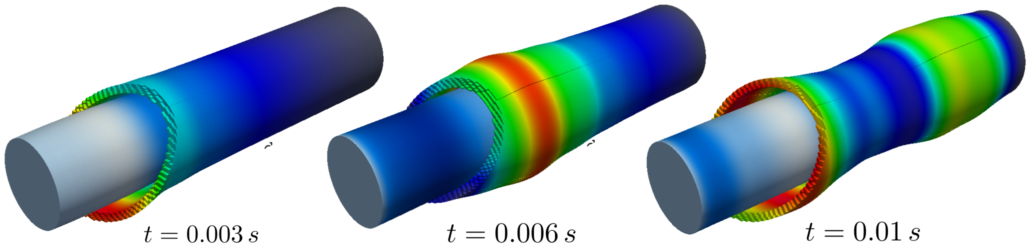


FIGURE 4.13 *3D tube scenario with multiple pressure pulses. The flow induced deformations of the flexible tube and the traveling pressure wave are visualized for the 3D tube scenario at three different time points. The pressure inlet is opened repeatedly to initiate multiple traveling pressure pulses. Image modified from [Sch17].*

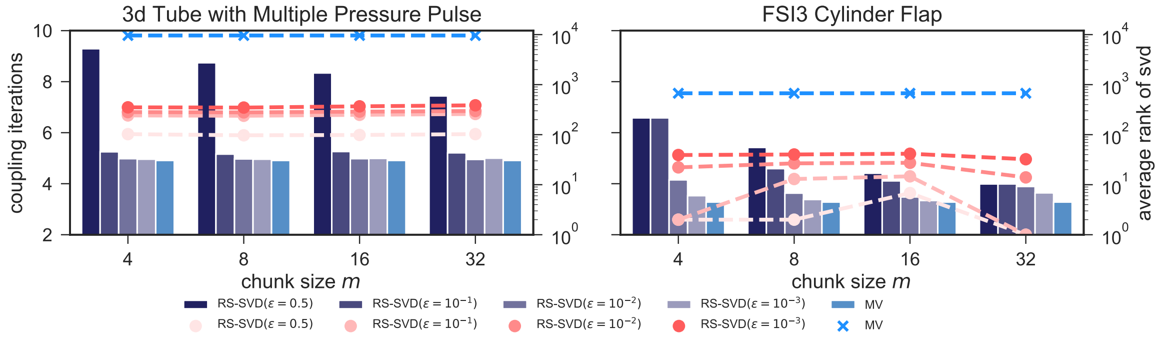


FIGURE 4.14 *SVD-restart strategy for the 3D flexible tube (with multiple pressure pulses) and the FSI3 benchmark scenario. We report the average number of required coupling iterations (left axis; shades of navy blue) using different restart periods $m \in \{4, 8, 16, 32\}$ and varying truncation thresholds $\varepsilon \in \{0.5, 1\text{E}{-}1, 1\text{E}{-}2, 1\text{E}{-}3\}$ applied to the subspace tracking strategy at restart. The average rank of the truncated SVD after restart is given as red dashed lines (right axis), dependent on the truncation threshold. The average iteration counts for the plain-vanilla MV method and the dimensionality of the corresponding Jacobian (blue dashed line) are outlined as a reference. A relative convergence criterion of $\varepsilon_r = 1\text{E}{-}4$ is used and a QR2 filter with threshold $\varepsilon_F = 1\text{E}{-}2$ is applied. Image modified from [Sch17].*

(see §4.1.2 and Fig. 4.13). This modified tube scenario is used to foster displacements of the tube wall that are harder to capture with quasi-Newton approaches. Further, results for the 1D tube scenario are evaluated. The numerical setup is identical to the experiments above.

Results. We report results for the convergence of the RS-SVD method (in terms of the number of coupling iterations; navy blue bars) and the resulting average rank (dashed red lines) for different choices of the restart period m and variations of the truncation threshold ε. Results for the one- and three-dimensional tube are given in Fig. 4.8 and Fig. 4.10, respectively. Results for the 3D tube with multiple pressure pulses and the FSI3 benchmark scenario are given in Fig. 4.14.

Observations. With respect to the restart period m, we observe that the method is quite insensitive for a proper choice of the truncation threshold ε. In theory, the restart period m controls the emphasis on the employed restart strategy compared to the MV properties for implicit reuse (cf. the RS-0 curves for different m in Fig. 4.10). Small choices for m stress the restart and require a sophisticated transmission of already captured information from the past across restart borders. From Fig. 4.8 (1D tube), Fig. 4.10 (3D tube), and Fig. 4.14 (FSI3 and 3D tube with multiple pulses), we observe that the restart period can be set to a small number $m \in \{4, 8\}$ without loss of convergence speed or robustness.

The truncation threshold ε controls the amount of retained information across restart boundaries, i.e., the restart quality, but also critically affects the runtime complexity of the method. The latter grows cubically with $\bar{\kappa}$. All experiments (Fig. 4.8, Fig. 4.10, Fig. 4.11, Fig. 4.12, and Fig. 4.14) exhibit a clearly identifiable jump in the number of coupling iterations when going from $\varepsilon = 10^{-2}$ to $\varepsilon = 10^{-1}$ or from $\varepsilon = 10^{-1}$ to $\varepsilon = 0.5$, respectively. We, thus, conclude that, for common FSI cases, most of the information lies within a radius of one or two orders of magnitude around the largest singular value. The truncation threshold can, therefore, be easily determined.

Conclusion. We conclude that (*i*) the RS-SVD method is powerful enough to allow for small restart periods m, which is directly reflected in a lower storage and runtime complexity of the method; (*ii*) the "correct" truncation threshold can be identified easily and does not involve a costly tuning process. This shows the "parameter-free" nature of the method.

Summarizing, we state that the RS-SVD method in fact has a complexity that is linear in the problem size N with a small constant factor and is expected to be well-suited and efficient for coupled FSI simulation without any tuning of parameters.

Sensitivity to Filtering Methods: Comparison of filtered LS(∞) and MV RS-SVD.

Purpose. With the goal to provide an efficient, fast and robust quasi-Newton acceleration method with minimal requirements for tuning of additional parameters, we developed two different approaches, (*i*) the filtered LS(∞) method with basically unbounded number of reused time steps, and (*ii*) the MV method with RS-SVD restart. In the remainder of this section, we compare both approaches with respect to convergence properties and robustness. We further compare different numbers of reused steps for the LS method, and, in particular, study the ability of two filter techniques – QR1 and QR2 – presented in §2.2.1 to stabilize the method.

Setup. We employ the 3D tube scenario (with one and with multiple pressure pulses) (see §4.1.2) and the FSI3 benchmark scenario (see §4.1.3). The numerical setting and solutions are identical to those in the previous experiment.

Results. We study and compare the convergence properties of the LS(∞) method and the MV restart variant RS-SVD for the 3D tube scenario with a single pressure pulse in Tab. 4.3. Thereby, we analyze the performance of two different filtering methods for varying filter thresholds ε_F. Results for the MV and the LS(0) method are given as a reference.

Observations. Most importantly, we observe that both presented methods, the LS(∞) and the MV RS-SVD method yield comparable and excellent convergence properties, with a slight advantage of MV RS-SVD over the optimally filtered LS method. Second, there is hardly any loss in convergence speed when switching from the expensive, quadratic complexity MV method to the efficient SVD restart counterpart.

With respect to the filtering method, we see that, for all configurations in Tab. 4.3, the combination with the QR1 filter shows poor performance. The method fails to improve the condition of the least-squares problem, and, thus, to accelerate the convergence speed of LS. On the contrary, the QR2 filter shows excellent stabilization properties and results in fairly good convergence rates, if combined with the LS method – in particular for the challenging case of extensive reuse of information in LS(∞). The approach successfully selects beneficial information. This supports our findings summarized in §4.2.1 and [Hae15]. The QR2 filter strategy, however, also results in higher runtime complexity than QR1, as it triggers a re-computation (rather than column-updating) of the QR-factorization in every iteration.

The MV method and its SVD restart counterpart perform best if no filter is applied (italic numbers in Tab. 4.3 indicate that no columns have been deleted). This is expected as MV only holds

TABLE 4.3 *3D flexible tube scenario and FSI3 cylinder flap benchmark scenario. Averaged number of iterations for the filtered LS(ξ), the plain-vanilla MV method and its efficient SVD-restart counterpart are reported. We investigate the sensitivity of all methods to a powerful filtering technique, and compare the two filtering variants QR1 and QR2, presented in §2.2.1. For the LS(ξ) method, we consider different amounts of reused steps, given in terms of retained time steps ξ. Italic numbers indicate that no columns have been deleted by the filtering. Slight differences in the iteration numbers for these cases between QR1 and QR2 are due to the different QR-decomposition algorithms used inducing different rounding errors. For the RS-SVD method, we employ a restart period of $m = 8$ and two truncation thresholds $\varepsilon \in \{1\text{E}{-2}, 1\text{E}{-1}\}$. Bold numbers indicate the best results for each method type.*

		QR1-Filter					*QR2-Filter*			
	Method	10^{-8}	10^{-7}	10^{-6}	10^{-5}	10^{-4}	10^{-4}	10^{-3}	10^{-2}	10^{-1}
3D Tube (single pulse)	LS(0)	*21.53*	*21.53*	*21.53*	21.52	21.99	*21.53*	*21.53*	**21.37**	24.41
	LS(8)	10.26	10.52	10.52	11.94	13.95	10.13	9.74	**9.21**	10.35
	LS(16)	9.53	9.82	9.82	11.65	15.00	9.25	8.27	**7.59**	8.86
	LS(∞)	15.04	15.97	15.97	33.21	21.42	12.16	7.55	**6.18**	6.33
	MV	***6.00***	*6.00*	*6.00*	6.32	6.76	***5.99***	*5.99*	*5.99*	6.95
	RS-SVD(1E−2)	***6.07***	*6.07*	*6.07*	6.36	6.77	***6.03***	*6.03*	*6.03*	7.06
	RS-SVD(1E−1)	*6.28*	*6.28*	*6.28*	6.42	7.05	*6.20*	*6.20*	*6.20*	7.23
3D Tube (mult. pulses)	LS(0)	*21.10*	*21.10*	*21.10*	21.22	21.46	*21.10*	*21.10*	21.14	23.54
	LS(∞)	14.05	14.39	19.45	41.61	37.07	9.78	7.11	5.74	**5.45**
	MV	***4.89***	*4.89*	*4.89*	5.03	5.23	***4.91***	*4.91*	*4.91*	5.48
	RS-SVD(1E−2)	***4.93***	*4.93*	*4.93*	5.08	5.30	***4.93***	*4.93*	*4.93*	5.55
	RS-SVD(1E−1)	***5.11***	*5.11*	*5.11*	5.29	5.57	***5.17***	*5.17*	*5.17*	5.69
FSI3 Benchmark	LS(0)	*10.65*	*10.65*	*10.65*	10.69	10.62	*10.65*	*10.65*	10.69	11.71
	LS(∞)	5.15	5.12	5.18	9.13	33.79	5.13	4.67	4.35	**4.13**
	MV	***3.27***	*3.27*	*3.27*	3.41	3.33	***3.28***	*3.28*	3.43	3.72
	RS-SVD(1E−2)	***3.57***	*3.57*	*3.57*	3.87	3.88	***3.69***	*3.69*	*3.69*	4.53
	RS-SVD(1E−1)	***4.64***	*4.64*	*4.64*	4.79	4.81	***4.49***	*4.49*	*4.49*	6.07

information from the current time step in the multi-secant equation, and, thus, the risk of linearly dependent information is reduced. This implies that neither the computation complexity of the MV method (or its restart alternative) is increased by additional filter operations, nor tuning of the filter threshold ε_F is necessary.

Conclusion. We conclude that the highly efficient, linear-complexity MV RS-SVD method achieves the good MV convergence and outperforms all filtered LS(ξ) variants. In particular, it presents good stability without necessity for additional filtering. This eliminates all tuning parameters for RS-SVD. The optimally filtered LS(∞) method also yields very good convergence, but introduces another tuning parameter in finding the correct filter threshold ε_F. The QR2 filter (for our scenarios) is a powerful stabilization technique, while QR1 fails. The former, however, triggers a re-computation of the QR-factorization in every iteration.

4.3 Parallel Scalability and Runtime Efficiency

Purpose. With the upcoming exa-scale era and the access to very large parallel machines, simulations that capture ever more physical effects and employ ever more complex models and feature fine resolutions become feasible. This demands for highly scalable algorithms. We consider surface coupled problems, which means that the coupling numerics act on a lower dimensional subspace. Thus, we can assume that the main computational effort is associated to the single-physics solvers, which need to be scalable to hundreds of thousands of cores. For the coupling numerics, we require to at least not deteriorate the parallel scalability of the single-physics solvers, i.e., for the herein developed algorithms, we desire good parallel scalability up to the number of processors that share the coupling interface.

Consequently, in this section, we benchmark the runtime and parallel efficiency of the two advanced quasi-Newton alternatives, LS(∞) and the MV RS-SVD, derived above. We limit our analysis to the two mentioned methods; detailed scalability results for the original MV method and all other involved algorithmic building blocks (such as, e.g., the QR-factorization and dense matrix-multiplications) can be found in [Bun16b; Uek16]. As ever higher resolutions are critical to resolve physical effects accurately, we are mainly interested in the parallel ***weak-scaling efficiency***.

Setup. We benchmark the parallel weak-scaling efficiency of the LS(∞) and MV RS-SVD method for the 3D flexible tube scenario (see §4.1.2). Due to the high computational cost of the 3D flow solver, the computational grid cannot be refined arbitrarily, as the expected runtime of the overall simulation problem becomes infeasible. In order to evaluate the performance and scalability of the interface numerics beyond this point in an isolated way, we therefore consider an additional artificial test case. We describe both settings below.

Experiment Description. For the LS method, we use information from $\xi = 30$ previous time steps[6]. For the MV RS-SVD method, we use a restart period of $m = 8$ and discard all singular values that don't lie within a radius of two orders of magnitude around the most dominant singular value (i.e., we chose $\varepsilon = 1\text{E}{-2}$). For a fair comparison, both methods are equipped with a QR2 filter method, using a threshold of $\varepsilon_F = 1\text{E}{-2}$. While the filter is absolutely critical for LS, it does not have an effect on the MV restart method[7]. We use a relative convergence criterion of $\varepsilon_r = 1\text{E}{-4}$ in (3.10) for the termination of the implicit coupling per time step.

We report the overall parallel runtime of both methods, and also measure the computational time spent in major algorithmic building blocks. The latter are subdivided into work per iteration and work per converged time step and outlined in Tab. 4.4. The table also indicates the hierarchy and relation of the measured components and gives references to the corresponding parallel runtime complexities, as summarized in Tab. 2.1. On every level, we report parallel runtimes averaged over five identical, independently executed experiments. Their standard deviation is indicated with error bars in the figures.

We scale up to $p = 128$ and $p = 2048$ processors at the interface for the 3D tube scenario and the artificial ASTE configuration, respectively. We'd like to remind the reader, that for the here considered experiments, the number of processors only refers to the number of processors involved at the coupling interface. The number of processors involved for the complete multi-physics simulation is much higher.

Test Cases. For our weak-scaling study, we consider the following test case configurations:

(a) ***The 3D flexible tube scenario (3DFT).*** We consider six levels of successively refined grids, doubling the number of unknowns N at the interface as well as the number of processors p at the interface in between each level. The initial workload on the lowest level L0 corresponds to $N = 9,600$ unknowns at the interface, distributed on $p = 4$ processors. This corresponds to $60,000$ inner unknowns for the 3D flow solver. The last level features $307,200$ unknowns at the

[6] Note, that, in order to eliminate the dependency on ξ and fully rely on the filter in combination with LS(∞), a reuse of $\xi = 30$ is still small. Thus, runtime results given here have to be considered as a lower bound for LS(∞) runtime.

[7] This is true in therms of convergence, but in terms of runtime, the filter increases the cost. Thus, runtime results given here have to be considered as an upper bound for MV runtime.

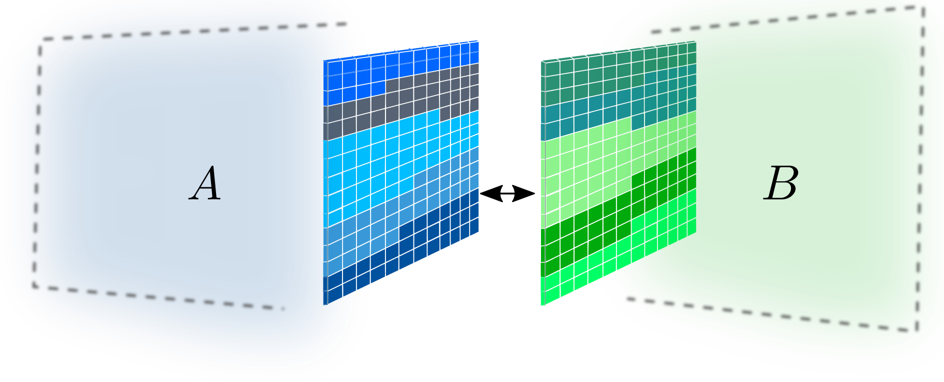


FIGURE 4.15 *Schematic drawing of the artificial solver testing environment (ASTE) and the row-wise ordering of the unknowns. Exemplarily, we show a decomposition into five sub-domains.*

coupling interface, distributed on $p = 128$ processors. For the fluid solver, this translates to a number of $10,137,600$ inner unknowns.

(b) ***The Artificial Solver Testing Environment (ASTE)***[8]. For the above considered FSI scenario, the flow solver becomes a major bottleneck when scaling beyond $N = 307,200$ unknowns at the interface (and $10,137,600$ inner flow solver unknowns). To evaluate the parallel performance of the coupling numerics beyond this point, we employ artificial solvers that generate random data at the coupling interface at a minimum computational load. Thereby, we isolate the coupling functionality from single solver effects. Analogously to real parallel single physics solvers, the coupling surface is decomposed, and unknowns reside on different processors. We consider a row-wise ordering of the unknowns on the 2D coupling surface. We disregard the volume-unknowns of the sub-domains. The grids for both "solvers" are identical. A schematic illustration of the artificial solver setup, coupled via preCICE, is shown in Fig. 4.15 with an exemplary data distribution on five processors.

To provide a realistic test environment for the benchmarked quasi-Newton solvers, we adopt the parameter settings and characteristics of the 3D tube scenario. That is, we use the artificial solvers to help emulating the behavior of the quasi-Newton methods for the 3DFT scenario. On average, the simulations of 3DFT result in $\eta = 230$ columns in the multi-secant equation for the employed LS(30) method, and in a rank $\overline{\kappa}$ of around $\overline{\kappa} = 210$ for the truncated SVD representation of $\boldsymbol{M}^{-1}$ (MV RS-SVD method). Further, we observe convergence after approximately eight (LS(30)), and seven (MV RS-SVD) coupling iterations, respectively, per time step. As a consequence, we fix the maximum size of the least-squares system, the rank of the SVD representation and the number of performed iterations per time step to these values. Apart from this, the numerical preCICE settings remain unchanged compared to the flexible tube scenario.

Using this solver setup, we perform a weak-scaling series over ten levels, ranging from $N = 8,192$ unknowns at the interface (distributed on $p = 4$ processors) to $N = 4,194,304$ unknowns at the interface (distributed on $p = 2,048$ processors).

Hardware. All experiments considered in this section were conducted on the Haswell nodes partition (phase II) of the Tier-1 `SuperMUC` machine at the Leibniz Supercomputing Center[9] in

[8] https://github.com/precice/aste
[9] https://www.lrz.de/

TABLE 4.4 *Relation between the algorithmic building blocks studied in our weak-scaling experiments (see Fig. 4.16 and Fig. 4.17). We give references to the parallel runtime complexities, derived in §2.2.2 and summarized in Tab. 2.1 for every algorithmic block.*

work/iteration		**work/time step**	
• quasi-Newton iteration		• converged time step	
– *apply filter*	(***QR***)	– *update* $\widetilde{\mathbf{W}}$	(***upWtil***)
– *QR solve*	(***backS***)/(***iV***)	– *update SVD* (after period m)	(***updSVD***)
– *compute qN update*	(***Mr***)/(***lmDx***)		

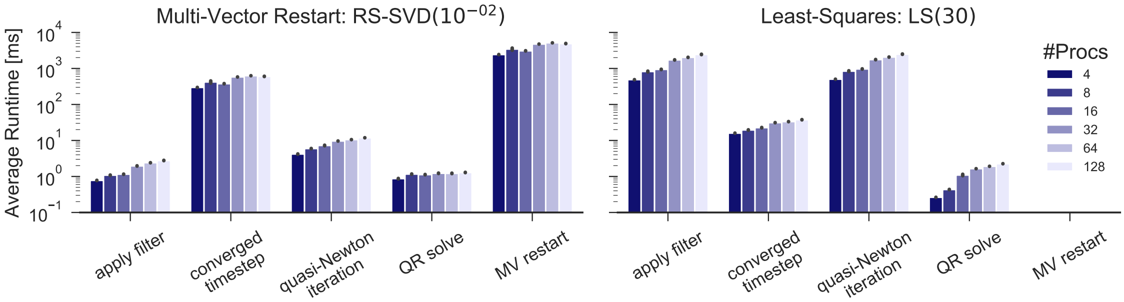


FIGURE 4.16 *Parallel weak scalability study for the 3D flexible tube scenario. We report parallel runtimes for the main algorithmic blocks of the LS(30) and the MV RS-SVD method. Both methods are stabilized with the QR2 filter. We consider six levels, and successively double the number of unknowns N at the interface (problem size) and the number of processors p between levels. The first level corresponds to $N = 9,600$ unknowns at the interface, distributed on $p = 4$ processors. Image modified from [Sch17].*

Garching. The system offers a peak performance of 3.58 Petaflops comprising 3,072 nodes with Haswell Xeon E5-2697 v3 processors, and 28 cores per node. The nodes are interconnected via an Infiniband FDR14 network.

Results. We report parallel runtimes for the two described weak scaling experiment series for 3DFT and ASTE. In Fig. 4.16 and in Fig. 4.17, we furthermore compare the time-to-solution for the parallel execution of the plain-vanilla MV method, the MV RS-SVD and the LS(30) method on the coarsest grid for the 3DFT scenario in Tab. 4.5.

Observations. The most important observation is that, for both experiments (see Tab. 4.16 and Tab. 4.17), the total execution time and, in particular, the parallel weak scaling efficiency for the MV RS-SVD method is considerably better than for the LS(30) alternative. The latter predominantly suffers from expensive QR re-computations, triggered by the QR2 filter technique.

From Tab. 4.16 and Tab. 4.17, it can be seen that about 95% of the computational time is spent in the `apply filter` block, which triggers a re-computation of the QR-decomposition. Due to the large number η of reused vectors, this not only dominates the cost per quasi-Newton iteration, but the implementation also shows a rather poor parallel scalability. The latter can be explained by the $\mathcal{O}(\eta^2 \log p)$ factor in the complexity for the re-computation of the QR-decomposition. If, on the other hand, a small number η of vectors is used, the cost for `apply filter` becomes negligible (compare respective runtimes for the MV RS-SVD method). However, we in general observe a non-optimal parallel scalability for all building blocks that include insertion of columns into the QR-decomposition or re-computation of the latter.

For the MV RS-SVD method, the most expensive part is the restart numerics itself. Updating the SVD representation of the inverse Jacobian with the information collected within the past MV-estimation era comprises several QR-decompositions and eventually the ***sequential*** computation of a

TABLE 4.5 *3D flexible tube scenario. Comparison of parallel runtimes for the plain-vanilla MV method, MV RS-SVD and LS*(30) *on the coarsest grid with* $N = 9,600$ *unknowns (at the interface) distributed on* $p = 4$ *processors.*

method/runtime	per iteration [ms]	per time step [ms]
MV	330	37
MV RS-SVD	3	291
LS(30) w. QR2	502	15

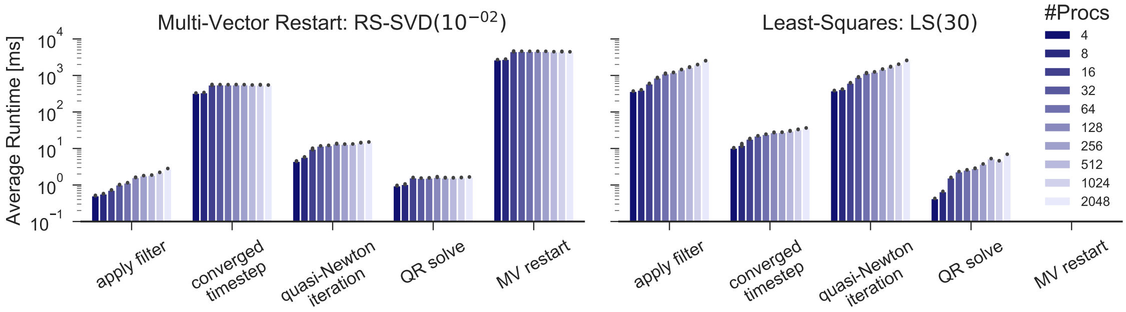


FIGURE 4.17 *Parallel weak scalability study for the artificial ASTE configuration. We emulate the behavior of the quasi-Newton methods as observed for the 3D tube scenario by adopting the parameter settings and characteristic from this previously studied example. We report parallel runtimes for the main algorithmic blocks of the LS*(30) *and the MV RS-SVD method. Both methods are stabilized with the QR2 filter. We consider ten levels of refinement from* $N = 2^{13}$ *unknowns on* $p = 4$ *processors up to* $N = 2^{22}$ *unknowns distributed on* $p = 2^{11}$ *processors. Image modified from [Sch17].*

small singular-value decomposition. We want to emphasize that restart is only performed after every $m = 8$ time steps; the computational cost for the restart numerics is entirely included in the (much lower) per-time-step cost depicted under `converged time step`. In particular, the MV RS-SVD cost ***per time step*** is even lower than the LS(30) cost ***per iteration*** (compare `converged time step` and `apply filter` for MV RS-SVD and LS(30), respectively in Tab. 4.16 and Tab. 4.17). For the study in Tab. 4.17, this translates to a per-iteration execution time of $\approx$ 2.60 s for LS(30) compared to a per-iteration runtime of $\approx$ 0.02 s and per (converged) time step runtime of $\approx$ 0.56 s for MV RS-SVD on the highest level. Lastly, the SVD update also shows very good parallel (weak scaling) efficiency. This especially holds for the artificial ASTE configuration, where, due to a fixed rank of the SVD, we observe perfect parallel scalability of the RS-SVD restart functionality. We furthermore note a perfect agreement of the measurements for levels 1–6 between the experiments for the 3DFT scenario and the artificial ASTE configuration, which justifies the choice of fixed parameters.

We close the performance considerations with a comparison of parallel runtimes for LS(30), MV RS-SVD, and the parallel runtime of the plain-vanilla MV method in Tab. 4.5 (results are obtained for the 3DFT scenario using $N = 9,600$ unknowns at the interface). Considering an average of seven iterations per time step for MV RS-SVD, the parallel runtime per iteration reduces by 87% to $\approx$ 44.50 ms for MV RS-SVD compared to MV. Moreover, we could produce any factor between the runtimes for the plain-vanilla MV method and its restart counterpart if we increase the number of unknowns N at the interface. The former suffers from a quadratic complexity with respect to N, while the restart counterpart exhibits linear runtime complexity in N. The cost per time step for the MV RS-SVD method does not grow remarkably when increasing the number of unknowns N at the interface, as the rank of the SVD is small and independent of N (cf. §4.2). For more detailed strong and weak scaling experiments for the LS and the MV method, we refer to [Bun16b], Fig. 16.

Conclusion. From the parallel weak scalability experiments, we conclude that the MV RS-SVD method (*i*) outperforms the LS(30) method in terms of total execution time, and (*ii*) exhibits excellent parallel (weak scaling) efficiency for the (runtime dominating) restart numerics. The LS(30) method suffers from very expensive re-computations of the QR-factorization (due to a large number of retained columns η in the least-squares system), triggered by the QR2 filtering method. In particular, this part does not scale well when increasing the problem size and the degree of parallelism. Thus, the MV RS-SVD method developed within this work clearly outperforms other state-of-the-art methods.

Part II

Coupling of Bio-Physical Brain-Tumor Models with Medical Image Registration

5 Scalable Biophysics-based Image Analysis

The second part of my thesis targets the coupling of a very different type of multi-component system, the volume-coupling between the inversion of biophysical brain tumor progression (data assimilation/model calibration) and medical image registration. Integrating biophysical modeling with medical imaging may improve diagnosis and clinical decision making, aid prognosis, and foster the design of new treatment protocols. Applications for medical image analysis algorithms include image segmentation, registration and calibration of biophysical models. These tasks are inverse problems, formulated as PDE-constrained non-linear optimization problems. Our goal is to design inversion methods that incorporate complementary data by tight integration of physics and imaging, and rigorously follow mathematical and physical principles, in an attempt to aid clinical decision making. Clinical application demands for robust and reliable, high accuracy algorithms with a short time-to-solution. The design of such algorithms is complicated by the ill-posed nature of inverse problems and the typically large dimensionality of the resulting optimality systems. We propose, evaluate and benchmark schemes and solvers that target the solution of this large-scale coupled inverse problem. Thereby, some of the findings from coupling fluid-structure interaction simulations and, in particular, insights about performance of different quasi-Newton methods from Chapter 2 are revisited and applied for this new type of coupled problem. The need for scalability is due to the increasing scanner resolution for medical imaging and the critical time-to-solution aspect for clinical use.

In this chapter, we motivate and present our target application. Both biophysically-augmented image analysis and image-driven biophysical model development are essentially large-scale data-assimilation inverse problems that involve non-linear partial differential equations (PDEs). The associated optimality systems are sets of non-linear multi-component PDEs, for which the development of efficient solvers is challenging. In §5.1, we give a brief introduction into the challenge of solving large-scale, non-linear inverse problems, followed by the more specific presentation of the coupled multi-component problem for our target application in §5.2. We discuss significance and motivation for the problem in §5.2.1, review related literature and the state-of-the-art in §5.2.2 and summarize the main contributions within the present work in §5.2.3. §5.3 briefly introduces the involved sub-components—inverse brain tumor progression simulation and medical diffeomorphic image registration. Benchmark results, demonstrating the scalability and parallel efficiency of our algorithms and solvers on massively parallel systems on distributed data conclude this chapter. In Chapter 6 we present the mathematical formulation of the coupled PDE-constrained optimization problem and present two different Picard-type coupling schemes: The so called ***moving-patient*** coupling scheme in §6.1 and the ***moving-atlas*** coupling scheme in §6.2. §6.3 deals with details on the numerical optimization and algorithmic methodology including parameter and grid-continuation schemes. Improved inexact Newton-type solvers to speedup the coupled non-linear optimization are discussed in §6.4. A thoroughly qualitative and quantitative analysis of the presented coupling schemes and algorithmic improvements based on numerical experiments for synthetic cases and actual clinical patient MR-data is presented in Chapter 7. It includes a comparison of the proposed schemes with respect to accuracy, clinical relevance, and time-to-solution, as well as careful analysis of the algorithmic improvements.

5.1 Inverse Problems: PDE-Constrained Optimization

5.1.1 Introduction

So far, we considered (time-dependent) simulation processes in part I, modeled by partial differential equations. Given some knowledge about the PDE parameters, e.g., initial conditions, boundary conditions, domain geometry, domain sources and material properties, the 'image' of the system in terms of defined state variables such as displacement, density, velocity, temperature field, or species concentrations is computed at a certain time point. This can be viewed as the forward evaluation of an abstract function, describing the targeted system. As opposed to these so called ***forward problems***, ***inverse problems*** describe the reversed task, i.e., parameter estimation of systems given an observation of the system's state (output of the forward problem). Typically, the considered systems are governed by PDEs. Examples can be found in various application fields, ranging from weather forecasting and global climate-change simulations via ocean and earthquake modeling, reservoir simulation, turbulence modeling, electromagnetic sensing, hazardous substance attacks and medical imaging to astrophysics.

The herein considered inverse problems can be formulated as non-linear optimization problems, governed by partial differential equations.

$$\min_{\boldsymbol{q}} \mathcal{J}[\boldsymbol{d},\boldsymbol{u},\boldsymbol{q}] = \mathcal{D}[\boldsymbol{d},\boldsymbol{u}] + \beta\mathcal{S}[\boldsymbol{q}] \qquad \text{subject to} \quad \mathcal{C}[\boldsymbol{u},\boldsymbol{q}] = 0, \tag{5.1}$$

We are given some partial observations $\boldsymbol{d}$ (target data) of the system, which we assume to be solutions of a system of PDEs $\mathcal{C}[\boldsymbol{u},\boldsymbol{q}] = 0$ for a suitable parameter vector $\boldsymbol{q}$. $\mathcal{C} \in \bar{\Omega} \times [0,T]$ is referred to as the forward model, describing the underlying system or process; solving the forward model results in the system's predicted state $\boldsymbol{u}$. Based on observations, we seek to estimate a subset of the underlying system parameter(-functions) $\boldsymbol{q}$ such that the discrepancy $\mathcal{D}$ between the (partially) observed states $\boldsymbol{d}$ and the solution $\boldsymbol{u}$ predicted by the model is as small as possible. This optimization goal is modeled in the objective function $\mathcal{J}$. The data-discrepancy measure can be defined as a simple L_2 distance $\mathcal{D} := 1/2\|\mathcal{O}\boldsymbol{u} - \boldsymbol{d}\|^2_{L^2(\Omega)}$. Here, the observation operator $\mathcal{O}$ projects the state variables $\boldsymbol{u}$ into the space of observed data $\boldsymbol{d}$. Computing the solution of large-scale inverse problems represents one of the most outstanding challenges in computational science and engineering. While the simulation process (forward problem) is usually well-posed,[1] the inverse problem can be ill-posed, meaning that typically the solution is not unique and, in addition, appears to be highly sensitive to changes in the data, as noise and high frequencies are amplified; the problem is, thus, highly ill-conditioned. It needs to be reformulated applying a suitable regularization scheme [Eng96], including prior knowledge and additional assumptions, such as smoothness of the solution. Within this work, we

[1] Well-posed in the sense of Hadamrd [Had02]: (*i*) A solution to the problem exists, (*ii*) the solution is unique, and (*iii*) the solution's behavior changes continuously with the initial conditions or input parameters.

consider a Thikonov-type regularization functional $\mathcal{S}$. The regularization parameter β balances data fidelity with regularity. For theory and analysis of PDE-constrained optimization problems compare [Ban89; Del11; Gun03; Lio72; Pir12].

To solve problem (5.1), we make use of a variational approach, translating the constrained problem into an unconstrained one by defining the Lagrangian function

$$\mathcal{L}\left[\boldsymbol{d},\boldsymbol{u},\boldsymbol{q},\boldsymbol{\lambda}\right] = \mathcal{J}\left[\boldsymbol{d},\boldsymbol{u},\boldsymbol{q}\right] + \boldsymbol{\lambda}^T\mathcal{C}\left[\boldsymbol{u},\boldsymbol{q}\right] \tag{5.2}$$

which enforces the constraints $\mathcal{C}$ via an inner product with the Lagrange multiplier $\boldsymbol{\lambda}$, or adjoint variable. Solving the optimization problem requires stationarity of the Lagrangian $\delta\mathcal{L} = 0$ with respect to $\boldsymbol{u}$, $\boldsymbol{q}$ and $\boldsymbol{\lambda}$. This results in a large, non-linear system which can be addressed with gradient-descent or Newton-type methods. The utilized solvers need to be robust, efficient, highly accurate, and tailored to the structure of the operators.

5.1.2 Computational Challenges and Solution Approaches for Large-Scale Inverse Problems

We consider large-scale inverse problems, formulated as (non-linear) PDE-constrained optimization problems. Due to their characteristics, these inverse problems are typically incomparably harder to solve than the forward simulation problem for the following reasons: (*i*) Inverse problems are usually ill-posed and feature dense ill-conditioned operators, which leads to non-unique solutions and noise amplification and, thus require regularization; (*ii*) solving inverse problems is computationally extremely expensive as multiple evaluations of the forward simulation model are required. Their complexity and high dimensional search space pose significant challenges for general-purpose optimization algorithms; (*iii*) while forward operators are usually evolution operators, the inverse operator introduces a space-time coupling, i.e., coupling the entire time history to the system's response; (*iv*) the optimization problem is often non-linear and non-convex, featuring multiple local minima and urges multi-scale grid- and parameter-continuation schemes to convexify the objective and prevent the solution process from getting stuck in local minima.

For their numerical solution, iterative solvers with efficient preconditioners and globalization schemes are inevitable. To foster the efficiency of the solution process, inexact solves are important to prevent over-solving, and the large-scale urges for parallel approaches; while there is a broad variety of highly efficient and parallel large-scale PDE solvers for the solution of the forward problem, design and implementation of numerical methods for the solution of the inverse problem is not straightforward. Developing solvers that feature good scalability to both large numbers of processors (parallel scalability[2]) and large problem sizes (algorithmic scalability[3]) is very challenging. The resulting algorithms need to be highly tailored towards the structure and characteristics of the underlying problem.

While Newton's method is a classical choice for the treatment of non-linear systems, it unveils significant difficulties when trying to solve optimality systems arising from large-scale non-linear optimization problems governed by PDEs. Newton's method requires exact Jacobian information (Hessian information of the Lagrangian in the context of optimization, as considered here) to solve for the Newton step. Due to the large dimensionality of the non-linear systems, not only assembling

[2] Parallel scalability refers to the speedup for the execution time when using p processors compared to single core execution. For ideal scalability, the speedup would be p.

[3] Algorithmic scalability refers to scalability of the used methods when increasingly finer discretizations are used. Often time, the condition number of systems depends on the discretization, causing e.g., iterative methods such as Newton or Krylov solvers to require considerably more iterations.

and storing the Hessian becomes infeasible, but also solving the resulting Newton system becomes intractable for direct or matrix-based solvers: Building the Hessian is prohibitively expensive as it involves numerous solves of the forward and adjoint model, its dense structure rules out sparse matrix formats, thus, it cannot be kept in memory, and the system is of the order of the number of inversion variables, which often depends on the mesh size of the PDE constraint. Thus, a popular remedy to overcome the extremely expensive or infeasible construction of the Hessian is to employ twofold inexactness by combining a Gauß-Newton approximation of the Hessian with a matrix-free Krylov-subspace method for the solution of the resulting inexact Newton system. Designing powerful preconditioners for these methods becomes increasingly important when more sophisticated regularizers are used to ensure well-posedness of the inverse problem, yet also more challenging. Typical examples include multi-level or quasi-Newton preconditioners. Lastly, a careful globalization scheme has to be used in order to prevent the method from getting stuck in local minima, or to prevent the solver from diverging due to the non-convex structure.

Due to the computational expense of Hessian based solvers, a large number of large-scale PDE-constrained optimization problems rely on gradient information only. The simplest of which is gradient-descent, yet it usually shows poor convergence. A more favorable gradient based method is non-linear conjugate gradients (NLCG) and even more advanced are quasi-Newton methods such as L-BFGS, which build a limited amount of curvature based on changes in the observed gradient values. Gradient-based methods build mostly upon PDE solves for the gradient evaluation and thus, yield a high parallel scalability due to the availability of highly efficient solvers for the forward and adjoint problem.

For recent algorithmic trends in large scale applications, we refer to [Bie03]. An excellent overview on parallel algorithms for PDE constrained optimization can be found in [Akc06; Bir99], and in particular Newton-Krylov schemes in [Bir05a; Bir05b; Kno04]. [Kel87a] investigate quasi-Newton for large-scale unconstrained optimization.

We will see, that the formulation and numerical solution of the herein considered application problem results in a large-scale, ill-conditioned, strongly coupled system of non-linear PDEs, featuring all of the above mentioned challenges. In an attempt to solve this formidable system, we go a step further and decompose it into smaller sub-components, whose characteristics are better understood. We employ a matrix-free, inexact, preconditioned Gauß-Newton-Krylov approach for the solution of each sub-system, and establish the coupling between the sub-components by means of two different Picard iterations, resembling a block-Newton-type solution of the overall coupled formulation. The coupling itself is subtle and requires tight integration of the sub-components; it asks for careful tuning of solver tolerances and continuation schemes to allow for a robust and computationally feasible solution. We additionally evaluate advanced quasi-Newton methods for one of the sub-component solvers.

5.2 A Coupled Multi-Component Problem: Biophysics-Based Tumor Models Coupled with Medical Image Registration

As mentioned in the introduction, this second part of the thesis targets a volume coupled multi-component problem, motivated by applications in brain tumor analysis and biophysics. The scope is to develop a framework targeting the analysis and patient specific simulation of the progression of Glioblastoma multiforme (GBM), one of the most common and most malignant forms of primary brain tumors. We approach this highly involved and sophisticated physiological phenomenon using a

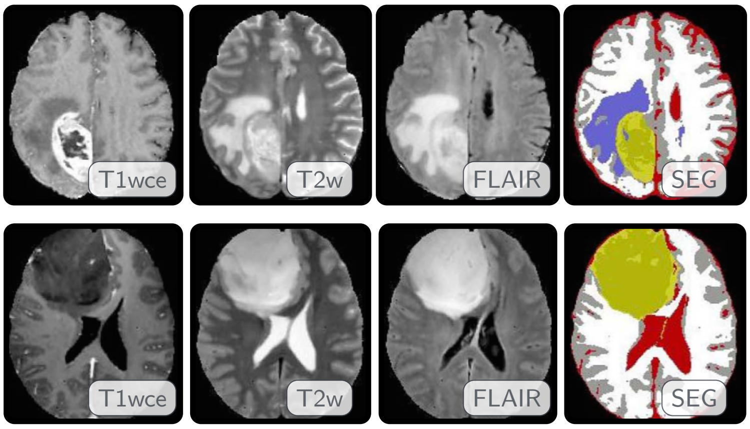


FIGURE 5.1 *The image shows different MRI modalities (left to right: T1-weighted (contrast-enhanced), T2-weighted, FLAIR and a segmentation into tissue labels) of two different patients diagnosed with Glioblastoma multiforme grade IV. The images illustrate the highly infiltrative and invasive nature of such tumors which leads to a poorly defined brain-tumor interface and a strong mass effect, i.e., deformation of the surrounding brain tissue as a result of tumor growth. Image modified from [Goo13].*

joint medical image registration and biophysical inversion strategy, which tightly integrates methods for non-rigid, large deformation image registration with biophysical models for (inverse) brain tumor growth simulation. The herein developed framework SIBIA (Scalable Integrated Biophysics-based Image Analysis;[Gho17c; Sch]) provides solvers and algorithms to address (*i*) biophysics aided medical image registration, by adding meaningful prior information about the underlying biophysical process, (*ii*) parameter estimation and biophysical model calibration, fostering prediction of tumor progression, and (*iii*) automated atlas-based segmentation; all of which targeting at facilitating diagnosis, aiding and supporting surgical planning, and, improving the efficacy of brain tumor therapy.

5.2.1 Motivation and Background

Pathology and Clinical Significance. Tumor cells emerge from healthy cells by occasionally occurring cell mutation, leading to erratic cell proliferation and growth. Tumor cells replace healthy tissue cells, which leads to an impairment to health due to organ malfunction and, as the pathology progresses, eventually leads to death. Tumors are classified according to their grade of malignancy: one distinguishes between benign and malign tumors of different grades. While benign tumors usually show slow growth rates and feature a clearly visible boundary between cancerous cells and healthy tissue, and, thus facilitate therapy through effective resection, malignant tumors are characterized by their rapid growth rate and massive infiltration into healthy tissue far beyond the visible tumor bulk. This highly infiltrative and diffusive character leads to an indefinite and obscure boundary between cancerous and healthy tissue [Man06], which strongly aggravates both diagnosis and efficient therapy.

Within this work, we consider Gliomas, the most common form of primary brain malignancies, with different degrees of aggressiveness and variable prognosis. They primarily originate from glial cells. Amongst them, we focus on one of the most malignant and most common forms, Glioblastoma multiforme (GBM), a WHO grad IV Glioma [Col98; Mar05; Wei12; Lou07]. Fig. 5.1 shows different magnetic resonance imaging (MRI) data for two different patients, diagnosed with GBM grade IV. GBM tumors are the most aggressive form of primary brain tumors, accounting for

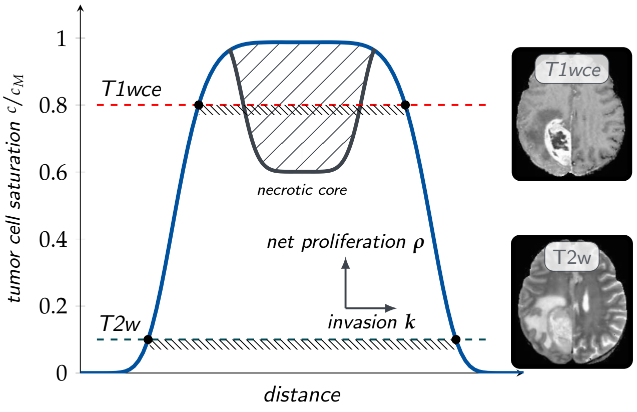


FIGURE 5.2
Schematic view of a typical cross section Glioblastoma cell density distribution as seen by MRI in T1wce (T1-weighted contrast-enhanced) and T2w (T2-weighted) imaging (c_M denotes tissue carrying capacity). The schematic thresholds for T1wce and T2w indicate the lowest cell concentration detectable by the respective modality. Image modified from [Man14; Har07].

about 50% of all Gliomas [Tov93; Hol01], and are characterized by their highly infiltrative nature and diffusive invasion into the surrounding healthy tissue (compare Fig. 5.1). Due to the rapid growth of GBM, tumor cells are exposed to acidosis and nutrient deficiency, which requires them to be extremely adaptable to changing conditions in their micro environment, further hampering efficient therapy. GBMs feature various heterogeneous histological sub-regions, such as a necrotic core (due to nutrient deficiency), an enhancing and non-enhancing core, and peritumoral edema (compare Fig. 5.3). Gliomas are almost impossible to cure, as they generally grow and invade extensively before the patient notes any symptoms. The tumor is CT detectable when it reaches a size comparable to a sphere of a diameter of 3 cm (death occurs at a diameter of approximately 6 cm). By this time peripheral cells already have invaded healthy tissue in remote areas, thus despite maximum treatment involving total resection of all obvious tumor, followed by radiation and chemotherapy, the tumor shows notorious recurrence either near the edge of resection or at more distant locations within the brain [Gas92; Lia98; Pri07]. The tumor's aggressive nature is reflected in its 100% fatality rate within 6-12 months after diagnosis and massive treatment; without therapy, survival time is usually three months.

Imaging. Medical imaging is absolutely crucial for clinical diagnosis and therapy. The most common imaging techniques related to brain tumor imaging are magnetic resonance imaging (MRI) [Bro14] and computer tomography (CT) [Buz08]. Within this work we only consider imaging data originating from MRI scans. Gliomas feature a highly heterogeneous structure, with various different sub-structures that represent different biological tumor properties (cf. Fig. 5.3). To visualize these highly diffuse sub-regions, different (contrast-enhancing) MRI modalities – visualizing different sub-regions – are employed. We give a brief summary of the most important modalities, for details see [Man14]. For *T1-weighted MRI (T1w)* the image contrast is based predominantly on the T1 (longitudinal) relaxation time of tissue; tissue with short T1 relaxation time appears brighter (hyperintense) (adipose appears brightest, followed by white matter, gray matter, and cerebrospinal fluid (CSF) has lowest signal). In *T2-weighted MRI (T2w)* the image contrast is based predominantly on the T2 (transverse) relaxation time of tissue, so tissue with long T2 relaxation time appears hyperintense (CSF is brightest followed by gray matter and white matter; muscle appears darkest). The *fluid-attenuated inversion-recovery MRI (FLAIR)* technique allows to suppress hyperintense signals from fluids, so the bright signal of the CSF (cerebrospinal fluid) is suppressed which allows for a better detection of small hyperintense lesions. These are potentially combined with different contrast-enhancing techniques. The different modalities are illustrated in Fig. 5.1 The *diffusion tensor imaging (DTI)* [Bas94] technique allows to extract diffusion anisotropy in white and gray matter,

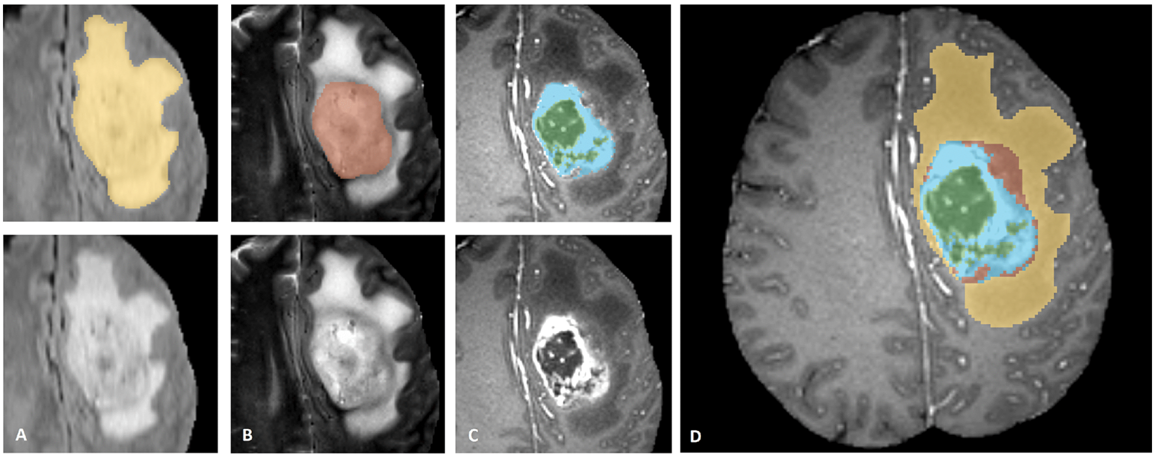


FIGURE 5.3 *Glioma sub-regions. The image shows Glioma tumor sub-structures, annotated in different modalities by human expert raters (top left) and the final segmentation for the entire detest (right). From left to right, the image shows (A) the whole tumor (yellow), as visible in T2-FLAIR MRI, (B) the tumor core (red) observable in T2, and (C) the enhancing tumor regions (light blue, T1Gd MRI) together with the necrotic core (green). The combination of which yields the final labels (right): edema (yellow), non-enhancing core (red), necrotic core (green), enhancing core (blue). Figure and explanation taken from BraTS IEEE TMI paper [Men15]*

which is of great importance for tumor simulation modeling, as tumor cells preferentially migrate along the direction of fiber tracts [Gie03; Gie96; Man06; Pri07; Pri04; Ale07].

Despite the ensemble of different modalities, the true extend of tumor invasion and boundaries of tumor sub-structures are still obscure and not clearly recognizable [Kel87b; Kel87c; Sil97]. Further, it is commonly accepted that there is no sharp borderline between tumorous and healthy tissue [Har07; Kon09; Kon10a; Swa08], which is illustrated in Fig. 5.2. Thus, biophysical tumor progression simulation has the potential to be of great value for clinical diagnosis and treatment.

Goals and Challenges. With recent advances in physiology and medical imaging, the demand for increasingly sophisticated and tightly integrated imaging algorithms and computational biophysical models has increased drastically to facilitate and aid diagnosis, and provide insights for prognosis and therapy, especially surgical-planning. We aim at providing a framework (SIBIA) that tightly couples biophysical models for the simulation of the progression of primary brain tumors with medical imaging analysis. Based on a volumetric segmentation of a magnetic resonance imaging (MRI) data-set of a Glioma patient, SIBIA provides functionality to register this image to a segmented MRI data-set of a normal patient, also referred to as ***atlas***, while simultaneously fitting a biophysical tumor growth model.

Applications and needs for such a joint registration-biophysical inversion approach for analyzing medical MRI data are manifold. We focus on three practical problems, for each of which the tight integration of biophysics with medical imaging within SIBIA is the core ingredient:

(1) ***Calibration of complex macroscopic biophysical PDE models by means of parameter estimation [Swa08; Rah17].*** Biophysical models enable predictive medicine, are beneficial in providing quantitative understanding of human physiology and cancer progression, and can furthermore foster the development and design of new treatments. Examples include models for cardiac electrophysiology and mechanics, flow and transport in micro-circulation, therapy models, atherosclerotic plaque growth, soft tissue and bone mechanics and tumor growth models. Especially when analyzing high grade Gliomas, calibrated models can be used for subsequent prediction of diffusive and invasive tumor margins as it progresses from the initially observed

boundary (CT/MRI imaging based). This information is crucial for surgical and/or radiotherapeutic treatment, as conventional imaging (CT, MRI) falls short in determining the degree of diffusive invasion of tumor cells peripheral to the bulk of the tumor mass. Furthermore biophysical model based simulation has the potential to reveal hidden information not accessible by visible inspection of the data only, and can be helpful in predicting sites of potential recurrence. It therefore has the capability to highly advance therapy without having to wait for reemergence on follow-up imaging.

Our work lays the basis for estimating and calibrating the unknown patient specific parameters in these models. One of the most important parameters is the exact position of the tumor's genesis, the initial tumor seed. The growth rate of GBM highly depends on the invaded tissue: it grows faster in white matter than in gray matter (10-100 times), and it is halted by the dura mater or the ventricles [Gie03; Man06; Pri07; Pri04; Ale07] (for a brief overview of the human brain anatomy, see §5.3.1). Thus, in order to get meaningful and highly diagnostic tumor growth simulation results (or, conversely, parameter estimates), the underlying brain structure (segmentation of tissue regions such as white matter (WM), gray matter (GM), glial matter (GLM), ventricles (VT) with cerebrospinal fluid (CSF) and cerebellum (CB)) of the patient needs to be known.

A stand-alone inverse tumor solver can not solve this problem for the following reason: In practice, information about the ***healthy patient*** brain structure is inaccessible, as imaging data usually only exists at the time the pathology already is present, and the tumor evolution drastically changes and destroys the initially healthy tissue. In addition, we are only given a single snapshot (i.e., we do not have patient-specific longitudinal (transient) data). Thus, in order to solve the time dependent inverse tumor-growth problem, we need to artificially create a second time point by means of a pre-segmented healthy atlas brain. This step requires image registration to establish a relation between patient and atlas brain. The approach is summarized in Fig. 5.4. Assuming we are given a healthy segmented brain image, a tumor growth model, and a deformation map, the process is most intuitively accessible by explaining the ***forward problem***. We grow an artificial tumor in the healthy atlas brain using the tumor growth model to produce a new segmentation, comprising both healthy tissue and tumor. This segmentation is then used to compute the deformation map between patient and atlas which results in a tumor-plus-deformation warped atlas.

Conversely, in the ***inverse problem*** we simultaneously seek to compute the biophysical tumor-growth model parameters and deformation parameters such that the tumor-plus-deformation warped atlas image matches the patient data. For model calibration, emphasis is put on getting meaningful parameters and realistic tumor growth simulations that reconstruct the patient's pathology in the best possible way. This may include the need for a slightly more complicated scheme and imposing additional constraints. This is explained in more detail in §6.2.

(2) ***Normal-to-abnormal registration [Moh06; Kwo14; Zac09].*** Here, we aim at registering normal MR images to MR images with abnormalities, i.e., we seek to compute a deformation map that registers topologically distinct images with the objective to get the best possible (visual) match between the warped atlas image and the patient's MR image showing a pathology (see Fig. 5.4). This problem appears to be a unusual hard (and ill-posed) because of the large deformations and topological differences due to the presence of the tumor. Standard intensity based deformable registration methods fall short solving this problem, showing up significant errors and resulting in implausible deformation maps, particularly in areas near the tumor. Thus, adding biophysical

constraints to the registration problem allows for biophysics induced deformations and aids the registration process [Hog08a; Goo13].

Normal-to-abnormal image registration is used to map structural and functional information from atlases to specific patients (atlas-based segmentation) and first and foremost finds application in (neuro-)surgical planning [Zac08b], but also in longitudinal studies, followups, or population studies. In atlas-based segmentation, first the patient MRI is co-registered to an already segmented normal-brain (or atlas) MRI to solve for a deformation map between the images, which is then used to transfer the labels in the normal brain to the patient [Goo13; Hog08a; Rik10]. Conversely, normal-to-abnormal registration is also applied to pool data from different brain tumor patients (and possibly information from different modalities such as MRI and CT) into a common stereotaxic space to construct statistical brain tumor atlases.

In contrast to the biophysical model calibration, normal-to-abnormal registration emphasizes the quality and match of the final registration result, and obtaining meaningful physical parameters and diagnostic significance of the tumor model is of secondary importance. The basic idea of the simulation and inversion scheme, however, remains unchanged, cf Fig. 5.4.

(3) ***Automated atlas-based image segmentation [Bak15; Goo13; Pra09].*** The imaging phenotype (appearance and shape) of Gliomas highly reflects their intrinsic heterogeneity, as their subregions are described by varying intensity profiles disseminated across multimodal MRI scans as can be seen from Fig. 5.3. This highly heterogeneous appearance and shape renders the segmentation of brain tumor in multimodal MRI scans to be one of the most challenging tasks in medical image analysis. In current clinical routine, MRI scans of Glioma patients are evaluated based on qualitative criteria only (e.g., indicating presence of characteristic hyper-intense tissue in T1-weighted MRI), or by rudimentary quantitative indicators such as the largest tumor diameter observable in axial slices of the pathology. Thus, image processing routines that are capable of *automatically* analyzing brain tumor scans from different modalities, that means, automatically provide highly accurate and reproducible measurements of the tumor substructures, including their segmentation, would be of tremendous potential value for improved diagnosis and treatment planning.
Developing such automated brain tumor segmentation techniques is a very challenging task. There is a broad variety of different approaches, the best of which are summarized in the "multimodal brain-tumor segmentation challenge (BraTS)" [Men15]. The approach relevant for this work combines the above mentioned atlas-based ***normal-to-abnormal*** biophysics-aided image registration with a *supervised machine learning* algorithm [Man17c]. The latter is used to create probability maps (spatial priors) for the target classes (white matter, gray matter, glial matter, ventricles, CSF, whole tumor, edema and enhancing tumor), which are then processed by the joint registration and biophysical inversion scheme (SIBIA) to implicitly impose spatial correlations. These steps can be iterated to further improve the result. The described approach is part of the BraTS'17 challenge and is based on [Goo13] and [Hog08a]. It uses SIBIA [Gho17c; Sch] as core building block and embodies a fully automated atlas-based segmentation technique. Related schemes that use similar approaches are [Goo13; Rik10; Cla05].
Within this work, we do not investigate the above described automated atlas-based segmentation technique any further, but only focus on it's main building block, SIBIA.

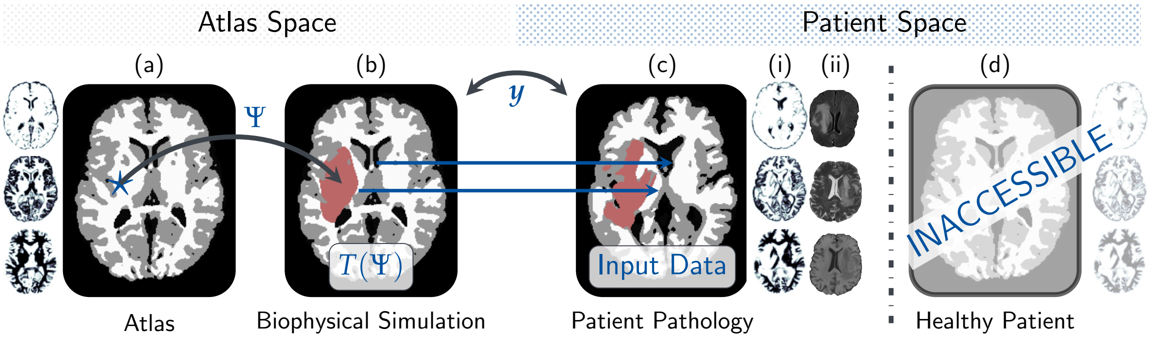


FIGURE 5.4 *The big picture. Illustration of the targeted problem and motivation for the need of joint registration and biophysical inversion approach. We would like to (1) solve for highly diagnostic tumor model parameters Ψ to aid clinical diagnosis and treatment, or (2) register normal-to-abnormal MRI scans. To solve (1), we don't have access to the healthy patient brain structure, thus we deploy an atlas-matching technique. For (2), the topological differences render this problem extremely hard and biophysical constraints aid the registration process to get meaningful deformation maps. Both solution schemes are based on the same joint approach, however, for (1) emphasis is put on getting meaningful physical parameters, while for (2) focus lies on getting best agreement applying the transformation map $\boldsymbol{y}$. The input data is based on the patients' MRI data (c). This could either be (prior) probability maps for distribution of different tissue regions (i), or different MRI modalities (ii). Output data is improved (posterior) probability maps of tissue distribution, tumor growth parameters Ψ and the deformation map $\boldsymbol{y}$. Lacking healthy patient data (d), a (healthy) atlas brain (a) is employed. The objective is to (simultaneously) solve for tumor growth parameters Ψ and registration parameters $\boldsymbol{y}$, such that the simulated tumor and (atlas) brain structures (c) match best with the patient input data.*

5.2.2 Related Work

The formulation and numerical solution of the targeted problem result in a highly non-linear mixed-type PDE-constrained optimization problem. This type of non-linear optimization problems is inherently ill-posed and hard to solve, posing significant numerical challenges. Thus, these problems demand for sophisticated algorithms, and highly efficient and accurate solvers. For an excellent introduction into general approaches for PDE-constrained optimization, we refer to [Bie03; Bor12; Her10; Hin09b], and to [Man17d] for a review on its importance and applications to medical image analysis. Profound surveys for medical image analysis in the context of brain tumor imaging can be found in [Ang07; Bau13]. In [Gho17c], we presented fast algorithms and the key computational kernels for the individual solver components of SIBIA and showcased their excellent scalability on very large image resolutions and distributed-memory architectures on up to thousands of cores. Our coupling scheme (the moving-patient scheme, see §6.1) for the tight integration of biophysical brain tumor models with medical image registration is presented in [Sch]. This is to our knowledge the first work approaching the coupled optimization problem using gradient-based optimization techniques and a state-of-the-art algorithm for constrained large deformation diffeomorphic image registration [Man18a; Man15; Man16a; Man16b; Man17a]. In what follows, we give a skim on related work and similar approaches, but limit ourselves to the work most closely related to ours.

A general overview on medical image registration is given in [Mod04; Sot13]. Here, we focus on the biophysics augmented image registration used for analysis and data assimilation in brain tumor MR imaging. We give a brief review on diffeomorphic image registration as such in §5.3.2. Normal-to-abnormal image registration (registering the atlas to the patient image) requires finding correspondences between images that are topologically different. The ill-defined correspondence arising from the presence of a tumor in only one of the images portrays the key challenge that has to be addressed in order to get meaningful image registration results. A straightforward strategy to overcome this issue is to simply mask the area affected by the tumor from the optimization, and thereby declare it as non-informative [Hen04; Ste04; Bre01]. A similarly motivated approach is to relax the registration in areas affected by the tumor [Par14]. However, these strategies are only practical for small lesions, and they will certainly give very poor registration results when having to deal with large pathologies and a severe deformation of healthy tissue. In [Li10; Li12], the authors presented an embedded maps approach that simultaneously inverts for the deformation field and a drift in intensity that represents the topological abnormality associated with the tumor. This approach has the advantage of being generic, i.e., it can easily be applied to other registration problems featuring abnormalities or topological differences between the images that do not originate from tumor growth (for instance, Alzheimer's, or pre- and postoperative image registration [Kwo14]). Considering the purpose of atlas-based segmentation and registration, i.e., goals (2) and (3), this approach may produce acceptable results. However, if we are interested in the biophysics, model calibration and prediction, this approach cannot be used.

A popular strategy to meet this goal is to augment an optical control PDE constrained optimization formulation for diffeomorphic image registration [Man18a; Man15; Man16a; Man16b; Man17b; Man17a] with more complex biophysics operators. Several approaches have been presented in literature for the biophysical model inversion and calibration of parameters to individual medical imaging data [Gho16a; Gho17a; Gho17c; Yan13; M M98; Rek13], among them a parabolic, non-linear, reaction-diffusion tumor progression model [Mur89; Swa02]. We review some different brain tumor models when presenting our cancer progression model in §5.3.1. Despite its phenomenological character, this model has been shown to generate simulations that are in good agreement with observations of abnormalities in clinical MR imaging data [Cla05; Hog07a; Har07; Kon10a; Kon10b; Le16; Lim16; Mos12; Men11; Man12; Rek13; Swa08]. The literature on related optimal control formulations for tumor progression simulation is numerous; we, e.g., refer to [Col14; Hog08b; Kno13; Kno17; Liu14; Man12; Qui15; Qui16; Won15].

This tightly integrated biophysical tumor progression models with deformable medical image registration has been previously targeted in [Goo13; Hog08a; Moh06; Zac08b; Zac08a; Zac09]. The work in [Moh06; Zac08b; Zac08a; Zac09] uses a purely mechanical model for tumor progression featuring some major limitations. The considered tumor progression model is oversimplified and not capable of generating or reconstructing tumors with complex shape. Furthermore, because of its purely mechanical character, the model does not provide information about progression and infiltration of cancerous cells in the surrounding healthy tissue. The primary focus of the work mentioned above is the atlas-based segmentation and registration, drawing an emphasis on achieving good registration results, i.e., similarity of the registered images at lowest computational cost. In [Zac08a] the authors showcase that the tumor model can be drastically simplified, yet the approach still shows good similarity results of the registered images. The solution of the considered coupled problem has been targeted employing various strategies: Besides PDE-constrained optimization formulations, a generative approach based on Bayesian inversion has been used [Men11]. In [Bak15; Kwo14; Hog08a; Goo13], the authors propose a framework for joint segmentation, registration and tumor modeling, which is very similar to our approach. Like ours, their methodology employs a PDE-constrained optimization approach with a non-linear, mixed-type reaction-diffusion-advection equation woven into the formulation as transport constraints, modeling the tumor progression. However, the previously mentioned approaches suffer from several shortcomings, such as requiring manual seeding for the tumor, monofocal tumors, oversimplified models and extremely long time-to-solution. The latter is critical in medical application. Reasons for longer response times in the above work are inferior algorithms based on either derivative free optimization, gradient descent algorithms, lack of powerful preconditioners, and an algorithm and implementation design which is not tailored for distributed data and massively parallel execution.

5.2.3 Contributions and Limitations

Contributions and Outline. Within this work, we developed the framework SIBIA [Gho17c; Sch], providing methods and algorithms for the analysis of clinical imaging data of primary brain malignancies. The goals and target applications (1)-(3) we aim at are described above. This work tightly integrates a sophisticated state-of-the-art large deformation diffeomorphic image registration method (CLAIRE; Constrained Large Deformation Diffeomorphic Image Registration [Man18a; Man15; Man16a; Man16b; Man17b; Man17a] following the pioneering work of [Chr96; Tro98; Beg05]) with a biophysical model inversion method for the simulation of brain tumor progression [Gho16a; Gho17a]. Both sub-problems are by their nature non-linear, PDE-constrained optimization problems, that have their validity and applications as standalone solvers. While medical image registration is a supremely generic tool of crucial importance in clinical routine (e.g. for cardiovascular diseases [Ser06a; Ser06b; Cos01; Ma13; Cot99; Won07], oncology [Cla05; Goo13; Goo11; Hog06; Yan13; Kon10a], and surgical-planning [Hin09a; Fer01; Ger01; War03]), the tumor inversion solver without image registration is only applicable when patient's longitudinal data (i.e., imaging data of multiple time instances, ideally including the healthy brain), is given [Rek13; Gho16a]. The sub-solver components have been developed prior to this work and are described in §5.3. The utilized methods and computational kernels representing the main building blocks of the sub-solvers are explained in §5.3.3. This work builds upon the tremendous previous effort of my collaborators, Andreas Mang (large deformation diffeomorphic image registration framework CLAIRE) and Amir Gholami (tumor inversion solver and computational kernels; Fast-Fourier Transform and cubic interpolation). While the registration code CLAIRE is kept as a sub-module of SIBIA with a clear defined interface, the biophysical model inversion solver has been redesigned, and, in joint work with Shashank Subramanian, a revised version has been tightly integrated into SIBIA.

Our work improves the approach in [Bak15; Kwo14; Hog08a; Goo13] in terms of formulation of the problem, employed algorithms and solvers, scalability, performance, and overcomes some of the shortcomings of the existing approaches. In particular, SIBIA is distinguished by (*i*) the utilized solvers and the highly efficient and scalable computational kernels, resulting in algorithms that scale to very large resolutions and thousands of cores [Man16b; Gho17c]: instead of using derivative-free-optimization methods [Goo13; Hog08a; Moh06; Zac08b; Zac08a; Zac09; Won15], we propose a Picard iteration-type solution on both control variables (tumor controls and registration controls) resulting in a block-Newton-like solver. We employ globalized inexact Newton methods, i.e., Gauß-Newton-Krylov and quasi-Newton methods, to solve the respective (modified) sub-problems—registration and biophysical inversion; (*ii*) two different formulations and respective optimality systems for the fully coupled problem; (*iii*) a parametrization of the tumor initial condition that not only allows us to represent multifocal tumors, but also significantly reduces the number of inversion parameters and simplifies the PDE constraints without loosing segmentation accuracy. Finally, (*iv*) our methodology integrates one of the most advanced state-of-the-art algorithms for constrained large deformation diffeomorphic image registration [Man18a; Man15; Man16a; Man16b; Man17a].

In §5.3.4, we present efficiency and scalability results for the computational kernels (FFT and interpolation), the registration solver and the inverse tumor solver for up to 16 thousand cores and 200 billion unknowns (problem size 64 times larger than state-of-the-art) on two x68 super computing systems – `HazelHen` at the Stuttgart High Performance Computing Center and `Lonestar5` at the

Texas Advanced Computing Center. For clinically relevant test cases, SIBIA is up to eight times faster than the state-of-the-art. These optimization and scalability results are a contribution within this work and arose from joint work with Amir Gholami, Andreas Mang, George Biros and Miriam Mehl (published in the proceedings of CM/IEEE Conference on Supercomputing'17 [Gho17c]).

An important contribution of this work is the formulation of the fully coupled PDE-constrained optimization problem and the presentation of an iterative Picard-type solution scheme to solve the strongly coupled problem. In Chapter 6, we present two different formulations of the problem. The first, ***moving-patient formulation (MP)*** in §6.1 focuses on the highest possible biophysics aided registration- and segmentation quality, and is best suited to accomplish goals (2) and (3). The second, ***moving-atlas formulation (MA)***, presented in §6.2, is more involved and allows for more diagnostically meaningful biophysical inversion results and is targeting goal (1). We use an adjoint approach and derive the first order optimality system for both formulations, respectively, and introduce two Picard iteration-type solution schemes to solve the respective system of strongly coupled, non-linear PDEs by means of modular sub-solvers. For both schemes, the sub-solvers need to be modified and tightly integrated into a tailored solution scheme, featuring inexact sub-problem solutions and parameter-continuation.

To this end, in §6.3.3 we propose several parameter-continuation schemes and investigate the convergence of the considered optimization scheme. As for the clinical application, the time-to-solution is an absolute critical factor; we further optimized and improved the employed algorithms and solvers. In §6.3.3, we present a multilevel based coarse-to-fine grid-continuation scheme which gradually improves a prolongated coarse level initial guess, reducing the overall time-to-solution by a factor of 4. We extended the tumor inversion solver by an advanced quasi-Newton variant, requiring only gradient information and renders the solution of the Hessian system for the Newton step redundant. For this alternative, we benefit from the small number of inversion variables for the biophysical model inversion, and intend to learn from the findings from the previous chapters. We compare quasi-Newton against Gauß-Newton-Krylov to evaluate if the significantly reduced costs per iteration outweigh the slower convergence of quasi-Newton. Combining all optimizations, we were able to make the overall solution process approximately $10 - 20$ times faster.

The two fully coupled inversion schemes, moving-patient and moving-atlas, are thoroughly analyzed and compared for synthetic and clinical imaging data. In Chapter 7, we present numerical experiments that demonstrate the quality, validity and efficiency of our solution schemes. Empirically, we show that the Picard iteration-type solution schemes reduce the gradient of the respective fully coupled formulation and converge to a local minimum. We conduct a mesh-independence study, and show that convergence of our scheme is not sensitive with respect mesh-size (i.e., the number of unknowns for both, $\boldsymbol{p}$ and $\boldsymbol{v}$) We analyze the quality of the moving-atlas results for diagnostic relevance. Timings and speedup factors are provided for all optimization variants and judged under consideration of the overall reconstruction quality and accuracy.

Within this work, a highly scalable and efficient joint image registration and biophysical inversion framework for image analysis and data assimilation in brain tumor imaging has been developed. At a glance, this comprises the following main contributions:

- *The derivation of two formulations for the fully coupled optimization problem, focusing on different aspects, i.e., (i) registration quality and image similarity, and (ii) biophysical parameter estimation.*
- *The derivation of the first order optimality conditions for both formulations of the joint registration and biophysical inversion for brain tumor growth.*
- *Two Picard iteration-type block-Newton solution schemes for both formulations, respectively, for solving*

the fully coupled optimization problem based on modular tumor and image registration components. The modular approach allows for great flexibility in the employed biophysical model and registration approach, but also the concrete realization of the underlying solvers. We empirically showcase convergence of the Picard iterations to a local minimum, monitoring the gradient of the fully coupled scheme.

- *Optimization and scalability of the utilized solvers and algorithms, allowing for extremely large resolutions and efficient parallel execution on thousands of cores.*
- *Significant reduction of time-to-solution (factor $10 - 20$) by improving and optimizing the employed solvers, usage of quasi-Newton methods and a multilevel based grid-continuation scheme, gradually improving coarse level solutions.*
- *Thorough numerical experiments and analysis on synthetic and real data that demonstrate the validity and efficiency of our approach. We compare results for the two schemes (i.e., different formulations of the coupled problem), present speedup factors for optimized solvers, and different use cases of the SIBIA framework. We further show a mesh-independent convergence rate for our solution scheme.*
- *Examination of the sensitivity of our solver on the choice of the tumor model variant and judgment of our schemes by the capability of producing diagnostically meaningful results.*

Limitations and Open Issues. Our current work features some limitations and unresolved issues remain. As of now, our framework only accepts segmented MRI data as input, or more precisely, probability maps, showing the distribution of cell concentration of the different tissue regions. Usually, one can only get tissue and tumor classes from segmentation, but not their concentration values [Man12]. Especially for goal (2) and (3) it is of great potential value to input and process different MRI modalities (T1w, T2w, FLAIR, etc.) and combine various feature spaces. One main limitation is that we do not have a theoretical proof for the convergence of our Picard iteration-type solution schemes, we only have empirical evidence; the gradient of the fully coupled problem continues to decrease.

Further, our tumor model is a rather simplistic and phenomenological reaction-diffusion model [Swa00; Swa02; Mur89] with very limited predictive capabilities that further currently does not include a tumor mass-effect (deformations of the parenchyma due to tumor growth). While the model yields very good quantitative results and is greatly useful for image analysis, segmentation and tumor characterization [Man12; Swa08; Jac15; Lim16], it might be too simplistic to provide diagnostic biophysical insights. Therefore, currently we mainly focus on goals (2) and (3). We want to state, however, that more complex tumor models accounting for effects such as mitosis, apoptosis, chemotaxis, and deformation of brain parenchyma (mass-effect) can theoretically be embedded into our framework due to a modular design pattern. Furthermore, we derived a more sophisticated problem formulation, presented as the ***moving-atlas coupling scheme***, which lays the basis for a sophisticated and predictive, data-driven biophysical model calibration from the methodological side. On the biophysical model side, straightforward improvements include anisotropy for the diffusion and enhancing the model to involve a tumor mass-effect, edema, necrosis, angiogenesis and chemotaxis [Eng15; Hab03; Haw13; Hog07b; Hog08b]. This is ongoing work. Notwithstanding the above, the biophysical model and tumor progression solver is integrated in a modular way within SIBIA and can be exchanged by a more predictive model in the future. Self-evidently, with these more complex models, further tests would be necessary to demonstrate convergence of the Picard iteration-type solution scheme. Typically, a more sophisticated model also drives the hardness and computational expenses of the model inversion. The fully coupled optimization problem we try to solve is highly non-convex and non-linear, therefore multiple solutions may exist. In order to reduce the chance of getting trapped in local minima, we employ a coarse-to-fine grid-continuation strategy together with a parameter continuation for the registration regularization parameter, which both gradually improve approximate initial guesses for the inversion variables. Nonetheless, our solver only guarantees convergence to a local minimum.

Within this work, we exclusively consider a deterministic problem formulation. In reality, however, all quantities and models, i.e., the measurement of the observed data, the mathematical model for the description of the underlying process, the model parameters/controls, and the inversion algorithm itself are subject to uncertainties due to noise, modeling error or numerical errors. Thus, instead of point estimates for our quantity of interest, confidence intervals are required, in order to allow for an uncertainty quantification and propagation from the input to the quantity of interest. This is of particular interest in the case where the seeked for model parameters are not the final quantity of interest, but only used to compute or predict a future quantity of interest (such as, e.g., degree of cancerous cell infiltration, an estimate on life expectancy, or the probability of tumor recurrence). To account for uncertainties, our deterministic inversion approach can be combined with statistical inference methods, such as Bayesian posterior sampling [Mar12; Pet14]. We only consider deterministic inversion.

Summarized, the limitations of this work are:

- *Our tumor model is very simplistic and not predictive. It does not model edema, necrosis, angiogenesis and chemotaxis and is lacking a description of tumor mass-effect.*
- *We do not have a theoretical proof for the convergence of our Picard iteration-type solution scheme, but provide empirical evidence for convergence. We present preliminary results, that are primarily a proof of concept, but show very promising characteristics and provide evidence that our scheme has great potential to be of clinical relevance if enhanced with more complex models.*
- *Our framework, as of now, cannot integrate and process multimodal data from different MRI modalities. Input is assumed to be a set of probability distributions for each brain tissue and actual tumor concentration values.*
- *We only guarantee convergence to a local minimum.*
- *We consider deterministic inversion methods and do not account for uncertainties in input data and models.*

5.3 Tumor Simulation and Medical Image Registration – Sub-Components

Our joint inversion method for analyzing brain tumor MR imaging data builds upon the two sub-components, a brain tumor progression solver for biophysical model estimation [Gho16a], and a large deformation diffeomorphic image registration solver (CLAIRE; [Man18a; Man15; Man16a; Man16b; Man17b; Man17a]). We present formulation, models, solver strategies and employed algorithms for both sub-components. The methods and algorithms for the sub-components have been developed for the past few years at the Institute for Computational Engineering and Sciences (ICES) at the University of Texas at Austin. This section lays the foundation of the contribution covered within the second part of this thesis and briefly reviews the work of my collaborators, Andreas Mang (image registration, CLAIRE), Amir Gholami (computational kernels, tumor solver) and George Biros. This description is not extensive and we limit ourselves to the most relevant parts that are crucial for the subsequent chapters; we refer to the above mentioned references for further details. This description introduces both components with a view towards the coupled formulation in the next chapter; it follows [Sch; Man18b]. In §5.3.4, we demonstrate the parallel scalability and efficiency of the involved sub-components. These result have been originally published at the Supercomputing'17 conference [Gho17c]; a joint effort including significant own contributions.

5.3.1 The Biophysical Model Inversion

Within this work, we rely on a phenomenological model to simulate brain tumor progression, which, due to its simplistic nature allows for inversion, and, thus, parameter estimation for model calibration. We give a brief abstract of brain anatomy and present the utilized reaction-diffusion brain tumor progression model along with some non-exhaustive view on related work. We then state the inverse problem, based on a PDE-constrained optimization formulation, with the goal to seek for estimates of the tumor seed, i.e., the initial concentration at time $t_T = 0$, and the rate of tumor cell invasion into different tissue regions of the human brain.

The biophysical model inversion for the simulation of the progression of Gliomas addresses multiple bio-medical challenges: (*i*) The highly infiltrative nature of the tumor well beyond the visible tumor bulk and the ambiguous imaging criteria make differential diagnosis difficult. Simulations of Glioma progression with a calibrated biophysical model may allow for a more reliable prediction of the extent of tumor infiltration. (*ii*) Clinical data is static, i.e., no information about past and future progression is available at the time of diagnosis. Sophisticated tumor growth models have the potential to enable prediction of patient specific individual outcome. (*iii*) The data space is very large and cannot be assessed by visual inspection only. Further insights and hidden information is likely to be revealed by designing a consistent and unbiased simulation-based framework for data analysis.

Synopsis of Brain Anatomy. The brain is a central organ, consisting of billions of neurons that govern the central nervous system and the peripheral nervous system. Thus, the brain controls most of our activities; it processes, integrates and coordinates information, makes decisions and sends control signals to the rest of the body. The brain is a highly complex structure. It consists of cerebrum, the brainstem and the cerebellum (CBM) and is contained in and protected by the skull. The cerebrum is connected to the spinal cord via the brainstem. The main components of the human brain are *white matter* (WM), *gray matter* (GM), the *ventricles* (VT) with *cerebrospinal fluid* (CSF), and blood. For a schematic illustration of these components, see Fig. 5.5 The cerebrum is the largest part of the brain, and divided into (nearly symmetric) left and right hemisphere, each of which is split into four lobes; the frontal, temporal, parietal, and occipital lobes. The outer part of the cerebrum, the cerebral cortex, is primarily made up of *gray matter*, and covers the core of *white matter*. The ventricular system lies within the cerebrum and consists of interconnected ventricles, filled with cerebrospinal fluid. The latter serves as suspension for the sensitive brain structures. *White matter* is primarily composed of glial cells and axonal nerve fibres. The nerve fibres are bundled to fiber tracts, and diffusion along these fiber tracts is significantly higher than those perpendicular to them. The brain surface is folded into ridges and grooves that present various manifestations across individuals. No two human brains are alike. While the basic anatomical structures are always encountered, their individual shapes and characteristics may vary greatly.

Brain Tumor Progression Models—Literature. The difficulty in brain tumor progression modeling is that we have to cope with a highly heterogeneous system with countless interacting constituents (e.g., cell division, nutrients, hypoxic cells, viable tumor cells, extracellular matrix, etc.). The challenge

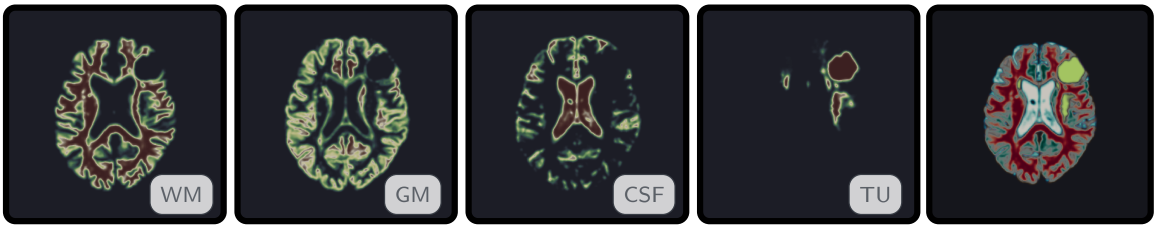


FIGURE 5.5 *Schematic view of the primary brain structures. From left to right, the probability maps for white matter (WM), gray matter (GM), cerebrospinal fluid with ventricles (CSF), and tumor (TU) are shown. The rightmost image is an overlaid combination of all previous probability maps.*

is to develop mathematical characterizations and relations that describe the biological phenomena responsible for emergence, growth, or decline of tumor (compare 8 hallmarks of cancer [Lim16]) and are consistent with basic physical laws. The variety of Glioma progression models is extensive. They can be classified into two main approaches: discrete [Kan00] and continuous [Ara04]. Models belonging to the discrete setting are on the microscopic scale and describe cellular division and migration. The spatio-temporal evolution of cells and the interaction between surrounding cells is described by probabilistic phenomenological rules to model, e.g., mitosis, apoptosis, chemotaxis, and random motion. Cellular automata [Kan00] and lattice-based models belong to this class. Continuous models don't track cancerous cells individually, but describe the evolution of local tumor cell density on the macroscopic scale, often times governed by the laws of continuum mechanics expressed via PDEs [Cri03; Lim16; Hog08a; S05; Tra99]. Most of these models assume cancer progression to be dominated by two main phenomena, cell division/mitosis and cell migration, reflected by reaction-diffusion-advection models [Jac15; Mur89; Swa00; Swa02], that effectively take into account the impact of the nature of the surrounding brain tissue. Attempts that consider locally changing diffusivity parameters effectively capture the infiltrative nature of brain tumors [S05; Pri04; Pri07; Ale07]. More complex macroscopic models based on multi-species mixture theory and porous media flow, that take into account cellular heterogeneity and incorporate mechanical effects in tissue, have been shown to be capable of generating very realistic simulations of tumor progression [Sci13; Ara05; Byr03; Gu12; Haw13; Rah17]. However, they are characterized by an excessive number of parameters and their complexity renders them impractical for parameter estimation and model calibration.

We employ a simplistic reaction-diffusion model, which by no means, is predictive on its own, but captures the most important phenomenological characteristics of Glioma growth. It is the de-facto standard approach for modeling brain tumor progression when linked to medical imaging [Swa00; Swa08; Cla05; Har07; Kon10a; Hog08a]. Despite its simplistic nature, some results available in literature accredit this model to offer some predictive capabilities when integrated with medical imaging [Swa00; Tom14]. The diffusion part of the model is a good fit to describe the highly infiltrative character and submicroscopic invasion into surrounding tissue and the small number of model parameters allows for effective model personalization, i.e., model calibration.

Mathematical Formulation

Notation. Our formulation is based on probability maps $m_i(\boldsymbol{x}, t) \in [0, 1]$, (i.e., tissue cell distributions) for all dominant tissue region types. Namely, we use probability maps for ***white matter*** (WM), ***gray matter*** (GM), ***ventricles*** with ***cerebrospinal fluid*** (CSF), and ***tumor***. Fig. 5.5 exemplarily shows a set of probability maps for a real data set. The tissue probability maps evolve in the space-time interval $\Omega \times T$ with $\Omega = (0, 2\pi]^d$, $T = [0, 1]$, $d = \{2, 3\}$ and spatial boundary $\partial\Omega$.

They are composed into a space-time vector field

$$\boldsymbol{m}(\boldsymbol{x},t) = (m_i(\boldsymbol{x},t))_{i=1,\dots,3} \in \mathbf{R}^3 \text{ with } m_1 = m_{GM}, m_2 = m_{WM}, m_3 = m_{CSF} \tag{5.3}$$

encoding the brain anatomy in the space-time domain. We interpret the normalized concentration of cancerous cells as probability distribution $c(\boldsymbol{x},t)\colon \Omega_B \times [0,1] \to [0,1]$, which represents the probability to encounter cancerous cells at a location $\boldsymbol{x}$ at time t. For better readability, we additionally define the space-time domains $U = \Omega \times (0,1]$, $U_B = \Omega_B \times (0,1]$ and $\tilde{U} = \Omega \times [0,1)$, $\tilde{U}_B = \Omega_B \times [0,1)$. To simplify notation, we assume that all probability maps are defined in Ω[4]. We will see, that the registration solver (cf. §5.3.2) also introduces a pseudo-time variable to model image intensity transport. For clarity of notation, we distinguish between the physical time $t_T \in T$ of tumor evolution and the pseudo-time $t_R \in T$ for image advection. The subscript may be omitted if obvious from the context. To reduce complexity, for most formulations we only explicitly state the dependency on time, but omit the dependency on the spatial position $\boldsymbol{x}$. For instance, we indicate the initial tumor probability map (the tumor *seed*) by $c(0)$, whereas $c(1)$ denotes the probability map of the 'grown' tumor at time $t_T = 1$ (solution of the forward problem).

The Tumor Growth Model (Forward Problem). We consider a non-linear mixed-type reaction-diffusion partial differential equation model, which has been widely adopted in literature [Swa00; Swa08; S05; Roc10; Har07]. As of now, we do not account for a tumor mass-effect, i.e., the deformations of parenchyma due to tumor growth. Our model describes the spatio-temporal tumor growth based on the normalized tumor density $c(\boldsymbol{x},t_T)$, which in our setup is also interpreted as cancerous cell distribution. The model encodes two main phenomena: ***proliferation*** (cell division/mitosis) of cancerous cells and the ***net migration*** of malignant cells into surrounding healthy tissue [Mur89]. The cell proliferation is modeled as a logistic growth function

$$R_{\boldsymbol{m}}(c) := \rho_{\boldsymbol{m}}\, c(1-c), \tag{5.4}$$

with ***reaction coefficient***

$$\rho_{\boldsymbol{m}}(\boldsymbol{x}) := \rho_f\, \rho_{\boldsymbol{m},0}(\boldsymbol{x}) = \rho_f\, \left(\rho_w\, m_{WM}(\boldsymbol{x}) + \rho_g\, m_{GM}(\boldsymbol{x})\right), \tag{5.5}$$

where ρ_f is a scaling of the characteristic growth rate parameters ρ_w and ρ_g in white and gray matter, respectively. The infiltrative migration model is given by an inhomogeneous, (potentially anisotropic) diffusion process

$$D_{\boldsymbol{m}}(c) := \nabla \cdot \boldsymbol{k}_{\boldsymbol{m}}\, \nabla c, \tag{5.6}$$

with ***diffusion coefficient***

$$\boldsymbol{k}_{\boldsymbol{m}}(\boldsymbol{x}) := k_f\, k_{\boldsymbol{m},0}(\boldsymbol{x})\boldsymbol{I} + k_a\, \boldsymbol{T}(\boldsymbol{x}) = k_f\, \left(k_w m_{WM}(\boldsymbol{x}) + k_g m_{GM}(\boldsymbol{x})\right) \boldsymbol{I} + k_a\, \boldsymbol{T}(\boldsymbol{x}), \tag{5.7}$$

$\boldsymbol{k}_{\boldsymbol{m}} \in \{\tilde{\boldsymbol{k}}\colon \Omega_B \to \mathbf{R}^{d,d} \mid \tilde{\boldsymbol{k}} > 0, \tilde{\boldsymbol{k}} = \tilde{\boldsymbol{k}}^T\}$, with $\boldsymbol{I} := \operatorname{diag}(1,\dots,1) \in \mathbf{R}^{d,d}$ and k_f and k_a representing the scaling factors for the *isotropic* and *anisotropic* parts of the diffusion tensor. Experimental biological studies [Gie96] have shown, that, due to a higher cell density in gray matter, Glioma cells have a

[4] The biophysical tumor progression model equations require Neumann boundary conditions on $\Omega_B \subset \Omega$, where Ω_B is the area of the brain, enclosed by the skull. By using a penalty approach, we extent the equations onto whole Ω. This is also known as fictitious domain method [Del03; Man12]; for details, see §5.3.3.

2-100-fold higher motility rate in white matter than in gray matter [Swa00]. Therefore, we consider an inhomogeneous diffusion model $k_{\boldsymbol{m},0}(\boldsymbol{x})$, based on the distribution of white matter and gray matter cells with their respective characteristic motility rates k_w and k_g [Swa00; Swa02; Hog07a; Sil97]. Further, it is commonly accepted that glial cells (thus tumor cells) preferably migrate along white matter fiber tracts [S05; Le 01; Pai13; Rek13; Eng15; Gie03; Gie96; Man06; Pri07; Pri04; Ale07]. This can be reflected by an anisotropic diffusion model with weighted diffusion tensor $\boldsymbol{T}(\boldsymbol{x}) \in \mathbf{R}^{d,d}$, derived from diffusion tensor imaging (DTI) data[5]. Although integration of anisotropy is straightforward, the experiments within this work only consider isotropic diffusion, i.e., we fix $k_a := 0$. We therefore omit details on the various formulations of the anisotropic diffusion tensor and refer to [Gho16a; S05; Cla05] for details. Our model is stated in dimensionless form. Typical values we use for proliferation and migration rate are $\rho_f = 10$ in non-dimensional form (with $\rho_w = 1$ and $\rho_g = 1/5$), which corresponds to a proliferation rate of $\rho = 0.003$ per day in dimensional form, and $k_f = 1\text{E}{-2}$ (with $k_w = 1$ and $k_g = 1/10$), corresponding to $k_f = 0.001\ mm^2$ per day in dimensional form.

Summarizing, the non-linear parabolic ***tumor forward model*** PDE with non-constant coefficients for the tumor concentration c is given by

$$\partial_t c - D_{\boldsymbol{m}}(c) - R_{\boldsymbol{m}}(c) = 0 \qquad \text{in } \Omega_B \times (0,1], \tag{5.8a}$$

$$\partial_n c = 0 \qquad \text{on } \partial\Omega_B \times (0,1], \tag{5.8b}$$

$$c(0) = \Phi \boldsymbol{p} \qquad \text{in } \Omega_B. \tag{5.8c}$$

For the tumor initial condition, we use a parametrization $c(0) := \Phi\boldsymbol{p}$, as stated in (5.8c), in an $n_{\boldsymbol{p}}$-dimensional space spanned by a Gaussian basis, i.e., $\boldsymbol{p} \in \mathbf{R}^{n_p}$, $\Phi\boldsymbol{p} := \sum_{i=1}^{n_p} \varphi_i p_i$ with Gaussian basis functions $\varphi_i : \Omega_B \to \mathbf{R}$, $\boldsymbol{x} \mapsto (2\pi\sigma^{1/2})^{-1}\exp(-1/2(x-x_i)^T\sigma^{-1}(x-x_i))$. For notational convenience, we represent the process of solving (5.8) based on the operator

$$\mathcal{T}^{\rightarrow}[\boldsymbol{p}] := c(1), \tag{5.9}$$

which maps the parametrization $\boldsymbol{p}$ of the initial conditions in (5.8c) to the tumor density $c(1)$ at physical (end-)time $t_T = 1$. An exemplary illustration of a tumor forward simulation obtained by our model is given in Fig. 5.6.

The Tumor Inverse Problem. A key issue for successful tumor progression modeling with the objective of clinical, patient-specific decision making is the personalization of the model, i.e., the ability to determine reasonable patient-specific parameters from medical imaging data. The most important parameters to be identified for our reaction-diffusion model are (*i*) a combination of the diffusion and reaction coefficient (i.e., characteristic (average) diffusion rates in white and gray matter, diffusion tensor information, anisotropy, and proliferation rate), (*ii*) the initial position and the initial concentration (cell distribution) of the tumor at physical time $t_T = 0$, and (*iii*) the tumor progression time horizon. The model calibration task does not have a unique solution, as different combinations of the above mentioned parameters may result in the same grown tumor result. Thus, not all parameters can be estimated simultaneously. In our approach, we fix the proliferation rate and the time horizon. From the (pre-processed/segmented) patient MR data, we then solve for

[5] Diffusion Tensor Imaging (DTI) is a modern magnetic resonance imaging method that helps to reveal human brain structure by tracing the diffusion behavior of injected water molecules [Mor08; Le 01; Pai13; Bas94; Pri04; Ale07]. The result of this measurement is a tensor-field that can be directly integrated in the formulation in (5.7)

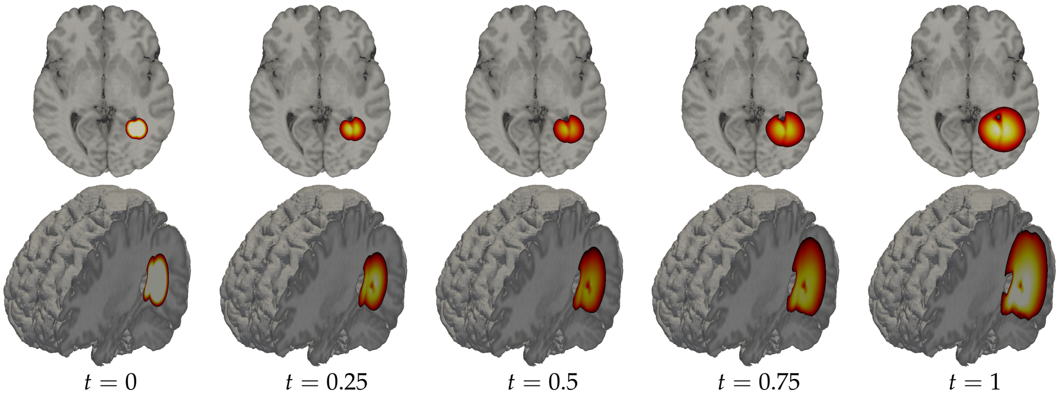


FIGURE 5.6 *Brain tumor progression simulation results as obtained from our reaction-diffusion forward model at different time instances: The leftmost image shows the initial condition; the rightmost image shows the grown tumor as solution of the forward model at physical time $t_T = 1$. The top row shows an axial slice cut through the tumor center; the bottom row shows a 3D illustration. Locations of high tumor cell density are visualized as white areas, while darker areas indicate low tumor density. Figure originally published in [Gho17c].*

estimates of the initial tumor concentration[6] as well as for average diffusivity rates in white matter and gray matter, respectively.

Our numerical optimization approach extends [Hog07a] by using second order (Hessian) information (instead of only using gradient information, or employing a derivative-free approach [Man12]). Most of the following has been introduced in [Gho16a]).

We state the inverse tumor problem as PDE-constrained optimization problem, where we seek to find coefficients $\boldsymbol{p}$ for the parametrization of an initial condition $c(0) = \Phi\boldsymbol{p}$ for the forward tumor problem and (isotropic) diffusivity rates $\bar{k} = (k_1, k_2, k_3)^T = (k_w, k_g, 0)^T$ in white matter and gray matter, respectively, that allow the model to recover a given patient tumor concentration c_D as good as possible, i.e., we solve the minimization problem

$$\min_{\boldsymbol{q}=(\boldsymbol{p},\bar{k})} \mathcal{J}_T[\boldsymbol{q}, c, c_D] := \mathcal{D}_c[c, c_D] + \frac{\beta_p}{2}\mathcal{S}_{\boldsymbol{p}}[\boldsymbol{p}] \tag{5.10a}$$

subject to

$$\begin{aligned} &\partial_t c - D_{\boldsymbol{m}}(c) - R_{\boldsymbol{m}}(c) = 0 && \text{in } \Omega_B \times (0,1], && (5.10b)\\ &\partial_n c = 0 && \text{on } \partial\Omega_B \times (0,1], && (5.10c)\\ &c(0) = \Phi\boldsymbol{p} && \text{in } \Omega_B. && (5.10d)\end{aligned}$$

The driving ***data similarity*** term

$$\mathcal{D}_c[c, c_D] := \frac{1}{2}\|c(1) - c_D\|^2_{L^2(\Omega_B)} \tag{5.11a}$$

is a squared L^2-distance that measures the discrepancy between the simulated tumor $c(1)$ at time $t_T = 1$ (solution of forward model (5.10b)–(5.10d)) and the target patient data c_D. The second term is a L^2-*Thikonov regularization* term with the regularization parameter β_p. Here, we choose to penalize

[6] The initial tumor concentration (initial condition) implicitly also yields the initial tumor position.

large values in the initial condition

$$\mathcal{S}_{\boldsymbol{p}}\left[\boldsymbol{p}\right] := \|\Phi\boldsymbol{p}\|^2_{L^2(\Omega_B)} \tag{5.11b}$$

Other new regularizations will be considered in Chapter 6. Regularization is inevitable when solving inverse problems, as they are usually ill-posed, instable (amplify noise and high frequencies), and/or do not have a unique solution per se. In particular, the null-space of the inversion operator contains high frequency components of the inversion variables. Our Thikonov-type regularization penalizes high variations in the reconstructed initial condition of the tumor, thus effectively eliminates the null-space of the inverse operator. The regularization term and its weighting parameter β_p usually have a large influence on the solution process. For theory and application of different regularizations, see [Eng96]. The solution of problem (5.10), i.e., the parameter tuple $\boldsymbol{q} = (\boldsymbol{p}, \bar{k})^T$ is referred to as ***control variables*** and determined based on performance goals defined in the ***objective function*** (5.10a). Together, this defines the ***inverse tumor operator***

$$\mathcal{T}^{\leftarrow}\left[c_D\right] := \boldsymbol{q} = \arg\min_{\hat{\boldsymbol{q}}} \mathcal{J}_T\left[\hat{\boldsymbol{q}}, c_D\right]. \tag{5.12}$$

5.3.2 The Registration Solver

The second sub-component of our joint inversion framework for image analysis is medical image registration. We consider non-rigid (deformable) registration, which allows for more complex deformations, as required in the analysis of brain tumor magnetic resonance imaging data. We assume images (in our case, probability maps) as continuously differentiable functions[7] with compact support on a domain $\Omega = [0, 2\pi)^d$ with closure $\bar{\Omega} = \Omega \cup \partial\Omega$ and data dimensionality $d \in \{2, 3\}$. Image registration is a classical task in medical image analysis, which essentially tries to establish a point-wise spatial correspondence $\boldsymbol{y} \colon \bar{\Omega} \to \mathbf{R}^d$, $\boldsymbol{x} \mapsto \boldsymbol{y}(\boldsymbol{x})$, between two images $m_T \colon \bar{\Omega} \to \mathbf{R}$, $\boldsymbol{x} \mapsto m_T(\boldsymbol{x})$ and $m_R \colon \bar{\Omega} \to \mathbf{R}$, $\boldsymbol{x} \mapsto m_R(\boldsymbol{x})$ of the same or a similar object, such that the ***warped*** (or transformed) ***template image*** $m(1) = m_T \circ y$ becomes similar to the ***reference image*** m_R [Mod04; Mod09]. In generic form, this can be formulated as finding a minimizer $\boldsymbol{y}$ of the variational problem

$$\min_{\boldsymbol{y}} \mathcal{D}_m\left[m_T \circ \boldsymbol{y}, m_R\right] + \beta\mathcal{S}_{\boldsymbol{y}}\left[\boldsymbol{y}\right] \qquad \text{subject to} \quad \mathcal{C}\left[m_T, \boldsymbol{y}\right] = 0 \tag{5.13}$$

driven by the data similarity measure, quantifying[8] $\mathcal{D}_m$ the proximity between m_R and $m_T \circ \boldsymbol{y}$; here, $\circ$ denotes the function composition. In absence of regularization, the above problem is usually ill-posed and lacking a unique solution. The regularization term, whose influence is weighted by the regularization parameter β, effectively eliminates the null-space of the inverse operator and ensures the required regularity of the solution. Its particular choice is highly dependent on the application at hand; typically it is given as a Sobolev norm of a certain order. Together with possible constraints $\mathcal{C}\left[m_T, \boldsymbol{y}\right] = 0$ (typically a set of partial differential equations), the regularization operator penalizes and rules out unwanted solutions and ensures ***plausible*** mappings.

In the context of medical image registration, a key requirement for a deformation map to be considered as plausible is to be a ***diffeomorphism***, i.e., a smooth, invertible, and continuously differentiable map with smooth inverse that maps Ω onto itself (automorphism). This is locally

[7] If required, we pre-smooth the input data or use appropriate mollifiers to obtain the required regularity.

[8] Different choices for the data similarity measure are conceivable [Mod04; Mod09; Sot13], the most common of which is a L^2-distance $\mathcal{D}_m = 1/2\|m_T \circ \boldsymbol{y} - m_R\|^2_{L^2(\Omega)}$ based on image intensities.

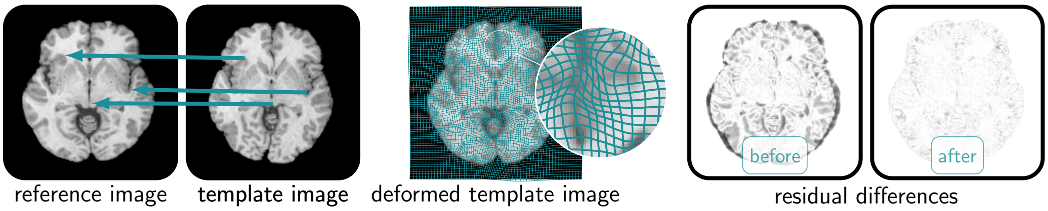


FIGURE 5.7 *Image registration. Given scalar intensity values of two images m_R and m_T of the same object (left), the inverse problem of image registration seeks for a deformable geometric transformation $\boldsymbol{y}$ that relates points in the reference image m_R to its corresponding points in the template image m_T (green arrows). The existence of a deformable geometric relation between the images is assumed and part of the model. As we seek for point correspondences based on intensity values, the solution of this problem may easily result in implausible mappings, if no further restrictions are imposed. In our formulation, we consider a mapping to be meaningful or plausible if it is a diffeomorphism, meaning it is unique, smooth, and invertible. The formulation may be further augmented by certain material properties and biophysical constraints. The transformation process is modeled as a transport equation. The deformed template image is illustrated in the middle image, and on the right, the residual between the images before and after the registration process, is shown. Figure originally published in [Gho17c].*

fulfilled if the functional determinant of the transformation map $\det(\nabla \boldsymbol{y})$ is strictly positive and can be guaranteed by a suitable regularization operator. Further complimentary data or prior knowledge about the underlying problem may be incorporated by adding additional constraints to the problem, e.g., incompressibility or elasticity properties of the material, or descriptions of physiological and biophysical processes.

By its nature, image registration is an inverse problem, which due to its vast number of unknowns, inherent instabilities and computational complexity poses significant mathematical and algorithmical challenges. As a result it asks for carefully designed, fast, robust, and highly scalable algorithms and solvers that allow for precise control of the physical and mathematical properties of the deformation map $\boldsymbol{y}$. In the sequel, we recapitulate the most important parts of the work of Andreas Mang [Man18a; Man15; Man16a; Man16b; Man17b; Man17a] with respect to the application targeted within this thesis. The image registration task is illustrated in Fig. 5.7. Before presenting the optimal control-based formulation for the large-deformation diffeomorphic image registration, we review an extract of the most important and most closely related literature on non-rigid image registration.

Non-Rigid Image Registration—Literature. There is a vast body of literature on image registration and no attempt is made for an extensive review. We focus on closely related work only, i.e., non-rigid approaches that view image registration as a problem of optimal control and model the deformation map via a velocity field. For a general introduction to the problem, including applications and algorithmic approaches, we refer to [Fis08; Mod04; Mod09; Sot13]. In the context of medical image registration, small deformation models [Bro81; Fis02; Fis03] reveal many limitations and lead to unsatisfying results. To overcome these shortcomings, viscous fluid mechanical models for diffeomorphic image registration have been proposed [Chr94; Chr96], which consider diffeomorphic registration as hyperbolic optimal control problem [Bor02; Che11a; Har09; Man15; Man16a; Man17a; Man16b]. The basic idea is to introduce a pseudo-time variable t_R, and instead of solving for the deformation map $\boldsymbol{y}$, we seek for a (possibly non-stationary) velocity $\boldsymbol{v}$, which parameterizes $\boldsymbol{y}$ via a hyperbolic transport or continuity equation. The latter explicitly appears as a constraint in the optimization problem. The relation between a diffeomorphic deformation map $\boldsymbol{y}$ and the regularity requirements on $\boldsymbol{v}$ have been studied in [Beg05; Che11b; You10]. Loosely speaking, a sufficiently smooth velocity $\boldsymbol{v}$ gives raise to a diffeomorphic map $\boldsymbol{y}$. The required regularity of $\boldsymbol{v} \in \mathcal{S} \subset H^k(\Omega)$

the Sobolev space of order $k \in \{1, 2, 3\}$ is typically attained by a Sobolev norm[9] in the regularization operator. This approach has been extended and embedded into a variational framework [Beg05; Dup98; Gre98a; Mil01; Tro98]. These formulations directly search for an optimizer on the manifold of diffeomorphic mappings by employing a geodesic distance[10] regularization functional on the velocity $\boldsymbol{v}$. However, solely restricting the search space to the manifold of diffeomorphisms does not necessarily result in physically meaningful deformations. Further, these models do not allow the control of geometric properties and might result in transformation maps that are close to being non-diffeomorphic. Thus, some applications may benefit from imposing further constraints[11] on the resulting deformation map. Related formulations that use constraints on the velocity $\boldsymbol{v}$ and the functional determinant $\det(\nabla \boldsymbol{y})$ of $\boldsymbol{y}$ can be found in [Lee10; Lee11].

The here considered PDE-constrained (Eulerian) formulation for diffeomorphic image registration does not operate on the space of diffeomorphisms, but models the transport of intensities in m_T via a linear advection equation. This only depends on the velocity $\boldsymbol{v}$, and, thus allows the augmentation of the formulation with additional constraints [Bor02; Che11b; Man15; Man16a; Man16b]. This formulation also lays the foundation to augment the image registration process by more complex biophysical operators [Goo13; Hog08a; Sun09; Zac08a; Zac08b; Zac09], which also is the objective of this thesis.

Within this work, we employ the registration framework CLAIRE [Man18a; Man15; Man16a; Man16b; Man17b; Man17a], which implements the above mentioned formulation. So far, only little effort has been made in the velocity-based diffeomorphic image registration community to design efficient state-of-the-art algorithms; thus, its advanced numerical methods, massively parallel scalability and distributed memory algorithms, as well as the utilized discretization scheme is what mainly sets CLAIRE apart from similar tools. While almost all existing related efforts exclusive make use of first-order derivative information for numerical optimization [Ash07; Beg05; Bor02; Che11b; Chr94; Chr96; Har09; Her09; Lee10; Lee11; Mus09; Via12; You07], the methodology considered here employs second-order information. As opposed to slowly converging gradient-descent methods, the use of Hessian information gives rise to extremely powerful, inexact Newton-type solvers featuring fast convergence rates. Here, preconditioners for the Hessian system become an indispensable presupposition to obtain a short time-to-solution; approaches therefore are discussed in [Ben11; Her18; Man15; Man17a; Man17b; Sim12] and find application in CLAIRE. Not least, it significantly reduces the memory footprint and features scalable distributed memory algorithms with highly optimized computational kernels [Man16b; Gho17c], to solve e.g. the transport equation and to compute derivatives. This allows for the registration of extremely large imaging data resulting from recently emerging high resolution medical imaging methods [Chu13; Kut17; Tom14]. For a more detailed literature review on diffeomorphic image registration, we refer to [Man15; Man18b].

[9] The regularity of the chosen Sobolev space determines the regularization of the solution. The order of the Sobolev norm is chosen depending on the smoothness of the input images m_T and m_R, the form of the transport equation and the existence of additional constraints, e.g., incompressibility, that is penalization of divergence of $\boldsymbol{v}$, or biophysical information of the underlying deformation process. To meet the key requirement of $\boldsymbol{y}$ being diffeomorphic, the Sobolev norm in the regularization operator needs to enforce $\boldsymbol{v} \in \mathcal{S} \subset H^k$ to be sufficiently smooth. The minimum required order k depends on the dimensionality d of the data space; for $d = 3$, the norm $\|(-\Delta + \boldsymbol{I})^{\hat{k}} \boldsymbol{v}\|^2_{L^2(\Omega)}$ with $\hat{k} > 1.5$ guarantees $\boldsymbol{y}$ to be a diffeomorphism[Beg05]

[10] The typical formulation for these approaches takes the form $\min_{\boldsymbol{v}} \frac{1}{2} \|m_T \circ \varphi(1)^{-1} - m_R\|^2_{L^2(\Omega)} + \frac{\beta}{2} \int_0^1 \|\boldsymbol{v}\|^2_{\mathcal{S}} \, \mathrm{d}t$ subject to $d_{t_R} \varphi = \boldsymbol{v}\varphi$ for $t_R \in (0, 1])$ and $\varphi = \boldsymbol{I}$ for $t_R = 0$, with the time integral of the squared Sobolev norm as regularization term, embedding the manifold of diffeomorphisms in a Riemann space. The Sobolev norm guarantees that the resulting transformation map as solution of $d_{t_R} \varphi = \boldsymbol{v}(\varphi)$ is a diffeomorphism.

[11] In medical image registration, incompressibility of tissue may be a reasonable additional constraint on the deformation map, which can be realized by enforcing $\det(\nabla \boldsymbol{y}) = 1$. Controlling the deviation of the functional determinant from 1, i.e., penalizing the divergence of $\boldsymbol{v}$ is also an alternative, if we allow for a bounded extent of compression or expansion.

Mathematical Formulation

Advection-Based Deformation Map. Within this work, reference and template images are magnetic resonance images, or probability maps of brain tissue regions of different brains. We use a Lagrangian formulation for the deformation map and parametrize the actual mapping $\boldsymbol{y}$ via a velocity field $\boldsymbol{v}\colon \bar{\Omega} \times [0,1] \to \mathbf{R}$, $(\boldsymbol{x}, t_R) \mapsto \boldsymbol{v}(\boldsymbol{x}, t_R)$ by introducing the pseudo-time variable $t_R \in [0,1]$. As a consequence, the deformation map $\boldsymbol{y}$ does not occur explicitly in our formulation but can be computed as $\boldsymbol{y} = \boldsymbol{x} - \boldsymbol{u}(1)$ by solving

$$\partial_t \boldsymbol{u} + \nabla \boldsymbol{u} \cdot \boldsymbol{v} = \boldsymbol{v} \;\text{ in } U, \qquad \boldsymbol{u}(0) = 0 \;\text{ in } \Omega \tag{5.14}$$

with periodic boundary conditions on $\partial\Omega$ for the displacement field $\boldsymbol{u}\colon \bar{\Omega} \times [0,1] \to \mathbf{R}^d$, $(\boldsymbol{x}, t_R) \mapsto \boldsymbol{u}(\boldsymbol{x}, t_R)$. Although CLAIRE provides support for a transient (variable in time) as well as for a stationary (constant in time) velocity field, we limit ourselves to stationary velocity fields, as it significantly reduces the dimensionality of the search space, and for medical image registration both approaches yield comparable results in terms of image similarity (mismatch) [Ars06; Ash07; Her09; Man16a; Man17b; Man16b].

Advective Image Transformation (Forward Problem). Given a stationary velocity field $\boldsymbol{v}(\boldsymbol{x})$, the forward problem describes the advective transformation of the template image towards the reference image in a pseudo-time interval $[0,1]$. The transported intensities of the ***warped template image*** $m(1) := m(\boldsymbol{x}, t_R = 1)$ are computed by solving

$$\partial_t m + \nabla m \cdot \boldsymbol{v} = 0 \qquad \text{in } U, \tag{5.15a}$$

$$m(0) = m_T \qquad \text{in } \Omega, \tag{5.15b}$$

a hyperbolic transport equation with periodic boundary conditions on $\partial\Omega$ forward in time. This formulation may be augmented by further constraints to control the geometric properties of the resulting deformation map. As mentioned above, a possibly beneficial augmentation for medical image registration is the ***incompressibility*** of tissue. This translates to enforcing a divergence free velocity field $\nabla \boldsymbol{v} = 0$ (equivalent to imposing $\det(\nabla \boldsymbol{y}) = 1$ on the functional determinant[12] of the deformation map $\boldsymbol{y}$). This constraint can be relaxed by introducing a mass-source map w to control the extent of compression and expansion of the deformation

$$\gamma(\nabla \cdot \boldsymbol{v} - w) = 0 \qquad \text{in } \Omega. \tag{5.15c}$$

The parameter $\gamma \in \{0,1\}$ is introduced for clarity, to enable or disable the incompressibility constraint. If prior information and complimentary data for the underlying physiological process is available, further biophysical constraints might be added to the formulation to give raise to a deformation map that respects actual biophysical processes. Approaches to do so are proposed and analyzed within this work. This is covered in Chapter 6. The set of equations in (5.15) defines an implicitly given operator, the parameter-to-observation map

$$\mathcal{R}^{\rightarrow}\,[\boldsymbol{v}, m_T] := m(1) \tag{5.16}$$

[12] The functional determinant $\det(\nabla \boldsymbol{y}) := \det(\boldsymbol{F}(1))$, with $\boldsymbol{F}\colon \bar{\Omega} \times [0,1] \to \mathbf{R}^{d,d}$ of the transformation map $\boldsymbol{y} \in C^2(\Omega)$ reflects local volume change and can be used to assess invertability of $\boldsymbol{y}$. In our formulation, the deformation tensor field $\boldsymbol{F}$ can be computed from the velocity $\boldsymbol{v}$ by solving

$$\partial_t \boldsymbol{F} + (\boldsymbol{v} \cdot \nabla)\boldsymbol{F} = (\nabla \boldsymbol{v})\boldsymbol{F}, \;\text{ in } U, \qquad \boldsymbol{F}(0) = \boldsymbol{I} \;\text{ in } \Omega,$$

with periodic boundary conditions on $\partial\Omega$ and the identity tensor $\boldsymbol{I} = \operatorname{diag}(1, \ldots, 1) \in \mathbf{R}^{d,d}$

mapping the template image m_T to the reference image m_R at pseudo-time $t_R = 1$.

Image Registration (Inverse Problem). The actual image registration problem is given by the inverse problem, where we seek a velocity field $\boldsymbol{v}(\boldsymbol{x})$ such that the transported intensities of the template image m_T at pseudo-time $t_R = 1$ (i.e., solution of (5.15)) match the intensity values in the reference image m_R as good as possible. That is, we solve for $\boldsymbol{v} \in \mathcal{V} \subset L^2(\Omega)$ and $w \in \mathcal{W} \subset L^2(\Omega)$ as follows

$$\min_{\boldsymbol{v}} \mathcal{J}_R\,[\boldsymbol{v}, w] \quad \text{with} \quad \mathcal{J}_R\,[\boldsymbol{v}] := \mathcal{D}_m\,[m, m_R] + \beta_v \mathcal{S}_v\,[\boldsymbol{v}] + \gamma \beta_w \mathcal{S}_w\,[w] \tag{5.17a}$$

subject to

$$\textit{advection} \qquad \partial_t m + \nabla m \cdot \boldsymbol{v} = 0 \qquad \text{in } U, \tag{5.17b}$$

$$m(0) - m_T = 0 \qquad \text{in } \Omega, \tag{5.17c}$$

$$\textit{incompressibility} \qquad \gamma(\nabla \cdot \boldsymbol{v} - w) = 0 \qquad \text{in } \Omega, \tag{5.17d}$$

for a minimizer $(\boldsymbol{v}, w)$ with the ***data similarity*** term

$$\mathcal{D}_m\,[m, m_R] := \frac{1}{2} \|m(1) - m_R\|^2_{L^2(\Omega)} \tag{5.18a}$$

as squared L^2-distance, quantifying the discrepancy between the data and the warped template. We want the velocity $\boldsymbol{v}$ to give raise to a ***plausible*** mapping. As discussed above, this is attained by a suitable regularization term; The type and weight of which are selected to drive the solution towards a diffeomorphic map $\boldsymbol{y}$ at acceptable cost and maintaining good registration quality in terms of data similarity. An appropriate choice depends on many factors such as smoothness of m_R and m_T, desired properties of the resulting deformation, or additional constraints, and needs to be made with care; a theoretical discussion can be found in [Bar16; Beg05; Bor02; Che11b; Man16a; Via12]. The regularization operators $\mathcal{S}_v\,[\boldsymbol{v}]$ and $\mathcal{S}_w\,[w]$ define the regularity of the Sobolev spaces $\mathcal{V}$ and $\mathcal{W}$, respectively; the ***regularization parameter*** $\beta_v > 0$ balances the influence of the smoothness penalty on $\boldsymbol{v}$, while $\beta_w > 0$ acts as a penalty on $\nabla \cdot \boldsymbol{v}$ and controls the extent of incompressibility.

We require $\boldsymbol{y}$ to be diffeomorphic, and thus $\boldsymbol{v}$ to be of sufficient regularity. Further, we consider $\gamma = 1$ and by adding (5.17d) we ask for a (nearly-) incompressible deformation map. As a result, we chose the regularization operator $\mathcal{S}_v\,[\boldsymbol{v}]$ to enforce smoothness on $\boldsymbol{v}$ as a H^1-semi-norm[13], and an H^1-norm in $\mathcal{S}_w\,[w]$, respectively

$$\mathcal{S}_v\,[\boldsymbol{v}] = \frac{1}{2} \int_\Omega \sum_{i=1}^{3} |\nabla v^i(\boldsymbol{x})|^2 \,\mathrm{d}\Omega, \qquad \mathcal{S}_w\,[w] = \frac{1}{2} \int_\Omega |\nabla w(\boldsymbol{x})|^2 + |w|^2 \,\mathrm{d}\Omega. \tag{5.18b}$$

Using an H^1-seminorm for the regularizer of $\boldsymbol{v}$ is motivated by the fact that upon solution of (5.17) using a variational approach, we obtain optimality conditions that are similar to Stokes equations in fluid mechanics. For discussion and numerical results for different choices[14] of theses operators we

[13] This choice is appropriate if incompressibility is considered, i.e., $\nabla \boldsymbol{v} = 0$, as suggested by experimental and analytical analysis [Che12]. In [Man16a] it is shown by experimental analysis, that the choice of an H^1-seminorm in combination with controlling the deformation gradient (parameter w and β_w), yields small mismatch values and good-natured deformation maps, while converging significantly faster than using an H^2 seminorm.

[14] Often times, an H^2-seminorm $\mathcal{S}_v\,[\boldsymbol{v}] = \|\Delta \boldsymbol{v}\|^2_{L^2(\Omega)^d}$ is used in large deformation diffeomorphic image registration [Beg05; Har09; Via12]. In [Man16a], the authors' suggestion is based on extensive numerical analysis, that an H^1-seminorm is favorable in terms of computational complexity. It reduces the time-to-solution significantly, while yielding plausible

refer to [Man15; Man16a; Man17b]. Overall, this defines the (inverse) registration operator

$$\mathcal{R}^{\leftarrow}\left[m_T, m_R\right] := v = \arg\min_{\hat{v}} \mathcal{J}_R\left[\hat{v}\right]. \tag{5.19}$$

5.3.3 Numerical Methods

In the sequel, we present the strategies, methods, and solvers to numerically solve the presented problems in §5.3.1 and §5.3.2. For numerical optimization we employ an adjoint approach and introduce ***Lagrange multipliers*** (adjoint variables) to transform the constrained optimization problems (5.10) and (5.17) into unconstrained ones by defining the respective ***Lagrange functional***. For the registration problem, we follow an ***optimize-then-discretize*** (OTD) strategy, i.e., we arrive at continuous equations for the first order optimality conditions after requiring stationarity of the Lagrangian and taking variations with respect to the state, adjoint and control variables. These optimality conditions are then discretize in a second step. This approach contrasts the ***discretize-then-optimize*** (DTO) approach, where the Lagrangian is first discretized and thereafter, optimality conditions for the discrete approximation are derived [Wil15], p. 55ff. For OTD, it is neither guaranteed that the discretized gradient is consistent to the discretized objective function, nor that the discretized forward and adjoint operators are transposes of each other; also, the discretized Hessian may not be symmetric. This can have negative effects on the convergence of the solvers[15]. For DTO, the operators are consistent by construction. However, the numerical accuracy of the solvers, e.g., forward and adjoint solver, may be different [Hag00]. For a discussion on the discretization strategies, we refer to [Gun03; Bor12].

First, we derive optimality conditions for the considered sub problems in §5.3.1 and §5.3.2. As a result, we arrive at large systems of coupled non-linear PDEs, which upon discretization, give raise to large, ill-conditioned, non-linear operators that are formidable to solve and pose significant numerical challenges and feature very high computational complexity. We summarize the discretization in space and time, numerical strategies, and utilized solvers to efficiently tackle these challenges. The employed numerical schemes are tailored to efficiently solve the occurring PDE operators. The vast number of unknowns and computation complexity urge for massively parallel distributed memory implementations that scale up to tens of thousands of cores. Two major computational kernels emerge, whose efficient parallel implementation is at the heart of the solvers. We shortly outline the methods and give references for a more in-depth description.

Optimality Conditions

Biophysical Model Inversion. A classical way to solve the constrained optimization problem (5.10) is to introduce a ***Lagrange multiplier*** field $\alpha\colon \Omega_B \times [0,1] \to \mathbf{R}$, and $\alpha(0) = \alpha(\boldsymbol{x}, t_T = 0)$, corresponding to the state c, and define the ***Lagrange functional***

$$\mathcal{L}_T\left[c, \alpha, \boldsymbol{q}\right] = \mathcal{J}_T\left[c, \boldsymbol{q}\right] + \alpha^T \mathcal{T}^{\rightarrow}\left[\boldsymbol{q}\right], \tag{5.20}$$

well-behaved deformation maps if the deformation gradient $\det(\nabla \boldsymbol{y})$ is bounded appropriately (i.e., β_w needs to be tuned accordingly). In this case, the resulting registration quality in terms of image mismatch is similar as if the more expensive H^2-regularizer was used.

[15] For our application, we found by numerical analysis, that a relative reduction of the gradient of more than three orders of magnitude is impractical. Thus, the discretization error lies below the required accuracy. For a more accurate solution, our solvers may fail to converge, and discrete operators and solvers that preserve the continuous properties are required.

which takes into account the constraints $\mathcal{T}^{\rightarrow} = 0$ (residuals of (5.8)) via an inner product with α. The Lagrange multiplier α is also called co-state or ***adjoint variable***. For the biophysical tumor progression model inversion problem (5.10) the ***Lagrangian*** is given by

$$\mathcal{L}_T\left[c,\alpha,\boldsymbol{q}\right] = \mathcal{J}_T\left[c,\boldsymbol{q}\right] + \int_0^1 \langle \alpha, \partial_t c - \nabla \cdot \boldsymbol{k} \nabla c - \rho c\,(1-c)\rangle_{L^2(\Omega_B)}\,\mathrm{d}t + \langle \alpha(0), c(0) - \Phi \boldsymbol{p}\rangle_{L^2(\Omega_B)} \tag{5.21}$$

As a necessary condition for minima, we require stationarity of the Lagrangian with respect to the state c, adjoint α and control variables $\boldsymbol{q}$. Stipulating vanishing first order variations

$$\delta\mathcal{L}_T := \left(\delta_\alpha\mathcal{L}_T, \delta_c\mathcal{L}_T, \delta_{\boldsymbol{q}}\mathcal{L}_T\right)^T = \left(g_\alpha, g_c, g_{\boldsymbol{q}}\right)^T = \boldsymbol{0}, \tag{5.22}$$

we can derive the following system of coupled, non-linear PDEs, often referred to as the strong form of the ***First Order Optimality Conditions***, or the Karush-Kuhn-Tucker (KKT) conditions [Noc06], p. 323ff:

$$\textit{state eq.:} \quad \delta_\alpha\mathcal{L}_T = 0: \quad \partial_t c - \nabla \cdot (\boldsymbol{k}\nabla c) - \rho c(1-c) = 0 \quad \text{in } U_B, \tag{5.23a}$$

$$\delta_{\alpha(0)}\mathcal{L}_T = 0: \quad c(0) - \Phi\boldsymbol{p} = 0 \quad \text{in } \Omega_B. \tag{5.23b}$$

$$\textit{adjoint eq.:} \quad \delta_c\mathcal{L}_T = 0: \quad -\partial_t\alpha - \nabla \cdot (\boldsymbol{k}\nabla\alpha) - \alpha\rho + 2\alpha\rho c = 0 \quad \text{in } \tilde{U}_B, \tag{5.23c}$$

$$\delta_{c(1)}\mathcal{L}_T = 0: \quad \alpha(1) - c_D + c(1) = 0 \quad \text{in } \Omega_B. \tag{5.23d}$$

$$\textit{inversion eq.:} \quad \delta_p\mathcal{L}_T = 0: \quad g_{\boldsymbol{p}} := \beta_{\boldsymbol{p}}\delta_{\boldsymbol{p}}\mathcal{S}_{\boldsymbol{p}}\left[\boldsymbol{p}\right] - \Phi^\star\alpha(0) = \boldsymbol{0} \quad \text{in } \Omega_B, \tag{5.23e}$$

$$\delta_{k_i}\mathcal{L}_T = 0: \quad g_k := \int_0^1 \int_{\Omega_B} \boldsymbol{m}_i (\nabla c)^T \nabla\alpha \,\mathrm{d}x\,\mathrm{d}t = 0 \quad i \in \{1,2\}, \tag{5.23f}$$

with $\delta_{\boldsymbol{p}}\mathcal{S}_{\boldsymbol{p}}\left[\boldsymbol{p}\right] = \Phi^\star\Phi\boldsymbol{p}$ for our particular choice (5.11b) of the regularization operator for $\boldsymbol{p}$. The ***state equations***, are given by the earlier defined forward simulation problem (5.8) of tumor progression. The ***adjoint equation*** (5.23a) originates from stationarity with respect to the state variable and is a final value problem, linear in the adjoint variable α. Note, that the final condition in (5.23d) at $t_T = 1$ depends on c, which is the solution of the forward problem (5.8). Finally, the inversion equations for the control variables are given by (5.23a) and (5.23f). The set of coupled, non-linear PDEs given by the optimality system (5.23) is formidable; it portrays high complexity, indefiniteness, ill-conditioning, and a very large number of unknowns.

For numerical optimization, we therefore employ a ***reduced space approach***, where the state and adjoint equations are assumed to be solved exactly (i.e., their gradients vanish, $g_c = g_\alpha = 0$), and the search space is reduced to the (low dimensional) space of the control variables $\boldsymbol{q}$. This not only reduces the system's size to a manageable one, but the reduced system (and its first order approximation, the reduced Hessian) typically also feature better-behaved numerical properties. Assuming we are given state and adjoint variables $\widetilde{c}$ and $\widetilde{\alpha}$ such that $g_\alpha = g_c = 0$, the ***reduced gradient*** defines the non-linear root finding problem

$$g_{\boldsymbol{q}}(\boldsymbol{q}) \overset{!}{=} 0 \;\text{ in } \Omega_B, \qquad \text{with} \quad g_{\boldsymbol{q}} = \left(g_{\boldsymbol{p}}, g_{\bar{k}}\right)^T\Big|_{\substack{g_\alpha(\boldsymbol{q},\widetilde{c})=0\\ g_c(c,\widetilde{\alpha})=0}}, \tag{5.24}$$

which is then to be solved by means of gradient-descent or Newton-type iterations. This is in contrast to the ***full space approach***, considering the system as is, and solves for updates of state, adjoint and control variables simultaneously. The evaluation of the reduced gradient (5.24) for a given trial $\boldsymbol{q}$ then involves the following steps: (i) Compute the system's state $c(1)$ by solving (5.23a) for the control $\boldsymbol{q} = (\boldsymbol{p}, \bar{k})$ forward in time; (*ii*) given the final state $c(1)$ (prediction of data), compute α from solving the adjoint equation (5.23c) backward in time. This final value problem depends on the predicted state $c(1)$ and models the data misfit (residual differences to the observed data) backwards in time. (*iii*) Using the data misfit $\alpha(0)$, the expression for the reduced gradient can then be evaluated to compute updates for the control $\boldsymbol{q}$ in a suitable numerical optimization scheme. For the latter, the canonical approach is to use gradient descent schemes. Within this work, we use inexact Newton method to solve (5.24), i.e., Gauß-Newton-Krylov and quasi-Newton. Details are provided in subsequent sections and §6.4.

Diffeomorphic Image Registration. From the PDE-constrained optimization problem (5.17) for diffeomorphic image registration, we obtain the ***Lagrangian*** functional by introducing the Lagrange multipliers $\lambda \colon \bar{\Omega} \times [0,1] \to \mathbf{R}$, $\lambda(0) \colon \Omega \to \mathbf{R}$, corresponding to m, and $\nu \colon \bar{\Omega} \to \mathbf{R}$, associated with w as

$$
\begin{aligned}
\mathcal{L}_R\,[m, \boldsymbol{v}, w, \lambda, \lambda(0), \nu] = {} & \frac{1}{2}\|m(1) - m_R\|^2_{L^2(\Omega)} + \frac{\beta_v}{2}\|\nabla \boldsymbol{v}\|^2_{L^2(\Omega)} + \frac{\beta_w}{2}\|w(\boldsymbol{x})\|^2_{H^1(\Omega)} \\
& + \int_0^1 \langle \lambda, \partial_t m + \nabla m\, \boldsymbol{v} \rangle_{L^2(\Omega)}\,\mathrm{d}t + \langle \lambda(0), m(0) - m_T \rangle_{L^2(\Omega)} \\
& + \int_0^1 \langle \nu, \nabla \cdot \boldsymbol{v} - w \rangle_{L^2(\Omega)}\,\mathrm{d}t.
\end{aligned}
\tag{5.25}
$$

We arrive at the ***First-Order Optimality Conditions*** in strong form [Man16a] by requiring stationarity of the Lagrangian, taking variations and applying integration by parts

$$
\begin{array}{llrrl}
\text{(5.26a)} & \textit{state eq.:} & \delta_\lambda \mathcal{L}_R = 0: & \partial_t m + \nabla m \cdot \boldsymbol{v} = 0 & \text{in } U, \\
\text{(5.26b)} & & \delta_{\lambda(0)} \mathcal{L}_R = 0: & m(0) - m_T = 0 & \text{in } \Omega, \\
\text{(5.26c)} & & \delta_\nu \mathcal{L}_R = 0: & \gamma(\nabla \cdot \boldsymbol{v} - w) = 0 & \text{in } \Omega, \\
\text{(5.26d)} & \textit{adjoint eq.:} & \delta_m \mathcal{L}_R = 0: & -\partial_t \lambda - \nabla \cdot (\boldsymbol{v}\lambda) = 0 & \text{in } \bar{U}, \\
\text{(5.26e)} & & \delta_{m(1)} \mathcal{L}_R = 0: & m_R - m(1) - \lambda(1) = 0 & \text{in } \Omega, \\
\text{(5.26f)} & \textit{inversion eq.:} & \delta_{\boldsymbol{v}} \mathcal{L}_R = 0: & \boldsymbol{g}_v := \beta_v \delta_v \mathcal{S}_v\,[\boldsymbol{v}] + \mathcal{K}[\int_0^1 (\nabla m)^T \lambda\,\mathrm{d}t] = \mathbf{0} & \text{in } \Omega.
\end{array}
$$

with the elliptic operator $\delta_v \mathcal{S}_v\,[\boldsymbol{v}] = \Delta \boldsymbol{v}$ for our particular choice (5.18b) of the regularization operator; for different choices, see [Man15; Man16a; Che11b]. The resulting optimality system (KKT-system), consists of three blocks of equations, all of which are strongly coupled, non-linear PDEs. (5.26a) is referred to as the ***state equation***, which based on the control parameters $(\boldsymbol{v}, w)$, (5.26c) and the initial condition (5.26b) yields the system's predicted state. The ***adjoint equation*** (5.26d) with final condition (5.26e) models the transport of the mismatch between the deformed template image $m(1)$ and the reference image m_R backward in time. As we approach an admissible solution of the system (5.26), λ will tend to vanish. Finally, (5.26f) is the ***control equation*** for the unknown velocity $\boldsymbol{v}$. Note that we haven't given an equation for $\delta_w \mathcal{L}_R$; the associated penalty constraint on the divergence of $\boldsymbol{v}$, $\gamma(\nabla \boldsymbol{v} - w) = 0$, controlling the local volume change, is eliminated from the formulation. As a result, we get the pseudo-differential operator $\mathcal{K}$ in (5.26f), projecting $\boldsymbol{v}$ onto the space of nearly-incompressible velocity fields. For the considered regularization terms, $\mathcal{K}$ takes the form

$$
\mathcal{K} = \boldsymbol{I} - \nabla \mathcal{M}^{-1} \Delta^{-1} \nabla \cdot, \quad \text{with} \;\; \mathcal{M} = \beta_v(\beta_w(-\Delta + \boldsymbol{I}))^{-1} + \boldsymbol{I}
\tag{5.27}
$$

for a non-zero mass-source map w. In case of incompressibility, i.e., $w = 0$, $\mathcal{M}$ simplifies to the identity operator, and $\mathcal{K}$ is given by the Leary projection operator $\mathcal{K} = \boldsymbol{I} - \nabla\Delta^{-1}\nabla$. For $\gamma = 0$, we obtain $\mathcal{K} = \boldsymbol{I}$ and the incompressibility constraint is dropped. For details on the elimination and derivation of $\mathcal{K}$ for the considered and other regularization schemes, we refer to [Man16a].

In the same manner as for the biophysical model inversion, we attempt a reduced space approach by assuming state and adjoint fields $\tilde{m}$ and $\tilde{\lambda}$ such that $g_\lambda = g_m = 0$, resulting in the (non-linear) root-finding problem for the reduced gradient

$$g_v(v) \overset{!}{=} \mathbf{0} \;\text{ in } \Omega, \qquad \text{with} \quad g_v = \beta_v \nabla_v \Delta v + \mathcal{K}\Big[\int_0^1 (\nabla\tilde{m})^T \tilde{\lambda}\, \mathrm{d}t\Big]\Bigg|_{\substack{g_\lambda(v,\tilde{m})=0\\ g_m(m,\tilde{\lambda})=0}} \tag{5.28}$$

For the image registration, evaluating the reduced gradient (5.28) for a given trial velocity v involves the following algorithmic steps: (*i*) We compute the deformed template $m(t)$ as a result of the forward operator $\mathcal{R}^{\rightarrow}[m_T, v]$ by solving equations (5.26a)-(5.26c) forward in time; (*ii*) we then solve the adjoint equation (5.26d) using the previously computed final state $m(1)$ in (5.26e) for λ backward in time. (*iii*) Finally, the expression for (5.28) can be evaluated given the state and adjoint variable $m(t)$ and $\lambda(t)$, respectively. The latter is an integro-differential operator, requiring the time history for the state and adjoint equation, i.e., checkpointing approaches to store the time history of these variables become necessary.

As before, we employ a reduced space, inexact, matrix-free Gauß-Newton-Krylov solver [Man15; Bir05a; Bir05b] for numerical optimization. The numerical optimization using a Gauß-Newton-Krylov solver involves methods and algorithms, parallel implementation, and main computational kernels, which we discuss next.

Numerical Optimization

Strategy. For both sub-problems, we have derived the first order optimality systems (5.23) and (5.26) by requiring stationarity of the Lagrangian functionals with respect to state, adjoint and control variables, and taking variations. The reduced space approach results in the non-linear equations (5.24) and (5.28). Abstracting from the specific sub-problems, we require the gradient with respect to the control variable to vanish, i.e., we arrive at the non-linear equation

$$g(w_\star) = 0, \qquad \text{with} \quad g \in \{g_q, g_v\} \quad \text{and} \quad w \in \{q_\star, v_\star\}\,, \tag{5.29}$$

for the reduced gradient g and control variable w. For an admissable solution $w_\star$ of the respective optimization problem, equation (5.29) needs to be solved by means of a suitable iterative solver. The canonical choice are gradient-descent methods, requiring only gradient evaluations, and thus, can be implemented with limited effort. Despite their poor convergence properties, gradient-descent schemes have been widely used in the context of PDE-constrained optimization and the considered application problems [Ash07; Beg05; Bor02; Che11b; Chr96; Har09; Her09; Lee11; Mus09; Via12]. With a view to large scale simulations and more sophisticated formulations (i.e., considering more expensive regularization operators, etc.), these methods become infeasible in terms of time-to-solution, or fail to converge due to increasingly ill-conditioned operators. Preconditioned, inexact-Newton methods have been shown to outperform the simpler gradient-descent schemes in these regimes, especially if we strive for high inversion accuracy [Man15; Akc06; Kel87a; Bir99; Bir05a; Bir05b]. A further, strong reason for using second-order derivatives for our optimization algorithm is the non-linearity of the inverse problem in the control variables[16] and the typically poor conditioning of the resulting operators. Here, a Newton linearization is used to solve the optimality systems (5.23) and (5.26). Following the aforementioned reduced space approach, the reduced Hessian operator corresponding to (5.29) is obtained as a Schur-complement for the control variables, i.e., a block-elimination of the state and adjoint variables from the full space KKT system. High-resolution volumetric images result in a vast number of unknowns and large-scale severely ill-conditioned operators. As a result, matrix free solvers and different degrees of inexactness for the Newton-type solvers as well as powerful preconditioners are inevitable.

Newton Step. A Newton-type solution for (5.29) requires second-order variational information of the Lagrangian. Adhering to the reduced-space approach, this results in the ***reduced Hessian*** $\mathcal{H}_{\boldsymbol{w}} \in \mathbf{R}^{n_w,n_w}$ (after block-elimination), given as the Schur-complement of the control $\boldsymbol{w}$ of the second-order optimality system. The system has reduced dimensionality of the number of control variables n_w. We employ a globalized, inexact, preconditioned, matrix-free, reduced-space Gauß-Newton-Krylov (GNK) solver for numerical optimization. Given an algorithm that evaluates the reduced gradient $\boldsymbol{g}_{\boldsymbol{w}}$, the Newton step reads

$$\mathcal{H}_{\boldsymbol{w}}^k \hat{\boldsymbol{w}}^k = -\boldsymbol{g}_{\boldsymbol{w}}^k, \quad \boldsymbol{w}^{k+1} = \boldsymbol{w}^k + \gamma^k \hat{\boldsymbol{w}}^k, \quad k = 1, 2, \dots \tag{5.30}$$

As we opt for a matrix-free, iterative Krylov solution to compute the Newton update $\hat{\boldsymbol{w}}^k = -(\mathcal{H}_{\boldsymbol{w}}^k)^{-1} \boldsymbol{g}_{\boldsymbol{w}}^k$ on the control $\boldsymbol{w}^k$, we only need the action of the reduced Hessian on the ***incremental control*** variable $\hat{\boldsymbol{w}}^k$, i.e., the ***Hessian matvec*** $\mathcal{H}_{\boldsymbol{w}}^k[\hat{\boldsymbol{w}}^k]$ in every Newton iteration k. For our considered sub-components, the concrete representations for the Hessian matvec are given by

$$\mathcal{H}_{\boldsymbol{p}}\,[\hat{\boldsymbol{p}}] := \beta_v \delta_{\boldsymbol{p}}^2 \mathcal{S}_{\boldsymbol{p}}\,[\boldsymbol{p}] - \Phi^T \tilde{\alpha}(0) = -\boldsymbol{g}_{\boldsymbol{p}} \tag{5.31a}$$

$$\mathcal{H}_v\,[\hat{v}] := \beta_v \delta_v^2 \mathcal{S}_v\,[\boldsymbol{v}] + \mathcal{K}[\int_0^1 \hat{\lambda} \nabla m + \lambda \nabla \hat{m}\,\mathrm{d}t] = -\boldsymbol{g}_v \tag{5.31b}$$

for the biophysical inversion and image registration problem, respectively. Note, that for the former, we only state second-order information for the inversion of the initial condition parametrization $\boldsymbol{p}$; for the inversion of the diffusivity, $\bar{k} = (k_W, k_G)$, we use a quasi-Newton approach without deriving Hessian expressions. Here, $\delta_{\boldsymbol{p}}^2 \mathcal{S}_{\boldsymbol{p}}\,[\boldsymbol{p}] = \hat{\boldsymbol{p}}$ and $\delta_v^2 \mathcal{S}_v\,[\boldsymbol{v}] = \Delta \hat{\boldsymbol{v}}$ are the second-order variations of the respective regularization operators. In order to evaluate these expressions, we have to solve an ***incremental state*** and ***incremental adjoint*** equation; this is a direct consequence of the reduced space approach and block-elimination. That means, in order to compute the reduced Hessian action $\mathcal{H}_{\boldsymbol{p}}\,[\hat{\boldsymbol{p}}]$ on the incremental control $\hat{\boldsymbol{p}}$ of the biophysical inversion problem, we first have to solve the incremental state equation

$$\partial_t \hat{c} - \nabla \cdot (\boldsymbol{k} \nabla \hat{c}) - \rho \hat{c}(1 - 2c) = 0 \qquad \text{in } U_B, \tag{5.32a}$$

$$\hat{c}(0) - \Phi \hat{p} = 0 \qquad \text{in } \Omega_B, \tag{5.32b}$$

with periodic boundary conditions on $\partial\Omega$ for $\hat{c}$, forward in time. In a second step, given the state c,

[16] While the constraints, i.e., the forward problem is linear in the decision or control variables, the inverse problem is not; it is non-linear.

incremental state $\hat{c}$, and adjoint λ, solve the ***incremental adjoint equation***

$$-\partial_t \hat{\alpha} - \nabla \cdot (\boldsymbol{k} \nabla \hat{\lambda}) - \rho \hat{\lambda}(1 - 2c - 2\alpha) = 0 \qquad \text{in } \bar{U}_B, \tag{5.33a}$$

$$\hat{\lambda}(1) + \Phi^T \Phi \hat{c} = 0 \qquad \text{in } \Omega_B, \tag{5.33b}$$

with periodic boundary conditions on $\partial\Omega$ for $\hat{\alpha}$, backward in time. The Hessian action can then be computed from evaluating the expression (5.31a).

In a similar manner, the Hessian matvec $\mathcal{H}_{\boldsymbol{v}} [\hat{\boldsymbol{v}}]$ for the registration problem is computed from first solving the ***incremental state equation***

$$\partial_t \hat{m} + \nabla \hat{m} \cdot \boldsymbol{v} + \nabla m \cdot \hat{\boldsymbol{v}} = 0 \qquad \text{in } U, \tag{5.34a}$$

$$\hat{m}(0) = 0 \qquad \text{in } \Omega, \tag{5.34b}$$

$$\gamma(\nabla \cdot \hat{\boldsymbol{v}} - \hat{w}) = 0 \qquad \text{in } \Omega, \tag{5.34c}$$

with periodic boundary conditions on $\partial\Omega$ based on a current (Newton-) iterate $\boldsymbol{v}$ for the incremental state $\hat{m}$, forward in time. This is followed by solving the ***incremental adjoint equation***

$$-\partial_t \hat{\lambda} - \nabla \cdot (\boldsymbol{v} \hat{\lambda} + \hat{\boldsymbol{v}} \lambda) = 0 \qquad \text{in } \bar{U}, \tag{5.35a}$$

$$\hat{m}(1) + \hat{\lambda}(1) = 0 \qquad \text{in } \Omega, \tag{5.35b}$$

with periodic boundary conditions on $\partial\Omega$ for $\hat{\lambda}$ backward in time. Again, the action of $\mathcal{H}_{\boldsymbol{v}}$ on $\hat{\boldsymbol{v}}$ is computed evaluating the expression (5.31b), an integro-differential equation with the incremental body force $\hat{\lambda}\nabla m + \lambda \nabla \hat{m}$. Note, that the reduced Hessian system not only depends on the incremental state, adjoint, and control variables $\hat{m}, \hat{\lambda}$, $\hat{\boldsymbol{v}}$, but also on m, λ and $\boldsymbol{v}$; it is a strongly coupled system. Similar to the first order optimality conditions, the incremental state and adjoint variables $\hat{m}$ and $\hat{\lambda}$ are functions of $\hat{\boldsymbol{v}}$ via (5.34) and (5.35). The same applies for $\hat{\boldsymbol{p}}$ via (5.32) and (5.33) for the biophysical inversion problem. Furthermore, the computation of the adjoint equations as well as the evaluation of the Hessian matvecs, require time histories for the state and/or adjoint fields. Suitable checkpointing schemes or domain-decomposition methods [Akc02; Gri92; Hei05] and tailored time-integration schemes allowing for few time-steps become necessary.

The step size γ for the Newton update (5.30) is obtained from an Armijo-backtracking [Noc06], p. 33,48ff line search technique to guarantee global convergence. Inexactness is introduced in two ways: (*i*) To solve the Newton system $\mathcal{H}^k_w \hat{\boldsymbol{w}}^k = -\boldsymbol{g}^k_w$, a matrix-free, preconditioned Conjugate Gradient (PCG) method is used with an inexact Eisentat-Walker criterion [Eis96; Noc06]; and (*ii*) a Gauß-Newton approximation of the reduced Hessian is exploited to guarantee positive definiteness of the latter far away from a solution. For the reduced Hessian expressions (5.31a) and (5.31b), we arrive at the Gauß-Newton approximations $\mathcal{H}^{GN}_{\boldsymbol{p}}$ and $\mathcal{H}^{GN}_{\boldsymbol{v}}$ by dropping all terms in the incremental state and incremental adjoint that include the adjoint variable α and λ, respectively.

In §6.4, the mentioned Gauß-Newton-Krylov solver is explained in more detail for a general problem including in particular the reduced space approach and block-elimination resulting in the reduced Hessian operator. There, we further review different granularities of inexactness of Newton-type solvers, present different manifestations of solvers, and in particular, draw a comparison of Gauß-Newton-Krylov versus quasi-Newton methods in the context of large-scale PDE-constrained optimization. Within this thesis, we extend the biophysical inversion sub-component by an advanced, robust quasi-Newton solver, which, for a series of test cases, outperforms the GNK solver. Details are given in §6.4 and §7.4.5.

Preconditioner. For the considered Newton-Krylov optimization, most work is spent in the inner Krylov method to iteratively solve the Hessian system. Per Krylov iteration, this involves the solution of two PDEs, which is costly, thus powerful preconditioners are obligatory. For the biophysical inversion problem, we use the inverse of the diffusion operator with constant approximation of the diffusion coefficient [Gho16a] to precondition the reduced Newton system. For the registration, we employ the inverse regularization operator as a preconditioner for the Hessian system [Man15; Man16a]. This is a standard approach in PDE-constrained optimization. A more advanced, two-level multi-grid inspired preconditioner for the registration has been developed in [Man17a]. As we employ a spectral approach (details follow below), the application of the preconditioners has vanishing costs, due to fast and accurate evaluations of spatial differential operators. Assuming that all image features are fully resolved spectrally, the preconditioners are theoretically mesh-independent[17], i.e., yield the same number of Krylov iterations for increasing mesh-resolutions. Medical imaging data, however, typically features high contrast and sharp edges; thus, a grid that fully resolves the diffusion coefficient is likely to become prohibitively large. As a result, the number of Krylov iterations to solve the Hessian system increases as the resolution is refined. Further, the utilized preconditioners are not β-independent; in other words, changing the regularization parameter changes the number of required Krylov iterations. An exhaustive description of the utilized preconditioners is beyond the scope of this thesis; details can be found in the references given above.

Discretization

We discretize the space-time domain $\Omega \times [0,1], \Omega = [0,2\pi)^d \subset \mathbf{R}^d, d = \{2,3\}$, using regular grids. In space, we consider $n = \prod_{j=1}^{d} n_j, n_j \in \mathbf{N}$ regular grid points, with $\boldsymbol{x_i} = {}^{2\pi \boldsymbol{i}}/_{n}$, $\boldsymbol{i} = (i_1, \dots, i_d) \in \mathbf{R}^d$ and $0 \leqslant i_j \leqslant n_j - 1$, $j = 1, \dots, d$. For all spatial operations, we use a spectral projection scheme [Man16b; Gho17c], that is we approximate all considered functions (and images) as

$$m(x) = \sum_{\boldsymbol{\ell}} \hat{m}_{\boldsymbol{\ell}} \exp(-\boldsymbol{\ell} \cdot \boldsymbol{x}), \quad \text{where } \boldsymbol{\ell} = (\ell_1, \dots, \ell_d) \in \mathbf{R}^d$$

is a multi-index with $-{}^{n_j}/_2 + 1 \leqslant \ell_j \leqslant {}^{n_j}/_2, j = 1, \dots, d$. A spatial function m is associated with its *spectral coefficients* $\{\hat{m}_{\boldsymbol{\ell}}\}$. The mapping between $\{\hat{m}_{\boldsymbol{\ell}}\}$ and $\{m_{\boldsymbol{i}}\}$ is done using forward and inverse ***Fast Fourier Transforms (FFT)***. We assume all functions/images in our formulation to be periodic, sufficiently smooth, and continuously differentiable in space. We apply appropriate filters and smoothers, and, if necessary, periodically extend (zero-padding) or mollify our discrete data to meet these requirements. All spatial derivatives and differential operators are then approximated by applying the appropriate weights in the spectral domain and using forward and inverse FFTs to map back and forth between spatial and spectral representation. This approach allows us to apply all spatial differential operators (and their inverse) in a stable and efficient way with spectral-accuracy and computational costs dominated by the FFT mappings.

The biophysical tumor progression problem in (5.8) requires Neumann boundary conditions on $\partial\Omega_B$. For numerical treatment and discretization, we apply periodic boundary conditions on $\partial\Omega$, and extend the diffusion coefficient $\boldsymbol{k}$ by a small parameter ϵ outside of Ω_B in Ω. The original homogeneous boundary conditions (5.8b) can be re-imposed on $\partial\Omega$ and will be satisfied on $\partial\Omega_B$ in

[17] A mesh-independent preconditioner is a key pre-requisite for good algorithmic scalability. For the results presented in §5.3.4, the diffusion coefficient is not fully resolved, thus the number of Krylov iterations increase as we refine the mesh resolution for the weak-scaling experiments. The algorithmic scalability is not optimal.

```
1 func [$c$] = StrangSplitting($\boldsymbol{p} \in \mathbf{R}^{n_p}, \boldsymbol{\Phi} \in \mathbf{R}^{n_p,n_p}, n_t$)
2     $c(0) = \boldsymbol{\Phi p}$,  $\delta t = 1/n_t$
3     for $j = 0, 1, \ldots, n_t$ do
4         solve for $\widetilde{c}$:            $(\boldsymbol{I} - \frac{\delta t}{4}\boldsymbol{D}^h)\widetilde{c} = (\boldsymbol{I} + \frac{\delta t}{4}\boldsymbol{D}^h)c(t^j)$      in $(t^j, t^j + \frac{\delta t}{2}]$
5         solve for $\widehat{c}$:                       $\partial_t \widehat{c} = \rho c(1-c)$              in $(t^j, t^{j+1}]$
6                                                $c(t^j) = \widetilde{c}$
7         solve for $c(t^{j+1})$: $(\boldsymbol{I} - \frac{\delta t}{4}\boldsymbol{D}^h)c(t^{j+1}) = (\boldsymbol{I} + \frac{\delta t}{4}\boldsymbol{D}^h)\widehat{c}$      in $(t^j + \frac{\delta t}{2}, t^{j+1}]$
```

ALGORITHM 5.1 *Second order strang splitting for the parabolic tumor growth equation (5.23a). Here,* $\boldsymbol{D}^h = [\nabla\cdot]^h \boldsymbol{k} \nabla^h$ *is the discretized diffusion operator D.*

the limit $\epsilon \to 0$ [Del03]. This penalty strategy is known as fictitious domain method [Del03; Man12; Gho16a]. It can be shown that it approximates (5.8), and eventually converges to the correct solution as the discretization is further refined.

In time, we use a nodal discretization, resulting in $n_t + 1$ discretization points. We use different time-integration schemes, dependent on the characteristics and nature of the respective equations. This is what we discuss next.

Numerical Time Integration

The optimization problems for biophysical inversion and image registration involve repeated solutions of the forward and adjoint operators, given as parabolic and hyperbolic partial differential equations. In the sequel, we sketch the time-integration schemes for both types of equations. Our solvers require the storage of time-histories for the adjoint and/or state variable. Unconditionally stable schemes that are not bounded by the CFL number and allow for larger time steps are a key part in order to limit the memory footprint and make the usage of high resolution imaging data feasible. For details, we refer to [Man16b; Gho17c; Gho16a].

Parabolic PDEs. To solve the parabolic equations in the tumor forward and adjoint operators, we employ a unconditionally stable, second order Strang-splitting method [G68; Hog08b]. Generally speaking, splitting methods are beneficial for PDEs that involve several differential operators, which might have to be treated differently. We exemplarily explain the method for (5.23a). Let $c(t^j)$ denote the tumor distribution at time $t^j = j \cdot \delta t$, $\delta t = 1/n_t$. We take advantage of the splitting approach by applying an implicit Crank-Nickolson method for the diffusion part and plugging in the analytical solution for the reaction part of the PDE. The method is outlined in Alg. 5.1.

To solve the diffusion part in line 4 and line 7 of Alg. 5.1 for half a time step, we use a ***preconditioned conjugate gradient (PCG)*** method with a fixed absolute tolerance of 1E−6. Similar to the preconditioner for the Hessian matvec, we use a constant approximation of the diffusion coefficient for the preconditioner, given by $\boldsymbol{P}_T^h := (\boldsymbol{I} - \frac{\delta t}{4}\tilde{\boldsymbol{D}}^h)$, where $\tilde{\boldsymbol{D}}^h = ([\nabla\cdot]^h \tilde{\boldsymbol{k}} \nabla^h)$ and $\tilde{\boldsymbol{k}}$ is the average diffusion coefficient. The application of this preconditioner (computing the action of ${\boldsymbol{P}_T^h}^{-1} x$ for a vector x) involves a forward and inverse Fourier transformation and an Hadamard product in the frequency domain; the costs are negligible. For sufficiently smooth images, the preconditioner is mesh independent, in practice however, the number of inner Krylov solves is expected to increase for a refined mesh resolution.

Hyperbolic PDEs. The solution of hyperbolic advection dominated equations, such as the transport equation (5.15) is a core step for the registration solver. [Ewi01] gives an overview of solvers for advection dominated systems. Examples include explicit high-order total variation diminishing schemes [Bor02], or explicit, pseudo-spectral second order Runge-Kutta schemes [Man15; Man16a].

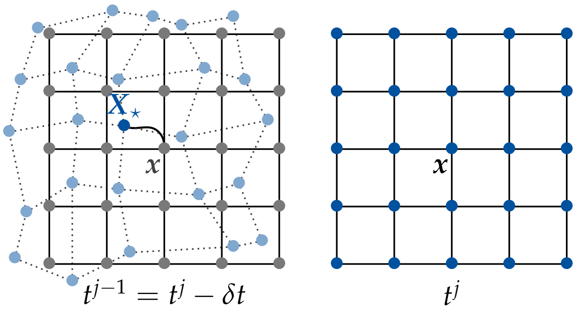


FIGURE 5.8 *Illustration of semi-Lagrangian scheme. Starting with a grid point $\boldsymbol{x}$ in the regular (Eulerian) grid at time point t^j (right figure), we compute its departure point $\boldsymbol{X}_\star = \boldsymbol{X}(t^{j-1})$ at time point t^{j-1}, by solving equation (5.36) for the characteristic $\boldsymbol{X}(t)$ (black line in left figure) backward in time. The value of the advected quantity $m^h(\boldsymbol{x})$ at the current time point t^j is then given by its value at the departure point $m^h(\boldsymbol{X}_\star)$, which is interpolated from computed grid point values of m^h at the previous time point t^{j-1}. The grid of departure points is illustrated in light blue, while the original is given in gray (left figure). Figure modified from [Man17a].*

These approaches are conditionally stable and suffer from possibly small time step limitations due to the CFL number restriction[18]. Since our solver requires the storage of time histories of the state and/or adjoint variable, using CFL restricted schemes is prohibitive (due to a possibly very small time step length). Thus, we opt for unconditionally stable schemes[19] allowing for large time steps[20] and thereby significantly reduce the memory footprint.

While Eulerian schemes suffer from the CFL limitation, unconditionally stable, pure Lagrangian schemes [Man17b] use a non-regular grid which is deformed over time and eventually may become highly irregular; as a result, we get potentially poor approximation quality of the solution due to non-uniform deformations. To combine the best of both worlds, we use a hybrid between Eulerian and Lagrangian, i.e., a high order, unconditionally stable semi-Lagrangian scheme, to solve the hyperbolic transport equations that occur in the registration problem. We consider the equation of the form $\partial_t m + \nabla m \cdot \boldsymbol{v} = f(m, \boldsymbol{v})$. The time-stepping for the advection term is based on an approximation of the total time derivative based on backward trajectories (characteristics) in time:

$$m(\boldsymbol{x}, t_j) = m(\boldsymbol{X}_\star, t_j - \delta t), \quad \text{with } \boldsymbol{X}_\star \approx \boldsymbol{x} - \int_{t_j - \delta t}^{t_j} \boldsymbol{v} dt.$$

As opposed to the Lagrangian scheme, semi-Lagrangian schemes trace back characteristics only for the past time step and revert to a uniform (Eulerian) grid via interpolation. For every time step t_j, two basic steps are involved: (i) We need to compute the departure point $\boldsymbol{X}_\star = \boldsymbol{X}(t_j - \delta t)$ of a transported particle, by solving the characteristic equation

$$\partial_t \boldsymbol{X}(t) + \boldsymbol{v} \cdot \nabla \boldsymbol{X}(t) = \boldsymbol{v}(\boldsymbol{X}(t)) \ \text{ in } (t_{j-1}, t_j], \quad \text{with } \boldsymbol{X}(t_j) = \boldsymbol{x} \tag{5.36}$$

for $\boldsymbol{X} \colon [t_{j-1}, t_j] \to \mathbf{R}^d$ backward in time. We use a second order Runge-Kutta scheme to solve (5.36). This requires evaluation of $\boldsymbol{v}$ at off-grid locations at Runge-Kutta stages, i.e., we have to interpolate $\boldsymbol{v}$ in space. (ii) In a second step, we need to compute the transported quantity m along the characteristic

[18] For conditionally stable time-integration schemes, the CFL condition defines an upper bound on the time step size to guarantee a stable solution. It is defined as $C = \frac{v\delta t}{\delta x} \leqslant 1$, which for high spatial resolutions severely limits the maximum time step size δt allowed to still guarantee a stable time-integration scheme.

[19] Examples include Lax-Friedrich [Ben11], Lagrangian [Man17b] and semi-Lagrangian [Beg05; Che11b; Man17a; Man16b] schemes.

[20] The time step is chosen based on accuracy reasoning, rather than stability requirements.

$\boldsymbol{X}(t)$, i.e., we have to solve an equation of the form

$$\partial_t m(\boldsymbol{X}(t), t) + \boldsymbol{v} \cdot \nabla m(\boldsymbol{X}(t), t) = f(m(\boldsymbol{X}(t), t), \boldsymbol{v}(\boldsymbol{X}(t))) \text{ in } [t_{j-1}, t_j] \tag{5.37}$$

along the characteristic $\boldsymbol{X}(t)$. This is done using a second order Runge-Kutta scheme. Once again, this involves evaluations of $\boldsymbol{v}$ and m at off-grid locations and interpolation is required. The method is outlined in Fig. 5.8. The accuracy of the semi-Lagrangian method strongly depends on the employed time integration scheme and the quality of the interpolation method. For the latter, we use a cubic spline interpolation model; details are given in the sequel. For a more detailed description of the utilized semi-Lagrangian solver, the time-integration, interpolation, and solution of transport equations with time-dependent velocities, we refer to [Man17a].

Computational Kernels

The image registration solver as well as the tumor inversion solver are both based on two main computational kernels; (*i*) the FFT which is used for all spatial differential operators in our spectral approach, and (*ii*) the cubic interpolation which is required to compute off-grid values of velocity and advected quantities in the semi-lagrangian scheme. From the numerical analysis in §5.3.4 we will see, that about 90% of the compute time is spent in FFT or interpolation routines. As a result, scalability and performance of CLAIRE and SIBIA is mainly inherited from these kernels. In the sequel, we briefly discuss implementation and optimization for each of the kernels. For a more thorough description of the specific optimization and implementation details, we refer to [Man16b; Gho17c]. For all linear algebra operations, PETSc [Bal16a] is used, and its toolbox TAO [Mun14] is used for the non-linear optimization. The 3D FFTs are executed in parallel using the library AccFFT [Gho17b].

Data Distribution and 3D Fast-Fourier Transform (AccFFT). We employ a spectral approach, which means that all spatial differential operators such as gradient, laplacian, or divergence are evaluated in the frequency space. This involves two Fast Fourier Transforms (FFT) and an Hadamard product in the frequency domain: A forward FFT is used to map from spatial into frequency domain; there, the differential operator is computed by applying an Hadamard transformation, and the result is mapped back to the spatial domain using an inverse FFT, i.e., to compute a x-derivative of a scalar field f, we have to compute

$$\partial_x f(\boldsymbol{x}) = \mathcal{F}_x^{-1}(-i\omega_x \mathcal{F}_x(f(\boldsymbol{x}))), \tag{5.38}$$

where $\mathcal{F}_x$ is the 1D FFT transform in x-direction. Large scale simulations and 3 dimensional imaging data urge for a distributed memory design. The data distribution among processors is defined by a ***pencil decomposition*** [Gra03] for 3D FFTs (see Fig. 5.9). For our three dimensional imaging data with $n_i, i = 1, 2, 3$ grid points per spatial dimension, this results in a distribution such that each of the $p = p_1 p_2$ MPI tasks, gets a portion $^{n_1}/_{p_1} \times {}^{n_2}/_{p_2} \times n_3$ grid points. To compute the 3D FFTs, we use the open-source library AccFFT [Gho17b; A15] which is based on 1D FFT calls from the FFTW [Fri17; Fri05] package. After a 1D FFT in the first coordinate, which due to our data partitioning can be computed in parallel without communication, we perform a 2D FFT in the second and third coordinate. Each of these subsequent FFTs requires a data re-distribution, involving $\sqrt{p}$ concurrent `alltoallv` communications between groups of $\sqrt{p}$ MPI tasks; details can be found in [A15; Gra03]. AccFFT supports CPU and GPU implementation for parallel 3D FFTs in double and single precision,

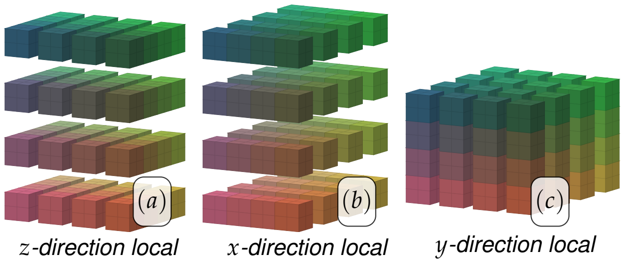


FIGURE 5.9
Parallel data partitioning, determined by the pencil decomposition [Gra03] for 3D FFTs. Each color represents the data distributed to an MPI task. The initial data distribution of a volumetric input image is given in (a); the z-direction is not distributed and owned locally. To perform FFT in x- and y-direction, a re-distribution of data, requiring $\sqrt{p}$ all-to-all communications of groups of $\sqrt{p}$ MPI tasks, is necessary. This is illustrated in figures (b) and (c), respectively. Figure modified from [Man16b; A15].

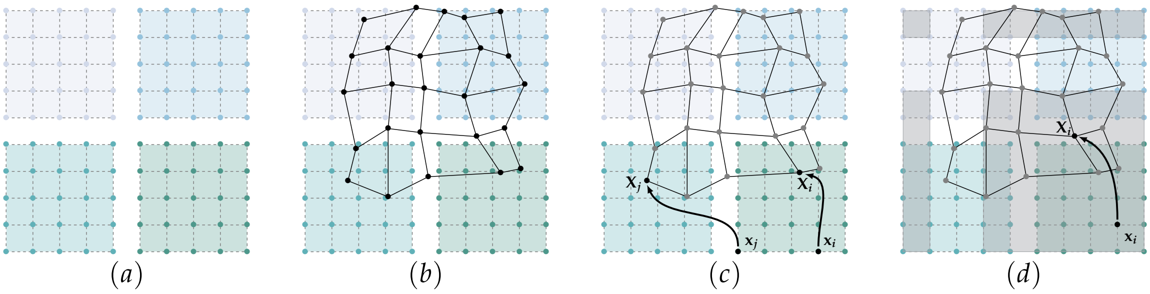


FIGURE 5.10 *Illustration of the data distribution and evaluation of the parallel interpolation kernel. The (2D) domain is decomposed into a 2×2 grid (i.e., we use $p_i = 2$); figure (a). The deformed grid corresponding to the domain of the bottom right processor, is overlaid; figure (b). In our parallel scheme, we encounter three different cases, illustrated in figure (c). The values to be interpolated, as given by the departure point $\boldsymbol{X}_{\star,i}$, are (i) owned by the processor (**owner** and **worker** MPI tasks are identical), (ii) owned by another processor (**owner** and **worker** MPI tasks are different), and (iii) not owned by a processor (i.e., the departure point is close to or on the boundary of two MPI processors; illustrated in figure (d)). The last two cases involve communication. For case one, we communicate the coordinates to the worker (three values), evaluate the interpolation kernel on the worker, and communicate the result back (one scalar value) to the owner. To handle the latter case, we introduce ghost points (figure (d); shaded in gray). Figure provided by Andreas Mang.*

and features optimization for gradient and divergence operators by avoiding unnecessary 3D FFTs and global transpose operations. As a result, the optimized algorithm to compute the gradient or divergence of a function reduces the communication volume by a factor of two (only 4 global transpose operations are required instead of 8) [Gho17c].

Parallel Semi-Lagrangian and Cubic Interpolation. The semi-Lagrangian scheme to solve the advection equation for the registration problem, requires interpolation of velocity and image data points at off-grid locations. We consider tricubic Lagrange-interpolation. The value of a scalar or vector field m at off-grid location (x, y, z) is computed as

$$(5.39) \qquad m(x,y,z) = \sum_{i=0}^{3}\sum_{j=0}^{3}\sum_{k=0}^{3} m_{ijk}\ell_i(x)\ell_j(y)\ell_k(z), \quad \text{with} \quad \ell_i(x) = \prod_{n=0,n\neq i}^{3} \frac{x - x_n}{x_i - x_n}$$

where m_{ijk} is the function value at grid point (ih_x, jh_y, kh_z), the sampling points for interpolation, and ℓ_i is the ith-Lagrange basis polynomial. The off-grid interpolation points are determined by back tracing fluid particle characteristics, computed from (5.36). Considering the parallel, distributed memory layout defined by the FFT pencil decomposition (Fig. 5.9), the corresponding departure point $\boldsymbol{X}_{i,\star}$ for a grid point $\boldsymbol{x}_i \in \Omega_{P_j}$ on MPI task P_i does not necessarily need to lie on the same MPI task, i.e., $\boldsymbol{X}_{i,\star} \in \Omega_{P_k}, j \neq k$. We encounter three different cases. An illustration is given in Fig. 5.10. The values to be interpolated as given by the departure point can either (i) be owned (i.e., reside on) locally, that means the owner and worker (i.e., the task performing the interpolation) MPI task are identical, or (*ii*) they can be owned by a different processor, i.e., owner and worker MPI task are different; lastly they can be (*iii*) not owned by any processor. This occurs if the departure point is close to or on the boundary between sub-domains of processors (note that the partitioning is non-overlapping). The last two cases require costly communication, which affects the scalability of the interpolation kernel [Man16b; Gho17c]. Prior to the computation of the transport equation (5.37), in the so called *scatter phase*, we need to send all points along the backwards characteristic that happen to lie in another processors domain to the corresponding MPI rank. An interpolation planner distributes this information via a sparse point-to-point `alltoallv` communication. The interpolation kernel has been further optimized [Gho17c] to significantly reduce cache misses by re-ordering the processing of departure points using a binning technique to group them by spatial proximity. Further optimization include AVX vectorization of the interpolation kernel and OpenMP support. Scalability results are presented in §5.3.4

5.3.4 Numerical Results: Optimizing Parallel Scalability and Efficiency

Recent advances in medical imaging lead to data sets with increasing resolution. Seven Tesla MRI scanners can reach a resolution of up to 0.5 mm which corresponds to $\approx 450^3$ voxels [Lüs13]; ultra high resolution CT scanners even capture a resolution of 0.25 mm with approximately 512^3 voxels [Kak15]. This translates to a drastic increase in memory consumption and high computational complexity. Consequently, distributed memory algorithms become inevitable and the critical time-to-solution constraint in a clinical application context urges for good strong-scaling efficiency. Highly optimized and scalable solvers allow for patient-specific simulations in the clinical setting; massively-parallel execution reduces the computation time to the scale of minutes, as opposed to days or weeks. Besides that, in biological imaging, biophysics and neuro-science, extreme high $\mathcal{O}(\mu\mathrm{m})$ resolutions ($\approx 2000^3$) voxels) of animal Micro-CT [Sta15], raise the need for good weak-scaling efficiency. A novel optical imaging technique in small animal neuro-imaging, called CLARITY [Tom14], delivers sub-micron resolution with images of $10 - 100$ billion voxels. Also in human clinical imaging, we observe a trend towards higher-resolution data in order to capture detailed brain (and other tissue) structure, which is decisive in tumor progression. All this manifests the need for HPC computing in medical image analysis and data assimilation software.

In this section, we present efficiency and scalability results of the presented sub-components, tumor inversion solver for biophysical model calibration and diffeomorphic image registration. We target strong-scaling performance on up to 16 thousand cores, and showcase results which demonstrate that our solver can be used to solve registration problems of unprecedented scale, $n = 4096^3$ resulting in approx 200 billion unknown—a problem size that is 64× larger than the state-of-the-art. The presented results have been previously published in [Gho17c], and are the result of joint work with my collaborators Andreas Mang, Amir Gholami, George Biros and Miriam Mehl. The scalability and parallel efficiency of the biophysical inversion solver is primarily own contribution and achieved within this thesis. We therefore focus on the discussion of the tumor inversion solver, but give results for the registration solver as well, for the sake of completeness. The optimization of the computational kernels 3D FFT and parallel interpolation kernel is not part of this thesis, and details can be found in [Gho17c].

Implementation, Hardware and Data. SIBIA and its components are written in C++ and use MPI for parallelism. PETSc [Bal16a] is used for linear algebra operations and its toolbox TAO [Mun14] for non-linear optimization. The library AccFFT [Gho17b] is used for computation of parallel 3D Fourier Transforms and PnetCDF [Uni] for parallel I/O. We benchmark the tumor inversion solver on the

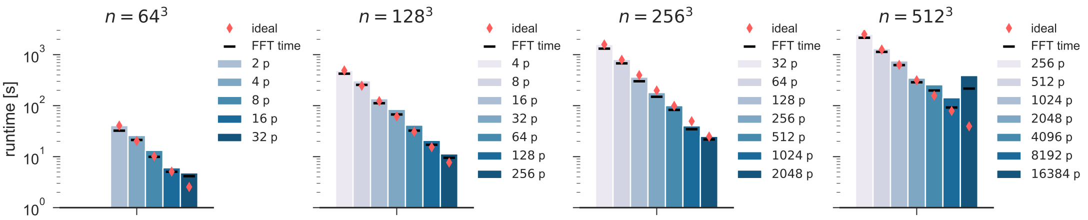


FIGURE 5.11 *Strong- and weak-scaling performance for the tumor inversion solver. We illustrate timings from Tab. 5.2, i.e., time to solution and time spent in the FFT kernel (in seconds), as a function of the number of unknowns (in space) and the number of tasks; modified from [Gho17c]*

Tier-1 supercomputer HazelHen at the High Performance Computing Center HLRS in Stuttgart[21], a Cray XC40 system with 7,712 noes with Xeon E5-269 v3 processors and 24 cores on two sockets per node, resulting in a peak performance of 7.42 Petaflops. Each node has 128 GB of memory and they are connected with an Aries interconnect. Additionally, we use the Lonestar5 (same nodes as HazelHen, but only 64 GB memory per node) HPC system of the Texas Advanced Computing Center (TACC)[22] to benchmark the registration solver.

For the tumor inversion runs, we use medical brain imaging data obtained from the University of Pennsylvania with a spatial resolution of $n = 256^3$. For weak- and strong-scaling experiments, we rescale the data to coarser or finer resolutions $n \in \{64^3, 128^3, 256^3, 512^3\}$ using cubic interpolation and band-limit the data by applying a Gaussian smoothing operator with spatial bandwidth $\sigma = 1/n_i$. For the registration runs, we use the open-access data repository ***'Non-rigid Registration Evaluation Project (NIREP)'*** [Chr06] with an original resolution of $(n_1, n_2, n_3) = (256, 300, 256)$. We apply zero padding and resampling for different resolutions.

Parallel Performance for the Biophysical Inversion Solver

Setup. We present strong- and weak-scaling results for the tumor inversion solver on HazelHen. We report the overall runtime, i.e., the time spent in the inversion, referred to as time-to-solution, and analyze its portions spent in the main computational kernels, subdivided into execution and communication time. We outline the time spent in the diffusion PCG solver, the Hessian matvec, and the 3D FFT kernel. All timings reflect the maximum time across all MPI tasks. For all runs, we use 12 MPI tasks per node. We benchmark the inversion solver for a regularization parameter of $\beta_p = 1\text{E}{-3}$ using a Gauß-Newton-Krylov solver for optimization. We limit the number of Newton iterations to three, and the number of Hessian matvecs per outer iteration to three (inner Krylov solver). We solve the diffusion operator up to an absolute tolerance of $1\text{E}{-6}$ for forward and adjoint, using a preconditioned conjugate gradient (PCG) method and $n_t = 4$ time steps for the numerical time-integration scheme. This setup results in an overall gradient reduction of one order of magnitude and 13% relative data mismatch for the predicted tumor at physical time $t_T = 1$ and the observed data. As data, we use a synthetically grown tumor, obtained by our forward tumor solver with $\rho_f = 15, k_f = 1\text{E}{-2}$.

Observations. We report strong- and weak-scaling results in the regime of $n \in \{64^4, 128^3, 256^3, 512^3\}$ numbers of unknowns in space and parallel execution on $p \in \{2^1, \ldots, 2^{14}\}$ MPI tasks. Timings for the overall time-to-solution for the inversion and a detailed breakdown into its main computational

[21] https://www.hlrs.de
[22] https://www.tacc.utexas.edu

kernels is given in Tab. 5.2. Across all runs, the 3D FFT kernel, which computes all spatial differential operators, consumes about 82% of the overall computation time. As a result, the parallel scalability of the tumor inversion solver is primarily inherited from the parallel performance of the 3D FFTs, in our case the scalability of the AccFFT library. Further, nearly all the computational time (98%) is spent in the PCG method to solve the diffusion part of the employed Strang-splitting method (we use an analytic solution for the reaction part; all other parts in the code have vanishing computational costs). About 65% of the runtime goes to the Hessian matvec and inner Krylov solve to compute the Newton step; the remaining percentage is spent in gradient and objective function evaluations.

We observe excellent ***strong-scaling*** results for $n = 256^3$ unknowns in space, with a strong-scaling efficiency of 98%, going from 32 to 2048 MPI tasks (runs #13 – #19). Similar observations can be drawn from the strong scaling series for $n = 128^3$ and $n = 64^3$ unknowns in space, yielding slightly worse but still acceptable parallel efficiency rates of 67% and 52%, going from 4 to 256, and 2 to 32 MPI tasks, respectively. We illustrate the strong-scaling performance in Fig. 5.11 and contrast the measured runtime to the ideal runtime, assuming 100% parallel efficiency (red diamonds). Considering the $n = 512^3$ runs, that represent a high level of parallelism going up to a total of 16384 MPI tasks, we observe a degradation of parallel efficiency beyond 2048 parallel tasks. Our analysis showed that this degradation in performance is due to an increasing communication overhead caused by MPI routines. Not least this is supported by the fact that the time consumed by the diffusion solver drops to 50%, as opposed to 98% before. By experimentally increasing the MPI buffer and the maximum message size for the MPI eager protocol, the performance of run #25 could be increased by approximately 30% (run #26). This indicates that the performance degradation is mainly due to non-optimal MPI settings and might be resolved with careful optimization of the HPC environment, or introducing hybrid parallelism with OpenMP.

We further analyze ***algorithmic*** and ***parallel weak-scaling*** efficiency, indicated as eff_W and $(\widetilde{\text{eff}}_W)$, respectively, in Tab. 5.2. We consider a setup, where the number of unknowns in space as well as the number of parallel MPI tasks is increased by a factor of eight in between weak-scaling levels; the series of runs #3, #10, #17, #24, #26 is exemplarily highlighted in Tab. 5.2. We observe an algorithmic scaling efficiency of 30% increasing the spatial resolution and number of MPI tasks by a factor of eight. As mentioned earlier, the utilized preconditioner requires the diffusion coefficient to be fully resolved spectrally in order to yield a mesh-independent condition number. For the considered data set, this is not the case, and, consequently the number of PCG iterations required to solve the diffusion equation increases from 10 iterations (on average) per diffusion solve on $n = 64^3$ to around 50 iterations per diffusion solve on a $n = 1024^3$ grid. Exact numbers are shown in Tab. 5.2. Fixing the number of allowed PCG iterations, results in a parallel weak-scaling efficiency of about 45%, increasing unknowns and MPI tasks by a factor of eight. We observe a perfect correlation of the FFT weak-scaling efficiency with the overall weak-scaling efficiency of the tumor inversion solver. For a increasing number of MPI tasks, the FFT becomes more and more dominated by the communication time, and as a result the parallel efficiency deteriorates. If we only use one MPI task per socket, i.e., two MPI tasks per node, the parallel weak-scaling efficiency increases to 80% going from $n = 64^3$ to $n = 256^3$. These runs, exploiting maximal communication bandwidth, are reported as runs #29-#32 in Tab. 5.2. Note, that there is always an inherent overhead for memory allocation and increased communication time due to larger message sizes when increasing the number of unknowns.

Conclusion *For the biophysical inversion solver, we obtain almost optimal scalability results up to* 2048 *MPI tasks with a parallel efficiency, ranging between* 60% *and* 100%*. Beyond* 2048 *MPI tasks, we encounter a fall off of parallel efficiency due to increased communication overhead of MPI routines. Optimizing the MPI environment and settings or hybrid parallelism with OpenMP are expected to solve this problem. With respect weak-scaling efficiency, we are bound by the weak-scaling efficiency of the underlying AccFFT library, which consumes about* 80% *of the overall compute time. Computing the 3D FFTs becomes increasingly expensive for larger numbers of unknowns due to increasing communication costs (global transpose and* `alltoallv` *operations).*
Without limiting the number of Newton and inner Krylov iterations, we solve the inverse tumor problem on clinical brain data with a resolution of $n = 256^3$ *up to a relative gradient of* 1.20E−4 *and a resulting final data mismatch of* 0.2% *in* 22 *minutes using* 512 *MPI tasks of* `HazelHen`*. The objective function is reduced by three orders of magnitude.*

Parallel Performance for the Image Registration Solver

Setup. We consider two sets of test-cases, clinical real brain MR data from the open access repository ***'Non-rigid Registration Evaluation Project (NIREP)'*** [Chr06], and smooth, synthetic images. For the latter, we set the template image as $m_T(\boldsymbol{x}) = (sin(x_1)^2 + sin(x_2)^2 + sin(x_3)^2)/3$, and generate a reference image via advection with the smooth velocity $\boldsymbol{v}(\boldsymbol{x}) = (v_1(\boldsymbol{x}), v_2(\boldsymbol{x}), v_3(\boldsymbol{x}))^T$ with $v_1(\boldsymbol{x}) = sin(x_3)cos(x_2)sin(x_2)$, $v_2(\boldsymbol{x}) = sin(x_3)cos(x_3)sin(x_3)$, $v_3(\boldsymbol{x}) = sin(x_2)cos(x_1)sin(x_1)$. We fix a number of parameters for the conducted experiments. We use a H^1-regularization with a penalty of $\beta_w = 1\text{E}{-4}$ on the divergence of $\boldsymbol{v}$ to control volume change, and experimentally determined the regularization parameter $\beta_v = 1.00\text{E}{-2}$. For the semi-Lagrangian, 4 time steps are used. The number of Newton and inner Krylov iterations are limited to 3 and 5 for the real data input images, and to 5 and 10 for the smooth synthetic images, respectively. We perform strong-scaling series for 4 resolution levels and resample the data accordingly, i.e., $\kappa_\ell \cdot (256, 300, 256), \kappa_\ell \in \{1/4, 1/2, 1, 2\}$. We report timings (in seconds) as maximum across all MPI tasks for (*i*) the overall runtime, (*ii*) the accumulated time spent in spectral operations, $\text{FFT}_{\text{total}}$, (*iii*) the FFT communication time FFT_{comm}, (*iv*) the accumulated time spent in the interpolation, $\text{INT}_{\text{total}}$, with (*v*) the respective time spent in interpolation kernel operations only $\text{FFT}_{\text{kernel}}$, and (*vi*) the portion of time spent in communication FFT_{comm}.

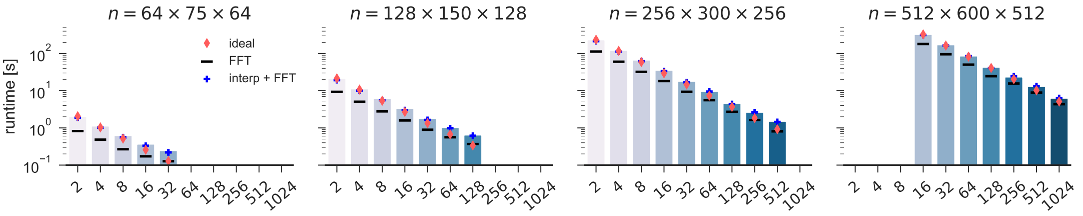


FIGURE 5.12 *Strong- and weak-scaling performance for diffeomorphic image registration. Illustrated results correspond to timings in Tab. 5.3. The time to solution and the time spent in the computational kernels (summarized) is shown (in sec.) as a function of the number of unknowns (in space), and the number of MPI tasks; modified from [Gho17c]*

Observations. Scalability results for the diffeomorphic registration on `Lonestar5` are reported in Tab. 5.3 and Fig. 5.12. More than 90% of the overall runtime is spent in the main computational kernels, FFT and interpolation for the semi-Lagrangian. Similar as for the tumor inversion solver, the

TABLE 5.1 *Computational performance of the tumor inversion solver for real data, on* `HazelHen`*. We report the time-to-solution and its portion spent in the Hessian matvec, 3D FFT kernel, and diffusion PCG solver, respectively. We give absolute timings in seconds and percentage for a strong- and weak-scaling setting as a function of number of unknowns and parallel MPI tasks. We show the strong-scaling efficiency eff_S and the algorithmic weak-scaling efficiency eff_W. To demonstrate the effective weak-scaling efficiency $\widetilde{eff}_W$, we fix the number of PCG iterations for the diffusion solver for all resolutions. We perform 3 Gauß-Newton iterations with 3 Hessian matvecs for the inner Krylov solver each. For runs marked with † we observed higher runtimes due to MPI problems arising from non-optimal settings. Improved performance was observed, increasing the MPI buffer and the maximum message size for the MPI eager protocol (run marked ‡). Results originally published in [Gho17c]*

	N	nodes	tasks	runtime	eff_S	eff_W	($\widetilde{eff}_W$)	FFT	([%])	H-matvec	([%])	diffusion	([%])
#1	64^3	1	2	4.07E+1	100.0	100.0	(100.0)	3.23E+1	(79.5)	2.57E+1	(63.2)	3.98E+1	(97.8)
#2		1	4	2.60E+1	78.2	100.0	(100.0)	2.12E+1	(81.7)	1.64E+1	(63.3)	2.55E+1	(98.1)
#3		1	8	1.33E+1	76.6	100.0	(100.0)	9.92	(74.7)	8.38	(63.1)	1.30E+1	(97.9)
#4		2	16	6.05	84.0	100.0	(100.0)	5.05	(83.5)	3.82	(63.2)	5.89	(97.4)
#5		3	32	4.81	52.8	100.0	(100.0)	4.14	(86.1)	3.02	(62.8)	4.68	(97.3)
#6	128^3	1	4	4.91E+2	100.0			4.24E+2	(86.3)	3.14E+2	(63.9)	4.87E+2	(99.2)
#7		1	8	3.07E+2	80.0			2.56E+2	(83.2)	1.96E+2	(63.9)	3.04E+2	(99.1)
#8		2	16	1.37E+2	89.6	29.7	(43.7)	1.12E+2	(81.5)	8.77E+1	(64.0)	1.36E+2	(99.2)
#9		2	32	8.46E+1	72.6	30.7	(45.2)	6.72E+1	(79.5)	5.41E+1	(63.9)	8.38E+1	(99.1)
#10		6	64	4.15E+1	73.9	32.0	(46.8)	3.24E+1	(78.1)	2.65E+1	(63.9)	4.11E+1	(99.0)
#11		11	128	2.08E+1	73.7	29.0	(42.0)	1.71E+1	(82.2)	1.33E+1	(63.8)	2.06E+1	(98.7)
#12		22	256	1.13E+1	67.7	42.5	(60.9)	9.49	(83.7)	7.20	(63.5)	1.11E+1	(98.0)
#13	256^3	3	32	1.58E+3	100.0			1.31E+3	(82.9)	1.02E+3	(64.4)	1.57E+3	(99.1)
#14		6	64	8.06E+2	98.2			6.75E+2	(83.7)	5.19E+2	(64.4)	7.99E+2	(99.1)
#15		11	128	3.63E+2	109.0	11.2	(22.9)	3.01E+2	(82.9)	2.35E+2	(64.7)	3.61E+2	(99.5)
#16		22	256	1.81E+2	109.6	14.4	(29.3)	1.49E+2	(82.6)	1.16E+2	(64.4)	1.80E+2	(99.5)
#17		43	512	1.00E+2	98.9	13.3	(26.6)	8.26E+1	(82.5)	6.45E+1	(64.4)	9.95E+1	(99.3)
#18		86	1024	4.03E+1	123.0	15.0	(31.0)	3.45E+1	(85.7)	2.55E+1	(63.4)	3.97E+1	(98.6)
#19		172	2048	2.50E+1	98.9	19.2	(38.5)	2.17E+1	(86.9)	1.59E+1	(63.5)	2.38E+1	(95.0)
#20	512^3	22	256	2.52E+3	100.0			2.15E+3	(85.2)	1.65E+3	(65.5)	2.50E+3	(99.4)
#21		43	512	1.32E+3	95.8			1.13E+3	(86.2)	8.61E+2	(65.5)	1.31E+3	(99.4)
#22		86	1024	7.54E+2	83.5	5.4	(13.6)	6.24E+2	(82.8)	4.93E+2	(65.4)	7.52E+2	(99.7)
#23		172	2048	3.47E+2	90.7	7.5	(19.0)	2.87E+2	(82.8)	2.32E+2	(66.9)	3.45E+2	(99.4)
#24		342	4096	2.57E+2	61.2	5.2	(11.6)	1.98E+2	(77.1)	1.66E+2	(64.6)	2.50E+2	(97.3)
#25†		683	8192	2.00E+2	39.3	3.0	(5.6)	1.30E+2	(65.1)	1.32E+2	(65.9)	1.71E+2	(85.5)
#26‡		683	8192	1.43E+2	55.1	4.2	(7.8)	9.20E+1	(64.4)	8.96E+1	(62.7)	1.15E+2	(80.7)
#27†		1366	16384	3.89E+2	10.1	1.2	(2.0)	2.16E+2	(55.6)	2.34E+2	(60.2)	2.69E+2	(69.2)
#28	1024^3	2732	32768	1.61E+3	100.0	0.8	(1.2)	6.58E+2	(40.9)	9.25E+2	(57.5)	8.19E+2	(50.8)
#29	64^3	2	4	2.20E+1		100.0	(100.0)	1.85E+1	(84.2)	1.40E+1	(63.5)	2.15E+1	(97.7)
#30	128^3	16	32	4.00E+1		55.0	(80.7)	3.49E+1	(87.2)	2.56E+1	(64.1)	3.94E+1	(98.6)
#31	256^3	128	256	6.96E+1		31.6	(64.8)	6.05E+1	(87.0)	4.52E+1	(65.0)	6.90E+1	(99.1)
#32	512^3	1024	2048	1.39E+2		15.9	(37.5)	1.14E+2	(82.3)	8.82E+1	(63.7)	1.36E+2	(98.2)

TABLE 5.2 *Number of PCG iteration to solve the diffusion problem up to an absolute tolerance of $10^{-6}6$ for different spatial resolutions along with the relative gradient and relative tumor mismatch after 3 Gauß-Newton with a maximum of 3 PCG iterations for the inversion of the Hessian.*

N^3	#its diffusion	rel. gradient	rel. mismatch
64^3	2599	4.20E−2	1.34E−1
128^3	4088	4.52E−2	1.38E−1
256^3	6110	4.63E−2	1.40E−1
512^3	8512	4.45E−2	1.43E−1
1024^3	13929	4.64E−2	1.41E−1

main computational time goes to the PDE solves, i.e., solving the transport equation (about 87% of the overall runtime). 4 of these PDE solves are done in gradient and objective function evaluation, respectively, and twelve PDE solves are carried out in the Hessian matvec routine.

For clinical relevant problem sizes, we observe excellent ***strong-scaling*** results with parallel efficiencies between 52.1% for run #5 and 80% for run #28; compare also Fig. 5.12. In these regimes, the FFT and interpolation kernel exhibit perfect scalability. Note, that the maximum degree of parallelism for the considered implementation is limited by the support of the interpolation kernel; for cubic interpolation a minimum of $3 \times 3 \times n_3$ points per MPI tasks are required. For the original resolution of the NIREP data we reduce the gradient by a relative factor of 3.17E−1 and the final data mismatch between the transported template image and the observed reference image by a factor of 1.89E−1 for the reported runs[23].

For the real brain NIREP data, we achieve a ***weak-scaling*** efficiency of 62.7%, 44.6% and 32.6% when comparing the runs #1, #9, and #19 through #28 in Tab. 5.3. Again we observe an increase of the FFT time, due to rising communication costs, as we approach larger scales (FFT is communication bound). This has negative effects on the weak-scaling performance; for run #28, e.g., 80% of the total FFT time is spent in communication. The interpolation on the other hand shows very good weak-scaling efficiency of about 50%, comparing runs #1 and #28. For the synthetic large-scale runs, we observe an excellent overall weak-scaling efficiency of 82%, across 128 MPI tasks for 1024^3 resolution (run #29) and 8192 MPI tasks for 4096^3 resolution (run #34).

Conclusion *For the diffeomorphic image registration, we solve an inverse problem with input images of* 4096^3 *resolution, resulting in* ≈ 200 *billion unknowns (only considering the unknowns in the velocity field and neglecting state and adjoint fields), a problem size that is* $64\times$ *larger than the state of the art. We obtain excellent strong-scaling performance which is critical for a feasible time-to-solution for 3D registration problems in a clinical application setting. We obtain a parallel strong scaling efficiency between* 52% *and* 80%. *The negative effects of the degrading FFT scalability are not as pronounced for the registration solver, as the overall computational time is roughly balanced between spectral operations and interpolation kernel. Overall this allows for application of the registration solver to high resolution images, arising e.g. from ultra high resolution* *CLARITY* *data [Kut16; Tom14; Kut17].*

[23] Note, that for real brain data sets and inter-subject registration, the data misfit cannot be reduced to zero due to incompatible topology and intensity variations.

TABLE 5.3 *Computational performance for the distributed-memory algorithm for diffeomorphic image registration, performed on TACC's* **Lonestar 5** *(runs #1 to #28; NIREP data-sets) and HLRS'* **Hazel Hen** *for the large-scale runs. (SYN data-sets). We use 12 MPI tasks per node. We set the upper limit for the Gauss–Newton iterations to three/five and the number of PCG iterations to five/ten for NIREP/SYN. We report (from left to right) the total time spent in the inversion (runtime), the strong-scaling efficiency, and the time spent in the computational kernels (spectral operations/FFT and interpolation), respectively (in seconds) as a function of the number of unknowns N (in space), and the number of nodes and tasks. All timings are accumulated. We report the max. value across all MPI tasks; Here FFT$_{total}$ corresponds to the time spent in all spectral operations; FFT$_{comm}$ is the communication time; INT$_{total}$ is the overall time spent in the interpolation; FFT$_{kernel}$ is the time spent on the execution of the interpolation operator, and FFT$_{comm}$ is the communication time for the interpolation. We also report the strong-scaling efficiency and the percentage of the total interpolation and FFT time with respect the overall runtime. Results originally published in [Gho17c]*

	N	nodes	tasks	runtime	eff.	FFT$_{\text{total}}$	([%])	FFT$_{\text{comm}}$	INT$_{\text{total}}$	([%])	INT$_{\text{kernel}}$	INT$_{\text{comm}}$
#1	(64, 75, 64)	1	2	2.07	100.0	8.21E−1	(39.7)	9.31E−2	1.05	(51.0)	7.18E−1	6.71E−2
#2		1	4	1.13	91.4	4.83E−1	(42.7)	7.36E−2	5.30E−1	(46.8)	3.70E−1	3.68E−2
#3		1	8	6.26E−1	82.6	2.67E−1	(42.7)	4.44E−2	2.87E−1	(45.9)	1.95E−1	1.78E−2
#4		2	16	3.70E−1	69.9	1.72E−1	(46.6)	6.88E−2	1.59E−1	(43.1)	9.84E−2	9.95E−3
#5		3	32	2.47E−1	52.3	1.27E−1	(51.3)	7.51E−2	9.28E−2	(37.5)	5.00E−2	5.54E−3
#6	(128, 150, 128)	1	2	2.15E+1	100.0	9.35	(43.4)	5.07E−1	9.91	(46.0)	6.83	8.22E−1
#7		1	4	1.13E+1	95.4	5.06	(44.9)	6.73E−1	5.05	(44.7)	3.42	4.28E−1
#8		1	8	6.19	86.9	2.79	(45.0)	3.89E−1	2.78	(44.9)	1.82	2.32E−1
#9		2	16	3.30	81.6	1.59	(48.2)	5.52E−1	1.44	(43.6)	9.24E−1	1.14E−1
#10		3	32	1.79	75.2	8.94E−1	(50.0)	3.86E−1	7.50E−1	(41.9)	4.63E−1	5.93E−2
#11		6	64	1.04	65.0	5.61E−1	(54.2)	3.68E−1	4.22E−1	(40.8)	2.32E−1	3.05E−2
#12		11	128	6.49E−1	51.8	3.69E−1	(56.9)	2.64E−1	2.47E−1	(38.1)	1.22E−1	1.90E−2
#13	(256, 300, 256)	1	2	2.36E+2	100.0	1.13E+2	(48.0)	6.40	1.02E+2	(43.4)	5.88E+1	1.05E+1
#14		1	4	1.22E+2	96.4	6.04E+1	(49.3)	8.33	5.25E+1	(42.9)	2.95E+1	5.51
#15		1	8	6.73E+1	87.6	3.23E+1	(48.0)	4.58	2.99E+1	(44.5)	1.56E+1	3.95
#16		2	16	3.59E+1	82.1	1.82E+1	(50.7)	7.80	1.55E+1	(43.2)	7.82	2.69
#17		3	32	1.81E+1	81.4	9.38	(51.8)	3.60	7.48	(41.3)	3.92	1.18
#18		6	64	9.70	76.0	5.57	(57.5)	3.30	3.93	(40.5)	1.99	7.07E−1
#19		11	128	4.63	79.5	2.72	(58.6)	1.50	1.69	(36.5)	9.98E−1	1.49E−1
#20		22	256	2.66	69.2	1.63	(61.1)	1.21	9.81E−1	(36.9)	4.98E−1	8.26E−2
#21		43	512	1.52	60.5	8.09E−1	(53.1)	6.15E−1	6.43E−1	(42.2)	2.54E−1	6.80E−2
#22	(512, 600, 512)	2	16	3.28E+2	100.0	1.81E+2	(55.0)	4.84E+1	1.35E+2	(41.2)	6.33E+1	2.50E+1
#23		3	32	1.73E+2	94.7	9.57E+1	(55.2)	2.85E+1	6.86E+1	(39.6)	3.17E+1	1.15E+1
#24		6	64	8.66E+1	94.7	5.04E+1	(58.2)	2.42E+1	3.48E+1	(40.2)	1.59E+1	6.31
#25		11	128	4.32E+1	94.9	2.46E+1	(56.9)	1.10E+1	1.63E+1	(37.8)	7.95	2.84
#26		22	256	2.36E+1	87.0	1.56E+1	(66.3)	1.08E+1	8.83	(37.5)	4.03	1.64
#27		43	512	1.31E+1	78.1	8.92	(67.9)	6.56	4.41	(33.6)	2.03	7.27E−1
#28		86	1024	6.35	80.7	4.35	(68.5)	3.52	2.02	(31.9)	1.02	2.08E−1
#29	1024^3	11	128	1.97E+2	100.0	1.20E+2	(60.9)	3.30E+1	6.90E+1	(35.0)	2.35E+1	2.23E+1
#30		22	256	9.88E+1	99.7	6.17E+1	(62.5)	2.15E+1	3.49E+1	(35.4)	1.16E+1	1.16E+1
#31	2048^3	86	1024	2.10E+2	100.0	1.37E+2	(65.0)	4.33E+1	7.21E+1	(34.3)	2.73E+1	2.40E+1
#32		171	2048	1.11E+2	94.8	7.17E+1	(64.7)	2.63E+1	3.64E+1	(32.8)	1.35E+1	1.05E+1
#33	4096^3	342	4096	4.42E+2	100.0	3.22E+2	(72.8)	1.31E+2	1.17E+2	(26.4)	4.20E+1	3.97E+1
#34		684	8192	2.38E+2	93.1	1.73E+2	(72.9)	8.27E+1	6.25E+1	(26.3)	2.10E+1	2.30E+1

6 Coupling Schemes for the Multi-Component Problem

The main contribution and research emphasis of the second part of this thesis is the development of methodologies for effective data-assimilation and image analysis in brain tumor imaging. The formidable and large-scale inverse problems involved within this task ask for highly scalable and robust parallel solvers. An exceptional challenge of our considered application (described in Chapter 5) lies in the necessity of combining inversion in two directions: the identification of point correspondences in two different images of similar objects, and the calibration of biophysical models for brain tumor progression simulations for specific patient data. A meaningful overall solution requires tight coupling of the sub-components. In this chapter, we present two different formulations for the general inversion problem described in Chapter 5. Both methodologies tightly couple image registration and inverse tumor growth simulation, yet target different aspects of brain tumor image analysis.

(1) The ***moving-patient strategy*** (presented in [Sch]; detailed in §6.1) is of particular interest for biophysics aided ***normal-to-abnormal*** image registration and the resulting clinically relevant task of automated segmentation of MR imaging data, as described earlier in Chapter 5. Such techniques are useful for mapping different functional regions of the brain from a labeled/segmented atlas brain to the patient anatomy for, e.g., surgical planning. The scheme allows for a straightforward decomposition into the sub-component solvers tumor inversion and registration (see §6.3.1). It results in good visual reconstruction and small errors for the observed data. However, it falls short in reconstructing meaningful biophysical model parameters if the anatomies of patient and statistical atlas differ significantly; the identified tumor parameters for this scheme "live" in the space of the atlas brain as opposed to the patient brain, and thus the biophysics part primarily aids the registration to find a plausible mapping in spite of topological differences. In addition, the registration also acts on the tumor concentration and can therefore assist to still produce a small tumor data mismatch (in case of an insufficient tumor model) by fitting the data to a potentially poor tumor reconstruction.

(2) With a view towards a more biophysically meaningful scheme that allows for calibration and personalization of brain tumor progression models, we designed a second, more advanced formulation of the problem. This so called ***moving-atlas strategy*** (detailed in §6.2) results in a coupling scheme that gradually improves on the approximation of the usually inaccessible healthy patient brain anatomy. As a result (*a*) the tumor inversion eventually operates in the space of the brain anatomy of the given patient at hand, yielding more reliable estimates for the biophysical parameters, whose clinical significance no longer is a function of the proximity of atlas and patient space; (*b*) the registration operates on the brain anatomy only, which forces the

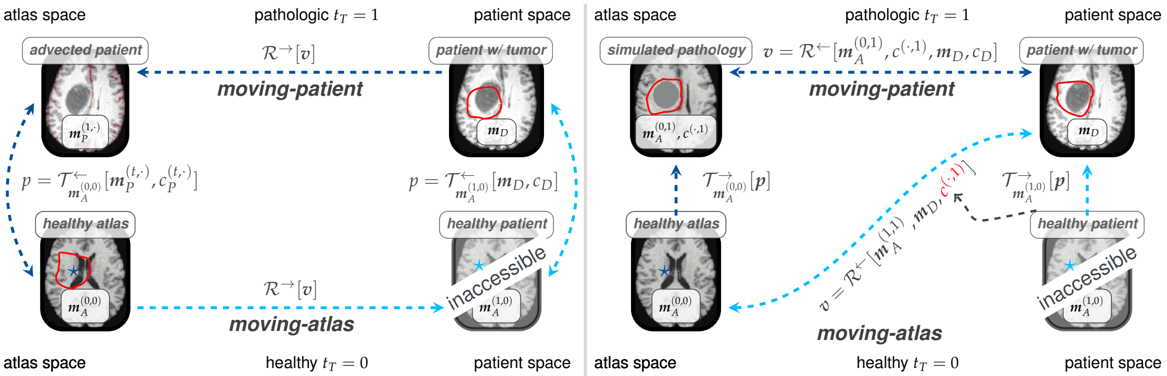

atlas space
pathologic $t_T = 1$
patient space
advected patient
$\mathcal{R}^{\rightarrow}[v]$
patient w/ tumor
moving-patient
$\boldsymbol{m}_P^{(1,\cdot)}$
$\boldsymbol{m}_D$
$p = \mathcal{T}^{\leftarrow}_{\boldsymbol{m}_A^{(0,0)}}[\boldsymbol{m}_P^{(t,\cdot)}, c_P^{(t,\cdot)}]$
$p = \mathcal{T}^{\leftarrow}_{\boldsymbol{m}_A^{(1,0)}}[\boldsymbol{m}_D, c_D]$
healthy atlas
healthy patient
inaccessible
$\boldsymbol{m}_A^{(0,0)}$
$\mathcal{R}^{\rightarrow}[v]$
$\boldsymbol{m}_A^{(1,0)}$
moving-atlas
atlas space
healthy $t_T = 0$
patient space
atlas space
pathologic $t_T = 1$
patient space
simulated pathology
$v = \mathcal{R}^{\leftarrow}[\boldsymbol{m}_A^{(0,1)}, c^{(\cdot,1)}, \boldsymbol{m}_D, c_D]$
patient w/ tumor
moving-patient
$\boldsymbol{m}_A^{(0,1)}, c^{(\cdot,1)}$
$\boldsymbol{m}_D$
$\mathcal{T}^{\rightarrow}_{\boldsymbol{m}_A^{(0,0)}}[p]$
$\mathcal{T}^{\rightarrow}_{\boldsymbol{m}_A^{(1,0)}}[p]$
$v = \mathcal{R}^{\leftarrow}[\boldsymbol{m}_A^{(1,1)}, \boldsymbol{m}_D, c^{(\cdot,1)}]$
healthy atlas
healthy patient
inaccessible
$\boldsymbol{m}_A^{(0,0)}$
$\boldsymbol{m}_A^{(1,0)}$
moving-atlas
atlas space
healthy $t_T = 0$
patient space

FIGURE 6.1 *Schematic view of the two formulations. The* ***moving-patient*** *formulation (dark blue arrows) grows a tumor in a healthy atlas brain and advects the patient image/pathology towards the resulting atlas with tumor; the tumor inversion then operates in the atlas space, i.e., the estimated biophysical parameters* ***live in the atlas space*** *(left figure). The velocity is chosen such that the distance between the simulated tumor in the atlas brain and the advected patient data is minimal (right figure). For* ***moving-atlas*** *(light blue arrows), we advect the healthy atlas brain towards the patient and grow a tumor in this approximation of the healthy brain geometry; the tumor inversion then yields* ***biophysical parameters in the patient space*** *(left figure). For the moving patient, the advection velocity is chosen such that the distance between the atlas brain with simulation of the pathology and the warped patient data is as small as possible (right figure; dark blue arrows). For the moving atlas, we intend to find correspondence relations between the healthy atlas brain and the pathologic patient brain. Due to the topological differences, we require the tumor solver to aid the registration (black arrow) and solve a modified registration formulation for the advection velocity (right figure; light blue arrows). Images modified from [Hog08b].*

tumor solver to produce a biophysically meaningful tumor that matches the observed input data as good as possible.

The two formulations are compared in Fig. 6.1 in two images, combining advection and tumor inversion (left) and registration with forward tumor growth (right); whereas the moving patient formulation utilizes image registration to transport the patient data to its corresponding locations in the atlas space, followed by tumor inversion, the moving-atlas strategy improves an approximation to the healthy patient brain anatomy starting from the healthy atlas. For moving-patient, the velocity used for transportation of brain anatomy and tumor concentration is chosen such that the distance between the simulated tumor in the atlas geometry and the warped-to-atlas-space patient data, is as small as possible (cf. Fig. 6.1, right). For the moving-atlas, a modified registration formulation that incorporates information of the tumor solver will become necessary to allow for registration of the healthy atlas with the pathologic patient data.

For numerical optimization of the two formulations of the multi-component problem, we follow a modular approach, resulting in two different iterative Picard schemes, respectively, that combine separate image registration and tumor inversion sub-components. The schemes can be viewed as block-Newton-type solvers, each of which fulfills a subset of the derived optimality conditions of the respective formulation.

Notation and Data

Before presenting the formulations for the considered multi-component problem, let us summarize, recapitulate and extend the required notation.

Tissue Labels and Probability Maps. Patient and statistical atlas are given by means of separate ***probability maps*** $m_X(\boldsymbol{x}, t) \in \Omega \times [0, 1], X \in \{WM, GM, CSF\}$ for the three primary healthy tissue types, i.e., ***white matter*** (WM), ***gray matter*** (GM), and ***cerebrospinal fluid*** (CSF; which includes the ventricles (VE)), giving the probability to encounter a cell of the respective tissue type at every spatial location $\boldsymbol{x}$. The probability maps are computed from MR image data or segmented MR data in a pre-processing step. We gather the probability maps in a space-time vector field

$$\text{(6.1)} \qquad \boldsymbol{m}(\boldsymbol{x},t) = (m_i(\boldsymbol{x},t))_{i=1,\dots,3} \in \mathbf{R}^3 \;\text{ with }\; m_1 = m_{GM}, m_2 = m_{WM}, m_3 = m_{CSF}$$

which we refer to as brain geometry. We further consider the normalized ***tumor cell concentration*** (saturation) $c\colon \Omega_B \times [0,1] \to [0,1], (\boldsymbol{x},t) \mapsto c(\boldsymbol{x},t)$ and likewise interpret it as the probability to encounter a cancerous cell at location $\boldsymbol{x} \in \Omega_B$ and time $t \in [0,1]$. Ω_B is the domain occupied by brain tissue (healthy or unhealthy) and is delimited by the skull; it is embedded[1] in the simulation domain $\Omega_B \subset \Omega = [0,2\pi)^3$. For completeness, we recapitulate the space-time domain definitions $U = \Omega \times (0,1]$, $\bar{U} = \Omega \times [0,1)$, $U_B = \Omega_B \times (0,1]$, and $\bar{U}_B = \Omega_B \times [0,1)$.

Segmentation labels, i.e., characteristic functions calculated based on given threshold values for the healthy tissue types and normalized tumor concentration are used only for computing overlap measures (such as Dice coefficients) for the reconstruction and image output of our solver.

Dependency on Time and Space. For simplicity, we omit the explicit dependency on the spatial position $\boldsymbol{x}$ for most formulations, but explicitly state only the dependency on the time. The considered probability maps evolve along two independent time scales; an illustration is given in Fig. 6.2. For the tumor evolution, we consider the physical time t_T, normalized to $[0;1]$, i.e., $c(0) = c(\boldsymbol{x}, t_T = 0)$ denotes the initial tumor probability map, whereas $c(t_T = 1) = c(\boldsymbol{x}, t_T = 1)$ is the grown tumor at final time $t_T = 1$ (solution of the forward simulation). For the diffeomorphic image registration, we introduce a ***pseudo-time variable*** $t_R \in [0,1]$ without physical meaning. The pseudo-time describes the transient advection process and probability maps $\boldsymbol{m}(0) = \boldsymbol{m}(\boldsymbol{x}, t_R = 0) = \boldsymbol{m}_T$ are associated with the undeformed ***template image*** and $\boldsymbol{m}(1) = \boldsymbol{m}(\boldsymbol{x}, t_R = 1)$ with its ***deformed*** representation. For convenience, we indicate by the negative time $t_R = -1$ the advection of the template image with the negative velocity in the opposite direction to obtain the action of the inverse deformation map[2], i.e., $\boldsymbol{m}(-1) = \mathcal{R}^{\rightarrow}[-\boldsymbol{v}, \boldsymbol{m}_T]$. For clarity of notation, we introduce an index tuple superscript (t_R, t_T) for all considered probability maps, where t_R denotes the pseudo-time associated to registration advection, and t_T the physical tumor progression time. Note, that all probability maps are still space-time objects $m \in \Omega \times [0,1]$ and evolve in only one direction – either "tumor growth" or "registration deformation" – at a time. The differentiation between physical tumor time and virtual registration time for the advection can therefore be as well omitted, however, the state and meaning of the evolved objects are better to grasp if stated explicitly.

Patient and Atlas Data. We annotate probability maps of brain tissue regions and cancerous cells with subscripts to indicate the space they originate from. A subscript D ($\boldsymbol{m}_D, c_D$) denotes input patient data, subscript A objects either represent the healthy statistical atlas $\boldsymbol{m}_A^{(0,0)}$, or are derived from direct calculation of the latter ($\boldsymbol{m}_A^{(0,1)}$; moving-patient, and $\boldsymbol{m}_A^{(1,1)}$, $\boldsymbol{m}_A^{(1,0)}$; moving-atlas), and subscript P ($\boldsymbol{m}_P^{(1,\cdot)}$, $c_P^{(1,\cdot)}$; moving patient only) objects are derived by direct calculation from the patient image data. The dependency on time (t_R, t_T) and space is summarized and illustrated in Fig. 6.2 for both schemes. Note that the registration advection process is working in opposite direction for both schemes: for the moving-patient formulation, the registration advects objects

[1] Although the tumor growth equations are formally defined in Ω_B, we discretize them in Ω using periodic boundary conditions in combination with extension methods to approximate the correct boundary conditions (e.g., Neumann conditions) on $\partial\Omega_B$.

[2] For velocity based image registration, the inverse of the deformation map can be easily accessed by solving the forward problem for the negative velocity $-\boldsymbol{v}$

description	*moving-patient*	*moving-atlas*
healthy atlas	$\boldsymbol{m}_A^{(0,0)}$	$\boldsymbol{m}_A^{(0,0)}$
patient brain	$\boldsymbol{m}_D$	$\boldsymbol{m}_D$
patient pathology	c_D	$\boldsymbol{m}_A^{(1,0)}$
healthy patient	$\star$	$\boldsymbol{m}_A^{(1,0)}$
initial pathology	$c^{(\cdot,0)}$	$c^{(\cdot,0)}$
sim. pathology	$c^{(\cdot,1)}$	$c^{(\cdot,1)}$
sim. brain	$\boldsymbol{m}_A^{(0,1)}$	$\boldsymbol{m}_A^{(1,1)}$
adv. patient brain	$\boldsymbol{m}_P^{(1,\cdot)}$	–
adv. pathology	$c_P^{(1,\cdot)}$	–

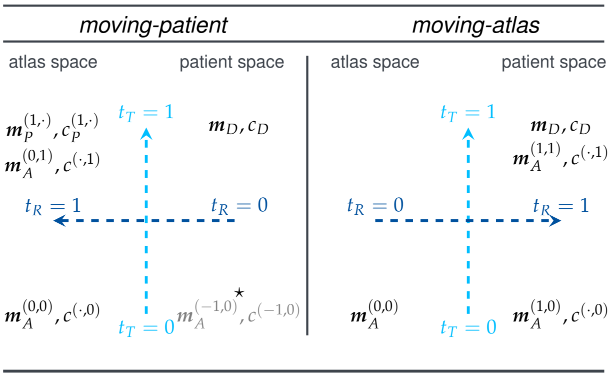

moving-patient
moving-atlas
atlas space
patient space
atlas space
patient space
$\boldsymbol{m}_P^{(1,\cdot)}, c_P^{(1,\cdot)}$
$\boldsymbol{m}_A^{(0,1)}, c^{(\cdot,1)}$
$t_T = 1$
$\boldsymbol{m}_D, c_D$
$t_R = 1$
$t_R = 0$
$\boldsymbol{m}_A^{(0,0)}, c^{(\cdot,0)}$
$t_T = 0$
$\boldsymbol{m}_A^{(-1,0)}, c^{(-1,0)}$ $\star$
$t_T = 1$
$\boldsymbol{m}_D, c_D$
$\boldsymbol{m}_A^{(1,1)}, c^{(\cdot,1)}$
$t_R = 0$
$t_R = 1$
$\boldsymbol{m}_A^{(0,0)}$
$t_T = 0$
$\boldsymbol{m}_A^{(1,0)}, c^{(\cdot,0)}$

FIGURE 6.2 *Notation for the* ***moving-patient*** *and the* ***moving-atlas*** *formulation and outline of dependency on time and space.* $\boldsymbol{m}$ *denotes the vector of probability maps for white matter, gray matter and cerebrospinal fluid defining the brain geometry. c denotes the probability map for tumor. Superscripts* (t_R, t_T) *denote the time associated to registration (*t_R*, virtual advection time, first index) and tumor growth (*t_T*, normalized to* $[0; 1]$*, second index). In cases where only one of them is applicable, we mark the second one with '·'. The dependency on the times* (t_R, t_T) *and the brain anatomy spaces A, P and D are outlined on the right. The subscript D denotes data given from the patient imaging data (after pre-processing), subscript A data derived by direct calculation from the statistical atlas brain, subscript P data derived by direct calculation from the patient image data. Entries marked by* $\star$ *have to be artificially derived and are no direct output of the respective formulation (see for example the healthy patient and initial condition in patient space for the moving-patient scheme, shaded light gray); entries marked by – are not used and calculated at all for the respective scheme.*

from the patient space into the atlas space (i.e., uses patient data as template image), whereas the registration forward direction for the moving-atlas scheme points from atlas to patient space, warping the healthy atlas towards an approximation of the healthy patient brain.

Vector Notation. Given a vector field $\boldsymbol{m} \in \mathbf{R}^3$, we compute $\nabla \boldsymbol{m} = (\partial_j \boldsymbol{m}_i)_{i,j=1,2,3}, \in \mathbf{R}^{3,3}$. That is, given a velocity field $\boldsymbol{v} \in \mathbf{R}^3$, $\nabla \boldsymbol{m}\ \boldsymbol{v}\ \in \mathbf{R}^3$ indicates a matrix-vector multiplication. The standard scalar product in $\mathbb{R}^3$ is denoted by "$\cdot$" and the outer product between two vector fields will be denoted by "$\otimes$". In addition, we define the following inner products:

$$(6.2)\qquad \langle \boldsymbol{m}, \tilde{\boldsymbol{m}} \rangle_{L^2(\Omega)^3} := \sum_{i=1}^{3} \langle m^i, \tilde{m}^i \rangle_{L^2(\Omega)}, \qquad \|\boldsymbol{m}\|^2_{L^2(\Omega)^3} := \sum_{i=1}^{3} \|m^i\|^2_{L^2(\Omega)}.$$

Distance Measures. We formulate the coupled schemes as PDE-constrained optimization problems. Their respective objective functionals feature the following two L^2-distance measures to judge proximity of the predicted data to the observed data:

$$(6.3)\qquad \mathcal{D}_c\left[c_1, c_2\right] := \frac{1}{2}\|c_1 - c_2\|^2_{L^2(\Omega)}, \quad \text{and} \quad \mathcal{D}_{\boldsymbol{m}}\left[\boldsymbol{m}_1, \boldsymbol{m}_2\right] := \frac{1}{2}\|\boldsymbol{m}_1 - \boldsymbol{m}_2\|^2_{L^2(\Omega)^3}$$

The chapter is structured as follows. We present the moving-patient formulation for the multi-component problem and corresponding first order optimality conditions in §6.1. This is contrasted with the moving-atlas formulation in §6.2 of the multi-component problem. For both schemes, we derive a Picard-iteration-type coupling algorithm for numerical optimization in §6.3, allowing for a modular combination of sub-component solvers. The Picard-type coupling is enhanced with a parameter- and grid-continuation scheme to prevent the solution process from getting stuck in local minima and allow for successively refined solution. A prerequisite for more biophysically meaningful results is the implementation of a sparsity constraint for the tumor inversion, such that the time horizon can be fixed to the time where we are only given point-source tumor seeds.

Furthermore, we add a limited memory BFGS (L-BFGS) quasi-Newton solver to the tumor inversion solver, allowing for significant speedups for the time-to-solution. We provide a generic explanation of different reduced-space methods and align their notation to ease comparison. The above mentioned features are presented in §6.4.

6.1 The Moving Patient Coupled Multi-Component Problem

We present a coupled formulation based on diffeomorphic image registration and inverse tumor growth simulation for the application problem introduced in §5.2. This ***moving-patient formulation*** aims at minimizing the difference between observed data (segmented MR imaging data of pathology) and reconstructed data (result of brain tumor progression simulations) simultaneously from two sides: (*i*) for the registration part, we consider the observed patient data (input) as ***template images*** and wish to advect it towards a ***reference image***, given a suitable velocity; (*ii*) the reference image itself is generated by synthetic tumor growth simulation in a healthy standard brain geometry (atlas brain; input) using the biophysical tumor progression model. In the inverse problem, we identify the tumor growth initial condition parametrization $\boldsymbol{p}$ and characteristic diffusivity $\bar{k} = k_f(k_w, k_g)$ for the tumor model and an image registration velocity $\boldsymbol{v}$ that together minimize the distance between the ***atlas with simulated tumor*** and the ***patient-to-atlas warped*** input data. In this sense, the image registration closes the gap that is caused by the fact that we can grow tumors (by simulation) only in atlas brain geometries for which we know the healthy geometry and, at the same time, have to match the observed tumor in the specific patient brain. Or, from a registration perspective, the tumor solver aids the registration in overcoming topological differences between atlas and patient. We simultaneously register four different images: three brain regions in $\boldsymbol{m}$ (see (6.1)) and the tumor concentration c. An illustration of the scheme is given in Fig. 6.3.

This formulation has been presented in our earlier work [Sch] and has been proven to yield very good results in terms of data similarity of reconstruction and observation, overlay measures (i.e., Dice coefficients), and registration quality. It is thus greatly suited for normal-to-abnormal registration tasks where the inverse biophysics solver aids the registration. For biophysics inversion, however, we observe the drawbacks mentioned above; as a result this formulation is prone to falling short in reconstructing reliable tumor progression parameters that allow for prediction and clinical decision making.

6.1.1 Formulation of The Coupled Problem

We consider a formulation based on optimal control theory. It results in a PDE-constrained, nonlinear optimization problem. We seek for a stationary velocity field $\boldsymbol{v}(\boldsymbol{x}) = (v_1(\boldsymbol{x}), v_2(\boldsymbol{x}), v_3(\boldsymbol{x}))^T$ (which establishes spatial correspondences between patient and atlas space), a mass-source map w (which controls the compressibility of the deformation pattern), and biophysical parameters in form of a coefficient vector $\boldsymbol{p}$ (which defines the tumor initial condition via $c^{(\cdot,0)} = \Phi\boldsymbol{p}$) and the characteristic diffusivity $\bar{k} = k_f(k_w, k_g)^T$ (controlling the tumor infiltration into surrounding tissue) as follows:

$$\min_{\boldsymbol{v},w,\boldsymbol{p},\bar{k}} \mathcal{J}_{\mathrm{MP}}\left[\boldsymbol{v}, \boldsymbol{p}, \bar{k}, w\right] \quad \text{with} \tag{6.4a}$$

$$\mathcal{J}_{\mathrm{MP}}\left[\boldsymbol{v}, \boldsymbol{p}, \bar{k}, w\right] := \mathcal{D}_c\left[c^{(\cdot,1)}, c_P^{(1,\cdot)}\right] + \mathcal{D}_{\boldsymbol{m}}\left[\boldsymbol{m}_A^{(0,1)}, \boldsymbol{m}_P^{(1,\cdot)}\right] + \beta_{\boldsymbol{p}}\mathcal{S}_{\boldsymbol{p}}\left[\boldsymbol{p}\right] + \beta_{\boldsymbol{v}}\mathcal{S}_{\boldsymbol{v}}\left[\boldsymbol{v}\right] + \beta_w\mathcal{S}_w\left[w\right]$$

subject to

$$\begin{aligned}
&\textit{tum:} && \partial_t c^{(\cdot,t)} - \nabla \cdot \boldsymbol{k}\nabla c^{(\cdot,t)} - R(c^{(\cdot,t)},\rho) = 0 && \text{in } U, && (6.4\text{b})\\
& && c^{(\cdot,0)} = \Phi p && \text{in } \Omega, && (6.4\text{c})\\
&\textit{reg:} && \partial_t \boldsymbol{m}_P^{(t,\cdot)} + \nabla \boldsymbol{m}_P^{(t,\cdot)} \boldsymbol{v} = \boldsymbol{0} && \text{in } U, && (6.4\text{d})\\
& && \boldsymbol{m}_P^{(0,\cdot)} = \boldsymbol{m}_D && \text{in } \Omega, && (6.4\text{e})\\
& && \partial_t c_P^{(t,\cdot)} + \nabla c_P^{(t,\cdot)} \cdot \boldsymbol{v} = 0 && \text{in } U, && (6.4\text{f})\\
& && c_P^{(0,\cdot)} = c_D && \text{in } \Omega, && (6.4\text{g})\\
& && \nabla \cdot \boldsymbol{v} = w && \text{in } U, && (6.4\text{h})\\
&\textit{cpl:} && \boldsymbol{m}_A^{(0,1)} = \boldsymbol{m}_A^{(0,0)}(1 - c^{(\cdot,1)}) && \text{in } \Omega. && (6.4\text{i})
\end{aligned}$$

with periodic boundary conditions on $\partial\Omega$. The optimization is driven by the following building blocks in the objective functional $\mathcal{J}_{\mathrm{MP}}$ in (6.4a):

(i) the ***two** L^2**-distance measures*** $\mathcal{D}_c[c^{(\cdot,1)}, c_P^{(1,\cdot)}]$ and $\mathcal{D}_{\boldsymbol{m}}[\boldsymbol{m}_A^{(0,1)}, \boldsymbol{m}_P^{(1,\cdot)}]$ (see (6.3) for a definition) that measure the discrepancy between the simulated tumor in atlas space $c^{(\cdot,1)}$ and the ***warped-to-atlas*** probability map $c_P^{(1,\cdot)}$ of cancerous cells for the input patient data, as well as the discrepancy between atlas brain regions with grown synthetic tumor $\boldsymbol{m}_A^{(0,1)}$ (computed according to (6.4i)) and the ***warped-to-atlas*** probability maps for patient brain regions $\boldsymbol{m}_P^{(1,\cdot)}$ ($c^{(\cdot,1)}$ is given as solution of the forward tumor growth simulation (6.4f) with initial condition (6.4g); $c_P^{(1,\cdot)}$ and $\boldsymbol{m}_P^{(1,\cdot)}$ are computed from (6.4f) and (6.4d), respectively),

(ii) three ***regularization operators*** balanced against the discrepancy measures $\mathcal{D}_c$ and $\mathcal{D}_{\boldsymbol{m}}$ based on regularization weights $\beta_j > 0$, $j \in \{v, w, p\}$ involving a regularization operator $\mathcal{S}_{\boldsymbol{v}}$ for $\boldsymbol{v}$, a regularization operator $\mathcal{S}_w$ for w (an H^1 Sobolev semi-norm[3] and an H^1 Sobolev norm[4], respectively; defined in (5.18b) in §5.3.2) and a regularization operator $\mathcal{S}_{\boldsymbol{p}}$ for $\boldsymbol{p}$ for which we combine two variants $\mathcal{S}_{\boldsymbol{p}}^i, i \in \{1,2\}$ defined as

$$\mathcal{S}_{\boldsymbol{p}}^1\,[\boldsymbol{p}] = \frac{1}{2}\|\boldsymbol{p}\|_{L^1(\Omega)} \text{ and } \mathcal{S}_{\boldsymbol{p}}^2\,[\boldsymbol{p}] = \frac{1}{2}\|\boldsymbol{p}\|_{\boldsymbol{W}}, \tag{6.5}$$

where $\|\boldsymbol{p}\|_{\boldsymbol{W}} = \|\boldsymbol{W}\boldsymbol{p}\|_{L^2(\Omega)}$ with $\boldsymbol{W} \in \mathbf{R}^{n_p \times n_p}$ is a weighted L^2-norm. For the special case $\boldsymbol{W} = \Phi$ we obtain the L^2-regularization on the initial condition[5] considered in (5.8) in §5.3.1. To favor sparse initial conditions/point sources for the tumor seed[6], we combine a L^1- and weighted L^2-regularization, where the weighting matrix $\boldsymbol{W}$ ensures that the L^2-regularization solve using $\mathcal{S}_{\boldsymbol{p}}^2$ penalizes $\boldsymbol{p}$-entries corresponding to Gaussian basis functions that have not been selected by a prior L^1-regularization solve using $\mathcal{S}_{\boldsymbol{p}}^1$. For details on the inverse tumor regularization strategies enforcing sparsity and the switching between L^1- and L^2-regularization solves, we refer to §6.3.3.

Notice, that for the registration, we use the probability maps from the input patient imaging data $\boldsymbol{m}_T := (\boldsymbol{m}_P^{(1,\cdot)}, c_D)^T$ as template images. The reference image $\boldsymbol{m}_R := (\boldsymbol{m}_A^{(0,1)}, c^{(\cdot,1)})^T$ is modified by

[3] The H^1 Sobolev semi-norm for $\boldsymbol{v}$ is given by $\mathcal{S}_{\boldsymbol{v}}\,[\boldsymbol{v}] = \frac{1}{2}\int_\Omega \sum_{i=1}^3 |\nabla v^i(\boldsymbol{x})|^2 \,\mathrm{d}\Omega$.

[4] The H^1 Sobolev norm for w is given by $\mathcal{S}_w\,[w] = \frac{1}{2}\int_\Omega |\nabla w(\boldsymbol{x})|^2 + |w|^2 \,\mathrm{d}\Omega$.

[5] The L^2-regularization on the initial condition $\Phi\boldsymbol{p}$ reads $\mathcal{S}_{\boldsymbol{p}}^2\,[\boldsymbol{p}] = \frac{1}{2}\|\Phi\boldsymbol{p}\|_{L^2(\Omega)}$.

[6] In other words we fix the time horizon of the inverse tumor growth such that the initial condition consists of point sources only.

the tumor sub-component solver. For the latter, recall, that the tumor progression is dictated by the diffusion coefficient $\boldsymbol{k}(\boldsymbol{m}_A^{(0,0)})$, a tensor field parametrized by scalar characteristic diffusivity weights for white matter and gray matter, respectively, i.e., $\boldsymbol{k}(\boldsymbol{m}) = \bar{k}\boldsymbol{m}(\boldsymbol{x})\boldsymbol{I} = k_f(k_w, k_g, 0)\boldsymbol{m}(\boldsymbol{x})\boldsymbol{I}$, and the reaction coefficient $\rho(\boldsymbol{m}_A^{(0,0)})$ for the logistic growth function $R(c,\rho) = \rho(\boldsymbol{m})c(1-c)$ with respective characteristic cell proliferation rates, i.e., $\rho(\boldsymbol{m}) = \bar{\rho}\boldsymbol{m}(\boldsymbol{x}) = \rho_f(\rho_w, \rho_g, 0)\boldsymbol{m}(\boldsymbol{x})$; see §5.3.1 for details. The geometric coupling condition (6.4i) combines the result of the tumor progression simulation with the surrounding healthy tissue by imprinting the probability of tumor cells into the healthy tissue probability maps. Note that (6.4i) does not consider mass effect, that is deformation of brain parenchyma due to the tumor growth. For the integration of mass effect, the condition reads $\boldsymbol{m}_A^{(0,1)} = \boldsymbol{m}_A^{(0,0)} \circ \boldsymbol{\vartheta} \cdot (1 - c^{(\cdot,1)})$ in Ω for a map $\boldsymbol{\vartheta}$ describing the deformations of healthy tissue due to tumor growth.

6.1.2 Optimality Conditions

For the numerical solution of (6.4) we employ an adjoint approach. Introducing ***Lagrange multiplier*** functions $\alpha\colon \Omega \times [0,1] \to \mathbf{R}$ for the tumor state $c^{(\cdot,t)}$, $\lambda_c\colon \bar{\Omega} \times [0,1] \to \mathbf{R}$ and $\boldsymbol{\lambda}_{\boldsymbol{m}}\colon \bar{\Omega} \times [0,1] \to \mathbf{R}^3$ for the advected maps $c_P^{(t,\cdot)}$ and $\boldsymbol{m}_P^{(t,\cdot)}$, respectively, $\nu\colon \bar{\Omega} \to \mathbf{R}$ associated with the penalization of compressibility (6.4h), and $\boldsymbol{\xi}\colon \bar{\Omega} \times [0,1] \to \mathbf{R}^3$ for the geometrical coupling condition (6.4i), we transform the constrained optimization into an unconstrained one.

The resulting ***Lagrangian functional*** reads

$$
\begin{aligned}
\mathcal{L}_{\text{MP}}\,[\boldsymbol{\Theta}] = {} & \mathcal{D}_c\left[c^{(\cdot,1)}, c_P^{(1,\cdot)}\right] + \mathcal{D}_{\boldsymbol{m}}\left[\boldsymbol{m}_A^{(0,1)}, \boldsymbol{m}_P^{(1,\cdot)}\right] + \beta_{\boldsymbol{p}}\mathcal{S}_{\boldsymbol{p}}\,[\boldsymbol{p}] + \beta_{\boldsymbol{v}}\mathcal{S}_{\boldsymbol{v}}\,[\boldsymbol{v}] + \beta_w\mathcal{S}_w\,[w] \\
& + \langle \alpha(0), (c^{(\cdot,0)} - \Phi\boldsymbol{p})\rangle_{L^2(\Omega)} + \int_0^1 \langle \alpha, \partial_t c^{(\cdot,t)} - \nabla\cdot\boldsymbol{k}\nabla c^{(\cdot,t)} - R(c^{(\cdot,t)},\rho)\rangle_{L^2(\Omega)}\,dt \\
& + \int_0^1 \langle \boldsymbol{\lambda}_{\boldsymbol{m}}, \partial_t \boldsymbol{m}_P^{(t,\cdot)} + \nabla\boldsymbol{m}_P^{(t,\cdot)}\ \boldsymbol{v}\rangle_{L^2(\Omega)^3}\,dt + \langle \boldsymbol{\lambda}_{\boldsymbol{m}}(0), \boldsymbol{m}_P^{(0,\cdot)} - \boldsymbol{m}_D\rangle_{L^2(\Omega)^3} \\
& + \int_0^1 \langle \lambda_c, \partial_t c_P^{(t,\cdot)} + \nabla c_P^{(t,\cdot)}\cdot\boldsymbol{v}\rangle_{L^2(\Omega)}\,dt + \langle \lambda_c(0), c_P^{(0,\cdot)} - c_D\rangle_{L^2(\Omega)} \\
& + \int_0^1 \langle \nu, \nabla\cdot\boldsymbol{v} - w\rangle_{L^2(\Omega)}\,dt + \langle \boldsymbol{\xi}, \boldsymbol{m}_A^{(0,1)} - (1 - c^{(\cdot,1)})\boldsymbol{m}_A^{(0,0)}\rangle_{L^2(\Omega)^3},
\end{aligned}
\tag{6.6}
$$

with the argument vector $\boldsymbol{\Theta} = (c^{(\cdot,1)}, c_P^{(1,\cdot)}, \boldsymbol{m}_P^{(1,\cdot)}, \boldsymbol{m}_A^{(0,1)}, \alpha, \lambda_c, \boldsymbol{\lambda}_{\boldsymbol{m}}, \nu, \boldsymbol{\xi}, \boldsymbol{p}, \bar{k}, \boldsymbol{v}, w)$, composed of the state fields $c^{(\cdot,1)}, c_P^{(1,\cdot)}, \boldsymbol{m}_P^{(1,\cdot)}$, and $\boldsymbol{m}_A^{(0,1)}$, the adjoint fields $\alpha, \lambda_c, \boldsymbol{\lambda}_{\boldsymbol{m}}, \nu$, and $\boldsymbol{\xi}$, and the inversion fields $\boldsymbol{p}, \bar{k}, \boldsymbol{v}$, and w. For an admissible solution, we require stationarity of the Lagrangian with respect to $\boldsymbol{\Theta}$, i.e., vanishing first-order variations.

The strong form of the ***first-order optimality conditions*** for (6.6) is given by the following system of equations:

tumor, state equation

$$\delta_\alpha\mathcal{L}_{\text{MP}} = 0: \qquad \partial_t c^{(\cdot,t)} - \nabla\cdot\boldsymbol{k}\nabla c^{(\cdot,t)} - R(c^{(\cdot,t)},\rho) = 0 \quad \text{in } U, \tag{6.7a}$$

$$\delta_{\alpha(1)}\mathcal{L}_{\text{MP}} = 0: \qquad c^{(\cdot,0)} - \Phi p = 0 \quad \text{in } \Omega. \tag{6.7b}$$

tumor, adjoint equation

$$\delta_{c^{(\cdot,t)}}\mathcal{L}_{\text{MP}} = 0: \qquad -\partial_t\alpha - \nabla\cdot\boldsymbol{k}\nabla\alpha - \partial_{c^{(\cdot,t)}}R^\star(c^{(\cdot,t)},\rho)\alpha = 0 \quad \text{in } \bar{U}, \tag{6.7c}$$

$$\delta_{c^{(\cdot,1)}}\mathcal{L}_{\text{MP}} = 0: \qquad c_P^{(1,\cdot)} - c^{(\cdot,1)} - \boldsymbol{\xi}\boldsymbol{m}_A^{(0,0)} - \alpha(1) = 0 \quad \text{in } \Omega. \tag{6.7d}$$

registration, state equation

$$\delta_{\boldsymbol{\lambda}} \mathcal{L}_{\text{MP}} = 0: \qquad \partial_t \boldsymbol{m}_P^{(t,\cdot)} + \nabla \boldsymbol{m}_P^{(t,\cdot)} \boldsymbol{v} = \boldsymbol{0} \quad \text{in } U, \tag{6.7e}$$

$$\delta_{\boldsymbol{\lambda}(0)} \mathcal{L}_{\text{MP}} = 0: \qquad \boldsymbol{m}_P^{(0,\cdot)} - \boldsymbol{m}_D = \boldsymbol{0} \quad \text{in } \Omega, \tag{6.7f}$$

$$\delta_{\lambda_c} \mathcal{L}_{\text{MP}} = 0: \qquad \partial_t c_P^{(t,\cdot)} + \nabla c_P^{(t,\cdot)} \cdot \boldsymbol{v} = 0 \quad \text{in } U, \tag{6.7g}$$

$$\delta_{\lambda_c(0)} \mathcal{L}_{\text{MP}} = 0: \qquad c_P^{(0,\cdot)} - c_D = 0 \quad \text{in } \Omega, \tag{6.7h}$$

$$\delta_{v} \mathcal{L}_{\text{MP}} = 0: \qquad \nabla \cdot \boldsymbol{v} - w = 0 \quad \text{in } U. \tag{6.7i}$$

registration, adjoint equation

$$\delta_{\boldsymbol{m}_P^{(t,\cdot)}} \mathcal{L}_{\text{MP}} = 0: \qquad -\partial_t \boldsymbol{\lambda}_{\boldsymbol{m}} - \nabla \cdot (\boldsymbol{\lambda}_{\boldsymbol{m}} \otimes \boldsymbol{v}) = \boldsymbol{0} \quad \text{in } \bar{U}, \tag{6.7j}$$

$$\delta_{\boldsymbol{m}_P^{(1,\cdot)}} \mathcal{L}_{\text{MP}} = 0: \qquad \boldsymbol{m}_A^{(0,1)} - \boldsymbol{m}_P^{(1,\cdot)} - \boldsymbol{\lambda}_{\boldsymbol{m}}(1) = \boldsymbol{0} \quad \text{in } \Omega, \tag{6.7k}$$

$$\delta_{c_P^{(t,\cdot)}} \mathcal{L}_{\text{MP}} = 0: \qquad -\partial_t \lambda_c - \nabla \cdot (\lambda_c \boldsymbol{v}) = 0 \quad \text{in } \bar{U}, \tag{6.7l}$$

$$\delta_{c_P^{(1,\cdot)}} \mathcal{L}_{\text{MP}} = 0: \qquad c^{(\cdot,1)} - c_P^{(1,\cdot)} - \lambda_c(1) = 0 \quad \text{in } \Omega. \tag{6.7m}$$

coupling, state equation

$$\delta_{\boldsymbol{m}_A^{(0,1)}} \mathcal{L}_{\text{MP}} = 0: \qquad \boldsymbol{m}_A^{(0,0)}(1 - c^{(\cdot,1)}) - \boldsymbol{m}_A^{(0,1)} = \boldsymbol{0} \quad \text{in } \Omega. \tag{6.7n}$$

coupling, adjoint equation

$$\delta_{\boldsymbol{m}_A^{(0,1)}} \mathcal{L}_{\text{MP}} = 0: \qquad \boldsymbol{m}_P^{(1,\cdot)} - \boldsymbol{m}_A^{(0,1)} - \boldsymbol{\xi} = \boldsymbol{0} \quad \text{in } \Omega. \tag{6.7o}$$

tumor, inversion equation

$$\delta_{\boldsymbol{p}} \mathcal{L}_{\text{MP}} = 0: \qquad g_{\boldsymbol{p}} := \beta_{\boldsymbol{p}} \delta_{\boldsymbol{p}} \mathcal{S}_{\boldsymbol{p}}\left[\boldsymbol{p}\right] - \Phi^T \alpha(0) = \boldsymbol{0} \quad \text{in } \Omega, \tag{6.7p}$$

$$\delta_{\bar{k}} \mathcal{L}_{\text{MP}} = 0: \qquad g_k := \int_0^1 \int_\Omega \boldsymbol{m}_A^{(0,0)} \left((\nabla c^{(\cdot,t)})^T \nabla \alpha \right) dx\, dt = \boldsymbol{0} \quad \text{in } \Omega, \tag{6.7q}$$

registration, inversion equation

$$\delta_{\boldsymbol{v}} \mathcal{L}_{\text{MP}} = 0: \quad \beta_v \delta_{\boldsymbol{v}} \mathcal{S}_v\left[\boldsymbol{v}\right] + \mathcal{K}[\int_0^1 (\nabla \boldsymbol{m}_P^{(t,\cdot)})^T \boldsymbol{\lambda}_{\boldsymbol{m}} + \nabla c_P^{(t,\cdot)} \lambda_c\, dt] = \boldsymbol{0} \quad \text{in } \Omega. \tag{6.7r}$$

Note that $(\nabla \boldsymbol{m})^T \boldsymbol{\lambda} = \sum_i^3 \lambda_i (\partial_1 m_i, \partial_2 m_i, \partial_3 m_i)^T$ and that we have not given an equation for w. As described in §5.3.3 we obtain a pseudo-differential operator $\mathcal{K}$ in (6.7r) from elminating w in (6.4h) from the set of optimality conditions. For $\nabla \cdot \boldsymbol{v} = w = 0$, we obtain the Leary projection operator $\mathcal{K}[\boldsymbol{u}] := \boldsymbol{u} + \nabla \Delta^{-1} \nabla \cdot \boldsymbol{u}$; for a non-zero w, the projection operator becomes slightly more involved; details are given in §5.3.3 and [Man15; Man16a]. The gradients $\delta_{\boldsymbol{p}} \mathcal{S}_{\boldsymbol{p}}\left[\boldsymbol{p}\right]$ and $\delta_{\boldsymbol{v}} \mathcal{S}_{\boldsymbol{v}}\left[\boldsymbol{v}\right]$ of the regularization functionals (6.5) in the inversion equations (6.7p) and (6.7r), respectively, are given by

$$\delta_{\boldsymbol{p}} \mathcal{S}_{\boldsymbol{p}}^2\left[\boldsymbol{p}\right] = \boldsymbol{W}^\star \boldsymbol{W} \boldsymbol{p} \qquad \text{and} \qquad \delta_{\boldsymbol{v}} \mathcal{S}_{\boldsymbol{v}}\left[\boldsymbol{v}\right] = \Delta \boldsymbol{v} \tag{6.8a}$$

Notice, that for $\boldsymbol{W} = \Phi$ we obtain the gradient on the L^2-regularization of the initial condition and that the L^1-regularization operator $\mathcal{S}_{\boldsymbol{p}}^1\left[\boldsymbol{p}\right]$ is not differentiable, and thus not given here; the strategy how to solve the L^1-regularized inverse tumor problem is detailed in §6.3.3.

Summarizing, the first order optimality system (6.7) comprises a set of coupled, non-linear PDEs. The system portrays high complexity and is challenging to solve as it features ill-conditioning, indefiniteness along with a very large number of unknowns; it contains 11 fields in addition to the tumor initial condition parameters $\boldsymbol{p}$. Given the input data $\boldsymbol{m}_D$, c_D, $\boldsymbol{m}_A^{(0,0)}$ of our joint registration and biophysical inversion scheme, the task is to solve the first-order optimality PDEs for the state, adjoint and inversion variables. Falling back on our sub-component solvers and the therein employed methods, we follow a ***reduced-space approach*** [Noc06; Bor12], which assumes the state and adjoint equations to be solved exactly; elimination of the latter then yields the reduced system with ***reduced-gradient*** $\boldsymbol{g} = (g_v, g_p, g_k)^T$ of the objective function. For details, we refer to §5.3.3 and §6.4, as well as the above references. At a stationary point, the reduced-gradient $\boldsymbol{g}$ vanishes. An iterative gradient-descent or Newton-type solution requires evaluations of the reduced-gradient to compute updates on the inversion variables $\boldsymbol{p}$, $\bar{k}$, and $\boldsymbol{v}$, respectively. However, we opt for a simpler and modular Picard-iteration, which by exploiting the sub-component (reduced-Newton-)solvers resembles a block-Newton-type iteration. To monitor convergence of this scheme, the reduced-gradient for (6.7) is evaluated and involves the following steps[7]:

(i) Solve the forward problems for tumor progression and registration (6.4b)–(6.4i) for the state fields $c^{(\cdot,t)}$, $\boldsymbol{m}_A^{(0,1)}$, $c_P^{(t,\cdot)}$, $\boldsymbol{m}_P^{(t,\cdot)}$.

(ii) Compute the data mismatch, or coupling adjoint variable ξ from equation (6.7o).

(iii) Solve the adjoint tumor equations (6.7c) and (6.7d) for $\alpha(t)$.

(iv) Solve the adjoint registration equations (6.7j)–(6.7m) for $\boldsymbol{\lambda}_{\boldsymbol{m}}(t)$ and $\lambda_c(t)$.

(v) Evaluate the gradients using the inversion equations (6.7p), (6.7q) and (6.7r) at $\boldsymbol{p}$, $\bar{k}$ and $\boldsymbol{v}$:

$$g_{\boldsymbol{p}} = \beta_{\boldsymbol{p}}\delta_{\boldsymbol{p}}\mathcal{S}_{\boldsymbol{p}}(\boldsymbol{p}) - \Phi^T\alpha(0) \qquad g_k := \int_0^1\int_\Omega \boldsymbol{m}_A^{(0,0)}\left((\nabla c^{(\cdot,t)})^T\nabla\alpha\right)\,dx\,dt \tag{6.9a}$$

$$g_{\boldsymbol{v}} = \beta_{\boldsymbol{v}}\delta_{\boldsymbol{v}}\mathcal{S}_{\boldsymbol{v}}(\boldsymbol{v}) + \mathcal{K}[\int_0^1 (\nabla\boldsymbol{m}_P)^T\boldsymbol{\lambda}_{\boldsymbol{m}} + \nabla c_P\ \lambda_c\ \mathrm{d}t]. \tag{6.9b}$$

Before we explain our Picard-type coupling scheme in §6.3, we present a second formulation for the application problem, with potentially better clinical relevance for the biophysical parameter estimation.

6.2 The Moving Atlas Coupled Multi-Component Problem

The above presented moving-patient formulation is prone to fall short in terms of reliability, clinical applicability and practical relevance of the estimated biophysical tumor parameters if the brain anatomy of the specific patient at hand differs greatly from the employed standard atlas brain; the latter serves as a proxy to carry out the tumor progression simulations. As a result, the estimated model parameters "live" in the standard brain anatomy and are, thus, possibly inadequate for the considered patient. This particularly becomes emerging when considering model parameters that cannot be simply advected to the opposing space such as, e.g., the characteristic diffusivity for white matter and gray matter.

In order to tackle this problem, we introduce a second, slightly more advanced formulation for the joint inversion, the so called ***moving-atlas formulation***. As its name implies, the main difference lies in the inversion of the registration direction, i.e., instead of translating the patient input data to the atlas space, we warp the healthy atlas brain towards the patient space and thereby generate an approximation for the healthy patient brain anatomy, the brain structure before occurrence of cancerous invasion. As opposed to the previous scheme, this formulation leaves the input data

[7] Our Picard-iteration algorithm, presented in §6.3, naturally executes steps (i) through (iv) within one iteration. A reduced-gradient evaluation for the coupled formulation thus only involves step (v), i.e., evaluation of equations (6.9)

"untouched" and tries to generate preferably similar data (brain structure plus tumor) by growing a synthetic tumor in the obtained healthy patient approximation. As the deformation map becomes more accurate, the tumor progression simulation, and thereby, the model parameter estimation is carried out in the patient space. As a result, the underlying brain structure defining the characteristics of tumor progression is changed whenever the approximation to the healthy patient is improved, i.e, the deformation map is refined. This results in repeatedly changing diffusion and reaction coefficients throughout the simulation, inducing a tight intermeshing of the sub-component solvers.

Further, the inverse registration sub-problem that establishes the correspondences between atlas and patient space via the velocity $\boldsymbol{v}$ is more involved for this scheme. In contrast to the moving-patient formulation, we set the patient input data as ***reference image*** and choose the ***healthy atlas*** image as ***template image*** that is to be aligned to the pathologic patient data. For this challenging normal-to-abnormal registration problem of topologically different brains, a modified formulation for the registration problem is derived from the optimality condition of the coupled problem formulation in §6.2.1. It integrates information from the biophysics solver, to aid the process of identifying meaningful correspondences. Further, in this formulation, the registration only acts on the brain anatomy and, thus, cannot make up for a poor biophysical model or tumor reconstruction to still obtain good data similarity. The formulation of the overall scheme, including the modified registration formulation, respective Lagrangian and optimality system, is what we discuss next. A schematic illustration of the moving-atlas scheme is given in Fig. 6.3 and the required algorithmic steps and used sub-modules are described in detail in §6.3.2.

6.2.1 Formulation of The Coupled Problem

We present the moving-atlas as PDE-constrained optimization problem formulation, where we seek for a stationary velocity field $\boldsymbol{v}(\boldsymbol{x}) = (v_1(\boldsymbol{x}), v_2(\boldsymbol{x}), v_3(\boldsymbol{x}))^T$ mapping between atlas and patient space, a mass-source map w, controlling the registration induced volume change, and biophysical tumor progression parameters $\boldsymbol{p}$ for the initial tumor concentration and location, along with characteristic diffusivity scales $\bar{k} = k_f(k_w, k_g)^T$ for white matter and gray matter, respectively, controlling the cancerous cell invasion into healthy tissue. That is, we are interested in a minimizer $(\boldsymbol{v}, w, \boldsymbol{p}, \bar{k})$, such that the predicted data by our scheme becomes similar to the observed data in the following sense:

$$\min_{\boldsymbol{v},\boldsymbol{p},\bar{k},w} \; \mathcal{J}_{\text{MA}}\left[\boldsymbol{v}, \boldsymbol{p}, \bar{k}, w\right] \quad \text{with} \tag{6.10a}$$

$$\mathcal{J}_{\text{MA}}\left[\boldsymbol{v}, \boldsymbol{p}, \bar{k}, w\right] := \mathcal{D}_c\left[c^{(\cdot,1)}, c_D\right] + \mathcal{D}_{\boldsymbol{m}}\left[\boldsymbol{m}_A^{(1,0)}(1 - c^{(\cdot,1)}), \boldsymbol{m}_D\right] + \beta_{\boldsymbol{p}}\mathcal{S}_{\boldsymbol{p}}\left[\boldsymbol{p}\right] + \beta_{\boldsymbol{v}}\mathcal{S}_{\boldsymbol{v}}\left[\boldsymbol{v}\right] + \beta_w\mathcal{S}_w\left[w\right]$$

subject to

$$\textit{tum:} \quad \partial_t c^{(\cdot,t)} - \nabla \cdot \boldsymbol{k}(\boldsymbol{m}_A^{(1,0)})\nabla c^{(\cdot,t)} - R(c^{(\cdot,t)}, \rho(\boldsymbol{m}_A^{(1,0)})) = 0 \qquad \text{in } U, \tag{6.10b}$$

$$\Phi(\boldsymbol{m}_A^{(1,0)})\boldsymbol{p} = c^{(\cdot,0)} \qquad \text{in } \Omega, \tag{6.10c}$$

$$\textit{reg:} \quad \partial_t \boldsymbol{m}_A^{(t,0)} + \nabla \boldsymbol{m}_A^{(t,0)}\, \boldsymbol{v} = \boldsymbol{0} \qquad \text{in } U, \tag{6.10d}$$

$$\boldsymbol{m}_T = \boldsymbol{m}_A^{(0,0)} \qquad \text{in } \Omega, \tag{6.10e}$$

$$\nabla \cdot \boldsymbol{v} = w \qquad \text{in } U, \tag{6.10f}$$

with periodic boundary conditions on $\partial\Omega$. The objective functional $\mathcal{J}_{\text{MA}}$ in (6.10a) consists of the following building blocks:

(i) the ***two*** L^2***-distance measures*** $\mathcal{D}_c[c^{(\cdot,1)}, c_D]$ and $\mathcal{D}_{\boldsymbol{m}}[\boldsymbol{m}_A^{(1,0)}(1-c^{(\cdot,1)}), \boldsymbol{m}_D]$ (see (6.3) for a definition) measuring the tumor cell distribution and brain region proximity of the reconstruction data to the observed input data, where $\mathcal{D}_c$ measures the discrepancy between the simulated tumor $c^{(\cdot,1)}$ (computed from (6.10b)–(6.10c) in the ***approximated patient space***) and the input tumor data c_D, $\mathcal{D}_{\boldsymbol{m}}$ measures the discrepancy between probability maps of the patient brain regions (input) and the ***warped-to-patient*** healthy atlas probability maps (computed according to (6.10d)–(6.10f)); to account for the topological differences, the simulated pathology is embedded into the advected healthy atlas probability maps for the registration misfit, which can be seen as a "masking" of the tumor region from the registration (i.e., for this scheme, the registration does not alter the tumor concentration nor the discrepancy measure $\mathcal{D}_c$),

(ii) the same ***regularization operators*** $\mathcal{S}_{\boldsymbol{p}}^1(\boldsymbol{p})$ or $\mathcal{S}_{\boldsymbol{p}}^2(\boldsymbol{p})$, respectively, for $\boldsymbol{p}$, $\mathcal{S}_{\boldsymbol{v}}(\boldsymbol{v})$ for $\boldsymbol{v}$, and $\mathcal{D}_w(w)$ for w as given in the moving patient formulation §6.1.1 and balanced with the regularization weights $\beta_j > 0$, $j \in \{v, w, p\}$.

For the registration, we use the probability maps for the healthy atlas brain tissue regions as ***template image*** $\boldsymbol{m}_T := \boldsymbol{m}_A^{(0,0)}$ and register them towards the probability maps of the pathologic patient brain tissue regions, i.e., $\boldsymbol{m}_R := \boldsymbol{m}_D$. Note, that the tumor concentration is not registered in this formulation, but is only used to "mask" the tumor region from the registration to allow for a meaningful deformation map for the normal-to-abnormal registration task. Details for the modified registration problem are given in §6.3.2. Furthermore, reaction and diffusion coefficient $\boldsymbol{k}(\boldsymbol{m}_A^{(1,0)})$ and $\rho(\boldsymbol{m}_A^{(0,0)})$, respectively, depend on the approximation of the healthy patient brain anatomy, which is used for tumor progression simulations. The scheme gradually improves this approximation and, thus, also alters the reaction and diffusion coefficients.

6.2.2 Optimality Conditions

As before, we consider the unconstrained optimization problem defined by the minimization of the ***Lagrangian functional***, which arises from the constrained optimization problem (6.10) by incorporation of the forward problem equations (constraints) (6.10b)–(6.10f) as inner products with the respective ***Lagrange multiplier*** fields, or adjoint variables of all state variables occurring in the forward problem equations.

The ***Lagrangian functional*** for the moving-atlas formulation reads

$$
\begin{aligned}
\mathcal{L}_{\mathrm{MA}}[\boldsymbol{\Theta}] = {} & \mathcal{D}_c\left[c^{(\cdot,1)}, c_D\right] + \mathcal{D}_{\boldsymbol{m}}\left[\boldsymbol{m}_A^{(1,0)}(1-c^{(\cdot,1)}), \boldsymbol{m}_D\right] + \beta_p \mathcal{S}_{\boldsymbol{p}}[\boldsymbol{p}] + \beta_v \mathcal{S}_{\boldsymbol{v}}[\boldsymbol{v}] + \beta_w \mathcal{S}_w[w] \\
& + \int_0^1 \langle \alpha, \partial_t c^{(\cdot,1)} - \nabla\cdot \boldsymbol{k}(\boldsymbol{m}_A^{(1,0)})\nabla c^{(\cdot,1)}) - R(c^{(\cdot,1)}, \rho(\boldsymbol{m}_A^{(1,0)}))\rangle_{L^2(\Omega)}\, dt \\
& + \langle \alpha(0), (c^{(\cdot,0)} - \Phi\boldsymbol{p})\rangle_{L^2(\Omega)} + \int_0^1 \langle \nu, \nabla\cdot\boldsymbol{v} - w\rangle_{L^2(\Omega)}\, dt \\
& + \int_0^1 \langle \boldsymbol{\lambda}, \partial_t \boldsymbol{m}_A^{(t,0)} + \nabla \boldsymbol{m}_A^{(t,0)}\ \boldsymbol{v}\rangle_{L^2(\Omega)^3}\, dt + \langle \boldsymbol{\lambda}(0), \boldsymbol{m}_A^{(0,0)} - \boldsymbol{m}_T\rangle_{L^2(\Omega)^3}
\end{aligned}
\tag{6.11}
$$

with the argument vector field $\boldsymbol{\Theta} = (c^{(\cdot,1)}, \boldsymbol{m}_A^{(t,0)}, \boldsymbol{m}_A^{(1,0)}, \alpha, \boldsymbol{\lambda}, \nu, \boldsymbol{p}, \bar{k}, \boldsymbol{v}, w)$ composed of the state fields $c^{(\cdot,1)}, \boldsymbol{m}_A^{(t,0)}$, and $\boldsymbol{m}_A^{(1,0)}$, the adjoint fields $\alpha, \boldsymbol{\lambda}$ and ν, and the inversion fields $\boldsymbol{p}, \bar{k}, \boldsymbol{v}$, and w.

As a necessary condition for a minimum, we require stationarity of the Lagrangian $\mathcal{L}_{\mathrm{MA}}$ with respect to $\boldsymbol{\Theta}$; stipulating vanishing first-order variations, we can derive the strong form of the

first-order optimality conditions for (6.11), given as the following KKT system:

tumor, state equation

$$\delta_\alpha \mathcal{L}_{\mathrm{MA}} = 0: \quad \partial_t c^{(\cdot,t)} - \nabla \cdot \boldsymbol{k}(\boldsymbol{m}_A^{(1,0)}) \nabla c^{(\cdot,t)} - R(c^{(\cdot,t)}, \rho(\boldsymbol{m}_A^{(1,0)})) = 0 \quad \text{in } U, \tag{6.12a}$$

$$\delta_{\alpha(1)} \mathcal{L}_{\mathrm{MA}} = 0: \quad c^{(\cdot,0)} - \Phi p = 0 \quad \text{in } \Omega, \tag{6.12b}$$

tumor, adjoint equation

$$\delta_{c^{(\cdot,t)}} \mathcal{L}_{\mathrm{MA}} = 0: \quad -\partial_t \alpha - \nabla \cdot \boldsymbol{k}(\boldsymbol{m}_A^{(1,0)}) \nabla \alpha + \partial_{c^{(\cdot,t)}} R^\star(c^{(\cdot,t)}, \rho(\boldsymbol{m}_A^{(1,0)}))\alpha = 0 \quad \text{in } \bar{U}, \tag{6.12c}$$

$$\delta_{c^{(\cdot,1)}} \mathcal{L}_{\mathrm{MA}} = 0: \quad c_D - c^{(\cdot,1)} + (\boldsymbol{m}_A^{(1,0)})^T (\boldsymbol{m}_A^{(1,0)}(1 - c^{(\cdot,1)}) - \boldsymbol{m}_D) - \alpha(1) = 0 \quad \text{in } \Omega. \tag{6.12d}$$

registration, state equation

$$\delta_{\boldsymbol{\lambda}} \mathcal{L}_{\mathrm{MA}} = 0: \quad \partial_t \boldsymbol{m}_A^{(t,0)} + \nabla \boldsymbol{m}_A^{(t,0)} \boldsymbol{v} = \boldsymbol{0} \quad \text{in } \bar{U}, \tag{6.12e}$$

$$\delta_{\boldsymbol{\lambda}(1)} \mathcal{L}_{\mathrm{MA}} = 0: \quad \boldsymbol{m}_A^{(0,0)} - \boldsymbol{m}_T = \boldsymbol{0} \quad \text{in } \Omega, \tag{6.12f}$$

$$\delta_{\boldsymbol{v}} \mathcal{L}_{\mathrm{MA}} = 0: \quad \nabla \cdot \boldsymbol{v} - w = 0 \quad \text{in } U, \tag{6.12g}$$

registration, adjoint equation

$$\delta_{\boldsymbol{m}_A^{(t,0)}} \mathcal{L}_{\mathrm{MA}} = 0: \quad -\partial_t \boldsymbol{\lambda} - \nabla \cdot (\boldsymbol{\lambda} \otimes \boldsymbol{v}) = \boldsymbol{0} \quad \text{in } \bar{U}, \tag{6.12h}$$

$$\begin{aligned} \delta_{\boldsymbol{m}_A^{(1,0)}} \mathcal{L}_{\mathrm{MA}} = 0: \quad & (c^{(\cdot,1)} - 1)(\boldsymbol{m}_A^{(1,0)}(1 - c^{(\cdot,1)}) - \boldsymbol{m}_D) + \partial_{\boldsymbol{m}_A^{(1,0)}} \Phi(\boldsymbol{p}) \alpha(0) \\ & - \int_0^1 (\nabla c^{(\cdot,t)})^T \nabla \alpha \, \partial_{\boldsymbol{m}_A^{(1,0)}} \boldsymbol{k}^\star(\boldsymbol{m}_A^{(1,0)}) + \partial_{\boldsymbol{m}_A^{(1,0)}} R^\star(c^{(\cdot,t)}, \rho(\boldsymbol{m}_A^{(1,0)}))\alpha \, dt - \boldsymbol{\lambda}(1) = \boldsymbol{0} \quad \text{in } \Omega, \end{aligned} \tag{6.12i}$$

tumor, inversion equation

$$\delta_{\boldsymbol{p}} \mathcal{L}_{\mathrm{MA}} = 0: \quad \boldsymbol{g}_{\boldsymbol{p}} := \beta_{\boldsymbol{p}} \delta_{\boldsymbol{p}} \mathcal{S}_{\boldsymbol{p}}[\boldsymbol{p}] - \Phi^T \alpha(0) = \boldsymbol{0} \quad \text{in } \Omega. \tag{6.12j}$$

$$\delta_k \mathcal{L}_{\mathrm{MA}} = 0: \quad g_k := \int_0^1 \int_\Omega \boldsymbol{m}_A^{(1,0)} \left((\nabla c^{(\cdot,t)})^T \nabla \alpha \right) dx\, dt = \boldsymbol{0} \quad \text{in } \Omega, \tag{6.12k}$$

registration, inversion equation

$$\delta_{\boldsymbol{v}} \mathcal{L}_{\mathrm{MA}} = 0: \quad \boldsymbol{g}_{\boldsymbol{v}} := \beta_{\boldsymbol{v}} \delta_{\boldsymbol{v}} \mathcal{S}_{\boldsymbol{v}}[\boldsymbol{v}] + \mathcal{K}[\int_0^1 (\nabla \boldsymbol{m}_A^{(t,0)})^T \boldsymbol{\lambda} \, dt] = \boldsymbol{0} \quad \text{in } \Omega. \tag{6.12l}$$

with adjoint variables α, $\boldsymbol{\lambda}$, ν associated to the state variables $c^{(\cdot,1)}$, $\boldsymbol{m}_A^{(t,0)}$, and $\boldsymbol{m}_A^{(1,0)}$. As before, w is eliminated which results in the pseudo-differential operator $\mathcal{K}$ in (6.12l). The gradients $\delta_{\boldsymbol{p}} \mathcal{S}_{\boldsymbol{p}}[\boldsymbol{p}]$, and $\delta_{\boldsymbol{v}} \mathcal{S}_{\boldsymbol{v}}[\boldsymbol{v}]$ of the regularization functionals in (6.12j) and (6.12l), respectively, remain unchanged compared to the moving-patient formulation and are given in (6.8a); additionally, we get the gradients for the diffusion and reaction coefficients in (6.12i) with respect to $\boldsymbol{m}_A^{(1,0)}$ which are given by

$$\partial_{\boldsymbol{m}_A^{(1,0)}} \boldsymbol{k} = \bar{k} = (k_w, k_g, 0) \qquad \text{and} \qquad \partial_{\boldsymbol{m}_A^{(1,0)}} R = \bar{\rho} \boldsymbol{m}_A^{(1,0)} (1 - 2c^{(\cdot,1)}) \tag{6.13a}$$

Analogously to the moving-patient scheme, we follow a ***reduced-space*** approach to solve the set of coupled non-linear PDEs (6.12). A block-iterative scheme that solves for $\boldsymbol{p}^\star$, $\bar{k}^\star$ and $\boldsymbol{v}^\star$ such that the reduced-gradient $\boldsymbol{g} = (\boldsymbol{g}_{\boldsymbol{v}}, \boldsymbol{g}_{\boldsymbol{p}}, g_k)^T$ vanishes is presented in §6.3.2.

The ***reduced-gradient*** for the moving-atlas formulation can be evaluated as follows:

(i) Solve the registration forward equations (6.12e)–(6.12g) for the healthy patient approx. $\boldsymbol{m}_A^{(1,0)}$.

(ii) Update the diffusion and reaction coefficients $\boldsymbol{k}(\boldsymbol{m}_A^{(1,0)})$ and $\rho(\boldsymbol{m}_A^{(1,0)})$, respectively as well as $\Phi(\boldsymbol{m}_A^{(1,0)})$; then solve the tumor progression forward problem (6.12a)–(6.12b) for $c^{(\cdot,t)}$.

(iii) Solve the adjoint tumor equations (6.12c)–(6.12d) for $\alpha(t)$.

(iv) compute the final condition (6.12i) for the registration using $c^{(\cdot,t)}, \alpha(t)$ and the gradients $\partial_{\boldsymbol{m}_A^{(1,0)}} \boldsymbol{k}$, $\partial_{\boldsymbol{m}_A^{(1,0)}} R$, and $\partial_{\boldsymbol{m}_A^{(1,0)}} \Phi$ and solve the adjoint equation (6.12h) for $\boldsymbol{\lambda}(t)$.

(v) Evaluate the gradients using the inversion equations (6.12j), (6.12k) and (6.12l) at $\boldsymbol{p}$, $\bar{k}$ and $\boldsymbol{v}$:

Summarizing, the moving-atlas formulation remedies two drawbacks of the moving-patient scheme. It (*i*) solves for estimates of the biophysical parameters with respect to the patient space rather then the atlas space, and (*ii*) it prohibits the registration from picking up the slack for the tumor solver in fitting observed data to a possibly poor tumor reconstruction/poor model, resulting in an overall small data mismatch. This, however, results in (mathematically) more intermeshed sub-component solvers and further requires the derivation of a tailored formulation for the registration module to fulfill the optimality conditions (6.12). Details on this as well as the employed decomposition of the formulation into sub-components and consulted solver strategies are given in §6.3.2. In order to exploit the full potential of the moving-atlas formulation, a sophisticated regularization scheme for the tumor inversion, which enforces sparse initial conditions, becomes necessary; this is explained further in §6.3.3.

6.3 Modular Iterative Solvers – Picard-Iterations

In this section, we present our iterative schemes to solve the formidable optimization systems (6.7) and (6.12) of strongly coupled, non-linear partial differential equations for the moving-patient and the moving-atlas formulation, respectively. Instead of solving the optimality systems using a gradient descent or Newton scheme, we employ a modular Picard-iteration-type approach, which interleaves separate sub-component solvers for tumor growth inversion and diffeomorphic registration models. We solve both optimization problems by iteratively improving estimates on the registration velocity $\boldsymbol{v}$, the tumor parameters $\boldsymbol{p}$ for the parametrization of the tumor initial condition, and (if switched on) the characteristic diffusivity scales $\bar{k} = (k_w, k_g, 0)$ for white matter and gray matter, respectively. This methodology allows for an easy, stable, modular and efficient way for numerical optimization of (6.4) and (6.10), respectively, where the submodules can be exchanged as required; for instance a more sophisticated tumor solver accounting for effects such as mitosis, apoptosis, chemotaxis, and deformation of brain parenchyma (mass-effect) can be plugged in as replacement of our quite simplistic model.

Our coupling schemes use sub-components for (*i*) tumor progression simulation, forward in time, (*ii*) inverse tumor growth simulation (parameter estimation), (*iii*) image advection, and (*iv*) solving for spatial correlations in the image registration problem. The respective iterative schemes are confronted schematically in Fig. 6.3. The sub-component solvers have been discussed in §5.3 and [Man15; Man16a; Gho17c; Gho16a; Gho16b; Man16b; Man17a].

Both schemes share the start-up phase of selecting initial guesses for the inversion fields $\boldsymbol{p} \in \mathbf{R}^{n_p}$, $\bar{k} \in \mathbf{R}^3$ and $\boldsymbol{v} \in \mathbf{R}^{n \times 3}$, and selection of the Gaussian basis functions for the parametrization of the tumor initial condition. Unless the schemes are run in ***warm-start-mode***, we choose a zero initial guess for $\boldsymbol{p}$, $\bar{k}$ and $\boldsymbol{v}$, respectively[8]. To reduce runtime, we use the solution for $\boldsymbol{p}$, $\bar{k}$, and $\boldsymbol{v}$ from the previous Picard-iteration as the initial guess for the current iteration. For the ***selection of Gaussian basis functions*** dependent on the patient's tumor concentration, two variants are supported and detailed in §6.3.3. Note that for the current schemes the initial selection of Gaussian

[8] The warm-start-mode is used together with a multi-scale approach and explained in more detail in §6.3.3.

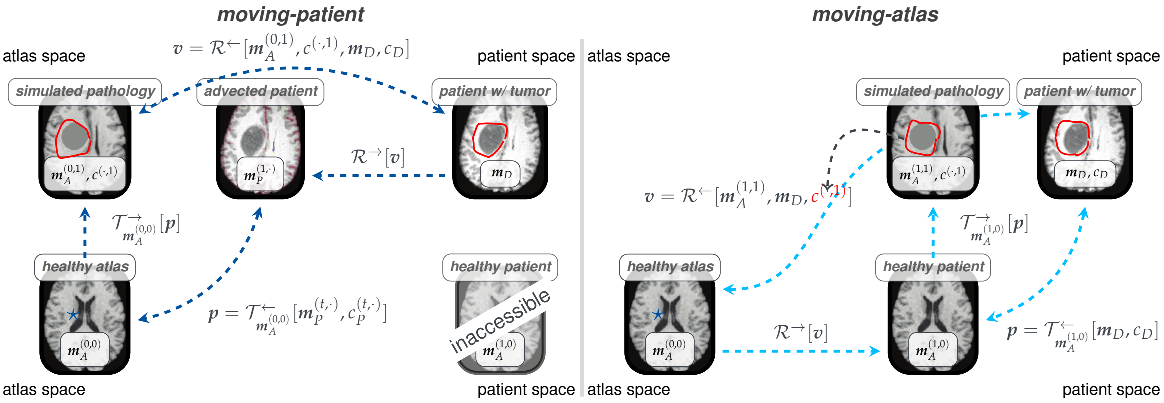

moving-patient
atlas space
$v = \mathcal{R}^{\leftarrow}[\boldsymbol{m}_A^{(0,1)}, c^{(\cdot,1)}, \boldsymbol{m}_D, c_D]$
patient space
simulated pathology
advected patient
patient w/ tumor
$\boldsymbol{m}_A^{(0,1)}, c^{(\cdot,1)}$
$\boldsymbol{m}_P^{(1,\cdot)}$
$\mathcal{R}^{\rightarrow}[v]$
$\boldsymbol{m}_D$
$\mathcal{T}^{\rightarrow}_{\boldsymbol{m}_A^{(0,0)}}[\boldsymbol{p}]$
healthy atlas
healthy patient
$\boldsymbol{p} = \mathcal{T}^{\leftarrow}_{\boldsymbol{m}_A^{(0,0)}}[\boldsymbol{m}_P^{(t,\cdot)}, c_P^{(t,\cdot)}]$
inaccessible
$\boldsymbol{m}_A^{(0,0)}$
$\boldsymbol{m}_A^{(1,0)}$
atlas space
patient space
moving-atlas
atlas space
patient space
simulated pathology
patient w/ tumor
$\boldsymbol{m}_A^{(1,1)}, c^{(\cdot,1)}$
$\boldsymbol{m}_D, c_D$
$v = \mathcal{R}^{\leftarrow}[\boldsymbol{m}_A^{(1,1)}, \boldsymbol{m}_D, c^{(\cdot,1)}]$
$\mathcal{T}^{\rightarrow}_{\boldsymbol{m}_A^{(1,0)}}[\boldsymbol{p}]$
healthy atlas
healthy patient
$\mathcal{R}^{\rightarrow}[v]$
$\boldsymbol{p} = \mathcal{T}^{\leftarrow}_{\boldsymbol{m}_A^{(1,0)}}[\boldsymbol{m}_D, c_D]$
$\boldsymbol{m}_A^{(0,0)}$
$\boldsymbol{m}_A^{(1,0)}$
atlas space
patient space

FIGURE 6.3 *Schematic view of the Picard-iterations for the two coupling-schemes. The **moving-patient Picard-iteration** (left figure) grows a tumor in a healthy atlas brain and computes a velocity, which warps the patient data towards the modified atlas brain along with the simulated pathology. For the tumor inversion, we use the patient-to-atlas warped patient pathology as input data to estimate the biophysical parameters in the atlas space. This process is iterated, resulting in a Picard block-Newton type iteration. For the **moving atlas Picard-iteration** (right figure), we advect the atlas brain towards the patient and grow a tumor in the resulting approximation of the healthy brain geometry; the simulated pathology is then used to aid the registration finding a velocity that establishes plausible correspondences for this normal-to-abnormal problem of registering the healthy atlas to the patient brain. A modified registration formulation is used here. The tumor inversion operates on the original patient data and seeks for biophysical parameters that live in the (approximated) patient space. The process is iterated, gradually refining the velocity, biophysical parameters and the healthy patient approximation. As a result, the tumor evolution properties (diffusion/reaction coefficient) of the tumor solver change in every iteration. Images modified from [Hog08b].*

basis functions remains unchanged throughout the Picard-iterations, i.e., neither re-selection nor advection of the basis functions in case of updated tumor data (moving-patient) or changed simulation geometry (moving-atlas), is considered. Furthermore, both schemes are embedded into a ***parameter-continuation*** scheme for the regularization parameter β_v of the registration sub-problem; starting off with a large regularization weight, it is gradually reduced throughout the solution process (we discuss this below in §6.3.3). On a limiting note, we do not have a proof for convergence of our Picard-iteration coupling schemes to a minimizer of the respective fully-coupled optimization formulations (6.4) and (6.10). However, thorough numerical analysis in Chapter 7 provides evidence that the schemes effectively reduce the respective gradients $\boldsymbol{g} = (\boldsymbol{g}_v, \boldsymbol{g}_p, \boldsymbol{g}_k)$ of the fully-coupled formulations. A rigorous proof of convergence remains subject to future work.

6.3.1 A Picard-Iteration Algorithm for the Moving Patient Scheme

The Picard-iteration algorithm for the moving-patient formulation explicitly iterates over the tumor inversion variables $\boldsymbol{p}$, $\bar{k}$ and implicitly iterates over the registration velocity $\boldsymbol{v}$. Given some $(\boldsymbol{p}, \bar{k})^k$, the $(k+1)$-th Picard-iteration is defined by the following sub-steps:

(1) Seed the atlas brain $\boldsymbol{m}_A^{(0,0)}$ with the initial tumor $\Phi \boldsymbol{p}^k$ and compute its temporal evolution as $c^{(\cdot,1)} = \mathcal{T}^{\rightarrow}_{\boldsymbol{m}_A^{(0,0)}}[(\boldsymbol{p}, \bar{k})^k]$, solving[9] equations (6.7a)–(6.7b).

(2) Imprint the simulated tumor into the healthy atlas geometry to obtain a synthetic brain with pathology at physical time $t_T = 1$. We define the operator $\mathcal{G}[c^{(\cdot,1)}] := (\boldsymbol{m}_A^{(0,0)}(1 - c^{(\cdot,1)}, c^{(\cdot,1)})$.

[9] This step is only required in the first Picard-iteration, if the scheme is run in warm-start mode (e.g., for grid-continuation, see §6.3.3). Otherwise, we use initial guess $\boldsymbol{p} = 0$ and for all subsequent iterations, the simulated $t_T = 1$ tumor $c^{(\cdot,1)}$ is already available as a side-product of the tumor inversion solve from the previous Picard-iteration. The step is stated explicitly for clarity.

(3) Register the input patient data $\boldsymbol{m}_T := (\boldsymbol{m}_D, c_D)^T$ to the generated synthetic brain with pathology $\boldsymbol{m}_R := (\boldsymbol{m}_A^{(0,1)}, c^{(\cdot,1)})^T$ via solving equations (6.7e)–(6.7m) and (6.7r). This yields the velocity $\boldsymbol{v} = \mathcal{R}^{\leftarrow}_{\mathrm{MP}}[\boldsymbol{m}_A^{(0,1)}, c^{(\cdot,1)}, \boldsymbol{m}_D, c_D]$.

(4) Advect[10] the input data for the patient pathology into the atlas space using the computed velocity $\boldsymbol{v}$; this gives us new input data for the tumor inversion via $c_P^{(1,\cdot)} = \mathcal{R}^{\rightarrow}[\boldsymbol{v}, c_D]$.

(5) Use the updated tumor data $c_P^{(1,\cdot)}$ to improve estimates for $\boldsymbol{p}$ and $\bar{k}$ via solving equations (6.7a)–(6.7d) and (6.7p)–(6.7q); that is, solve the inverse tumor growth problem $(\boldsymbol{p}, \bar{k})^{k+1} = \mathcal{T}^{\leftarrow}_{\boldsymbol{m}_A^{(0,0)}}[c_P^{(1,\cdot)}]$.

(6) Check for convergence (see §6.3.3). If the convergence criteria are fulfilled, stop. If not, go back to step 1 (i.e., continue iterating).

A schematic illustration of the scheme is given in Fig. 6.3. Summarized, the moving-patient coupling scheme can be written as the following Picard-iteration:

$$(\boldsymbol{p}, \bar{k})^{k+1} = \mathcal{T}^{\leftarrow}_{\boldsymbol{m}_A^{(0,0)}} \circ \mathcal{R}^{\rightarrow}\left[\mathcal{R}^{\leftarrow}_{\mathrm{MP}}\left[\mathcal{G} \circ \mathcal{T}^{\rightarrow}_{\boldsymbol{m}_A^{(0,0)}}[(\boldsymbol{p}, \bar{k})^k], \boldsymbol{m}_D, c_D\right], c_D\right] \tag{6.14}$$

Note, that steps (1) and (4) in the above Picard-iteration algorithm are obsolete, as the result of the forward simulation is already given as side-product by the respective inverse problems. Thus, assuming we are given $(\boldsymbol{m}_A^{(0,1)}, c^{(\cdot,1)})$ after application of the geometrical coupling equation(6.7n), we have to solve

(MP.1) the ***inverse registration problem*** $\mathcal{R}^{\leftarrow}_{\mathrm{MP}}$, defined as

(6.15)
$$\mathcal{R}^{\leftarrow}_{\mathrm{MP}} \begin{cases} \min_{\boldsymbol{v},\boldsymbol{w}} \ \mathcal{J}_{\mathrm{R_{MP}}}[\boldsymbol{v}, \boldsymbol{w}] := \mathcal{D}_c\left[c^{(\cdot,1)}, c_P^{(1,\cdot)}\right] + \mathcal{D}_{\boldsymbol{m}}\left[\boldsymbol{m}_A^{(0,1)}, \boldsymbol{m}_P^{(1,\cdot)}\right] + \beta_{\boldsymbol{v}}\mathcal{S}_{\boldsymbol{v}}[\boldsymbol{v}] + \beta_{\boldsymbol{w}}\mathcal{S}_{\boldsymbol{w}}[\boldsymbol{w}] \\ \textit{subject to} \quad \text{the advection problem (6.7e)–(6.7i) for } c_P^{(t,\cdot)} \text{ and } \boldsymbol{m}_P^{(t,\cdot)} \end{cases}$$

and

(MP.2) the ***inverse tumor growth problem*** $\mathcal{T}^{\leftarrow}_{\boldsymbol{m}_A^{(0,0)}}$, defined as

(6.16)
$$\mathcal{T}^{\leftarrow}_{\boldsymbol{m}_A^{(0,0)}} \begin{cases} \min_{\boldsymbol{p},\bar{k}} \ \mathcal{J}_{\mathrm{T}_{\boldsymbol{m}_A^{(0,0)}}}[\boldsymbol{p}, \bar{k}] := \mathcal{D}_c\left[c^{(\cdot,1)}, c_P^{(1,\cdot)}\right] + \beta_{\boldsymbol{p}}\mathcal{S}_{\boldsymbol{p}}[\boldsymbol{p}] \\ \textit{subject to} \quad \text{the tumor growth simulation (6.7a)–(6.7b) for } c^{(\cdot,t)} \text{ with } c^{(\cdot,0)} = \Phi\boldsymbol{p}^k \end{cases}$$

Upon iterative solution of these sub-problems and convergence of the iteration, we fulfill all first order optimality conditions (6.7) for the moving-patient formulation, except from (6.7o) and (6.7d). Here, we replace the final condition for the tumor adjoint equation (6.7d) by

$$c_P^{(1,\cdot)} - c^{(\cdot,1)} - \alpha(1) = 0 \qquad \text{in } \Omega, \tag{6.17}$$

i.e., lacking the term $\boldsymbol{\xi}\boldsymbol{m}_A^{(0,0)}$ originating from the ***simplified objective*** for $\mathcal{T}^{\leftarrow}_{\boldsymbol{m}_A^{(0,0)}}$ which only considers the tumor data misfit $\mathcal{D}_c$ and neglects the data misfit $\mathcal{D}_{\boldsymbol{m}}$ for the brain tissue labels. Minimization of the registration objective $\mathcal{J}_{\mathrm{T}_{\boldsymbol{m}_A^{(0,0)}}}$, however, eventually enforces $\boldsymbol{m}_A^{(0,1)} \approx \boldsymbol{m}_P^{(1,\cdot)}$ and, thus, $\xi \approx 0$.

[10] Notice, that this data is already available as a side-product of the inverse registration process and is only explicitly stated here for clarity. These steps involves no further computations.

A Block-Newton Scheme. Alternatively we define the tumor inverse operator $\widehat{\mathcal{T}}^{\leftarrow}_{\boldsymbol{m}_A^{(0,0)}}$, considering the full data misfit $\mathcal{D}_c$ and $\mathcal{D}_{\boldsymbol{m}}$ in the objective function as

$$\text{(6.18)}\quad \widehat{\mathcal{T}}^{\leftarrow}_{\boldsymbol{m}_A^{(0,0)}} \left\{ \begin{array}{ll} \min_{\boldsymbol{p},\bar{k}} & \widehat{\mathcal{J}}_{\text{TMP}}[\boldsymbol{p},\bar{k}] := \mathcal{D}_c\left[c^{(\cdot,1)}, c_P^{(1,\cdot)}\right] + \mathcal{D}_{\boldsymbol{m}}\left[\boldsymbol{m}_A^{(0,1)}, \boldsymbol{m}_P^{(1,\cdot)}\right] + \beta_{\boldsymbol{p}}\mathcal{S}_{\boldsymbol{p}}[\boldsymbol{p}] \\ \textit{subject to} & \text{the tumor growth simulation (6.7a)–(6.7b) for } c^{(\cdot,t)} \text{ with } c^{(\cdot,0)} = \Phi\boldsymbol{p}^k \end{array} \right.$$

which, if plugged into the moving-patient Picard-iteration algorithm above, enhances it to a ***block-Newton iterative solver*** for the optimality conditions (6.7). This modified Picard-iteration, fulfilling all optimality conditions, is referred to as ***full-objective block-Newton scheme*** for the moving-patient formulation.

6.3.2 A Picard-Iteration Algorithm for the Moving Atlas Scheme

In a similar way, the moving-atlas Picard-iteration coupling scheme iterates over the inversion variables $\boldsymbol{p}$, $\bar{k}$, and $\boldsymbol{v}$ and performs the following sub-steps in the $(k+1)$-th iteration, given previous iterates $(\boldsymbol{p},\bar{k},\boldsymbol{v})^k$:

(1) Compute an approximation to the healthy patient brain geometry by solving[11] the advection equation (6.12e)–(6.12g) for the healthy atlas geometry $\boldsymbol{m}_A^{(0,0)}$ and the velocity $\boldsymbol{v}^k$ as $\boldsymbol{m}_A^{(1,0)} = \mathcal{R}^{\rightarrow}[\boldsymbol{v}, \boldsymbol{m}_A^{(0,0)}]$.

(2) Update the reaction and diffusion coefficients $\rho(\boldsymbol{m}_A^{(1,0)})$ and $\boldsymbol{k}(\boldsymbol{m}_A^{(1,0)})$, respectively with the approximation of the healthy patient brain geometry $\boldsymbol{m}_A^{(1,0)}$ and seed the latter with the initial tumor $\Phi\boldsymbol{p}^k$; simulate the tumor progression as $c^{(\cdot,1)} = \mathcal{T}^{\rightarrow}_{\boldsymbol{m}_A^{(1,0)}}[(\boldsymbol{p},\bar{k})^k]$, solving[12] equations (6.12a)–(6.12b).

(3) Solve equations (6.12a)–(6.12d) and (6.12j)–(6.12k) for improved estimates on $\boldsymbol{p}$ and $\bar{k}$ in patient space, using the unmodified input data of the patient's tumor c_D; that is, solve the inverse tumor growth problem $(\boldsymbol{p},\bar{k})^{k+1} = \mathcal{T}^{\leftarrow}_{\boldsymbol{m}_A^{(1,0)}}[c_D]$.

(4) Solve the normal-to-abnormal registration problem between the healthy atlas and pathologic patient geometry, resulting in a warped-to-patient atlas geometry, the healthy patient. This includes solving equations (6.12e)–(6.12i) and (6.12l), defining a modified registration operator $\boldsymbol{v}^{k+1} = \mathcal{R}^{\leftarrow}_{\text{MA}}[\boldsymbol{m}_A^{(1,0)}, \boldsymbol{m}_D, c^{(\cdot,1)}]$.

(5) Check for convergence (see §6.3.3). If the convergence criteria are fulfilled, stop. If not, go back to step 1 (i.e., continue iterating).

A schematic illustration of the scheme is given in Fig. 6.3. As before, the forward solves in step (1) and (2) are implicitly included in steps (4) and (3), respectively, and are, thus, redundant; we explicitly state them for clarity. In order to solve the optimality system (6.12) in a modular way, consisting of the primary blocks tumor inversion and registration, we need to define the following ***modified*** sub-component formulations: Given an approximation for the healthy patient (initially this is the healthy atlas for $\boldsymbol{v} = \boldsymbol{0}$), we solve

[11] This step is only required in the first Picard iteration, if the scheme is run in warm-start mode (e.g., for grid-continuation, see §6.3.3). Otherwise, we either start with an initial guess $\boldsymbol{v} = \boldsymbol{0}$, or the warped template (healthy patient approximation) is already available as by-product of the inverse registration solve in step (4) from the previous Picard-iteration.

[12] This step is only required in the first Picard-iteration. For all subsequent iterations, the simulated $t_T = 1$ tumor $c^{(\cdot,1)}$ is already available as a side-product of the tumor inversion solve from the previous Picard-iteration. The step is stated explicitly for clarity.

(MA.1) the ***inverse tumor growth problem*** $\mathcal{T}^{\leftarrow}_{\boldsymbol{m}_A^{(1,0)}}$, defined as

(6.19)

$$\mathcal{T}^{\leftarrow}_{\boldsymbol{m}_A^{(1,0)}} \begin{cases} \min_{\boldsymbol{p},\bar{k}} \ \mathcal{J}_{\mathrm{T}_{\boldsymbol{m}_A^{(1,0)}}}[\boldsymbol{p},\bar{k}] := \mathcal{D}_c\left[c^{(\cdot,1)}, c_D\right] + \beta_{\boldsymbol{p}}\mathcal{S}_{\boldsymbol{p}}[\boldsymbol{p}] \\ \textit{subject to} \quad \text{the tumor growth simulation (6.12a)–(6.12b) for } c^{(\cdot,1)} \text{ with } c^{(\cdot,0)} = \Phi \boldsymbol{p}^k \end{cases}$$

and

(MA.2) the ***modified inverse registration problem*** $\mathcal{R}^{\leftarrow}_{\mathrm{MA}}$, defined as

(6.20)

$$\mathcal{R}^{\leftarrow}_{\mathrm{MA}} \begin{cases} \min_{\boldsymbol{v},w} \ \mathcal{J}_{\mathrm{R_{MA}}}[\boldsymbol{v},w] \ \text{ with} \\ \mathcal{J}_{\mathrm{R_{MA}}}[\boldsymbol{v},w] := \mathcal{D}_{\boldsymbol{m}}\left[\boldsymbol{m}_A^{(1,0)}(1-c^{(\cdot,1)}), \boldsymbol{m}_D\right] + \int_\Omega \boldsymbol{q}^T \boldsymbol{m}_A^{(1,0)}\,dx + \beta_v\mathcal{S}_v[\boldsymbol{v}] + \beta_w\mathcal{S}_w[w] \\ \textit{subject to} \quad \text{the advection problem (6.12e)–(6.12g) for } \boldsymbol{m}_A^{(t,0)} \text{ and } \boldsymbol{m}_T := \boldsymbol{m}_A^{(0,0)} \end{cases}$$

where $\boldsymbol{q} = (q_1, q_2, q_3)^T$ defined as

$$\boldsymbol{q}(\boldsymbol{x}) = \int_0^1 \bar{k}((\nabla c^{(\cdot,1)})^T \nabla \alpha) + \bar{\rho} c^{(\cdot,1)}(1-c^{(\cdot,1)})\alpha\,dt. \tag{6.21}$$

This formulation for the registration problem with the particular choice for the ***modified objective function*** $\mathcal{J}_{\mathrm{R_{MA}}}$ allows us to sub-divide the solution of (6.12) into the above defined modular sub-component solvers while reproducing the first-order optimality conditions (6.12h) and (6.12i) of the fully-coupled moving-atlas formulation. Note that the registration problem $\mathcal{R}^{\leftarrow}_{\mathrm{MA}}$ does not modify the tumor probability map $c^{(\cdot,1)}$ and, thus, doesn't act on the data misfit term $\mathcal{D}_c$ for the tumor. To account for the topological differences between the registered brains, a frozen representation of the grown tumor from the previous iteration is used in the objective function $\mathcal{J}_{\mathrm{R_{MA}}}$ to "mask" this area from the registration.

After convergence of the outlined Picard-iteration algorithm, the solution fulfills all first-order optimality conditions of (6.12) for the moving-atlas formulation, except for the final conditions of the adjoint equations for tumor inversion and registration, i.e., equations (6.12d) and (6.12i), respectively. For the latter, the variation of the Gaussian basis functions with respect to the changing healthy patient geometry $\partial_{\boldsymbol{m}_A^{(1,0)}}\Phi(\boldsymbol{m}_A^{(1,0)})\boldsymbol{p}\alpha(0)$ is neglected and (6.12i) is replaced by

$$(c^{(\cdot,1)}-1)(\boldsymbol{m}_A^{(1,0)}(1-c^{(\cdot,1)}) - \boldsymbol{m}_D) - \int_0^1 \bar{k}((\nabla c^{(\cdot,1)})^T\nabla\alpha) + \bar{\rho}c^{(\cdot,1)}(1-c^{(\cdot,1)})\alpha\,dt - \boldsymbol{\lambda}(1) = \boldsymbol{0} \quad \text{in } \Omega,$$

As for the moving-patient Picard-iteration, our tumor inversion problem only considers the data misfit for the tumor probability maps $\mathcal{D}_c$, which is reflected in the first-order optimality conditions by neglecting the term $(\boldsymbol{m}_A^{(1,0)})^T(\boldsymbol{m}_A^{(1,0)}(1-c^{(\cdot,1)}) - \boldsymbol{m}_D)$ in the final-condition for the adjoint solve, i.e., equation (6.12d) is replaced by

$$c_D - c^{(\cdot,1)} - \alpha(1) = 0 \qquad \text{in } \Omega.$$

Upon convergence of the scheme, the neglected terms become very small, as the registration problem minimizes the distance between $(\boldsymbol{m}_A^{(1,0)})^T(\boldsymbol{m}_A^{(1,0)}(1-c^{(\cdot,1)})$ and $\boldsymbol{m}_D)$.

A Block-Newton Scheme. In a same manner as before, the tumor adjoint final-condition (6.12d) in the first-order optimality conditions of the moving-atlas scheme can be reproduced by introducing the modified inverse tumor operator $\widehat{\mathcal{T}}^{\leftarrow}_{\boldsymbol{m}_A^{(1,0)}}$, which adds the data misfit $\mathcal{D}_{\boldsymbol{m}}$ of the brain tissue probability maps to the objective function:

$$\text{(6.22)}\quad \widehat{\mathcal{T}}^{\leftarrow}_{\boldsymbol{m}_A^{(1,0)}} \begin{cases} \min_{\boldsymbol{p},\bar{k}}\ \widehat{\mathcal{J}}_{\mathrm{TMA}}\left[\boldsymbol{p},\bar{k}\right] := \mathcal{D}_c\left[c^{(\cdot,1)}, c_D\right] + \mathcal{D}_{\boldsymbol{m}}\left[\boldsymbol{m}_A^{(1,0)}(1-c^{(\cdot,1)}), \boldsymbol{m}_D\right] + \beta_{\boldsymbol{p}}\mathcal{S}_{\boldsymbol{p}}\left[\boldsymbol{p}\right] \\ \textit{subject to}\quad \text{the tumor growth simulation (6.12a)–(6.12b) for } c^{(\cdot,t)} \text{ with } c^{(\cdot,0)} = \Phi\boldsymbol{p}^k \end{cases}$$

Replacing the earlier defined tumor inverse operator $\mathcal{T}^{\leftarrow}_{\boldsymbol{m}_A^{(1,0)}}$ by the modified operator $\widehat{\mathcal{T}}^{\leftarrow}_{\boldsymbol{m}_A^{(1,0)}}$ in the Picard-iteration algorithm, again yields a ***block-Newton iterative solver*** for the optimality conditions (6.12); upon convergence, it fulfills all optimality conditions, except the neglecting of the basis function variation with respect to $\boldsymbol{m}_A^{(1,0)}$. This scheme is referred to as ***full-objective block-Newton scheme*** for the moving-atlas formulation.

6.3.3 Shared Ingredients of the Coupling Schemes

The challenging, highly non-linear, non-convex optimization problems and their presented respective Picard iteration-type solution approaches require further concepts to foster the solution process. In particular, we need to address (*i*) the non-linear and non-convex nature of the optimization problem, (*ii*) the immense computational complexity and significant time-to-solution, and (*iii*) the balancing of sub-solver accuracies to prevent over-solving and yet ensure an overall accurate precision. In the sequel, we discuss strategies and concepts employed in SIBIA to deal with these issues. A ***parameter-continuation*** scheme for the regularization parameters β_v and $\beta_{\boldsymbol{p}}$ as well as an F-cycle ***grid-continuation*** scheme are used to convexify the problem, prevent convergence to non-optimal local minima, find the suitable extent of regularization, and speedup the time to solution by providing sophisticated initial guesses on problems with higher data fidelity. Furthermore, we discuss a set of ***convergence criteria***, which is designed and tuned to prevent over-solving. Taking steps towards a fully automated brain tumor image analysis tool, we present strategies to automatically ***place the Gaussian basis functions*** for the parametrization of the initial tumor condition, dependent on the patient input data. Finally, we discuss a different regularization operator for the tumor inversion solver, which enforces ***sparsity of the initial condition parametrization***. This fixes the inversion time horizon to the stage of a point-source tumor seed and allows for biophysically more meaningful parameter estimates. In the following, we describe these approaches in more detail.

Parameter Continuation. The highly non-linear and non-convex nature of the considered optimization problem poses challenges on the employed solvers. Due to prominent noise level, non-convexity, and oscillations, the solvers (in particular the registration solver) are prone to fail for uninformed initial guesses (typically far from the optimum). Further, the proper extent of regularization, i.e., the balance of imposed regularity on the solution and data matching, varies for specific instantiations of the problem and is a priori unknown. Thus, we embed the Picard-iteration solution process into a parameter-continuation scheme for the regularization parameter β_v for the regularizer $\mathcal{S}_v[v] = {}^1\!/_2\|\nabla v\|_{L^2(\Omega)}$ of the registration velocity v. This scheme effectively addresses the above mentioned issues, i.e., it (*i*) convexifies the optimization problem, (*ii*) stabilizes the solution process and prevents it from getting trapped in local minima, (*iii*) reduces the time-to-solution, and (*iv*) identifies an optimal regularization parameter $\beta_v^\star$ for the given data.

In more detail, starting with a highly regularized (nearly convex and easy to solve optimization problem), we solve a series of problems with gradually reduced regularization including more and more data fidelity and details, while allowing for more complex solutions. The solutions from previous solves are used as initial guesses for subsequent solves with smaller regularization parts. This alleviates two problems: it (*i*) reduces the chance of getting trapped in local minima, introduced by more data fidelity in the objective function, and it (*ii*) attenuates the indefiniteness of the Hessian and fosters the solution of increasingly ill-conditioned operators[13]. Lastly, the optimal parameter $\beta_v^\star$ to balance the data mismatch and regularization depends on the input images and is not known a priori. We are interested in a solution with the lowest possible contribution from the regularizer to still yield an optimization problem with an acceptable condition and a unique solution. We use the parameter-continuation scheme to find that optimal weighting parameter $\beta_v^\star$ by monitoring the determinant of the deformation gradient $\det(\nabla \boldsymbol{y})$ as follows: We specify a lower admissable bound ε_∇ on the deformation gradient[14] and start the Picard iteration with $\beta_v^0 = 1$, i.e., a highly regularized optimization problem with smooth solution, omitting any details. In every Picard-iteration k, the candidate β_v^k for the regularization parameter is reduced by one order of magnitude[15], until the lower bound for the determinant of the deformation gradient $\det(\nabla \boldsymbol{y}) < \varepsilon_\nabla$ is breached for a candidate β_v^k. In this case, the registration solver disregards the obtained solution and restarts the inversion process with adapted regularization weight $\beta_v^k \leftarrow \beta_v^{k-1} - (\beta_v^{k-1} - \beta_v^k)/2$ and the solution from the previous Picard-iteration as initial guess. This β_v-backtracking algorithm is repeated until the lower bound on the determinant of the deformation gradient is no longer violated. If during the registration solve no violation of the lower bound criterion for the determinant of the deformation gradient was detected, the current Picard-iteration is finalized and the solution scheme proceeds with the next iteration. A pseudo code for the β_v-continuation is given in Alg. 6.1.

Remark. For the moving-patient formulation, the described β_v-continuation scheme is (besides the aforementioned reasons) additionally required to prevent unwanted data-fitting and establish a meaningful coupling between the sub-component solvers: In order to prevent the registration from fitting the tumor input data to an initially poor tumor reconstruction/prediction of the biophysical model, we wish to limit the registration induced deformations (registration velocity is almost zero, i.e., the patient tumor is merely copied to the atlas space) by imposing a high regularization weight $\beta_v \approx 1$ in an early stage of the Picard-iteration solution process. As the estimation for the parameters of the tumor progression models improves and the reconstruction becomes more accurate, we allow for a more complex deformation by gradually reducing the regularization parameter β_v.

Remark. Analogous to the β_v-continuation scheme for the regularization operator of the registration velocity $\boldsymbol{v}$, a continuation scheme on the regularization parameter $\beta_{\boldsymbol{p}}$ for the regularizer on $\boldsymbol{p}$ of the tumor inversion solver, is conceivable. Numerical examination of different variants showed dependence of the solution on the considered parameter-continuation scheme (i.e., β_v-continuation or combined β_v-$\beta_{\boldsymbol{p}}$-continuation) and the step length for the reduction in between Picard-iterations. We observe best overall results if only continuation on the registration regularization parameter β_v is performed.

[13] A direct solve of the marginally regularized optimization problem would not only result in a non-optimal solution due to non-convexity of the objective, but is also most likely to diverge due to break down of the employed solvers in the face of severe ill-conditioning and indefiniteness of involved operators

[14] The determinant of the deformation gradient $\det(\nabla \boldsymbol{y})$ is a measure for the local volume change induced by the transformation, i.e., volume compression/expansion. A lower bound on this quantity controls the maximum allowed local volume change.

[15] In our numerical analysis, we found that reducing the regularization weight by 10 in between parameter-continuation steps yields robust results while allowing for an acceptable time-to-solution.

```
func picard-mp($v^0, p^0, \bar{k}^0, \beta_v^0, \beta_v^{\mathtt{lo}}, \varepsilon_\nabla, j_{\mathtt{final}}$)
  repeat
    $c^{(\cdot,1)} = \mathcal{T}^{\rightarrow}_{m_A^{(0,0)}}[p^k, \bar{k}^k]$          ⇝ (6.7a)–(6.7b)
    $m_A^{(0,1)} = m_A^{(0,0)}(1 - c^{(\cdot,1)})$          ⇝ (6.4i)
    for $\iota = 1, \ldots, 5$ do
      $v^{k+1} = \mathcal{R}^{\leftarrow}_{\mathtt{MP}}[m_A^{(0,1)}, c^{(\cdot,1)}, m_D, c_D, \beta_v^k]$
      if det$(\nabla y) < \varepsilon_\nabla$ then
        break
      $\beta_v^k = \beta_v^{k-1} - (\beta_v^{k-1} - \beta_v^k)/2$
      stop = true
    $c_P^{(1,\cdot)} = \mathcal{R}^{\rightarrow}[v, c_D]$          ⇝ (6.4f)–(6.4g)
    $p^{k+1}, \bar{k}^{k+1} = \mathcal{T}^{\leftarrow}_{m_A^{(0,0)}}[c_P^{(1,\cdot)}]$          ⇝ (6.16)
    if stop or $\beta_v^k \leqslant \beta_v^{\mathtt{lo}}$ then
      $j_{\mathtt{final}} = j_{\mathtt{final}} - 1$
    else
      $\beta_v^{k+1} = \beta_v^k/10$
    $k = k + 1$
  until $j_{\mathtt{final}} < 0$
```

```
func picard-ma($v^0, p^0, \bar{k}^0, \beta_v^0, \beta_v^{\mathtt{lo}}, \varepsilon_\nabla, j_{\mathtt{final}}$)
  repeat
    $m_A^{(1,0)} = \mathcal{R}^{\rightarrow}[v, m_A^{(0,0)}]$          ⇝ (6.12e)–(6.12g)
    update $\rho(m_A^{(1,0)})$ and $k(m_A^{(1,0)})$
    $c^{(\cdot,1)} = \mathcal{T}^{\rightarrow}_{m_A^{(1,0)}}[p^k, \bar{k}^k]$          ⇝ (6.12a)–(6.12b)
    $p^{k+1}, \bar{k}^{k+1} = \mathcal{T}^{\leftarrow}_{m_A^{(1,0)}}[c_D]$          ⇝ (6.19)
    for $\iota = 1, \ldots, 5$ do
      $v^{k+1} = \mathcal{R}^{\leftarrow}_{\mathtt{MA}}[m_A^{(1,0)}, m_D, c^{(\cdot,1)}, \beta_v^k]$
      if det$(\nabla y) < \varepsilon_\nabla$ then
        break
      $\beta_v^k = \beta_v^{k-1} - (\beta_v^{k-1} - \beta_v^k)/2$
      stop = true
    if stop or $\beta_v^k \leqslant \beta_v^{\mathtt{lo}}$ then
      $j_{\mathtt{final}} = j_{\mathtt{final}} - 1$
    else
      $\beta_v^{k+1} = \beta_v^k/10$
    $k = k + 1$
  until $j_{\mathtt{final}} < 0$
```

ALGORITHM 6.1 *Pseudo-code for the moving-patient (left) and moving-atlas (right) Picard-iteration algorithm with embedded β_v-continuation scheme. The regularization weight is reduced by a factor of* 10 *in every Picard-iteration. The optimal regularization parameter $\beta_v^\star$ is found via a backtracking strategy based on the lower bound ε_∇.*

Warm-Starts and Grid-Continuation. For both Picard-iterations, SIBIA supports a warm-start mechanism which is useful for an F-cycle type multi-scale ***grid-continuation*** approach, as known from full approximation schemes (FAS). Starting on a very coarse spatial grid, we solve a series of optimization problems with gradually refined spatial resolution, allowing for the representation of more and more details in the input images along with more accurate and complex parameter estimations. I.e., we consider a sequence of initially convex but increasingly oscillatory approximations of the objective function, which are minimized over a sequence of increasingly finer discretizations of state, adjoint, and parameter spaces. The solutions for the inversion variables v^k, p^k and $\bar{k}^k$ from previous coarse-grid solves are ***prolongated***[16] to the refined mesh and used as initial guess for the inversion solvers. Similar to the parameter-continuation scheme for the regularization weight, this multi-scale approach convexifies the optimization problem by gradually including more details and image features, once the parameter estimates approaches the solution[17]. This further aids to prevent sub-optimal local minima solutions and provides good initial guesses for high accuracy solves to keep the Newton-type methods within the basin of attraction of the global minimum and the regime of fast convergence. On the other hand, it drastically ***reduces the time-to-solution*** due to a significantly reduced number of unknowns on coarser grids. Both properties are of great value for the coupled solution schemes and are an important ingredient to tackle the large-scale, highly non-linear and non-convex problems.

[16] Note that only the velocity vector field v needs to be prolongated from a coarse grid representation to the refined grid. We do this with high-order interpolation schemes (order three or higher). Due to the parametrization of the initial condition, the tumor parameters do not need to be prolongated in between grids, unless the Gaussian basis functions remain unchanged; the obtained parameter solution vector p from a coarser grid is copied as is to the finer grid.

[17] Sharp resolution of features, interfaces and reconstructions may hinder the optimization process in an early stage, causing it to get trapped in a local minimum, which might be far from the global minimum.

```
func grid-continuation($(\beta_{v,\ell}^{0})_{\ell=0}^{\texttt{fine}}$, $(\beta_{v,\ell}^{\texttt{lo}})_{\ell=0}^{\texttt{fine}}$, $j_{\texttt{final}}$)
    $\boldsymbol{v}_0^0 = \mathbf{0}, \boldsymbol{p}_0^0 = \mathbf{0}, \bar{k}_0^0 = \mathbf{0}$
    for $\ell = 0, \ldots,$ fine do
        prolongate $\boldsymbol{v}_\ell^0 = P\boldsymbol{v}_{\ell-1}^k, \boldsymbol{p}_\ell^0 = \boldsymbol{p}_{\ell-1}^k, \bar{k}_\ell^0 = \bar{k}_{\ell-1}^k$
        picard-<ma/mp>($\boldsymbol{v}_\ell^0$, $\boldsymbol{p}_\ell^0$, $\bar{k}_\ell^0$, $\beta_{v,\ell}^0$, $\beta_{v,\ell}^{\texttt{lo}}$, $\varepsilon_\nabla$, $j_{\texttt{final}}$)
```

ALGORITHM 6.2 *Pseudo-code for the grid-continuation scheme for the coupled Picard-iteration solution process. Grid- and parameter-continuation are intermeshed such that increasingly oscillatory optimization problems are solved with educated initial guesses for the inversion variables obtained from prolongated coarse-grid/high-regularization solves.*

The grid-continuation scheme needs to be intertwined with the parameter-continuation for the regularization weight such that high regularization weights correspond to coarse spatial grid resolutions. With this approach, we effectively tackle the problem of over-solving in an early stage of the coupled solution scheme, as the limited solution accuracy due to initially high regularization weights does not justify full resolution solves. Hence, we invest less work in a solution stage where details and features are intentionally omitted to convexify the optimization process. For the combination of grid- and parameter-continuation, different stepping schemes, i.e., different combinations of regularization parameter β_v and corresponding spatial grid resolution are possible. The accuracy of the final result slightly depends on the choice of the stepping scheme, but remains robust if we avoid solving coarse spatial resolutions with a regularization parameter that is too small. Generally speaking, we start with a high regularization weight $\beta_v^0 = 1$ to solve the optimization discretized on the coarsest grid (e.g., $n = 32^3$) and follow the Picard-iteration scheme with embedded parameter-continuation until the prescribed β_v^{lo} for the current spatial resolution is reached. The resulting velocity is then prolongated to a finer resolution $\boldsymbol{v}_\ell^0 = P\boldsymbol{v}_{\ell-1}^k$ and together with the tumor parameters $\boldsymbol{p}_\ell^0 = \boldsymbol{p}_{\ell-1}^k, \bar{k}_\ell^0 = \bar{k}_{\ell-1}^k$ is used to warm-start the coupled solver on a finer discretization level ℓ. The initial regularization weight $\beta_{v,\ell}^0$ on this grid is chosen as the final weight $\beta_{v,\ell-1}^{\text{lo}}$ obtained after convergence of the Picard-iteration (specified minimum regularization weight for this level reached) on the previous grid. Alg. 6.2 shows pseudo-code for a solution scheme combining grid- and parameter-continuation. A typical scheme for instance reads

level 0	$n = 32^3$	$\beta_v^0 = 1\text{E}0$	$\beta_v^{\text{lo}} = 1\text{E}{-1}$
level 1	$n = 64^3$	$\beta_v^0 = 1\text{E}{-1}$	$\beta_v^{\text{lo}} = 1\text{E}{-2}$
level 2	$n = 128^3$	$\beta_v^0 = 1\text{E}{-2}$	$\beta_v^{\text{lo}} = 1\text{E}{-4}$
level 3	$n = 256^3$	$\beta_v^0 = 1\text{E}{-4}$	$\beta_v^{\text{lo}} = 1\text{E}{-4}$

Convergence Criteria. The Picard iteration-type solution schemes given in Alg. 6.1 and presented in detail in §6.3.1-§6.3.2 consist of various sub-solver modules. To obtain a prescribed overall accuracy, the respective sub-solver accuracies and stopping conditions need to be carefully tuned and geared to each other. Neither sub-optimal precision of the solution due to insufficiently accurate sub-results (accuracy bottleneck) nor over-solving, i.e., continuing to reduce the residual for a solver if the error is already in the noise level of the overall solution process, is desirable. The oppositional objectives of high overall solution accuracy and short time-to-solution need to be balanced. We summarize the set of stopping conditions for various components and sub-solvers employed within SIBIA. The actually utilized convergence criteria, i.e., the imposed tolerances, are given in Chapter 7 for the respective numerical experiments.

Picard-Iteration. We finalize the Picard-iteration scheme for our coupled solvers if either the registration regularization parameter reaches its prescribed minimum[18] $\beta_v \leqslant \beta_v^{\mathrm{lo}}$ in the parameter-continuation scheme or if the candidate β_v^k results in a violation of the user defined tolerance ε_∇ for the determinant of the deformation gradient, i.e., $\det(\nabla \boldsymbol{y}) \leqslant \varepsilon_\nabla$ is breached. After termination of the parameter-continuation scheme, we perform j_{final} post-convergence iterations, keeping the regularization weights and grid resolution fixed. Our numerical analysis showed that for real brain MR imaging data[19], the objective function value does not decrease significantly if more than one final iteration is performed. In this stage we typically reached the noise level[20]., primarily introduced by an order $\mathcal{O}(1\mathrm{E}{-}2)$ error caused by solving the advection equation, cf. §7.2. We therefore fix the number of final iterations to one for most experiments; compare also Alg. 6.1.

Optimizer – Quasi-Newton and Newton-Krylov. We use inexact Newton-type methods to solve the set of first order optimality conditions for the registration and tumor inversion, respectively. Two different sets of convergence criteria for the Newton-type optimization iteration are used within SIBIA, both addressing the reduced gradient $\boldsymbol{g}^j$ defined in (6.9) after Newton-iteration j. The $(C1)$ stopping conditions

$$
\begin{aligned}
&(C1.i) && \|\boldsymbol{g}^j\|_2 < \mathrm{opttol}\|\boldsymbol{g}^0\|_2,\\
&(C1.ii) && \|\boldsymbol{g}^j\|_2 < 1\mathrm{E}{-}10,\\
&(C1.iii) && j > \mathrm{maxit}_N
\end{aligned}
$$

with $(C1.i) \vee (C1.ii) \vee (C1.iii)$ terminate the iteration if the ***relative gradient norm*** $\|\boldsymbol{g}^j\|_{\mathrm{rel}} := \|\boldsymbol{g}^j\|_2/\|\boldsymbol{g}^0\|_2$ drops below a user specified tolerance opttol. The $(C2)$ stopping conditions

$$
\begin{aligned}
&(C2.i) && \mathcal{J}[\boldsymbol{x}^{j-1}] - \mathcal{J}[\boldsymbol{x}^j] < \mathrm{opttol}(1 + \mathcal{J}[\boldsymbol{x}^0]),\\
&(C2.ii) && \|\boldsymbol{x}^{j-1} - \boldsymbol{x}^j\|_2 < \sqrt{\mathrm{opttol}}(1 + \|\boldsymbol{x}^0\|_2),\\
&(C2.iii) && \|\boldsymbol{g}^j\|_2 < \sqrt[3]{\mathrm{opttol}}(1 + \mathcal{J}[\boldsymbol{x}^0]),\\
&(C2.iv) && \|\boldsymbol{g}^j\|_2 < 1\mathrm{E}{-}10,\\
&(C2.v) && \|\boldsymbol{g}^j\|_2 < \kappa\|\boldsymbol{g}^0\|_2,\\
&(C2.vi) && j > \mathrm{maxit}_N
\end{aligned}
$$

with $\{(C2.i) \wedge (C2.ii) \wedge (C2.iii)\} \vee (C2.iv) \vee \{(C2.vi) \wedge (C2.v)\}$ are inspired by [Man15; Mod09] (for a discussion see [Gil81], p. 305ff) and use a combination of the relative change of (i) the norm of the gradient $\boldsymbol{g}^j$, (ii) the objective function $\mathcal{J}$, and (iii) the inversion variable $\boldsymbol{x} \in \{\boldsymbol{v}, (\boldsymbol{p}, \bar{k})\}$; all three conditions must be fulfilled at the same time and are controlled by a single, user defined parameter opttol > 0. As a safeguard, iterations with either set of stopping conditions always terminate in case the absolute norm of the gradient drops below 1E−10. Furthermore, to limit excessive time-to-solution, to control the solvers' inexactness, and to prevent stagnation, a maximum number of allowed Newton iterations maxit_N can be prescribed. For the $(C2)$ set of stopping conditions, we always enforce a relative gradient reduction of at least κ, $0.1 \leqslant \kappa < 1$.

If Newton-Krylov is used for optimization, we solve the linearized, reduced space KKT system

[18] This is a lower bound which serves as safeguard against numerical instabilities that might occur if $\boldsymbol{v}$ becomes highly irregular and is chosen based on numerical experience for a set of input images. For a discussion, see [Man17a]

[19] Real brain imaging data is typically non-smooth, detailed, and has high frequency features. Furthermore, the brain geometry defines the diffusion coefficients and thus, complex brain structure hardens the diffusion solve due to sharp coefficients and coefficient jumps. The expected accuracy for real data is therefore lower.

[20] For very smooth data we can show further convergence and reduction of the objective function value for an increasing number of final iterations, see §7.4.2

in each Newton-step inexactly based on an Eisenstat/Walker convergence criterion [Dem82; Eis96] with a forcing sequence[21] that yields super-linear or quadratic convergence[22]. Details on the inexact Krylov solve and convergence criterion are given in §6.4. To prevent a prohibitively high number of inner Krylov iterations, stagnation, or runtime limitation, the maximum number of allowed Krylov iterations can be specified by maxit_K.

The utilized set of stopping conditions used for the numerical results in Chapter 7 is indicated therein; for almost all cases, we use the gradient reduction based convergence criterions $(C1)$ with the parameters $\text{opttol}_T, \text{maxit}_{N,T}, \text{maxit}_{K,T}$ for the tumor inversion, and the $(C2)$ stopping conditions for the registration controlled by the parameters $\text{opttol}_R, \text{maxit}_{N,R}, \text{maxit}_{K,R}$. In our numerical studies, we found that reducing the gradients for the tumor and registration solver by more than three and two orders of magnitude, respectively does not improve the overall accuracy if real brain data is considered, as we already reached the noise level[23]. The reference gradient $\boldsymbol{g}^0$ for both, registration and tumor inversion solver, are computed with respect to a zero initial guess, i.e., $\boldsymbol{g}^0 = \boldsymbol{g}_{\boldsymbol{v}}(\boldsymbol{v} = 0)$ and $\boldsymbol{g}^0 = \boldsymbol{g}_{\boldsymbol{p}}(\boldsymbol{p} = 0)$, respectively. We keep the reference gradients fixed throughout the Picard-iterations, i.e., the requested accuracy for the sub-solvers does not increase as the solution process proceeds.

Data-driven Selection of Gaussian Basis Functions. For the biophysical tumor inversion solver, the initial condition $c(0)$ is parametrized by a linear combination of Gaussians φ_j represented by the matrix Φ. In a discrete setting, the entries of $\Phi \in \mathbf{R}^{n,n_p}$, $n = n_1 n_2 n_3$, $n_i \in \mathbf{N}$, are given by

$$\Phi_{i,j} = \varphi_j(\boldsymbol{x}_i) \tag{6.23}$$

for $i = 1, 2, \ldots, n$ and $j = 1, \ldots n_{\boldsymbol{p}}$, and $\boldsymbol{x}_i \in \mathbf{R}^3$ denotes the i-th grid coordinate of a regular grid of n points stored in lexicographical ordering, cf. §5.3.1. With this parametrization, we can encode the initial conditions with a small number of parameters $\boldsymbol{p} \in \mathbf{R}^{n_p}$, and, in addition, it allows us to effectively deal with multi-focal tumors. We set the Gaussians in CSF to zero to prevent a spurious diffusion of cancerous cells into the area associated with CSF; that is, we define a filter $m_F(\boldsymbol{x}) = 1$ if $(m_{WM}(\boldsymbol{x}) > 0.1 \vee m_{GM}(\boldsymbol{x}) > 0.1) \wedge (m_{CSF}(\boldsymbol{x}) < 0.8)$ and $m_F(\boldsymbol{x}) = 0$ otherwise, and compute

$$\varphi_j(\boldsymbol{x}) = \exp(-(\sigma \|\boldsymbol{x} - \boldsymbol{x}_j\|_2)^2) m_F(\boldsymbol{x}),$$

for all $\boldsymbol{x} \in \Omega$. Here, $\sigma > 0$ is the standard deviation and $\boldsymbol{x}_j \in \mathbf{R}^3$ is the center of the j-th Gaussian basis. In the following, we characterize Gaussians by the tuple $(\boldsymbol{x}_{\boldsymbol{i}}, \sigma)$, i.e., by their center $\boldsymbol{x}_{\boldsymbol{i}}$ and their standard deviation σ.

We are interested in keeping the dimensionality $n_{\boldsymbol{p}} \in \mathbf{N}$ of the inversion variable $\boldsymbol{p}$ comparatively small, which is why using a set of Gaussian basis functions that covers the entire simulation domain is prohibitive. Yet, we opt for a fully automated and user friendly inversion process without any necessity for manual tuning. We therefore place a set of Gaussians covering the area in the brain which is afflicted by the tumor. We provide different strategies to select the number, spatial positioning, and standard deviation σ of the basis functions, based on the probability map for tumor cells for the given patient. Note that we place the basis functions, based on input tumor data in the patient space, though we use them to simulate tumor growth in the atlas space for the

[21] The forcing sequence depends on the gradient norm and the termination of the inner Krylov solve is not based on a user specified tolerance.

[22] To obtain quadratic convergence, the Hessian system needs to be solved more accurately. Our numerical experiments showed that the overall time-to-solution reduces for a slightly worse super-linear convergence rate of the Newton solver, which requires much less work in the inner Krylov iteration.

[23] The solution of the advection equation introduces an order $\mathcal{O}(1\text{E}{-2})$ error for typically obtained registration velocities and input images; see §7.2.

moving-patient scheme.[24] Furthermore, the initial spatial positioning of the Gaussians remains unchanged throughout the coupled inversion process: Only for the moving-atlas scheme, we update the above mentioned filtering of the basis functions, but not their spatial positioning. In the sequel, we briefly discuss two available alternatives within SIBIA to automatically define the Gaussian basis function based on given tumor data.

Tumor Diameter-Based Regular Grid Selection. Using this selection mode, the Gaussians are arranged on a regular cubic grid with spacing δ in each coordinate direction, covering the area of the patient tumor. To this end, we first compute the center of mass of the patient tumor, given by the probability map c_D as $\boldsymbol{x}_{CM} = \int_\Omega {}^{c_D(\boldsymbol{x})\boldsymbol{x}}/_{M}\, d\boldsymbol{x}$ with $M = \int_\Omega c_D(\boldsymbol{x})\, d\boldsymbol{x}$ and the (approximation of the) diameter $d_T := \max\{\|\boldsymbol{x}_i - \boldsymbol{x}_j \mid c_D(\boldsymbol{x}_i, c_D(\boldsymbol{x}_j) > 0.5\}$. We then uniformly distribute a specified number of n_p Gaussians within a cube d_T^3 around the center of mass $\boldsymbol{x}_{CM}$. The condition number $\kappa(\Phi^T\Phi)$ of the matrix Φ affects the conditioning of the non-linear optimization for the tumor solver; we generate Gaussian bases with condition numbers that are invariant with respect to the tumor size and shape. The condition number of Φ depends on the number n_p of basis functions[25], but is invariant with respect to the tumor diameter d_T if the relative overlap of the Gaussians is fixed. We therefore set the spacing between Gaussian centers to $\delta = 1.5\sigma$ and choose $\sigma = {}^{d_T}/_{1.5(\lfloor \sqrt[3]{n_p} \rfloor - 1)}$ dependent on the tumor expansion d_T. Thus, the accuracy and degree of fidelity to represent initial conditions as a superposition of Φ depends on the size and shape of the patient tumor in relation to the number n_p of allowed Gaussians. As a result, the value for n_p to obtain sufficient interpolation accuracy for the parametrization needs to be chosen manually in a pre-processing step. An exemplarily illustration for this method is given in Fig. 6.4.

Tumor Shape- and Intensity-Based Adaptive Selection. Starting from an equally scattered candidate set

$$\mathcal{B}^\delta = \left\{ (\boldsymbol{x_i}, \sigma) \mid \boldsymbol{x} = ({}^{\delta}/_{2} + {}^{2\pi i_j}/_{\delta})_{j=1}^3, i_j \in \{1, \ldots, {}^{2\pi}/_{\delta}\} \right\}$$

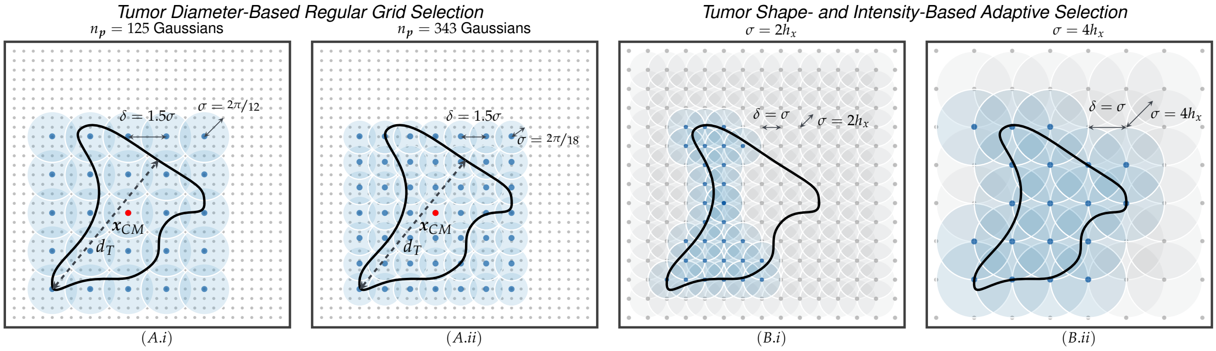
of possible Gaussian basis function locations in the entire simulation domain, this approach adaptively selects those basis functions for which the amount of tumor cells (volume fraction) within the standard deviation ball $B^\sigma(\boldsymbol{x_i})$ around the candidate location $\boldsymbol{x_i}$ exceeds a prescribed threshold τ_V. In other words, for all candidates $(\boldsymbol{x_i}, \sigma) \in \mathcal{B}^\delta$, we compute the ***volume fraction*** of tumor cells within the standard deviation σ, and activate or select the candidate, if

$$\frac{\int_{B^\sigma(\boldsymbol{x_i})} c_D(\boldsymbol{x})\, d\boldsymbol{x}}{\int_{B^\sigma(\boldsymbol{x_i})} 1\, d\boldsymbol{x}} > \tau_V$$

Otherwise, the candidate function is discarded. This results in a set Φ of n_p basis functions, resembling the shape of the tumor input data. The number n_p of basis functions can be controlled by (*i*) requiring a higher volume fraction τ_V of tumor cells within $B^\sigma(\boldsymbol{x_i})$, (*ii*) varying the standard deviation σ of the Gaussians, and by (*iii*) varying the spacing $\delta = \iota h_x$, $\iota = 1, 2, \ldots$ in multiples of the grid resolution h_x. The latter significantly influences the condition number of Φ and is therefore typically fixed to $\delta = \sigma$ or $\delta = 2\sigma$. If the number of selected basis functions exceeds a pre-defined upper bound $n_p > n_p^{\text{hi}}$, the process is restarted with a higher choice for the volume fraction threshold τ_V or larger σ. An illustration is given in Fig. 6.4.

[24] Advecting Gaussian basis function to the atlas space would significantly complicate the formulation and the respective Picard iteration-type solution scheme.

[25] For a standard deviation of $\sigma = {}^{\pi}/_{10}$ and spacing $\delta = 1.5\sigma$, we exemplarily obtain condition numbers $\kappa(\Phi^\star\Phi)$ of 436, 4212, and 12096 for $n_p = 27$, $n_p = 125$, and $n_p = 343$ basis functions, respectively.



$(A.i)$ $(A.ii)$ $(B.i)$ $(B.ii)$

FIGURE 6.4 *Illustration of the different Gaussian selection modes.* ***(A.i) and (A.ii): diameter-based regular grid selection.*** *The Gaussian basis functions are placed as a regular grid around the center of mass of the tumor input concentration. The size of the bounding-box is determined from the maximum diameter of connected tumor tissue; the standard deviation σ of the Gaussians is chosen such that a prescribed number of n_p equally spaced Gaussian basis functions cover the tumor region within the bounding-box (compare $(A.i)$ with $\sigma = 2\pi/12$ for a diameter of $d_T = \pi$ and $n_p = 125$ basis functions and $(A.ii)$ with $\sigma = 2\pi/18$, $d_T = \pi$ and $n_p = 343$; the relative spacing is fixed to $\delta = 1.5\sigma$).* ***(B.i) and (B.ii): shape- and intensity-based adaptive selection.*** *Starting from a standard deviation $\sigma = 2h_x$ and spacing of $\delta = \sigma$, the entire domain is seeded with candidates $(\boldsymbol{x_i}, \sigma) \in \mathcal{B}^\delta$ for Gaussian basis functions (light gray shaded circles). If for a candidate point $\boldsymbol{x_i}$ the volume fraction for tumor cells within the σ-ball exceeds a specified threshold, i.e., $\int_{B^\sigma(x_i)} c_D(x)\,dx / \int_{B^\sigma(x_i)} 1\,dx > \tau_V$, the respective Gaussian is activated and added to Φ (light blue shaded circles); otherwise, the candidate location is discarded. If the number of activated Gaussians exceeds a specified upper bound $n_p > n_p^{hi}$, the process is restarted with a higher required volume fraction, or doubled standard deviation $\sigma^{4h} = 2\sigma^{2h}$ and a new set of candidate locations $\mathcal{B}^{2\delta}$, figure $(B.ii)$.*

Both selection modes are implemented in parallel on distributed data. For the tumor shape- and intensity-based adaptive selection, this involves communication with all neighboring processes whose domain is overlapped by the standard deviation ball of a location $\boldsymbol{x_i}$ owned by the local processor.

A Sparsity Constraint for the Tumor Inversion. The tumor inversion problem is non-unique in the sense that different combinations of (i) the initial condition for the tumor concentration, (ii) the characteristic diffusivities for white and gray matter, i.e., k_w and k_g, respectively, (iii) the net proliferation rate ρ, and (iv) the considered time horizon for tumor simulation may result in identical simulated final state tumor cell distributions. For biophysical parameter estimation we therefore need to fix some values and invert for a subset of the parameters only, as discussed earlier. Generally speaking, when inverting for the initial condition of the tumor cell distribution, we are interested in the initial condition which appears to be a point source, i.e., as sparse as possible, representing the true initial cell mutation of a glial cell into a cancerous cell. With this condition, the time horizon for the tumor evolution is automatically fixed, which eliminates a further degree of freedom. The L_2 Thikonov-type regularization considered in the formulation of our inverse tumor problem in §5.3.1 alleviates bad conditioning, but does not result in sparse initial conditions; the solution process arrives at a richer initial condition that does not reflect the time point of the initial cell mutation.

We therefore enhance the tumor inversion solver by a ***sparsity constraint*** to acquire the desired effect. The sparsity constraint is implemented as a L_1-regularization operator $\mathcal{S}^1_{\boldsymbol{p}}\,[\boldsymbol{p}] = \frac{1}{2}\|\boldsymbol{p}\|_{L^1(\Omega)}$ (see (6.5)) penalizing the number of non-zero entries in the coefficient vector $\boldsymbol{p}$ of the initial condition parametrization $\Phi\boldsymbol{p}$. This regularization operator is non-differentiable and cannot be solved with our derivative-based Newton-type approaches. To solve the L^1-regularized optimization problem, we therefore employ the General Iterative Shrinkage Thresholding (GIST) [Gon13] algorithm, based on repeated solutions of a representative ***proximal operator problem***:

$$\boldsymbol{p}^{j+1} = \mathcal{P}(\boldsymbol{p}^j, \alpha^j) \quad \text{with} \;\; \mathcal{P}(\boldsymbol{p}^j, \alpha^j) = \arg\min_{\boldsymbol{p}} \; \frac{1}{2}\|\boldsymbol{p} - \boldsymbol{p}^j_{sd}\|^2_{L^2(\Omega)} + \alpha^j \frac{\beta_{\boldsymbol{p}}}{2}\|\boldsymbol{p}\|_{L^1(\Omega)}, \tag{6.24a}$$

where $\boldsymbol{p}^j_{sd} = \boldsymbol{p}^j - \alpha^j \nabla \mathcal{D}_c(c^{(\cdot,1)}, c_T)$ is the gradient-descent guess in the direction $\nabla \mathcal{D}_c(c^{(\cdot,1)}, c_T)$ with step length α^j. Depending on the formulation, the target tumor data is either the warped-to-atlas patient data $c_T = c_P^{(1,\cdot)}$ (moving patient formulation) or the original data $c_T = c_D$ (moving atlas formulation). For L^1-regularization, the solution of the proximal operator problem can be computed analytically:

$$\mathcal{P}(\boldsymbol{p}^j, \alpha^j) = \operatorname{sign}(\boldsymbol{p}^j_{sd}) \max(0, |\boldsymbol{p}^j_{sd}| - \alpha^j \beta_{\boldsymbol{p}}) \tag{6.24b}$$

We use Armijo line-search to determine the step length α^j, using the following stopping criteria

$$|\mathcal{J}_T[\boldsymbol{p}^{j+1}] - \mathcal{J}_T[\boldsymbol{p}^j]| < \tau_J(1 + \mathcal{J}_T[\boldsymbol{p}^0)] \tag{6.25a}$$

$$\|\boldsymbol{p}^{j+1} - \boldsymbol{p}^j\|_{L^\infty(\Omega)} < \sqrt{\tau_J}(1 + \|\boldsymbol{p}^{j+1}\|_{L^\infty(\Omega)}) \tag{6.25b}$$

where $\mathcal{J}_T = \mathcal{J}_{\mathrm{T}_{\boldsymbol{m}_A^{(0,0)}}}$ for moving patient and $\mathcal{J}_T = \mathcal{J}_{\mathrm{T}_{\boldsymbol{m}_A^{(1,0)}}}$ for moving atlas. $\tau_J > 0$ is a user defined tolerance.

To estimate a suitable value of the regularization weight $\beta_{\boldsymbol{p}}$, we employ a parameter-continuation scheme. Starting with a large value for $\beta_{\boldsymbol{p}}$ in the order of the initial gradient $\|\nabla \mathcal{D}_c(c^{(\cdot,1)}, c_T)\|_{L^\infty(\Omega)}$ of the distance measure, we perform a binary search in subsequent GIST iterations based on the Hoyer sparsity measure [Hur08], $\mathcal{H}_s(\boldsymbol{p}) \in [0, 1]$, defined as

$$\mathcal{H}_s(\boldsymbol{p}) = \frac{\ell(\boldsymbol{p}) - \|\boldsymbol{p}\|_{L^1(\Omega)}/\|\boldsymbol{p}\|_{L^\infty(\Omega)}}{\ell(\boldsymbol{p}) - 1}, \tag{6.26}$$

where $\ell(\boldsymbol{p})$ is the length of $\boldsymbol{p}$. A high value of $\mathcal{H}_s(\boldsymbol{p})$ indicates a sparse parameter vector $\boldsymbol{p}$; we use a tolerance of 0.95 to classify sparse solutions. We reduce the regularization weight (bisection), while the obtained solution is classified as sparse ($\mathcal{H}_s(\boldsymbol{p}) > 0.95$) and restart the optimization with new regularization weight using the previous solution as initial guess; in case of insufficient sparsity of the solution, the regularization weight is increased.

Due to the slow convergence of these first order techniques (gradient-descent), we employ a two step solution strategy: We first obtain a sparse initial guess $\boldsymbol{p}_{L^1}$ for the tumor initial condition by solving the L^1-regularized optimization problem for a large regularization weight $\beta_{\boldsymbol{p}}$ obtained from the above described parameter-continuation scheme. A large regularization weight pronounces the convex part of the objective functional and fosters fast convergence: The obtained solution is not accurate with respect to intensity, i.e., the absolute coefficient values in the parameter vector $\boldsymbol{p}$. However, it effectively captures the sparsity (non-zero entries in $\boldsymbol{p}$ are penalized stronger using L^1-regularization). From the sparse coefficient vector $\boldsymbol{p}_{L^1}$, we identify the Gaussian basis functions that are important for the reconstruction of the observed patient tumor. This information is used to define a ***weighting matrix*** applied in a second ***weighted L^2-regularization*** solve to improve the accuracy of the solution (identify the exact values/intensities of the coefficient vector $\boldsymbol{p}$). The modified L^2-regularizer is given by an induced weighted L^2-norm

$$\|\boldsymbol{p}\|_W := \left(\sum_{i=1}^{n_p} w_i \cdot p_i^2 \right)^{\frac{1}{2}} \tag{6.27}$$

which penalizes entries in $\boldsymbol{p}$ that do not correspond to selected basis functions from the L^1-step, i.e., (almost) zero entries in $\boldsymbol{p}_{L^1}$. The weights w_i are given as

$$w_i = \begin{cases} w^{\text{large}} & \text{if } |p_{L^1,i}| < \tau_S \|\boldsymbol{p}_{L^1}\|_{L^\infty(\Omega)} \\ w^{\text{small}} & \text{else.} \end{cases}$$

where τ_S is a user defined tolerance. This approach allows us to speed up convergence by relying on second-order methods for the differentiable L^2-regularized optimization problem, while retaining the sparse nature of the initial condition. The enhancement of the tumor inversion solver by the sparsity constraint including the modified solver for the non-differentiable L^1-regularization is a result of joint effort with Shashank Subramanian; a joint paper is in preparation.

6.4 Newton Methods for the Reduced Systems

In the previous section, we considered mechanisms and concepts required for steering, stabilizing and automating the coupled Picard-iteration. We further described specific modifications or enhancements of sub-component solvers to facilitate the coupling. In the following we focus on the sub-component systems and review reduced space methods, in particular reduced Newton methods, in a generic formulation. Apart from accurate and meaningful results (whether biophysically diagnostic parameters or meaningful image registration mappings), the ***time-to-solution*** is extremely critical for our application. Our methodology is intended to ultimately be used in clinical routine, where compute power is limited and time is precious. Therefore, we have to further optimize solvers and algorithms to reduce the response time of the simulation pipeline. The grid- and parameter-continuation schemes presented in the previous section already help to reduce the time-to-solution significantly, however, we are furthermore interested in investigating if approaches exploiting different degrees of inexactness for Newton-type iterations further speed up the simulation. Specifically, we examine the performance of quasi-Newton methods for the tumor inverse problems (6.16) and (6.19).

A standard approach to cope with the system of coupled, non-linear PDEs that establishes the optimality conditions of a PDE-constrained optimization problem is to employ a reduced space method, resulting in a reduced system of only the dimensionality of the inversion variables. We mentioned the basic concept of such a method, combined with a Gauß-Newton-Krylov solution scheme, in §5.3.3. Here, we abstract from the given problem and describe the Newton method for large-scale PDE constrained optimization in a more generic and detailed fashion in §6.4.1. We describe general inexact Newton approaches for the reduced system in §6.4.2. In §6.4.5, we point out how the Gauß-Newton-Krylov (GNK) approach can be effectively replaced by quasi-Newton type solvers and analyze their computational complexity. Numerical experiments, that compare GNK versus quasi-Newton methods based on their time-to-solution in relation to the quality of the final solution, can be found in §7.4.5.

6.4.1 Newton Methods for the Lagrangian Gradient

In the sequel, we abstract from specific optimization problems to a generic formulation and describe the Newton method for the solution of the optimality system arising from stationarity of the Lagrangian of a generic optimal control problem. Our description is based on [Pet11; Akc06]. Assume we are given the following abstract PDE constrained optimization problem

$$\min_q \mathcal{J}(u,q) \qquad \text{subject to} \quad c(u,q) = 0, \tag{6.28}$$

where u are the state variables, q the decision variables, $\mathcal{J}$ is the objective function, and c are the PDE constraints (state equations, or forward model). The constrained optimization problem (6.28) can be translated into an unconstrained one by defining the Lagrangian function

$$\mathcal{L} = \mathcal{J}(u,q) + \lambda^T c \tag{6.29}$$

which takes into account the constraints c via an inner product with the Lagrange multiplier λ. The latter is also called ***adjoint*** variable. We require stationarity of the Lagrangian $\mathcal{L}$ with respect to u, q and λ. Taking variations yields the (non-linear) system of first order optimality conditions (KKT-system):

$$\left\{ \begin{array}{c} g_u := \delta_u \mathcal{L} \\ g_q := \delta_q \mathcal{L} \\ g_\lambda := \delta_\lambda \mathcal{L} \end{array} \right\} = \left\{ \begin{array}{c} \delta_u \mathcal{J}(u,q) + (\delta_u c(u,q))^T \lambda) \\ \delta_q \mathcal{J}(c,q) + (\delta_q c(u,q))^T \lambda) \\ c(u,q) \end{array} \right\} = \left\{ \begin{array}{c} \mathcal{J}_u + c_u^T \lambda \\ \mathcal{J}_q + c_q^T \lambda \\ c \end{array} \right\} = \mathbf{0}. \tag{6.30}$$

With, g_u, g_q and g_λ we denote the first order variations or gradients of $\mathcal{L}$ with respect to u, q, and λ, respectively. $\mathcal{J}_u$ and $\mathcal{J}_q$ are the variations of the objective function $\mathcal{J}$ and c_u, c_q are the gradients of the state equations c with respect to u and λ, respectively. The optimality system (6.30) portrays a set of coupled, non-linear PDEs and is typically characterized by ill-conditioning, indefiniteness, and a large number of unknowns. The simplest strategy of using a globalized gradient-descent approach fails in most practical applications due to its slow convergence properties. Thus, we consider Newton-type methods that are characterized by their fast convergence, but require second order variational-derivatives of the Lagrangian. Newton-type methods solve the Newton system

$$H\hat{\boldsymbol{x}} = -\boldsymbol{g} \tag{6.31}$$

for increments $\hat{\boldsymbol{x}} = (\hat{u}, \hat{q}, \hat{\lambda})^T$ and update the variables with $(u,q,\lambda) := (u,q,\lambda) + \alpha(\hat{u},\hat{q},\hat{\lambda})$. H is the Hessian operator, i.e., second order variation $\delta^2\mathcal{L}$ of the Lagrangian and $\boldsymbol{g} = (g_u, g_q, g_\lambda)^T$ is the first order variation $\delta\mathcal{L}$ of the Lagrangian, i.e., the detailed system (6.31) reads

$$\begin{bmatrix} \delta_u\delta_u\mathcal{L} & \delta_u\delta_q\mathcal{L} & \delta_u\delta_\lambda\mathcal{L} \\ \delta_q\delta_u\mathcal{L} & \delta_q\delta_q\mathcal{L} & \delta_q\delta_\lambda\mathcal{L} \\ \delta_\lambda\delta_u\mathcal{L} & \delta_\lambda\delta_q\mathcal{L} & \delta_\lambda\delta_\lambda\mathcal{L} \end{bmatrix} \left\{ \begin{array}{c} \hat{u} \\ \hat{q} \\ \hat{\lambda} \end{array} \right\} = - \left\{ \begin{array}{c} \delta_u\mathcal{L} \\ \delta_q\mathcal{L} \\ \delta_\lambda\mathcal{L} \end{array} \right\}. \tag{6.32}$$

Replacing the abstract terms above by the respective discretized variations and functions, yields the following Newton scheme in block matrix notation

$$
\begin{bmatrix} J_{uu} + c_{uu}^T\lambda & J_{uq} + c_{uq}^T\lambda & c_u^T \\ J_{qu} + c_{qu}^T\lambda & J_{qq} + c_{qq}^T\lambda & c_q^T \\ c_u & c_q & 0 \end{bmatrix} \begin{Bmatrix} \hat{u} \\ \hat{q} \\ \hat{\lambda} \end{Bmatrix} = \begin{bmatrix} W_{uu} & W_{uq} & U^T \\ W_{qu} & W_{qq} & Q^T \\ U & Q & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \hat{u} \\ \hat{q} \\ \hat{\lambda} \end{Bmatrix} = - \begin{Bmatrix} g_u \\ g_q \\ g_\lambda \end{Bmatrix} \tag{6.33}
$$

We denote the discretized functions and operators in boldface notation, i.e., $\boldsymbol{u}$, $\boldsymbol{q}$, $\boldsymbol{\lambda}$ are the discretized functions u, q and λ (likewise for the increments $\hat{\boldsymbol{u}}, \hat{\boldsymbol{q}}, \hat{\boldsymbol{\lambda}}$), and $\boldsymbol{J}_{xy}$, c_x, c_{xy}, $x, y \in \{\boldsymbol{u}, \boldsymbol{q}\}$ denote the discretized first and second order variational derivative operators of the objective function $\mathcal{J}$ and the constraints c, respectively. For better readability we replace the components containing second order variational-derivatives of the Lagrangian by $\boldsymbol{W}_{uu}$, $\boldsymbol{W}_{uq}$, $\boldsymbol{W}_{qu}$, and $\boldsymbol{W}_{qq}$ and the Jaobians of the state equations, with $\boldsymbol{c}_u =: \boldsymbol{U}$, and $\boldsymbol{c}_q =: \boldsymbol{Q}$, respectively. $\boldsymbol{g}_u$, $\boldsymbol{g}_q$, and $\boldsymbol{g}_\lambda$ are the discrete gradients of the Lagrangian.

This linear system is known as KKT system. Assuming the discretized state and adjoint fields to be of dimensionality n with n_q control parameters, i.e., $\boldsymbol{u}, \boldsymbol{\lambda} \in \mathbf{R}^n$, $\boldsymbol{q} \in \mathbf{R}^{n_q}$, the KKT coefficient matrix $\boldsymbol{H}$ is of dimensionality $(2n + n_q) \times (2n + n_q)$. For most realistic scenarios, n is extremely large as state and adjoint fields represent 3D data, while n_q either may be in the same order, or significantly smaller if a good parametrization of the control space is applicable. In our case, n_q is significantly smaller than the problem dimensionality n for the tumor solver, while, for the registration solver $n_q = 3n$. Either way, direct solvers for the system (6.33) are ruled out and most iterative full space solvers that simultaneously solve for incremental state, adjoint and control variables struggle with the ill-conditioning, indefiniteness, and large scale of the KKT system. These difficulties motivate the use of *reduced space methods*.

6.4.2 Reduced-Space Newton Methods

An excellent overview on reduced space methods for PDE constraint optimization can be found in [Akc06; Akc02; Bir05a; Bir05b; Bir99; Pet11]. This class of solvers assumes the state and adjoint equations to be solved exactly, i.e., it assumes the gradients $\boldsymbol{g}_\lambda$, and $\boldsymbol{g}_u$ to be zero. Algorithmically, this means we compute the gradient of the reduced system by first solving the state equation $\boldsymbol{c}(\boldsymbol{u}, \boldsymbol{q}) = \mathbf{0}$ for $\boldsymbol{u}$, the adjoint $\boldsymbol{J}_u + \boldsymbol{c}_u^T\boldsymbol{\lambda} = \mathbf{0}$ for $\boldsymbol{\lambda}$, and then evaluate the reduced gradient $\boldsymbol{J}_q + \boldsymbol{c}_q^T\boldsymbol{\lambda}$ for this $\boldsymbol{\lambda}$. To solve the reduced gradient system with a Newton-type solver, a block-elimination of the incremental state and adjoint variables $\hat{\boldsymbol{u}}$ and $\hat{\boldsymbol{\lambda}}$ from (6.33) is performed, which yields the ***reduced Hessian system***

$$
\bar{\boldsymbol{H}}\hat{\boldsymbol{q}} = -\boldsymbol{g}_q \tag{6.34}
$$

with coefficient matrix $\bar{\boldsymbol{H}} \in \mathbf{R}^{n_q \times n_q}$. This can be solved for an update $\hat{\boldsymbol{q}}$ of the control variables $\boldsymbol{q}$, resulting in a *reduced Newton* (RN) method. Besides the (usually) significantly smaller dimensionality of the reduced system, which potentially makes the use of direct solvers eligible, the state and adjoint solvers are at the heart of this method, for which existing, efficient, and parallel PDE solvers can be exploited. However, reduced space methods assume exact solves for the state and adjoint equation, which might result in high computational costs and urges for good preconditioners of the respective operators. While for steady-state problems, full space approaches generally outperform reduced space methods; this is highly unclear for optimization problems that are governed by time-dependent PDEs, such as those considered here. Experimental results show good convergence for reduced space methods, even if the state and adjoint equations are not solved up to machine precision [Akc06;

```
func $[g_{q_k}, q_k, u_k, \lambda_k] =$ reducedNewton($q_0$)
  for $k = 0, 1, \ldots$ do
    $g_{q_k} =$ evalgradRN($q_k$)
    solve:  $\bar{H}\hat{q}_k = -g_{q_k}$   for $\hat{q}_k$     $\rightsquigarrow \bigstar$ (6.31)
    repeat
      $\alpha =$ linesearch($q_k, \hat{q}_k$)
    until linesearch successful
    $q_{k+1} = q_k + \alpha\hat{q}_k$
    if $\|g_{q_{k+1}}\| < \varepsilon_{opt}\|g_{q_0}\|$ then
      break
```

ALGORITHM 6.3 *Reduced Newton method. Algorithmic steps taken for the reduced Newton solver, globalized with a non-specified* **linesearch** *method. The core step is the solution of the reduced Hessian system ★; different derived methods can be discriminated in terms of their respective solution strategy. The line-search routine guarantees global convergence and the iteration terminates if the reduced gradient has been sufficiently decreased.*

Akc02]. We solve

$$\boldsymbol{U}\hat{\boldsymbol{u}}_k = -\boldsymbol{Q}\hat{\boldsymbol{q}}_k \qquad \Rightarrow \qquad \hat{\boldsymbol{u}}_k = -\boldsymbol{U}^{-1}\boldsymbol{Q}\hat{\boldsymbol{q}}_k := \bar{\boldsymbol{M}}_u\hat{\boldsymbol{q}}_k \tag{6.35a}$$

$$\boldsymbol{U}^T\hat{\boldsymbol{\lambda}}_k = -(\boldsymbol{W}_{uu}\hat{\boldsymbol{u}}_k + \boldsymbol{W}_{uq}\hat{\boldsymbol{q}}_k) \qquad \Rightarrow \qquad \hat{\boldsymbol{\lambda}}_k = -\boldsymbol{U}^{-T}(\boldsymbol{W}_{uu}\bar{\boldsymbol{M}}_u + \boldsymbol{W}_{uq})\hat{\boldsymbol{q}}_k := \bar{\boldsymbol{M}}_\lambda\hat{\boldsymbol{q}}_k \tag{6.35b}$$

derived from (6.33) for the incremental state and adjoint variables $\hat{\boldsymbol{u}}_k$ and $\hat{\boldsymbol{\lambda}}_k$. Notice that this uses the fact that $\boldsymbol{g}_u = \boldsymbol{g}_\lambda = 0$ for the given iterates $\boldsymbol{u}_k$ and $\boldsymbol{\lambda}_k$. Inserting $\hat{\boldsymbol{u}}$ and $\hat{\boldsymbol{\lambda}}$ from (6.35) into the control variable equation block of (6.33) eliminates the first and third block of equations therein, i.e.,

$$\begin{aligned} \boldsymbol{W}_{qq}\hat{\boldsymbol{q}}_k &= -\boldsymbol{g}_{q_k} - \boldsymbol{W}_{qu}\hat{\boldsymbol{u}}_k - \boldsymbol{Q}^T\hat{\boldsymbol{\lambda}}_k \\ \boldsymbol{W}_{qq}\hat{\boldsymbol{q}}_k &= -\boldsymbol{g}_{q_k} - \boldsymbol{W}_{qu}\bar{\boldsymbol{M}}_u\hat{\boldsymbol{q}}_k - \boldsymbol{Q}^T\bar{\boldsymbol{M}}_\lambda\hat{\boldsymbol{q}}_k, \end{aligned} \tag{6.36}$$

and yields the reduced Hessian $\bar{\boldsymbol{H}}$ as the Schur-complement of the matrix $\boldsymbol{W}_{qq}$; we state the reduced Hessian matvec as

$$\bar{\boldsymbol{H}}\hat{\boldsymbol{q}}_k := (\boldsymbol{W}_{qq} + \boldsymbol{W}_{qu}\bar{\boldsymbol{M}}_u + \boldsymbol{Q}^T\bar{\boldsymbol{M}}_\lambda)\hat{\boldsymbol{q}}_k. \tag{6.37}$$

Inserting the definitions of $\boldsymbol{W}_{qq}$, $\boldsymbol{W}_{qu}$, $\bar{\boldsymbol{M}}_u$, $\bar{\boldsymbol{M}}_\lambda$, $\boldsymbol{U}$, $\boldsymbol{Q}$, $\boldsymbol{W}_{uu}$, and $\boldsymbol{W}_{uq}$ yield the reduced Hessian in more detail

$$\begin{aligned} \bar{\boldsymbol{H}} &= \boldsymbol{W}_{qq} + \boldsymbol{W}_{qu}\bar{\boldsymbol{M}}_u + \boldsymbol{Q}^T\bar{\boldsymbol{M}}_\lambda \\ &= \boldsymbol{W}_{qq} - \boldsymbol{W}_{qu}\boldsymbol{U}^{-1}\boldsymbol{Q} + \boldsymbol{Q}^T\boldsymbol{U}^{-T}(\boldsymbol{W}_{uu}\boldsymbol{U}^{-1}\boldsymbol{Q} - \boldsymbol{W}_{uq}) \\ &= \boldsymbol{J}_{qq} + \boldsymbol{c}_{qq}^T\boldsymbol{\lambda} - (\boldsymbol{J}_{qu} + \boldsymbol{c}_{qu}^T\boldsymbol{\lambda})\boldsymbol{c}_u^{-1}\boldsymbol{c}_q + \boldsymbol{c}_q^T\boldsymbol{c}_u^{-T}((\boldsymbol{J}_{uu} + \boldsymbol{c}_{uu}^T\boldsymbol{\lambda})\boldsymbol{c}_u^{-1}\boldsymbol{c}_q - \boldsymbol{J}_{uq} + \boldsymbol{c}_{uq}^T\boldsymbol{\lambda}). \end{aligned} \tag{6.38}$$

The reduced space Newton algorithm to solve the non-linear optimality system (6.30) via the linearization (6.33) and subsequent block-elimination for the reduced system (6.33) and (6.37) is shown in Alg. 6.3. To guarantee global convergence, the method is globalized via a suitable ***line-search*** method. Popular choices are Armijo-backtracking line-search [Noc06] or Moré-Thuente line-search [Mor94; Noc06]. More details on the line-search methods are given later. Every Newton iteration requires at least (an approximation to) the solution of the linear system (6.31), one evaluation of the reduced gradient, and one objective function evaluation for the line-search. If the latter does not succeed, multiple gradient and objective function evaluations might become necessary. The call to **evalgradRN** involves two PDE solves – one for the state equation and one for the adjoint equation

– corresponding to lines 3 and 1 in (6.30). The solution for the state equation solve can be stored to be used for the objective function evaluation. The computation of the reduced gradient is summarized in Alg. 6.4.

```
1 func $[g_{q_k}] =$ evalgradRN($q_k$)
2   solve:    $c(q_k, u_k) = 0$ for $u_k$                                      $\rightsquigarrow$ ($g_\lambda = 0$, state equation)
3   solve:    $c_u^{-1}(q_k, u_k)^T \lambda_k = -J_{q_k,u_k}(u_k)$ for $\lambda_k$   $\rightsquigarrow$ ($g_c = 0$, adjoint equation)
4   compute:  $g_{q_k} = J_q(u_k, \lambda_k) + c_q(u_k, q_k)\lambda_k$               $\rightsquigarrow$ ($g_{q_k}$, reduced gradient)
```

ALGORITHM 6.4 *Reduced gradient evaluation routine. The core components are efficient solvers for state and adjoint equations (lines 2 and 3) with high accuracy.*

6.4.3 Inexact Reduced Newton Methods

Clearly, the solution of the reduced Hessian system is at the heart of the reduced Newton method. Depending on the pursued strategy, different reduced Newton-type methods can be derived:

A) The classical reduced ***Newton method*** solves the Newton system

$$\bar{\boldsymbol{H}}_k \hat{\boldsymbol{q}}_k = -\boldsymbol{g}_q(\boldsymbol{q}_k) \quad \text{exactly and sets} \quad \boldsymbol{q}_{k+1} = \boldsymbol{q}_k + \alpha \hat{\boldsymbol{q}}_k.$$

The inversion of the Hessian matrix $\bar{\boldsymbol{H}}_k$ is usually obtained via direct solvers such as LU-factorization. No approximations to $\bar{\boldsymbol{H}}_k$ are made, the matrix needs to be explicitly built and stored. Usually, this method is not used due to the prohibitive computational complexity necessary to solve for a search direction $\hat{\boldsymbol{q}}$ exactly. The optimal step length for the Newton step is $\alpha = 1$ if the Newton direction is solved exactly, i.e., $\bar{\boldsymbol{H}}_k^{-1}\boldsymbol{g}_q(\boldsymbol{q}_k)$ is computed exactly.

B) The family of reduced ***inexact Newton-type methods*** [Dem82] allows the Newton update to be solved ***inexactly***. In every inexact Newton step, we seek for a search direction $\hat{\boldsymbol{q}}_k$ and a forcing term $\eta_k \in [0, 1)$, such that

$$\|\boldsymbol{g}_q(\boldsymbol{q}_k) + \bar{\boldsymbol{H}}_k \hat{\boldsymbol{q}}_k\| \leqslant \eta_k \|\boldsymbol{g}_q(\boldsymbol{q}_k)\| \quad \text{is fulfilled and set} \quad \boldsymbol{q}_{k+1} = \boldsymbol{q}_k + \alpha \hat{\boldsymbol{q}}_k. \tag{6.39}$$

The ***forcing term*** η_k (relative error bound) needs to be picked prior to every iteration. Thereby, η_k forces $\|\boldsymbol{g}_q(\boldsymbol{q}_k) + \bar{\boldsymbol{H}}_k \hat{\boldsymbol{q}}_k\|$ to be small in a certain way ($\eta_k = 0$ enforces exactness and yields Newton's method). As a result, the forcing sequence $\{\eta_k\}$ controls the methods' convergence properties and needs to be chosen carefully: while solving inexactly, we don't want to loose the rapid convergence properties of Newton's method. Proof of guaranteed convergence properties for specific choices of $\{\eta_k\}$ can be found in [Dem82; Eis96].

Inequality (6.39) constitutes a quite generic discrimination of inexact Newton methods. Inexactness of the solution, while still fulfilling (6.39), can be obtained in three different ways: (1) The Newton system $\bar{\boldsymbol{H}}_k \hat{\boldsymbol{q}}_k = -\boldsymbol{g}_q(\boldsymbol{q}_k)$ can be solved ***inexactly*** in an iterative way, utilizing exact Hessian information, or (2) it can be solved by inverting an ***approximation*** $\boldsymbol{M}_k \approx \bar{\boldsymbol{H}}_k$ to the Hessian matrix or directly computing an approximation $\boldsymbol{M}_k^{-1} \approx \bar{\boldsymbol{H}}_k^{-1}$ to the inverse Hessian, or to $-\bar{\boldsymbol{H}}_k^{-1}\boldsymbol{g}_q(\boldsymbol{q}_k)$. (3) Lastly, the most typical approach is to combine both forms of inexactness, i.e., solving the system $\boldsymbol{M}_k \hat{\boldsymbol{q}}_k = -\boldsymbol{g}_q(\boldsymbol{q}_k)$ inexactly with $\boldsymbol{M}_k \approx \bar{\boldsymbol{H}}_k$. We consider the following manifestations of inexact Newton-type methods:

B.1) The reduced ***Newton-Krylov*** method utilizes a ***matrix-free Krylov-subspace*** method to solve the reduced Hessian system inexactly in every Newton iteration, using exact Hessian information, but only providing a function for the matrix-vector product $\bar{\boldsymbol{H}}_k \boldsymbol{g}_q(\boldsymbol{q}_k)$ instead of an explicit representation of $\bar{\boldsymbol{H}}_k$. Such matrix-free variants become inevitable in the face of large scale applications, where neither building nor storing the reduced Hessian is feasible.

B.2.a) The reduced ***quasi-Newton method*** directly maintains an approximation $\boldsymbol{M}_k^{-1}$ to the ***inverse Hessian***, which renders the subsequent solution of the linear system redundant. A Newton update is then obtained from

$$\boldsymbol{q}_{k+1} = \boldsymbol{q}_k - \alpha \boldsymbol{M}_k^{-1} \boldsymbol{g}_q(\boldsymbol{q}_k)$$

with $\|\boldsymbol{I} - \boldsymbol{M}_k^{-1}\bar{\boldsymbol{H}}_k\| \leqslant \eta_{max} < 1$ and similar reasoning as above. The quasi-Newton approximation does not discard all second-order variational-derivatives of the Lagrangian, but assumes them to be constant. An initial guess for the inverse Hessian is iteratively improved by observed gradient and iterate information. If enhanced with a line-search method satisfying the Wolfe conditions, the resulting inverse Hessian approximation is positive definite [Noc06].

B.2.b) The reduced ***Gauß-Newton*** method ***approximates*** the reduced Hessian with $\boldsymbol{M}_k$, solves

$$\boldsymbol{M}_k \hat{\boldsymbol{q}}_k = -\boldsymbol{g}_q(\boldsymbol{q}_k) \quad \text{and sets} \quad \boldsymbol{q}_{k+1} = \boldsymbol{q}_k + \alpha \hat{\boldsymbol{q}}_k$$

in every iteration, with the Gauß-Newton Hessian approximation, obtained by discarding parts of the second-order variational-derivatives of the Lagrangian. The step size α is obtained from a line-search technique. $\boldsymbol{M}_k$ must satisfy

$$\|\boldsymbol{I} - \boldsymbol{M}_k^{-1}\bar{\boldsymbol{H}}_k\| \leqslant \eta_{max} < 1,$$

and with

$$\begin{aligned} \|r\| = \|\boldsymbol{g}_q(\boldsymbol{q}_k) + \bar{\boldsymbol{H}}_k \hat{\boldsymbol{q}}_k\| &= \|\boldsymbol{g}_q(\boldsymbol{q}_k) + \bar{\boldsymbol{H}}_k[-\boldsymbol{M}^{-1}\boldsymbol{g}_q(\boldsymbol{q}_k)]\| \\ &\leqslant \|\boldsymbol{I} - \boldsymbol{M}_k^{-1}\bar{\boldsymbol{H}}_k\|\|\boldsymbol{g}_q(\boldsymbol{q}_k)\| \leqslant \eta_{max}\|\boldsymbol{g}_q(\boldsymbol{q}_k)\| \end{aligned}$$

equation (6.39) is fulfilled and convergence results for specified η_{max} can be applied [Den68]. Computing a suitable approximation of the Hessian is favorable for two reasons: (*i*) Computing the exact Hessian terms is usually computationally expensive, and (*ii*) in our context, the reduced Hessian suffers from indefiniteness which poses a significant difficulty for most solvers. The Gauß-Newton approximation addresses both issues by significantly reducing the computational complexity, discarding second order derivative terms, and, thereby, constructing a positive definite approximation. Furthermore, the optimal step length for the Gauß-Newton step is close to 1 [Noc06]. Typically, the resulting system is solved via an inner Krylov iteration, giving raise to

B.3) The ***Gauß-Newton-Krylov method***, which combines inexactness in the Hessian with an inexact iterative solution of the resulting linear system.

In the sequel, we consider the above described methods in more detail, and in particular, work out strengths and weaknesses for the GNK method when compared to the quasi-Newton method. A quantitative comparison for the tumor inversion problem is given in §7.4.5.

6.4.4 The Reduced Newton-Krylov Method

The reduced Hessian involves the inverse state and adjoint operator, and thus is a dense matrix. If the number of control variables n_q is large, building and storing the matrix is intractable, and solving (6.34) requires a matrix-free iterative Krylov-subspace solver (such as conjugate gradients (CG)). Even if n_q is small, an iterative method might be favorable as building $\bar{\boldsymbol{H}}$ requires at least n_q forward and adjoint solves, while an iterative method possibly converges in less than n_q iterations. Such solvers need the matrix-vector product (matvec) (6.37) of the reduced Hessian with a vector. The involved algorithmic steps are given in Alg. 6.5.

```
func $[\underline{\bar{\boldsymbol{H}}\hat{\boldsymbol{q}}}^{\ell}] =$ matvecRH($\hat{\boldsymbol{q}}_k^{\ell}, \hat{\boldsymbol{u}}_k^{\ell}, \hat{\boldsymbol{\lambda}}_k^{\ell}, \boldsymbol{q}_k, \boldsymbol{u}_k, \boldsymbol{\lambda}_k$)
    solve:   $\hat{\boldsymbol{u}}_k^{\ell} = \bar{\boldsymbol{M}}_u \hat{\boldsymbol{q}}_k^{\ell}$                                              ⇝ (incr. state)     ▷ (6.35a)
    solve:   $\hat{\boldsymbol{\lambda}}_k^{\ell} = \bar{\boldsymbol{M}}_\lambda \hat{\boldsymbol{q}}_k^{\ell}$                                              ⇝ (incr. adjoint)   ▷ (6.35b)
    compute: $\underline{\bar{\boldsymbol{H}}\hat{\boldsymbol{q}}}^{\ell} = \boldsymbol{W}_{qq}\hat{\boldsymbol{q}}_k^{\ell} + \boldsymbol{W}_{qu}\hat{\boldsymbol{u}}_k^{\ell} + \boldsymbol{Q}^T\hat{\boldsymbol{\lambda}}_k^{\ell}$   ⇝ (matvec)          ▷ (6.38)
```

ALGORITHM 6.5 *Reduced Hessian Matvec Routine.*

Solving the Newton system iteratively with a certain accuracy for the residual leads to inexactness in the Newton direction, and, thus, may affect the convergence properties. Convergence of such methods is controlled by the forcing sequence $\{\eta_k\}$, which is reflected in the termination criterion for the inner Krylov solver. While reducing computational costs, we would like to preserve rapid Newton-like convergence. Closely related to this is the issue of over-solving the Newton system: If far away from a solution, solving the Newton system very accurately by imposing a high accuracy on the approximation of the Newton step may lead to a significant disagreement of the gradient $\boldsymbol{g}_q(\boldsymbol{q}_k + \hat{\boldsymbol{q}})$ and its local linear model $\boldsymbol{g}_q(\boldsymbol{q}_k) + \bar{\boldsymbol{H}}(\boldsymbol{q}_k)\hat{\boldsymbol{q}}$. Thus, over-solving results in little to no reduction in the gradient and entails lavished costs. A less accurate approximation to the Newton step in an early stage may be both cheaper and more effective. Thus we employ an Eistenstat-Walker convergence criterion [Eis96; Noc06] for the inner Krylov solver

$$\varepsilon_{EW,k} = \eta_k \|\boldsymbol{g}_{q_k}\| \qquad \text{with the forcing sequence} \qquad \eta_k = \max(1\mathrm{E}{-}2, \min(0.5, \sqrt{\|\boldsymbol{g}_{q_k}\|})),$$

choosing the imposed accuracy relative to the gradient norm. This choice for the forcing sequence yields super-linear convergence. If $\eta_k = \max(1\mathrm{E}{-}2, \min(0.5, \|\boldsymbol{g}_{q_k}\|))$ is used, quadratic convergence is expected. A profound convergence analysis and proofs of guaranteed convergence can be found in [Dem82; Eis96; Den68].

Remark. Powerful preconditioners are absolutely inevitable and of great importance to efficiently tackle the exceptionally hard problems and give raise to fast converging solvers. For the computation of the Newton step using Newton-Krylov, for example, a sophisticated preconditioner has to be used to alleviate poor conditioning of the Hessian matrix for the Krylov solve. Within this work, however, we do not focus on design, implementation or comparison of such preconditioners. The utilized methods are described very briefly in §5.3.3. More details are given in [Man15; Man16a].

The Reduced Gauß-Newton(-Krylov) Method. Amongst other possible approaches to ensure positive definiteness of the Hessian, the *Gauß-Newton* addresses the issue by neglecting the second-order variational derivatives of the Lagrangian. More precisely, it neglects all terms from the reduced Hessian in (6.38) involving $\boldsymbol{W}_{uq} = \boldsymbol{J}_{uq} + \boldsymbol{c}_{uq}^T\boldsymbol{\lambda}$ and $\boldsymbol{W}_{qu} = \boldsymbol{J}_{qu} + \boldsymbol{c}_{qu}^T\boldsymbol{\lambda}$, as the terms $\boldsymbol{c}_{uq}^T$, $\boldsymbol{c}_{qu}^T$ and $\boldsymbol{c}_{uu}$ are second order derivatives and, thus, dropped by the Gauß-Newton strategy, and

```
func $[\hat{q}_{k+1}, r_k] =$ krylovSolverRH($g_{q_k} q_k, u_k, \lambda_k$)
    $\bar{H}^0_{\hat{q}} =$ matvecRH($\hat{q}^0_k, \hat{u}^0_k, \hat{\lambda}^0_k, q_k, u_k, \lambda_k$)   $\rightsquigarrow$ (6.37)
    for $\ell = 0, 1, \ldots$ do
        $r^\ell = -g_{q_k} - \bar{H}^\ell_{\hat{q}}$
        if $\|r^\ell\| < \varepsilon_{EW,k}$ then
            break
        $\hat{q}^{\ell+1}_k =$ KSP($\bar{H}^\ell_{\hat{q}}, g_{q_k}$)
        $\bar{H}^{\ell+1}_{\hat{q}} =$ matvecRH($\hat{q}^{\ell+1}_k, \hat{u}^{\ell+1}_k, \hat{\lambda}^{\ell+1}_k, q_k, u_k, \lambda_k$)
```

ALGORITHM 6.6 *Krylov solver for the reduced Hessian system. Sketch of a matrix-free Krylov-subspace method to solve (6.31), utilizing the **matvecRH** routine defined above. The **KSP** call is a placeholder to compute the iterate on the control increment based on a Krylov method such as CG or GMRES. An Eisenstat-Walker convergence criterion [Eis96; Noc06] with adaptive threshold ε_{EW} is utilized to prevent over-solving and foster the efficiency of the method.*

$\boldsymbol{J_{uq}} = \boldsymbol{J_{qu}} = 0$. Further, $\boldsymbol{W_{qq}} = \boldsymbol{J_{qq}} + \boldsymbol{c}^T_{\boldsymbol{qq}}\boldsymbol{\lambda}$ and $\boldsymbol{W_{uu}} = \boldsymbol{J_{uu}} + \boldsymbol{c}^T_{\boldsymbol{uu}}\boldsymbol{\lambda}$ simplify[26] to $\hat{\boldsymbol{W}}_{\boldsymbol{qq}} = \hat{\boldsymbol{J}}_{\boldsymbol{qq}}$ and $\hat{\boldsymbol{W}}_{\boldsymbol{uu}} = \hat{\boldsymbol{J}}_{\boldsymbol{uu}}$, respectively. Thereby it also reduces the computational costs for the Hessian matvec considerably. The resulting Gauß-Newton reduced Hessian approximation reads

$$\boldsymbol{M}_{GN} = \hat{\boldsymbol{W}}_{\boldsymbol{qq}} + \boldsymbol{Q}^T\boldsymbol{U}^{-T}\hat{\boldsymbol{W}}_{\boldsymbol{uu}}\boldsymbol{U}^{-1}\boldsymbol{Q} \tag{6.40a}$$

$$= \hat{\boldsymbol{J}}_{\boldsymbol{qq}} + \boldsymbol{c}^T_{\boldsymbol{q}}\boldsymbol{c}^{-T}_{\boldsymbol{u}}\hat{\boldsymbol{J}}_{\boldsymbol{uu}}\boldsymbol{c}^{-1}_{\boldsymbol{u}}\boldsymbol{c}_{\boldsymbol{q}} \tag{6.40b}$$

Remark. For the tumor and image registration problem, we are given an objective function of the form $\mathcal{J}[\boldsymbol{u}, \boldsymbol{q}] = {}^1\!/_2\|\boldsymbol{r}(\boldsymbol{u})\|^2_2 + {}^\beta\!/_2\|L\boldsymbol{q}\|^2_2$ with linear operators $\boldsymbol{r}$ (point-wise mismatch) and L (differential operator in regularization). Thus, we have $\boldsymbol{J_{uq}} = \boldsymbol{J_{qu}} = \boldsymbol{0}$, $\boldsymbol{J_{uu}} = ({}^{\partial r_i}/_{\partial u_j})^T_{i,j}({}^{\partial r_i}/_{\partial u_j})_{i,j} + ({}^{\partial^2 r_i}/_{\partial u_j \partial u_k})^T_{i,j,k} \cdot \boldsymbol{r} = \boldsymbol{0}$ since $\boldsymbol{r}$ is linear, and $\boldsymbol{J_{qq}} = \beta({}^{\partial L_i}/_{\partial q_j})^T_{i,j}({}^{\partial L_i}/_{\partial q_j})_{i,j} + ({}^{\partial^2 L_i}/_{\partial q_j \partial q_k})^T_{i,j,k} \cdot \boldsymbol{r} = \boldsymbol{0}$, since L is linear (i.e., no terms are neglected in $\boldsymbol{J_{uu}}$, $\boldsymbol{J_{uq}}$, $\boldsymbol{J_{qu}}$ and $\boldsymbol{J_{qq}}$ if applying Gauß-Newton to our application problems).

Typically, the resulting Gauß-Newton system is then solved inexactly using a Krylov method as in Alg. 6.6. This approach is known as Gauß-Newton-Krylov solver. A typical choice for the inner Krylov solver is a PCG method [Noc06], p. 199, Alg. 5.3, which for every inner iteration requires the Hessian response on a vector (matvec). The corresponding simplified Hessian matvec lacks the terms involving $\boldsymbol{W_{uq}}$ and $\boldsymbol{W_{qu}}$, but still requires one incremental state and one incremental adjoint solve, which is a direct consequence of the block-elimination of the reduced space approach. This can be seen from the operators $\boldsymbol{U}^{-1}$ and $\boldsymbol{U}^{-T}$ in equation (6.40a), respectively. The computational steps to compute the Gauß-Newton Hessian matvec thus involve: (*i*) to solve the incremental state equation $\hat{\boldsymbol{u}}^\ell_k = \bar{\boldsymbol{M}}_{\boldsymbol{u}}\hat{\boldsymbol{q}}^\ell_k$ (*ii*) then solve the incremental adjoint equation $\hat{\boldsymbol{\lambda}}^\ell_k = -\boldsymbol{U}^{-T}\boldsymbol{W_{uu}}\hat{\boldsymbol{u}}^\ell_k$ and finally (*iii*) compute $\boldsymbol{M}_{GN}\hat{\boldsymbol{q}}^\ell = \boldsymbol{W_{qq}}\hat{\boldsymbol{q}}^\ell_k + \boldsymbol{Q}^T\hat{\boldsymbol{\lambda}}^\ell_k$. Note that $\boldsymbol{W_{uq}}$ and $\boldsymbol{W_{qu}}$ are proportional to the adjoint since $\boldsymbol{J_{uq}} = \boldsymbol{J_{qu}} = \boldsymbol{0}$, thus, near a solution, where the adjoint variable is zero, we obtain near-quadratic Newton convergence despite neglecting blocks in the Hessian.

6.4.5 The Reduced Quasi-Newton Method

The Gauß-Newton(-Krylov) method discussed in the previous section is very powerful and is the de facto standard approach to solve large-scale, non-linear PDE-constrained optimization problems [Bir05a; Bir05b; Kno04]. The matrix-free nature of the Newton-Krylov approach effectively tackles the memory problem when going to large-scale systems, and inexactness helps to reduce

[26] Note, that in case $\boldsymbol{J_{qq}}$ or $\boldsymbol{J_{uu}}$ contain non-zero second order derivatives; these terms are also dropped for the GN approximation.

computational cost. Yet, the latter remains to be the method's main drawback: The significant computational expense per Newton-iteration. For the problems considered here, the Hessian matrices typically feature poor conditioning and the solution of the Newton system poses challenges for the Krylov solvers. Despite only requiring inexact solves, most of the computational time is spent in the Krylov-iterations. This motivates to examine solvers that avoid direct Hessian information and cope without the expensive solve of the Hessian system for the Newton-step. Quasi-Newton describes a family of methods that precisely achieve this by building an approximation $\boldsymbol{M}^{-1}$ to the ***inverse Hessian*** matrix, requiring only gradient information. In Chapter 2 we have thoroughly discussed various quasi-Newton methods with different manifestations tailored to specific application problems, analyzed their properties and presented information on an efficient implementation on distributed data. In particular, §2.3 targets quasi-Newton methods for non-linear optimization. We encourage the reader to recapitulate the findings from Chapter 2 before continue reading.

The algorithmic core of quasi-Newton results in cheap matvec operations to compute the Newton-update, and renders a linear system solve redundant. In general, $\boldsymbol{M}^{-1}$ is only a rough (low-rank) approximation to the true inverse Hessian matrix of the considered system. Compared to the Newton-Krylov method, the computed Newton-step direction is of lower quality, which is manifested in a slower, typically super-linear, convergence rate. Thus, condensed quasi-Newton solvers usually require a much larger number of non-linear iterations than Newton-Krylov methods, yet the computational cost per iteration is reduced significantly. The interesting question is, whether the overall time-to-solution reduces by replacing the Hessian-based Newton-Krylov (precisely GNK; Jacobian-based) with gradient-based quasi-Newton alternatives, and if the latter scales better when going to large-scale problems (despite the increased number of iterations required). The answer to this question depends on the characteristics and structure of the given problem, and the employed quasi-Newton variants with their sophisticated modifications, tailored to overcome known shortcomings. In [Sar13], quasi-Newton is effectively used to solve a large-scale PDE-constrained optimization problem and is shown to outperform the alternative Newton-Krylov method both in terms of memory usage and computational complexity for the considered problem. Obviously, for the latter, the computational effort to solve the Hessian system increases drastically, when increasing the number of unknowns (mesh-refinement); on the other hand, the plain vanilla[27] quasi-Newton method is known to suffer from a degradation of convergence speed, due to the discrepancy between the low rank approximation of the inverse Hessian and the possibly increasing information content in the true system Hessian (more resolved details; full rank). Mesh-independent convergence rates for the BFGS quasi-Newton method can be shown for some unconstrained optimal control problems if certain assumptions on the initial guess for the inverse Hessian are met [Kel87a].

In the following, we recapitulate the LBFGS method from §2.3 and compare it against the established GNK method. Thereby, we analyze computational cost and point out main differences between the methods. Furthermore, we resume the dominant factors that influence convergence and give some specific suggestions for the initial guess of the inverse Hessian approximation for the here considered tumor inversion problem. A quantitative comparison of quasi-Newton and GNK applied to the tumor inversion problem with respect to the time-to-solution is given in §7.4.5.

The Limited-Memory BFGS Method. Let us shortly recapitulate the ***BFGS quasi-Newton method*** from §2.3 and its extension to the limited memory LBFGS variant. For more details and a general discourse on quasi-Newton methods, we refer to Chapter 2. Based on collected iterate- and gradient-

[27] By this we mean the application of a black-box and generic quasi-Newton method, which is not tailored to the specific problem at hand. For L-BFGS for example, the number of vectors stored and used for the Hessian update as well as the initial guess of the inverse Hessian approximation, drastically affect the convergence properties of the method.

observations, quasi-Newton establishes an linear approximation of the inverse Hessian matrix motivated by a locally quadratic approximation of the objective function and the reasoning

$$\Delta g(q^k_{k-1}) = g(q^k) - g(q^{k-1}) \approx H(q^k)\Delta x q^k_{k-1},$$

This leads to the multi-secant equation $M^{-1}V_k = W_k$ which is solved for the inverse Hessian estimator M^{-1} with the matrices V_k and W_k as defined in (2.40) storing the modes of iterate- and gradient-observation differences. To obtain uniqueness of the estimator, we impose minimization in the weighted Frobenius norm. Also, we require symmetry of the inverse Hessian estimation. Summarizing, the BFGS estimator is the matrix M^{-1} which solves the following optimization problem [Noc06]

$$\min_{M^{-1}} \left\| M^{-1} - M^{-1}_{prev} \right\|^2_W \qquad \text{subject to} \qquad M^{-1}v_k = w_k \;\text{ and }\; M^{-T} = M^{-1} \tag{6.41}$$

and reads

(6.42)

$$\text{(BFGS)}\quad M^{-1,k+1} = \left(I - \rho_k w_k v_k^T\right) M^{-1,k} \left(I - \rho_k v_k w_k^T\right) + \rho_k w_k w_k^T = U_k^T M^{-1,k} U_k + \rho_k w_k w_k^T,$$

where $\rho_k := (v_k^T w_k)^{-1}$, $U_k = (I - \rho_k v_k w_k^T)$ and $M^{-1,k}$ is the inverse Hessian approximation from the previous quasi-Newton iteration which serves as initial guess for the approximation of the current iteration. When going to large-scale, storing the matrix M^{-1} becomes infeasible and limited memory algorithms with implicit matrix representation are required. The limited memory BFGS (LBFGS) alternative, known as one of the most efficient quasi-Newton methods [Noc06; Den77; Gil89] remedies this problem by keeping a sliding window of only a small number of μ previous difference vectors in memory. The inverse Hessian estimator is represented implicitly. For details, we refer the reader to §2.3. An efficient recursive formulation of the LBFGS update is given in equation (2.51) and can be computed from a two-loop recursion in $\mathcal{O}(4\mu n_q)$ operations, compare Alg. 2.4. Due to the limited memory strategy, the initial guess M^{-1}_{prev} for the inverse Hessian approximation can be chosen independently in every iteration.

Convergence & Dominant Factors. In order to tailor the LBFGS method to the considered inverse tumor problem, we need to understand the most dominant factors affecting its converges properties. In Chapter 2, §2.3 we discussed the role of (*i*) the number μ of stored vectors used for the limited memory update, and discussed (*ii*) different choices for the initial guess for the inverse Hessian approximation.

Summarizing the discourse in §2.3.1, we can typically improve convergence by increasing the ***number μ of stored vectors***, which in turn increases the computational cost per iteration. For successive rank-1 updates linearly dependent vectors are not a problem, however, wrong or contradicting information can slow down convergence and the parameter μ remains to be problem dependent. Within the context of our joint registration and biophysical inversion approach and its Picard-iteration-type solution scheme, an interesting question is, if reuse of vectors obtained from sub-component solves of previous Picard-iterations is beneficial. If the true Hessian of the optimization problem does not change significantly throughout Picard-iterations, this approach might be worthwhile. We examine the performance of this ***recycling strategy*** by numerical analysis in §7.4.5.

The choice for the ***initial guess M^{-1}_{prev} of the inverse Hessian*** approximation critically affects the method's robustness and convergence properties; in fact, it is the most dominant factor. A good initial guess balances approximation quality and computational effort. In §2.3.1, Chapter 2 we classify numerous strategies into

A. *Generic but uninformed approaches.*

B. *Tailored approaches that incorporate complimentary a priori knowledge about the problem.*

C. *Generic approaches that incorporate knowledge about the problem in an automated or implicit way.*

In addition to the generic strategies discussed in §2.3.1, we now give some specific manifestations for the choice of $\boldsymbol{M}_{prev}^{-1}$, which are categorized under item B:

B.1 ***Use informed approximation of Hessian*** $\boldsymbol{M}_{prev}^{k} = \boldsymbol{Z}$ and solve $\boldsymbol{M}_{prev}^{k}\Delta\boldsymbol{x}^{k} = \boldsymbol{r}$ for $\Delta\boldsymbol{x}^{k}$ in line 6 of Alg. 2.4. This approach involves the solution of a linear system of equations per iteration. For a moderately changing Hessian, the initial Guess may be factorized in an offline phase and then applied at (nearly) no cost. For our tumor inversion problem, we propose the following tailored choices, considering the L^2-regularization model $\mathcal{S}_p^2$ in (6.5) with $\boldsymbol{W} = \Phi$; the basic idea is given in the generic notation, introduced in §6.4.1:

(i) use $\boldsymbol{M}_{prev}^{k} = \hat{\boldsymbol{J}}_{\boldsymbol{qq}} \overset{\text{here}}{=} \beta_{\boldsymbol{p}}\Phi^T\Phi$ and solve $\boldsymbol{M}_{prev}^{k}\Delta\boldsymbol{x}^{k} = \boldsymbol{g}^{k}$ by PCG iterations. For the preconditioner, we use a LU-decomposition of $\beta_{\boldsymbol{p}}\Phi^T\Phi$ computed in an initialization phase.

(ii) use $\boldsymbol{M}_{prev}^{k} = \hat{\boldsymbol{J}}_{\boldsymbol{qq}} - \boldsymbol{Z}^T\boldsymbol{Z} \overset{\text{here}}{=} \beta_{\boldsymbol{p}}\Phi^T\Phi - \boldsymbol{Z}^T\boldsymbol{Z}$ and solve $\boldsymbol{M}_{prev}^{k}\Delta\boldsymbol{x}^{k} = \boldsymbol{g}^{k}$ by PCG iterations. $\boldsymbol{Z}$ is a low rank approximation of the missing part of the GN approximation (6.40b) of the reduced Hessian, i.e.,

$$\left(\boldsymbol{c}_{\boldsymbol{q}}^T\boldsymbol{c}_{\boldsymbol{u}}^{-T}(\hat{\boldsymbol{J}}_{\boldsymbol{uu}})^{T/2}\right)\left(\underbrace{\hat{\boldsymbol{J}}_{\boldsymbol{uu}})^{1/2}\boldsymbol{c}_{\boldsymbol{u}}^{-1}\boldsymbol{c}_{\boldsymbol{q}}}_{:=\boldsymbol{Z}}\right) \overset{\text{here}}{=} \Phi^T\boldsymbol{U}^{-T}\underbrace{\boldsymbol{U}^{-1}\Phi}_{:=\boldsymbol{Z}} = \Phi^T\hat{\alpha}(0), \tag{6.43}$$

where $\boldsymbol{U}^{-T}$ and $\boldsymbol{U}^{-1}$ represent the solution of the adjoint and forward equations, respectively. We can e.g., obtain $\boldsymbol{Z}$ from sampling $\boldsymbol{x} = (\boldsymbol{U}^{-1}\Phi)\tilde{\boldsymbol{q}}$ for a small number of random vectors $\tilde{\boldsymbol{q}}$ and add the term $\boldsymbol{Z}^T\boldsymbol{Z} = \boldsymbol{R}^T\boldsymbol{Q}^T\boldsymbol{Q}\boldsymbol{R} = \boldsymbol{R}^T\boldsymbol{R}$ obtained by QR-decomposition of $\boldsymbol{Z}$ to the initial guess for the Hessian. Within our Picard-iteration scheme, for example, we can sample and decompose $\boldsymbol{Z}$ after the first Picard-iteration using GNK for the tumor inversion to construct the initial guess $\boldsymbol{H}_k^0$. Then use LBFGS with $\boldsymbol{H}_k^0$ for all subsequent inverse tumor solves.

Lastly, the employed ***line-search method*** matters. Since quasi-Newton methods establish the Hessian approximation based on observations throughout the iterations, sub-optimal step-directions stemming from insufficient line-search conditions can cause problems. Once added, wrong information gets corrected slowly. Therefore, for our LBFGS algorithm, we employ the Moré-Thuente line-search method, which satisfies the Wolfe conditions (2.50) but requires an additional gradient evaluation in every trial step. For the GNK solver, the less expensive Armijo line-search method suffices.

Quasi-Newton for the Tumor Inversion. Generally speaking, for quasi-Newton methods to work adequately, we either (*i*) require the system matrix (Hessian) to encode low or limited information content, i.e., to be of either low dimension or low rank, such that it can effectively be represented by a low rank approximation, or (*ii*) we need an excellent initial guess for the inverse Hessian, which already captures big parts of the information encoded in the true Hessian. The latter typically is a difficult problem and involves a linear system solve[28] itself, or is aggravated by the fact that in general even for low rank or sparse Hessian matrices, the inverse is a dense matrix. Under these

[28] To invert a "constant" or invariant part of the Hessian to be used as an initial guess for the quasi-Newton approximation.

TABLE 6.1 *Comparison of computational effort of quasi-Newton versus Gauß-Newton-Krylov solver.*

		Quasi-Newton	#state	#adjoint	**Gauß-Newton-Krylov**	#state	#adjoint
COST PER ITERATION	grad. eval.		1	1		1	1
	obj. eval.		0	0		0	0
	line-search	$r\times$ (wolfe-ls)	r	r	$r\times$ (armijo-ls)	r	0
	Newton step	$\hat{\boldsymbol{q}}_k = -\boldsymbol{M}^{-1}\boldsymbol{g}_k$	0	0	$m\times$ Krylov matvec	m	m
GEN-ERAL	cost line-search	obj. eval. + grad. eval.			obj. eval.		
	performance	dependent on init. guess $\boldsymbol{M}_{prev}^{-1}$			dependent on Krylov forcing seq.		
	convergence	typically super-linear			typically near-quadratic		

considerations, quasi-Newton does not seem to be a promising alternative for GNK to solve the registration problem. Here, the system matrix is of dimensionality $3n \times 3n$, $n = n_1 \times n_2 \times n_3$ and due to the nature of the problem we can not hope for an inverse Hessian matrix that can be adequately approximated by a low-rank matrix. For the inverse tumor problem on the other hand, we are given a low-dimensional[29] Hessian $\boldsymbol{H} \in \mathbf{R}^{n_q \times n_q}$ and a low-rank quasi-Newton approximation is promising to yield Newton step directions, resulting in an acceptable convergence rate. In the following, we only consider GNK and quasi-Newton for the tumor inversion problem.

Gauß-Newton-Krylov vs. Quasi-Newton. We summarize the most important differences and give a very rough comparison of the methods' computational effort in terms of number of required PDE solves (state equation and adjoint equation) per Newton-iteration in Tab. 6.1. Per non-linear iteration, the Gauß-Newton-Krylov solver requires the evaluation of the objective function $\mathcal{J}_k := \mathcal{J}[\boldsymbol{q}_k]$, the evaluation of the reduced gradient $\boldsymbol{g}_k := \boldsymbol{g}(\boldsymbol{q}_k)$, and the solution of the Hessian system $\boldsymbol{H}\hat{\boldsymbol{q}} = -\boldsymbol{g}_k$ for $\hat{\boldsymbol{q}}$, requiring $m < n_q$ inner Krylov iterations. The evaluation of the objective function requires the solution of the state equation. Typically, we need to evaluate the objective function and the gradient for the current iterate $\boldsymbol{q}_k$; together, this requires to solve the state and adjoint equations, i.e., two PDE solves. If the step size does not satisfy the Armijo line-search conditions, we need additional evaluations of the objective function, i.e., additional solves of the state equation. Each Krylov iteration, i.e., each Hessian matvec, requires two PDE solves; one for the incremental state equation and one for the incremental adjoint equation. Assuming m Krylov iterations and r line-search steps, the cost of a single Gauß-Newton-Krylov iteration can be estimated by $\text{cost}_{GNK} = 2(m+1) + r$ PDE solves. As mentioned above, the quasi-Newton method only requires gradient information and forgoes the inner Krylov solve. For the line-search, however, Armijo is not sufficient and the stricter Wolfe-conditions need to be fulfilled; the latter require gradient information at line-search test probes. Summarized, this results in $\text{cost}_{QN} = 2(r+1)$ PDE solves per quasi-Newton iteration. Obviously, these cost are a function of the number of required inner Krylov-solves and line-search probes. For the overall cost, the usually larger number of iterations for quasi-Newton methods needs to be taken into consideration: while with GNK we typically obtain near-quadratic convergence, super-linear convergence rates are observed for most problems solved with quasi-Newton methods.

Also noteworthy is a further difference of the methods which lies in the nature of the Hessian approximation: Both methods employ inexactness in approximating the Hessian, either by the Gauß-Newton approximation or by an updated approximation of the inverse Hessian. The former discards all second order derivative information from the Hessian (Jacobian based), while the latter only assumes them to be constant. Thus, theoretically, quasi-Newton has the potential to compute a more accurate approximation of the Hessian than GNK, given enough linear independent observations for the Hessian update. Note, however, that quasi-Newton needs to also approximate

[29] This is a direct result of our parametrization $c^{(\cdot,0)} = \Phi \boldsymbol{p}$ for the tumor initial condition.

the first order derivative terms from the Hessian, which are represented exactly in the Gauß-Newton approximation.

7 Evaluation of Coupling Schemes and Advanced Methods

This chapter contains an extensive study and numerical analysis of our methodology. We target different objectives for evaluation, the most prominent of which are

- *validity of our scheme*, i.e., convergence analysis and plausibility considerations *for synthetic test cases* where the ground truth is known a priori;[1]
- *performance of our scheme* with respect to good data similarity, good overlap measures, and meaningful deformation maps *for biophysics-aided. normal-to-abnormal registration* for synthetic cases and real patient data, see §7.2 (primarily moving-patient formulation);
- *performance and quality of biophysical inversion* of our scheme as a proof of concept, primarily focusing on methodological aspects using synthetic cases, see §7.3 (primarily moving-atlas formulation);
- *efficiency*, i.e., overall time-to-solution, speedups obtained through various modifications (see §7.4 and §7.4.5) *and parallel scalability* (see §5.3.4).

The chapter is organized as follows. An overview of the general setting, including a description of the considered test cases, the utilized imaging data, parameters, and solver settings as well as definitions of monitored performance measures is given in §7.1. In §7.2, we study the eligibility of our joint inversion scheme for the task of registering a normal brain to a brain with abnormalities (i.e., a pathology). Here, we mainly focus on the moving-patient formulation and investigate the solution scheme's reconstruction quality and convergence for synthetic cases and real patient imaging data.

§7.3 targets the task of biophysical model calibration. In other words, the estimation of certain parameters of the biophysical model used to describe tumor progression in the human brain. For the reasons given in the preliminary remarks of Chapter 6, we mainly focus on the moving-atlas formulation due to its improved qualification for biophysical analysis. As mentioned above (cf. §5.3.1), our utilized tumor model is a very simplistic reaction-diffusion process and is purely phenomenological; it describes at best a crude approximation of the underlying process and the observed cancer progression. The results are therefore to be seen as preliminary, and they primarily serve as a proof of concept for the proposed solution strategy. Thus, in great part, we rely on synthetic cases (i.e., assume knowledge of the true model for the underlying biophysical process) where the ground truth for model parameters and healthy patient brain is known. Finally, in §7.4 we analyze advanced methods to improve the solution process with respect to time-to-solution and quality of

[1] Note that we do not validate the correctness of our biophysical model or the plausibility of obtained biophysical parameters by our joint inversion scheme against real data. We run experiments for real data, however, we have no knowledge about any ground truth parameters for these data sets.

the obtained solution. In particular, we investigate the eligibility of advanced quasi-Newton methods for this kind of non-linear optimization problem, and, thereby link to the first part of this thesis.

7.1 General Setup of Biophysics-Based Image Analysis Test Cases

We evaluate, study, and test our methodologies with respect to convergence, performance, validity, and reconstruction quality for synthetically generated and clinical test cases. Both are based on real MR neuro-imaging data, i.e., we consider realistic brain geometries. In the sequel, we summarize the overall setup introducing the test cases in brief; we give more details and specific purposes for each experiment when presenting the results.

7.1.1 Test Cases

We consider two classes of test cases. To assess the quality and validity of our methodology as well as convergence of our Picard-iteration solution schemes to a known ground truth, we use synthetic test cases, based on real, normal brain data. For the generation of these test cases, we use our tumor model to artificially grow a target tumor for which we know the correct model and ground truth parameters. We either grow the artificial tumor in the atlas space and obtain the patient data via application of a known deformation map (ATAV), or we directly grow the pathology in the patient space (ATRV). For these cases, we either partially or fully know the ground truth for the coupled optimization problem (up to discretization errors). The second class of test cases uses actual clinical imaging data obtained from the study in [Goo13; Bak15], where we test our methodology for real patient data (RTRV).

In a little more detail, the considered test cases are:

Test Case ATAV

Analytic tumor and analytic velocity (ATAV) test case (fully synthetic; true tumor parameters and velocity are known, $\boldsymbol{p} = \boldsymbol{p}_A^\star$, $\boldsymbol{v} = -\boldsymbol{v}^\star$); canonical synthetic test case for the moving-patient formulation with full knowledge of tumor parameters and deformation velocity; see also §7.2.2.

Purpose. *Proof of concept. In particular, we test the numerical accuracy of our scheme, identify lower bounds on the expected inversion accuracy (i.e., identify the accuracy we can ideally expect which is bound by the error introduced in image advection, and study the convergence of our solver. We judge the result of our joint inversion approach against the known (by construction) ground truth values.*

Setup. *ATAV is based on real brain geometries, but uses the registration advection and forward tumor solver to synthetically generate a pathological patient image. In particular, we use a resolution of $n_i = 128$, choose $\boldsymbol{p} = \boldsymbol{p}_A^\star$, which defines $c^{(\cdot,0)} = c^{(\cdot,0)^\star}$, and grow a tumor in the healthy atlas geometry, resulting in $c^{(\cdot,1)^\star}$. We choose $\boldsymbol{v} = \boldsymbol{v}^\star$, from registration of the atlas brain to a second tumor-free brain (offline, in a pre-processing step using β_v-continuation with $\beta_v^{lo,\star} = 1\text{E}{-4}$) and generate a pathological patient brain by advection of the atlas brain with incorporated grown tumor with the negative velocity $\boldsymbol{v}^\star$. This gives the synthetic patient target data $\boldsymbol{m}_D$, c_D for our joint inversion. An illustration is given in Fig. 7.1.*

Test Case ATRV

Analytic tumor and real velocity (ATRV) test case (true tumor model and parameters known; real brain images of two different individuals, $\boldsymbol{p} = \boldsymbol{p}_{\mathrm{P}}^{\star}$, $\boldsymbol{v} = \mathrm{N/A}$); canonical synthetic test case for moving-atlas with full knowledge of tumor parameters and healthy patient; see also §7.3.2.

Purpose. *For the moving-patient formulation, we test our algorithm in the context of multi-subject registration for real brain data of two different individuals. Most importantly, however, we evaluate the performance and quality of our biophysical inversion in conjunction with the moving-atlas formulation. For this test case, the true tumor parameters are known in the patient space and we have access to the healthy patient geometry.*

Setup. *The patient data is no longer generated from the atlas image. We use an image of a different individual instead. Accordingly, we no longer know the ground truth velocity field. We consider a resolution of $n_i = 128$, choose $\boldsymbol{p} = \boldsymbol{p}_{\mathrm{P}}^{\star}$, which defines $c^{(\cdot,0)} = c^{(\cdot,0)^{\star}}$ (in the patient space), and grow an artificial tumor c_{D} in a segmented, real patient brain $\boldsymbol{m}_{\mathrm{D}}$. An illustration is given in Fig. 7.1.*

Test Case RTRV

Real tumor and real velocity (RTRV) test case (ground truth inaccessible, $\boldsymbol{p} = \mathrm{N/A}$, $\boldsymbol{v} = \mathrm{N/A}$); see also §7.2.4.

Purpose. *We test our approach, i.e., moving-patient and moving-atlas formulation on the real data of patients diagnosed with glioma tumors and study the registration quality for a variety of parameter choices for the tumor growth model. Knowledge of neither the healthy patient geometry, nor any of the biophysical model parameters, i.e., tumor growth rate or time horizon, is accessible. Furthermore, our biophysical tumor progression model is, at best, a crude approximation of the actual dynamics.*

Setup. *This test scenario consists of real patient brains with real tumors for which we do not know any parameters.*

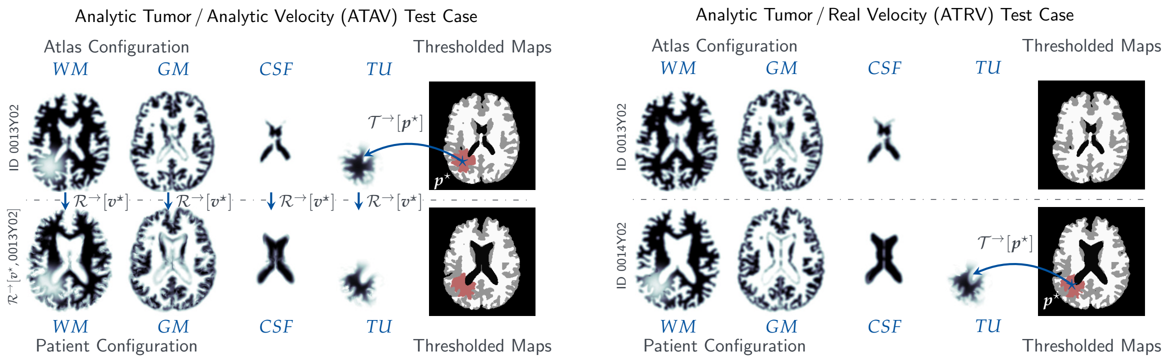

Analytic Tumor / Analytic Velocity (ATAV) Test Case
Atlas Configuration
Thresholded Maps
WM
GM
CSF
TU
ID 0013Y02
$\mathcal{T}^{\rightarrow}[p^{\star}]$
$p^{\star}$
$\mathcal{R}^{\rightarrow}[v^{\star}]$
$\mathcal{R}^{\rightarrow}[v^{\star}]$
$\mathcal{R}^{\rightarrow}[v^{\star}]$
$\mathcal{R}^{\rightarrow}[v^{\star}]$
$\mathcal{R}^{\rightarrow}[v^{\star}, 0013Y02]$
WM
GM
CSF
TU
Patient Configuration
Thresholded Maps
Analytic Tumor / Real Velocity (ATRV) Test Case
Atlas Configuration
Thresholded Maps
WM
GM
CSF
TU
ID 0013Y02
ID 0014Y02
$\mathcal{T}^{\rightarrow}[p^{\star}]$
$p^{\star}$
WM
GM
CSF
TU
Patient Configuration
Thresholded Maps

FIGURE 7.1 *Illustration of ATAV and ATRV test cases.* ***Left:*** *Fully synthetic analytic tumor/analytic velocity (ATAV) test case. Shown are the probability maps for white matter (WM), gray matter (GM), cerebrospinal fluid (CSF) with a synthetically grown tumor (TU) for an atlas brain (ID 0013Y02). The target patient data is generated from advection of these maps with a ground truth velocity $\boldsymbol{v}^{\star}$ (obtained from pre-registration of the healthy brains ID 0013Y02 and ID 0014Y02). The ground truth tumor parameters $\boldsymbol{p}^{\star}$ "live" in the atlas space.* ***Right:*** *Semi-synthetic analytic tumor / real velocity (ATRV) test case. Real brain data ID 0013Y02 and ID 0014Y02 is considered for atlas and patient geometry, respectively; the registration velocity between the two individuals is unknown. The target patient data is constructed by growing an artificial tumor with parameters $\boldsymbol{p}^{\star}$ in the patient space.*

7.1.2 Data

For the generation of most of our synthetic test cases, we use (real) normal[2] brain imaging data obtained at the Perelman School of Medicine at the University of Pennsylvania. We also consider real clinical imaging data of glioma patients presented in the study in [Goo13; Bak15]. We use the output data available after the first iteration of GLISTR [Goo11; Goo13; Bak15], translating MR imaging data into probability maps for the brain tissue regions. We consider six data-sets from this repository (patient IDs: AAMH, AAAN, AAAC, AAMP, AAQD and AAWI). The original data-sets have more labels than we use in our Picard iterations. In particular, they contain background (BG), white matter (WM), gray matter (GM), cerebellum (CB), cerebrospinal fluid (CSF), ventricles (VE), edema (ED), enhancing tumor (ENH), and necrotic tumor (NEC). We construct the labels (WM), (GM), (CSF), (BG) and (TU) by integrating (*i*) (CB) into (BG), (*ii*) (VE) into (CSF), and (*iii*) (ENH), (NEC) and (ED) into (TU).

We wish to have ***partition of unity property*** across all probability maps for each $\boldsymbol{x}$ in Ω, i.e., all labels have to sum up to one

$$\forall \boldsymbol{x} \in \Omega : c^{(\cdot,1)}(\boldsymbol{x}) + \sum_{i=1}^{4} m_i^{(\cdot,1)}(\boldsymbol{x}) \overset{!}{=} 1$$

The background (BG) probability map m_{BG} is introduced for technical reasons; it is not used within our formulation and is not considered for the registration problem. Glial matter is integrated in BG.

7.1.3 Common Parameters

We give an overview of the specific model and numerical parameters for the different test cases in Tab. 7.2. A more detailed listing of the parameters and test case series is given when presenting the results. For all conducted experiments, we fixed certain parameters[3] to choices found by numerical analysis of the methodology; these common parameters are listed in Tab. 7.1. We briefly discuss some of the more involved choices:

TABLE 7.1 *Common parameters used in all test cases: opttol_R, opttol_T are the convergence tolerances for registration and tumor inversion; β_p is the regularization parameter for the tumor inversion; $\mathcal{S}_v[\boldsymbol{v}]$ is the regularization operator for the image registration (with H^1-semi norm); β_v^0 and β_v^{lo} are the initial and final values for the β-continuation scheme as described in §6.3.3 applied in image registration, determined based on which values have been shown to yield best results in numerical tests for the RTRV test problem; ε_∇ is the bound on the variation of the deformation gradient $\det(\nabla \boldsymbol{y})$ used in the continuation scheme.*

opttol_R	opttol_T	β_p	$\mathcal{S}_v[\boldsymbol{v}]$	β_v^0	β_v^{lo}	ε_∇
1E−3	1E−3	2.50E−4	$\|\nabla \boldsymbol{v}\|_{L^2(\Omega)}$	1	1E−4	1E−3

For all experiments we use the $(C1)$ stopping conditions defined in §6.3.3 (based on relative change of gradient) for the tumor inversion and the $(C2)$ stopping conditions (based on relative change of gradient, objective function value and iterate difference) for the registration sub-problem, respectively. Throughout all experiments, we demand optimization tolerances of $\text{opttol}_R = \text{opttol}_T =$ 1E−3 for registration and tumor inversion, respectively (note that, in combination with the applied stopping conditions, this results in a required relative gradient of 1E−3 for the tumor inversion, but a gradient reduction of only about two orders of magnitude for the registration). Due to high

[2] That is, healthy brain imaging data without a pathology.
[3] If any of these parameters is changed for a specific experiment, this is clearly indicated.

TABLE 7.2 *Summary of the parameters for the generation of the synthetic test cases and the inversion. We report values for the following parameters: n_i, $i = 1,2,3$ denotes the grid size used for the discretization of the problem; ρ_w and ρ_g are the characteristic reaction factors for white and gray matter, ρ_f is the overall reaction scaling factor; k_w and k_g are the characteristic diffusion parameters for white and gray matter, k_f the overall scaling parameter for the isotropic part of the inhomogeneous diffusion coefficient for net migration of cancerous cells into surrounding tissue; $n_{\boldsymbol{p}}$ is the number of Gaussian for the parametrization of the tumor initial condition, σ is the standard deviation of the associated Gaussian basis functions, δ denotes the spacing in between centers of adjacent Gaussians; $\mathcal{S}_{\boldsymbol{p}}[\boldsymbol{p}]$ specifies the employed regularizer on $\boldsymbol{p}$ for the tumor inversion solver; and $maxit_i = (maxit_{i,N}, maxit_{i,K})$ denotes the maximum number of Newton iterations and Krylov iterations (for the KKT system) for the tumor inversion ($i = T$) and registration ($i = R$), respectively,*

	ATAV	ATRV	RTRV
n_i	128	{128,256}	256
ρ_w	1	1	1
ρ_g	0	0	0.2
ρ_f	$\{5, 10, 15\}$	$\{5, 10, 15, 20\}$	$\{0, 5, 10, 15\}$
k_w	$\{0, 1\}$	1	1
k_g	0	0	0.1
k_f	$\{0, 1\text{E}{-2}, 1\text{E}{-1}\}$	$\{5\text{E}{-1}\text{–}1\text{E}{-3}\}$	$\{0, 1\text{E}{-2}\}$
$n_{\boldsymbol{p}}$	$\{8, 125\}$	$\{125, n_{\boldsymbol{p}}\}$	$\{125, 343, n_{\boldsymbol{p}}\}$
σ	$\{\pi/10, \pi/15\}$	$\{\pi/15, \pi/128, \text{auto}\}$	$\pi/128, \text{auto}\}$
δ	$1.5 \cdot \sigma$	$\{1.5 \cdot \sigma, 2 \cdot \sigma\}$	$\{1.5 \cdot \sigma, 2 \cdot \sigma\}$
$\mathcal{S}_{\boldsymbol{p}}[\boldsymbol{p}]$	$\\|\Phi \boldsymbol{p}\\|_{L^2(\Omega)}$	$\\|\boldsymbol{p}\\|_{L^i(\Omega)}, i = 1,2$	$\\|\boldsymbol{p}\\|_{L^i(\Omega)}, i = 1,2$
maxit$_T$	$(50, 100)$	$(50, 100)$	$(30, 30)$
maxit$_R$	$(50, 80)$	$(50, 80)$	$(10, 20)$

upper bounds on the allowed number of Newton (and Krylov) iterations (cf. Tab. 7.2), we always reach these tolerances for our synthetic test cases (ATAV/ATRV). For the real data test cases (RTRV), however, the number of allowed Newton (and Krylov) iterations is strictly limited to bound the overall time-to-solution and the tolerances may not be reached on rare occasions. Numerical analysis for these real patient data sets show, that investing more compute time, i.e., further reducing the gradient, does not result in an improved overall accuracy. The obtained accuracy with limited solver iterations is in the order of what can be expected for standalone real patient data based image registration.

The regularization parameter $\beta_{\boldsymbol{p}}$ for the inverse tumor problem has been determined experimentally from an L-curve study for a synthetic test case, using $n_{\boldsymbol{p}} = 125$ Gaussian basis functions and an image resolution of $n = 128^3$. Notice that we use a constant relation $\sigma/\delta = 1.5$ for the spacing of the Gaussian basis functions, which leads to an invariant condition number. We observed that for variations of the image resolution n and the number of basis functions $n_{\boldsymbol{p}}$, the tumor inversion is not very sensitive[4] with respect to the regularization parameter $\beta_{\boldsymbol{p}}$, i.e., smaller values for $\beta_{\boldsymbol{p}}$ did not further reduce the tumor mismatch. Consequently, unless stated otherwise, we fix the parameter $\beta_{\boldsymbol{p}}$ for all experiments. Likewise, the lower bound β_v^{lo} for the regularization parameter β_v in combination with the employed H^1-semi norm regularizer of the deformation velocity has been determined based on extensive numerical analysis for different synthetic and real brain data sets [Man16a; Man17a]. Furthermore, we designed the β-continuation scheme (cf §6.3.3) experimentally and found that reducing the regularization weight by a factor of 10 in every Picard-iteration is sufficient[5].

As described in §6.3.3 SIBIA provides different modes to automatically initialize a set of Gaussian basis functions for the parametrization of the tumor initial condition, based on the patient specific input data. We use both approaches for our real patient data (RTRV) and synthetic data (cases ATAV/ATRV); for the latter, we also use a set of a priori chosen, manually placed Gaussians to facilitate the analysis of our methodology and convergence towards ground truth solutions. In any

[4] This holds true within a certain regime $1\text{E}{-2} \leqslant \beta_{\boldsymbol{p}} \leqslant 1\text{E}{-5}$ of the regularization parameter. Note that, this insensitivity is fostered by the fact that we control the conditioning of Φ for our basis function selection modes, described in §6.3.3.

[5] Using a smaller reduction or more Picard-iterations per fixed regularization weight did not improve the overall result.

case, the choice for n_p depends on the appearance of the patient's tumor, i.e., its size and shape. For the manually placed Gaussians, this number is chosen a priori on a case-by-case basis, such that the pathological domain is covered sufficiently; we consider choices $n_p \in \{2^3 = 8, 5^3 = 125, 7^3 = 343\}$; automatic selection typically results in a number of $8^3 \leqslant n_p \leqslant 20^3$.

7.1.4 Performance Measures

For the numerical analysis and benchmarking of our joint registration and tumor inversion framework, we wish to examine different measures and ratings to analyze, study, and compare the different problem formulations, solver strategies, and numerics. In particular, we want to assess

(i) the convergence towards solutions with low mismatch both in the brain geometry $\boldsymbol{m}$ (quantified by μ_{B,L^2}, see (7.1a)) and the tumor c (quantified by μ_{T,L^2}, see (7.1a));

(ii) the quality of the reconstruction of the healthy brain geometry, quantified by μ_{B_0,L^2}, see (7.1a);

(iii) the quality of visual reconstruction of brain region labels in terms of overlay measures, i.e., DICE_B, DICE_T, and DICE_{B_0} (see (7.1b) quantifying the overlay of patient brain, pathology, and healthy patient brain[6] with its respective reconstruction;

(iv) the quality of biophysical parameter estimation, i.e., the distance e_{c0,L^2} between the reconstructed tumor initial condition and a known ground truth (see (7.1d)), and the proximity of the diffusion coefficient to a known ground truth (for synthetic cases only);

(v) the reduction of the gradient of the coupled formulation $\|\boldsymbol{g}\|_{\text{rel}}$ (see (7.1c));

(vi) the time-to-solution T^{tot} and elapsed wall-time during a Picard-iteration T^{it}, registration solve T^{reg}_{inv}, and tumor inversion T^{tu}_{inv}.

The definition and computation of these quantities is detailed below. We report all performance measures in patient space, i.e., with respect to the patient anatomy and not the atlas anatomy, since the patient space is the relevant space from an applications point of view.[7] For the moving-atlas formulation, tumor simulation and parameter estimates are employed in an approximation of the healthy patient space, thus all occurring objects naturally "live" in the patient space. For the moving patient formulation, tumor simulation is carried out in the atlas space and pathology, brain geometry and reconstructed initial condition need to be warped to the patient space to quantify their proximity to the patient data. This is achieved by applying the inverse deformation map from atlas space to patient space, which for velocity based registration translates to advection with the negative velocity $-\boldsymbol{v}$.

The relative mismatch/residual between patient anatomy and reconstructed anatomy, between the reconstructed healthy patient anatomy and the ground truth, and between patient tumor and simulated tumor in the patient domain are defined as:

$$\text{(7.1a)} \quad \mu_{B,L^2} := \frac{\|\boldsymbol{m}_A^{(1,1)} - \boldsymbol{m}_D\|_{L^2(\Omega)^3}}{\|\boldsymbol{m}_A^{(0,0)} - \boldsymbol{m}_D\|_{L^2(\Omega)^3}}, \quad \mu_{B_0,L^2} := \frac{\|\boldsymbol{m}_A^{(1,0)} - \boldsymbol{m}_P^\star\|_{L^2(\Omega)^3}}{\|\boldsymbol{m}_A^{(0,0)} - \boldsymbol{m}_P^\star\|_{L^2(\Omega)^3}}, \quad \mu_{T,L^2} := \frac{\|c_A^{(1,1)} - c_D\|_{L^2(\Omega)^3}}{\|c_D\|_{L^2(\Omega)^3}}.$$

As mentioned above, we monitor consistent distance measures in the patient space to assess the reconstruction quality for the schemes, although the data misfit terms $\mathcal{D}_c[c^{(\cdot,1)}, c_P^{(1,\cdot)}] + \mathcal{D}_{\boldsymbol{m}}[\boldsymbol{m}_A^{(0,1)}, \boldsymbol{m}_P^{(1,\cdot)}]$

[6] This measure is only accessible for data that includes the healthy brain imaging data before occurrence of cancerous tissue or synthetic test cases where a ground truth for the healthy patient brain geometry is known.

[7] If we perform cohort studies, this is different (see [Goo13] for an example).

and $\mathcal{D}_c[c^{(\cdot,1)}, c_D] + \mathcal{D}_{\boldsymbol{m}}[\boldsymbol{m}_A^{(1,0)}(1 - c^{(\cdot,1)}), \boldsymbol{m}_D]$ in (6.4a) and (6.10a) differ for both formulations. Consequently, for the moving-patient formulation, the healthy patient brain needs to be generated[8] explicitly by advection with $-\boldsymbol{v}$.

To assess visual similarity of data and reconstruction based on the cardinality $|\cdot|$ of a set and a selection function $H(\boldsymbol{u}) := \{u_i \geqslant 0.5\}$ with threshold 0.5, we calculate Dice coefficients for the individual label maps associated with the probability maps for $\ell \in \{WM, GM, CSF\}$ and their average across labels for the patient and the atlas anatomy, i.e.,

$$\text{(7.1b)} \qquad \text{DICE}_{\ell,B} := 2\frac{|H(\boldsymbol{m}_{A,\ell}^{(1,1)}) \cap H(\boldsymbol{m}_{D,\ell})|}{|H(\boldsymbol{m}_{A,\ell}^{(1,1)})| + |H(\boldsymbol{m}_{D,\ell})|}, \qquad \text{DICE}_B = \sum_{\ell=1}^{3} \text{DICE}_{\ell,B}/3;$$

In the same manner, we compute Dice coefficient values for the healthy patient geometry, denoted by DICE_{B_0}, and for the probability maps of the tumor, denoted by DICE_T. Both, Dice and mismatch metrics, $\boldsymbol{m}_A^{(1,1)}$, $\boldsymbol{m}_A^{(1,0)}$, and $c_A^{(1,1)}$ are calculated from $\boldsymbol{m}_A^{(0,1)}$, $\boldsymbol{m}_A^{(0,0)}$, and $c^{(\cdot,1)}$ via advection with negative velocity $-\boldsymbol{v}$ for the moving patient formulation. Furthermore, we monitor the relative change of the gradient for the coupled problem formulations moving-patient and moving-atlas to check for convergence of the respective (block-Newton-type) Picard-iteration solution schemes:

$$\text{(7.1c)} \qquad \|\boldsymbol{g}\|_{\text{rel}} := \|\boldsymbol{g}^k\|_{L^2(\Omega)} / \|\boldsymbol{g}^0\|_{L^2(\Omega)}.$$

Here, $\boldsymbol{g}^k$ is the gradient of the coupled optimization problem (6.4), or (6.10) after the kth Picard-iteration and $\boldsymbol{g}^0$ the reference gradient with respect to the initial guess. To quantify the quality of the estimated biophysical parameters, we consider the relative L^2-error for the initial condition with respect to a known ground truth

$$\text{(7.1d)} \qquad e_{c0,L^2} := \|\Phi p^\star - c^{(1,0)}\|_{L^2(\Omega)} / \|\Phi p^\star\|_{L^2(\Omega)}.$$

The initial condition $c^{(1,0)} = \Phi \boldsymbol{p}$ for the moving-atlas formulation naturally "lives" in the patient space and is calculated via advection with velocity $-\boldsymbol{v}$ from $\Phi \boldsymbol{p}$ for the moving-patient formulation. The ground truth tumor seed $\Phi p^\star$ is only known for synthetic cases.

7.1.5 Hardware and Setup

All numerical results were obtained from the described cases, executed on the Tier-1 supercomputer `HazelHen` at the High Performance Computing Center HLRS in Stuttgart (`www.hlrs.de`), a Cray XC40 system with a peak performance of 7.42 Petaflops comprising 7,712 nodes with Xeon E5-2680 v3 processors and 24 cores on two sockets and 128 GB memory per node. The nodes are connected via an Aries interconnect. SIBIA is written in `C++` and uses MPI for parallelism. It is compiled using the Intel 17 compiler. We use PETSc's implementations for linear algebra operations and PETC's TAO package for the non-linear optimization [Bal16a; Mun15], AccFFT for Fourier transforms [Gho16a; Gho17b], and PnetCDF for I/O [Uni]. If not stated otherwise, we use 3 nodes with 64 MPI tasks for data sizes of $n_i = 128$, $i = 0, 1, 2$, and 11 nodes and 256 MPI tasks for resolutions $n_i = 256$, $i = 0, 1, 2$.

[8] Computing the healthy patient geometry for the moving-patient formulation is no longer possible once mass effect, i.e., deformation of brain parenchyma by the tumor evolution, is included into the model.

7.2 Evaluation of Normal-to-Abnormal Registration Performance and the Moving-Patient Iteration

In this section, we focus on the task of registering healthy (normal) brains to subjects with pathologies (abnormalities) and evaluate the applicability and aptitude of our joint registration and biophysical inversion approach. We are mainly interested in the quality of the normal/abnormal inter-subject registration, i.e., visual similarity and overlap measures of the deformed template image and the observed data, as well as good-natured[9] and plausible inter-subject mappings. We thus mainly focus on the moving-patient formulation (§6.1) and report numerical results for synthetic cases and actual clinical data of glioma patients.

7.2.1 Baseline: Registration vs. Joint Inversion

Before immersing into the numerical analysis of our scheme, we'd like to draw a baseline for normal-to-abnormal registration by considering the registration-only performance (i.e., standalone registration without biophysical enhancement/aid) for a representative subset of our considered test cases. For comparison, we anticipate the results for our joint inversion scheme from the following sections.

Purpose. As mentioned before, normal-to-abnormal registration poses significant challenges on the registration due to topological differences between the registered objects. Enhancing the registration formulation by a model that describes the dynamics of pathology evolution has the potential to aid the registration process considerably and foster biophysically meaningful deformation mappings. We want to study the extent to which the coupling of biophysical inversion with registration enhances the registration quality compared to the results obtained from standalone registration.

Setup. For this experiment, we consider an instantiation of the semi-synthetic ATRV test case (§7.2.3 and §7.3.2) with a synthetically grown tumor using our forward model and two of the clinical real patient imaging data sets (RTRV; §7.2.4, ID's AAAN/AAMH). For a description of the considered cases, we refer to §7.1.1. For the registration-only experiments, we employ the same continuation scheme on the regularization parameter β_v as we use for our joint inversion scheme (see §6.3.3) and apply the same settings, solver tolerances, and allowed number of Newton and Krylov iterations as in the respective joint inversion runs using SIBIA (i.e., we invest the same effort on the registration side).

Results. We report quantitative results for 3 different cases of plain-vanilla registration with parameter-continuation of a healthy atlas brain to an abnormal brain in Fig. 7.2. The performance for the biophysically enhanced registration, i.e., the joint inversion moving-patient Picard iteration solution scheme, is given in Tab. 7.3. For the latter, we additionally report results for different dimensionalities of the basis for the parametrization of the tumor initial condition and different solver alternatives (i.e., also the moving-atlas Picard iteration solution scheme, investigated in §7.3). Qualitative results are given in Fig. 7.3.

Observations.

From Fig. 7.2, we observe that the plain-vanilla registration yields registration results of rather poor quality in the vicinity of the pathology for all three considered test cases. In five consecutive registration solves, the initial L_2-data-misfit is reduced by $37\% - 63\%$, which corresponds to a final

[9] We use the deviation of the determinant of the deformation gradient $\det(\nabla y)$ from 1 as an indicator or measure for good-natured mappings (that is, volume compression or expansion should be moderate to small).

		ATRV-CM3		RTRV-AAAN		RTRV-AAMH	
It	β_v	μ_{B,L^2}	DICE$_B$	μ_{B,L^2}	DICE$_B$	μ_{B,L^2}	DICE$_B$
initial	–	1.00	5.45E−1	1.00	4.37E−1	1.00	4.36E−1
1	1	9.81E−1	5.50E−1	9.23E−1	4.80E−1	9.29E−1	4.75E−1
2	1E−1	9.34E−1	5.68E−1	7.24E−1	6.08E−1	7.59E−1	5.79E−1
3	1E−2	8.62E−1	6.06E−1	5.05E−1	7.57E−1	5.75E−1	7.03E−1
4	1E−3	7.66E−1	6.58E−1	4.09E−1	8.22E−1	4.97E−1	7.58E−1
5	1E−4	6.33E−1	7.17E−1	3.77E−1	8.42E−1	4.72E−1	7.73E−1

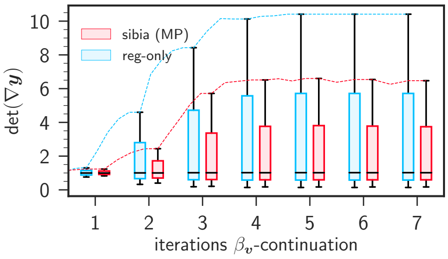


FIGURE 7.2 *Plain vanilla normal-to-abnormal registration. The table shows numerical results for normal-to-abnormal image registration using only the registration sub-component solver (see §5.3.2) without biophysical enhancement for synthetic (**ATRV**) and real data (**RTRV**) test cases. For the registration, we use the same parameter continuation scheme for β_v as used for our joint inversion approach: Beginning with $\beta_v = 1$, we repeatedly solve the registration problem while reducing the regularization weight by a factor of 10 until the lower bound of 1E−4 is reached. We use the same tolerances and upper bounds for the Newton and Krylov iterations as for the respective joint inversion experiment, i.e., we use opttol$_R$ = 1E−3, maxit$_{N,R}$ = 50, and maxit$_{K,R}$ = 80 for the synthetic test case **ATRV-CM3** and opttol$_R$ = 1E−2, maxit$_{N,R}$ = 10, and maxit$_{K,R}$ = 10 for the real data test cases **RTRV-AAAN** and **RTRV-AAMH**, respectively. For every test case, we report the average mismatch for the probability maps for the brain tissue labels μ_{B,L^2} and the mean Dice coefficient for brain tissue DICE$_B$ for the initial configuration and after every iteration of the continuation scheme. The figure shows the deviation of the determinant of the deformation gradient det(∇y) from one for the plain-vanilla registration solve (blue) and the joint inversion scheme (red) for the AAAN data set. This is a measure for the volume change induced by the registration; a value smaller than 1 indicates local compression, a value larger than 1 local expansion of tissue. The horizontal black bars indicate minimum, average and maximum of det$(\nabla y(x))$ over all $x \in \Omega$, colored bars indicate the standard deviation; the evolution of the maximum value is emphasized with additional dashed lines.*

TABLE 7.3 *Comparison: Quantitative results for joint registration and biophysical inversion. Numerical results for our joint inversion approach (moving-patient; §6.1) for synthetic (**ATRV**) and real data (**RTRV**) test cases. The two last rows show the joint inversion result after the last iteration of the Picard scheme of data sets AAAN and AAMH for adaptive selection of Gaussian basis functions; here, we additionally report results for the moving-atlas formulation §6.2. We use the same tolerances and iteration limits as in Fig. 7.2. For every test case, we report the average mismatch for the probability maps for the brain tissue labels μ_{B,L^2} and the mean Dice coefficient for brain tissue DICE$_B$ for the initial configuration and after every iteration of the continuation scheme.*

	ATRV-CM3				RTRV-AAAN				RTRV-AAMH			
β_v	μ_{B,L^2}	DICE$_B$	μ_{T,L^2}	DICE$_T$	μ_{B,L^2}	DICE$_B$	μ_{T,L^2}	DICE$_T$	μ_{B,L^2}	DICE$_B$	μ_{T,L^2}	DICE$_T$
initial	1.00	5.45E−1	1.00	0.00	1.00	4.37E−1	1.00	0.00	1.00	4.36E−1	1.00	0.00
1	9.23E−1	5.74E−1	1.00	0.00	9.55E−1	4.63E−1	1.00	0.00	9.59E−1	4.62E−1	1.00	0.00
1E−1	7.33E−1	6.85E−1	1.66E−1	8.81E−1	7.82E−1	5.61E−1	5.16E−1	6.94E−1	7.42E−1	5.60E−1	3.90E−1	8.43E−1
1E−2	4.90E−1	8.12E−1	1.36E−1	9.06E−1	5.28E−1	7.23E−1	4.50E−1	8.06E−1	5.05E−1	7.01E−1	3.13E−1	9.03E−1
1E−3	3.78E−1	8.69E−1	1.36E−1	9.14E−1	4.06E−1	8.01E−1	4.07E−1	8.69E−1	3.91E−1	7.66E−1	2.52E−1	9.39E−1
1E−4	3.71E−1	8.73E−1	1.35E−1	9.16E−1	3.72E−1	8.23E−1	3.90E−1	8.82E−1	3.60E−1	7.84E−1	2.20E−1	9.51E−1
1E−4	3.71E−1	8.73E−1	1.35E−1	9.16E−1	3.54E−1	8.35E−1	3.77E−1	8.91E−1	3.45E−1	7.92E−1	1.95E−1	9.57E−1
moving-patient, L_2-reg. (tumor), Φ-adaptive, $\sigma = 2\pi/128$					3.43E−1	8.37E−1	1.04E−1	9.72E−1	3.37E−1	7.94E−1	8.65E−2	9.78E−1
moving-atlas, L_2-reg. (tumor), Φ-adaptive, $\sigma = 2\pi/128$					3.11E−1	8.59E−1	2.36E−1	9.09E−1	3.07E−1	8.14E−1	1.03E−1	9.69E−1

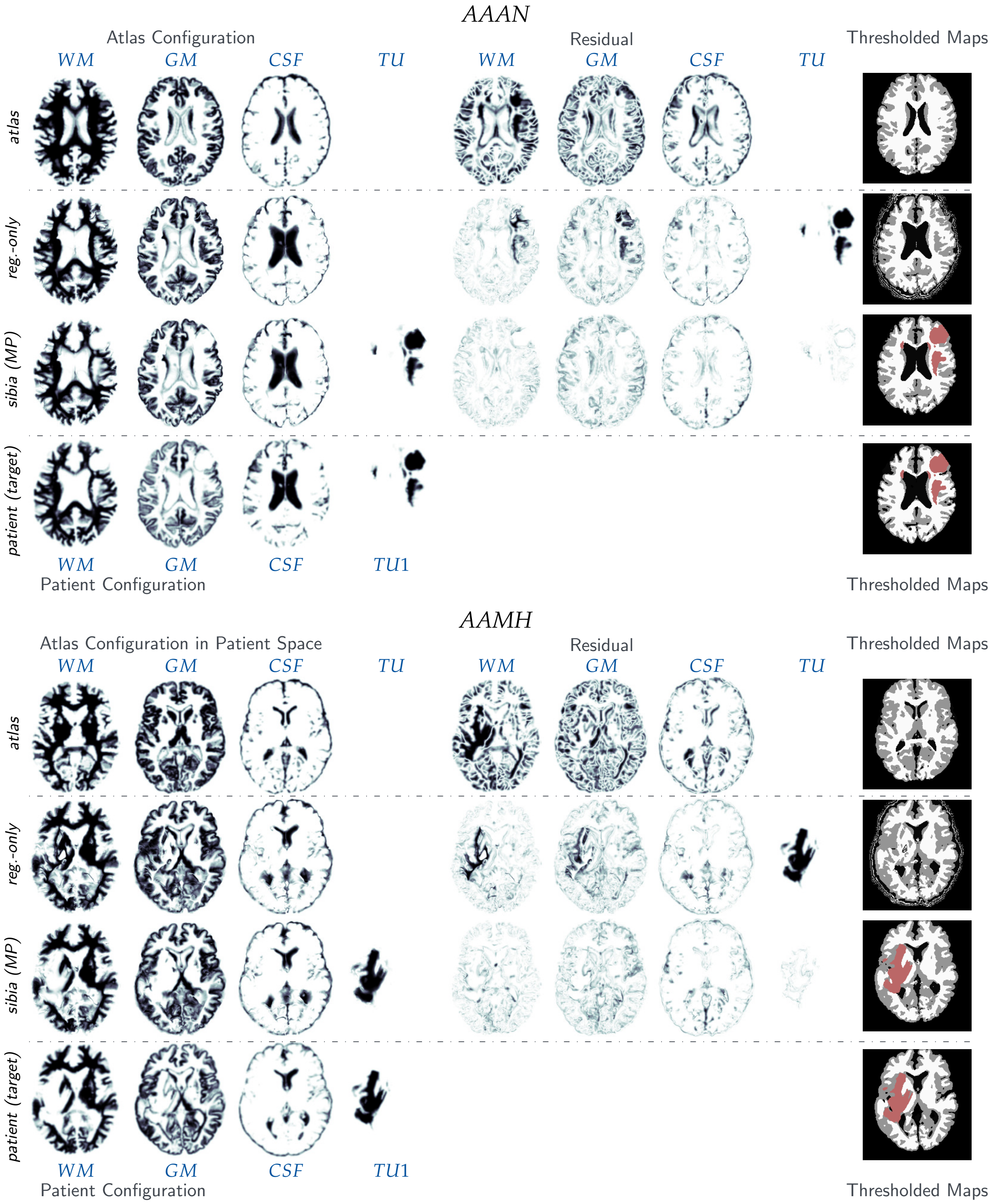


FIGURE 7.3 *Comparison: Qualitative results for plain-vanilla registration and the biophysically enhanced registration approach solved by the moving-patient Picard iteration scheme (§6.3.1) for the RTRV (ID's AAMH/AAAN) test case. We show simulation results for the per iteration detailed experiments for the registration-only experiments in Fig. 7.2 (second line) and the joint registration and biophysical inversion experiments in Tab. 7.3 (third line). The first line shows the configuration for the healthy atlas brain; the last line shows the configuration for the patient (target data).*

Dice score of 7.17E−1 for the semi-synthetic case (ATRV-CM3) and Dice scores of 8.42E−1 and 7.73E−1 for the real patient data sets AAAN and AAMH, respectively. From visual inspection of the qualitative registration results in Fig. 7.3, we observe that (as expected) the registration fails in the area of the pathology and in its vicinity; further away from the abnormality, the similarity/overlay of the deformed healthy brain and the abnormal brain for the plain-vanilla registration solve is acceptable and quite similar to the quality obtained with our joint inversion approach. The area occupied by the pathology and its immediate vicinity, however, is the region we are most interested in, if we consider segmentation or surgical planning (automated labeling) as target applications. The registration results obtained from the plain-vanilla registration are not meaningful in this area and the registration quality is not sufficient (tumor tissue is, in large parts, replaced by gray matter, which can be arbitrarily wrong). Furthermore, numerical artifacts can be identified in the images for the deformed template, indicating numerical problems due to the topological differences of the registered subjects and the ill-posedness of the resulting problem. This is reassured by monitoring the determinant of the deformation gradient $\det(\nabla \boldsymbol{y})$. The plot in Fig. 7.2 illustrates the deviation of $\det(\nabla \boldsymbol{y})$ from one for the plain-vanilla registration and our joint inversion scheme for the AAAN data set. $\det(\nabla \boldsymbol{y})$ varies greatly for the sole-registration solve, indicating a pronounced expansion of tissue; this behavior is rather non-physical. On the contrary, the joint registration and biophysical inversion approach developed within this thesis exhibits excellent normal-to-abnormal registration quality for the same cases. In particular, registration quality is still excellent in the area occupied by the pathology (cf. Fig. 7.3) As can be seen from Tab. 7.3, for the semi-synthetic case (ATRV-CM3), we reduce the initial L_2-data-misfit by 63% and improve the image overlay from an initial Dice score of 5.45E−1 to a Dice score of 8.73E−1 after biophysics-aided registration. Similar quality is obtained for the clinical data sets of real glioma patients: For the multi-focal AAAN case, we observe an error reduction of 65% and an increase in Dice score from initially 4.36E−1 to 8.35E−1; very similar results are obtained for the AAMH data set. Using the coupled approach, we further obtain a reconstruction of the tumor itself; this information is inaccessible by the plain-vanilla registration. We want to emphasize that the visual reconstruction of the tumor is excellent, yielding very high Dice scores of 9.16E−1 for ATRV-CM3 and 8.91E−1 and 9.57E−1 for ID's AAAN and AAMH, respectively.

As we will see in §7.2.4, the reconstruction quality of the pathology for our joint inversion scheme depends on the number n_p and the placement of the Gaussian basis functions that parametrized the tumor initial condition. Using the adaptive selection mode of Gaussian basis functions (described in §6.3.3), the tumor reconstruction quality can be further improved to an overlay quantified by Dice scores of 9.72E−1 and 9.78E−1 for patient ID's AAAN and AAMH which corresponds to a reduction of the L_2-data-misfit of more than 90% with respect to the initial configuration. The improved reconstruction of the tumor area only slightly affects the registration quality in terms of data-misfit/overlay of the surrounding healthy brain tissue. Using the moving-atlas formulation we can further improve the registration quality for the healthy brain tissue.

Conclusion *Plain-vanilla registration exhibits severe difficulties in registering normal brains to brains with abnormalities (such as, e.g., a grown tumor) due to topological differences. While registration quality is acceptable or even good far away from the pathology, the obtained result is not meaningful, misleading, and impractical in the vicinity of the abnormality. Using our joint registration and biophysical inversion approach, which augments the registration formulation by further biophysical models/constraints that capture the underlying dynamics of abnormality evolution, improves the obtained normal-to-abnormal registration quality considerably. The deformations in the region around the pathology remain especially meaningful.*

We arrive at improved Dice scores between 8.73E−1 *and* 8.14E−1 *for synthetic and real data in what is an extremely challenging problem featuring multi-focal tumors and highly complex pathologies. Furthermore, our scheme provides a reconstruction of the observed pathology as obtained from forward simulations using the calibrated biophysical model. We conclude that our moving-patient Picard iteration solution scheme for the joint inversion for brain tumor biophysical model parameters and registration velocity is a very powerful tool for inter-subject normal-to-abnormal registration, such as used in automated segmentation tools.*

7.2.2 Test Case ATAV: Analytic Tumor and Analytic Velocity

Purpose. The here considered experiments serve as a proof of concept. We examine the moving-patient Picard iteration solution scheme with respect to its normal-to-abnormal registration performance, quality of data reconstruction (similarity), and convergence of the methodology to a known ground truth. We further analyze the sensitivity of our approach with respect to perturbations in the model and model parameters (i.e., reaction-diffusion (ATAV-DIF) vs. reaction-only (ATAV-REAC), and different model parameters for the inversion).

Setup. A general description of this fully synthetic setting is given in §7.1.1; we complete the description with some specific information. The center of mass for the synthetic tumor is set to $(x^1, x^2, x^3) = 2\pi \cdot (0.285, 0.36, 0.5)$. As initial condition for the artificial tumor generation, we enable two of the Gaussians at the center of the grid of Gaussians. Whereas we disable diffusion ($k_f = 0$) in ATAV-REAC, we use the full tumor model, including diffusion ($k_f \neq 0$) for ATAV-DIF. The same growth rates ρ_w, ρ_g with scaling ρ_f are used for growing the tumor and for the inversion to reconstruct the initial condition. In ATAV-DIF, we use values for the reaction and diffusion coefficients for the inversion in the Picard iterations, that are either the same or differ from those used for the generation of the tumor. See §7.1.3 and Tab. 7.2 for further details on the parameters.

Results. As a baseline, we report results for the sole registration of healthy anatomy (i.e., neglecting the tumor forward solve to generate the data) in Tab. 7.4. In addition to that, we quantify the numerical error of our scheme for solving the transport equations. This is done by solving the forward problem twice, once with the original and once with the negative velocity. The associated error is given by

$$\frac{\|c^{(\cdot,1)^\star}\varepsilon_{\text{adv}} := -\mathcal{R}^{\rightarrow}(-\boldsymbol{v}^\star, \mathcal{R}^{\rightarrow}(\boldsymbol{v}^\star, c^{(\cdot,1)^\star}))\|_2}{\|c^{(\cdot,1)^\star}\|_2} = 9.37\text{E}{-2} \tag{7.2}$$

Results for the inversion using a reaction-only tumor model (ATAV-REAC) are presented in Fig. 7.5 and assess the reconstruction quality (Dice and residuals) with respect to the iteration index. We also report the error between the ground truth $c_A(0)^\star = \Phi \boldsymbol{p}^\star$ and $\boldsymbol{v}^\star$ and the estimated iterates, as well as the relative norm of the gradient of the fully coupled problem in (6.4).

For inversion with non-zero diffusion (ATAV-DIF), quantitative results are shown in Tab. 7.6 In addition to reconstruction quality and gradient reduction, we list the runtime for the Picard scheme per iteration and the percentage spent in each individual solver (tumor and registration), respectively. Note that tumor and registration runtimes do not add up to 100% as further parts of the code (such as the calculation of the reduced gradient and the steering of the Picard iteration) are not included in the measurements.

Observations. The most important observations are (*i*) that the reconstructed data (tumor and registered anatomy) is in excellent agreement with the patient data, and (*ii*) we are able to reduce

TABLE 7.4 *Results for the **analytic tumor / analytic velocity (ATAV)** test case. We report registration-only results between healthy atlas and (advection generated) healthy patient. The table shows values for the relative mismatch for the geometry (μ_{B,L^2}) and the associated Dice coefficient DICE$_B$ as well as the relative ℓ^2-error for the reconstruction of the velocity field e_{v,L^2} with respect to the ground truth $\boldsymbol{v}^\star$.*

maxit$_R$	μ_{B,L^2}	DICE$_B$	e_{v,L^2}
$(50, 80)$	1.78E−1	9.37E−1	3.59E−1
$(10, 20)$	1.68E−1	9.36E−1	3.14E−1

TABLE 7.5 *Quantitative results for the **analytic tumor / analytic velocity reaction-only (ATAV-REAC)** test case; ground truth: ($\rho_f = 15$, $\rho_w = 1$, $\rho_g = 0$, $k_f = 0$, $\boldsymbol{p} = \boldsymbol{p}^\star$, $\boldsymbol{v} = -\boldsymbol{v}^\star$). We report the average mismatch for the probability maps for the brain tissue labels μ_{B,L^2} and the tumor μ_{T,L^2}, the mean Dice coefficient for brain tissue DICE$_B$ and tumor DICE$_T$. The reconstruction quality is given in terms of convergence of $\boldsymbol{v}^k$ and $c_A^k(0)$ towards the ground truth $\boldsymbol{v}^\star$ and $c_A^\star(0)$ (e_{v,L^2} and e_{c0,L^2}). We can not expect this error to go to zero for several reasons. First, we loose information when we construct the test case (zero gradients in the intensity of the image). Second, our numerical solver introduces errors (in particular, the solver for the transport equations). We, in addition, report the change in update in the velocity $\boldsymbol{v}$ across successive iterations $\delta\boldsymbol{v} = \|\boldsymbol{v}^k - \boldsymbol{v}^{k-1}\| / \|\boldsymbol{v}^{k-1}\|$. Finally, we also list the relative norm of the gradient for the coupled problem in (6.4) ($\|\boldsymbol{g}\|_{rel}$).*

iteration	β_v	μ_{B,L^2}	DICE$_B$	μ_{T,L^2}	DICE$_T$	e_{v,L^2}	$\delta\boldsymbol{v}$	e_{c0,L^2}	$\|\boldsymbol{g}\|_{\text{rel}}$
initial	–	1.00	7.10E−1	1.00	0.00	1.00	–	1.00	1.00
1	1	9.37E−1	7.27E−1	1.00	0.00	9.60E−1	–	2.54E−1	9.98E−1
2	1E−1	6.30E−1	8.01E−1	2.26E−1	8.63E−1	8.50E−1	9.77E−1	2.05E−1	5.91E−2
3	1E−2	3.67E−1	8.88E−1	1.47E−1	9.19E−1	6.31E−1	9.33E−1	1.65E−1	3.07E−2
4	1E−3	1.71E−1	9.40E−1	7.91E−2	9.66E−1	3.35E−1	6.62E−1	1.40E−1	1.95E−2
5	1E−4	1.57E−1	9.47E−1	6.55E−2	9.74E−1	3.59E−1	3.65E−1	1.36E−1	1.32E−2
6	1E−4	1.57E−1	9.47E−1	6.39E−2	9.73E−1	3.59E−1	1.45E−5	1.33E−1	1.21E−2
7	1E−4	1.57E−1	9.47E−1	6.29E−2	9.73E−1	3.59E−1	1.45E−5	–	1.21E−2

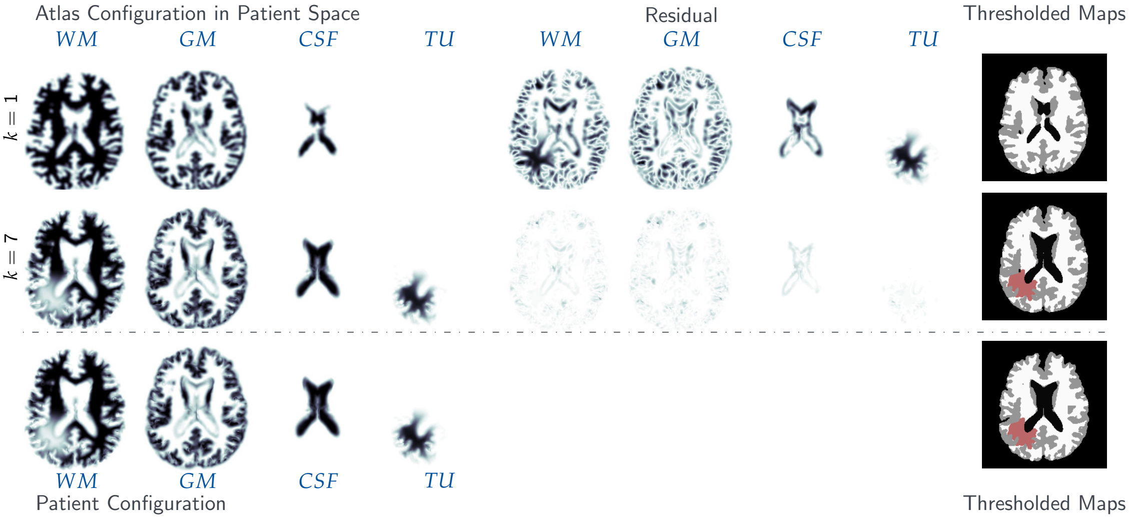


FIGURE 7.4 *Qualitative results for the **analytic tumor / analytic velocity (ATAV)** test case. The figure shows probability maps for the labels of the healthy atlas brain and the patient brain with a tumor generated from a tumor grown in the atlas and known atlas to patient advection velocity (see text for details; axial-slice 64 of a 3D volume). We show the initial configuration for the problem (top row; iteration $k = 1$), the final configuration after joint registration and tumor inversion (middle row; iteration $k = 7$; the atlas image probability maps are transported to the patient space), and the target patient data (reference image; bottom row). Each row contains (from left to right) the probability maps for WM, GM, CSF, and TU, the residual differences (if available) between the probability maps, and a hard segmentation based on the given probabilities for the individual tissue classes.*

TABLE 7.6 *Results for the* ***analytic tumor / analytic velocity with non-zero diffusion (ATAV-DIF)*** *test case; ground truth: ($\rho_f = 10$, $\rho_w = 1$, $\rho_g = 0$, $k_f = 1.00\text{E}{-2}$, $k_w = 1$, $k_g = 0$, $\boldsymbol{p} = \boldsymbol{p}^\star$, $\boldsymbol{v} = -\boldsymbol{v}^\star$). We show values for the (summed) norm of the residual between the respective probability maps for the different brain tissue classes μ_{B,L^2} and tumor μ_{T,L^2} in patient space, the mean Dice coefficient for brain tissue DICE$_B$ and tumor DICE$_T$, respectively, as well as the relative norm of the gradient $\|\boldsymbol{g}\|_{rel}$ for the global coupled problem (6.4). We report results for different values of $\rho_f \in \{5, 10, 15\}$ used in the inversion ($\rho_f = 10$ is the ground truth). We list the time spent per iteration (in seconds; top run) or in total (in seconds; bottom runs) for the entire Picard inversion and the amount of that time spent in the tumor inversion and image registration (in percent (top run), in seconds (bottom runs)). Note that the latter sums up to less than 100% as we do not explicitly measure time spent in additional coupling functionality and forward solvers. These runs are performed using 64 MPI tasks on three nodes of `HazelHen` (see §7.1.5 for details). The top block shows the inversion with respect to the Picard iteration index for the correct parameters (ground truth) for ρ_f and k_f. The four rows on the bottom show the final result for our Picard scheme for different parameter and model combinations.*

iterations	β_v	μ_{B,L^2}	DICE$_B$	μ_{T,L^2}	DICE$_T$	$\|\boldsymbol{g}\|_{\text{rel}}$	T^{it} [s]	T^{tu}_{inv} [%]	T^{reg}_{inv} [%]
initial	–	1.00	7.14E−1	1.00	0.00	1.00	–	–	–
1	1	9.32E−1	7.32E−1	1.00	0.00	9.94E−1	2.11E+3	99.7	0.2
2	1E−1	6.52E−1	8.01E−1	2.08E−1	8.56E−1	7.45E−2	2.88E+2	96.0	2.1
3	1E−2	3.79E−1	8.88E−1	1.47E−1	9.12E−1	3.87E−2	3.94E+2	91.8	6.9
4	1E−3	1.75E−1	9.39E−1	8.08E−2	9.63E−1	2.45E−2	3.98E+2	72.7	26.1
5	1E−4	1.60E−1	9.48E−1	5.93E−2	9.78E−1	1.77E−2	2.00E+2	35.4	62.1
6	1E−4	1.60E−1	9.48E−1	5.77E−2	9.77E−1	1.72E−2	1.34E+2	46.6	49.6
7	1E−4	1.60E−1	9.47E−1	5.63E−2	9.76E−1	1.73E−2	7.10E+1	0.0	93.0
ρ_f	k_f	μ_{B,L^2}	DICE$_B$	μ_{T,L^2}	DICE$_T$	$\|\boldsymbol{g}\|_{\text{rel}}$	T^{tot} [s]	T^{tu}_{inv} [s]	T^{reg}_{inv} [s]
$\rho_f = 5$	$k_f^\star = 1\text{E}{-2}$	1.61E−1	9.46E−1	6.48E−2	9.67E−1	2.21E−2	4.37E+3	3.95E+3	3.97E+2
$\rho_f^\star = 10$	$k_f^\star = 1\text{E}{-2}$	**1.60E−1**	**9.47E−1**	**5.63E−2**	**9.76E−1**	1.73E−2	3.60E+3	3.17E+3	4.01E+2
$\rho_f = 15$	$k_f^\star = 1\text{E}{-2}$	1.61E−1	9.48E−1	6.39E−2	9.72E−1	1.41E−2	3.96E+3	3.53E+3	4.00E+2
$\rho_f = 10$	$k_f = 0$	1.60E−1	9.48E−1	5.95E−2	9.71E−1	1.74E−2	**4.25E+2**	**2.25E+1**	4.01E+2

the reduced gradient (6.9) by two orders of magnitude in 7 iterations of our Picard scheme for the reaction-only as well as the reaction-diffusion experiments.

The numerical error for the advection in (7.2) is 9.37E−2. We can see in Tab. 7.5 that the relative mismatch for the anatomy obtained for our iterative Picard scheme is in the order of the advection error for the forward image registration problem. Furthermore, the reconstruction of $c^{(\cdot,0)^\star}$ and $c^{(\cdot,1)^\star}$ seems to be bounded by this error. In fact, due to the advection error that leads to a mismatch in this order in the atlas domain, this is the best we can expect without over-fitting the data. Similar observations can be made if we compare the inversion results with the results obtained for the registration of healthy brains (neglecting the tumor simulations) reported in Tab. 7.4. Hence, the quality of tumor reconstruction is comparable to the quality of pure image registration between the healthy geometries. This is an excellent result that clearly demonstrates the potential of our approach. The obtained Dice coefficient for the brain anatomy is in the order of what we see for the sole registration of healthy anatomies.[10] Note that the mismatch between the true velocity $\boldsymbol{v}^\star$ and the recovered velocity $\boldsymbol{v}^k$ reported for ATAV-REAC is due to the fact that image registration is an inherently ill-posed problem: the velocity can only be reconstructed exactly in image areas with non-zero gradients and if there are only non-zero intensity differences between the images to be registered in areas that do correspond to one another[11]. In addition, we ask for the reconstruction of a vector field from scalar data.

The Dice coefficient for the brain anatomy increases from 7.10E−1 to 9.47E−1 in ATAV-REAC, where we obtain a final Dice coefficient of 9.73E−1 for the tumor. These are results of excellent quality for normal-to-abnormal registration, the main objective of the here evaluated moving-patient

[10] In fact, it is even slightly better for the reaction-only experiments (ATAV-REAC). This slight increase might be a consequence of numerical inaccuracies in our scheme, and discrepancies in the number of Newton-steps and Krylov iterations taken.

[11] As in any formulation based on an L^2-distance functional, it is the mismatch between the reference and template image and the gradient of the deformed template image that drive the optimization (see (5.26f)).

joint inversion scheme. The results for the reaction-diffusion experiments (ATAV-DIF) show that for the moving-patient scheme and the used regularizers, the specifically employed tumor model does not have a significant impact on the quality of the inversion (Dice and mismatch); the registration solver compensates for smaller variations in the tumor model.

We can furthermore see that we can significantly reduce the norm of the reduced gradient (6.9) to 1.21E−2. We can also see that once we have reached the target regularization parameter $\beta_v = 1\text{E}{-4}$, we do not make any more progress. The update for the velocity tends to zero, the changes in the reduced gradient are small, and the error measures (residual and Dice) do not longer change significantly.

For ATAV-DIF, we see that we obtain a slightly better mismatch (5.63E−2) and Dice coefficient (9.76E−1) for the tumor probability map if we use the correct ρ_f. However, for the employed moving-patient scheme, the mismatch results for moderately wrong tumor parameters are quite indistinguishable. Even if switching off the diffusion operator (bottom line in Tab. 7.6), we do not observe a significant deterioration of the obtained proximity of the model prediction to the observed data. This underpins the hypothesis that for the moving-patient scheme we obtain excellent agreement between patient data and atlas data (i.e., normal-to-abnormal registration quality) irrespective of the model choice; the registration eventually compensates insufficient proximity of the tumor growth simulation to the observed data. This can be explained by the parametrization of the initial condition in conjunction with the employed Φ-weighted L_2-regularization of $\boldsymbol{p}$. We can reconstruct complex tumor shapes even with a simple model. Overall, this indicates that using a reaction-only model might be sufficient for pure diffeomorphic image registration with the objective to obtain good normal-to-abnormal registration quality. A comparison of the total runtime for the last row in Tab. 7.6 to the total runtimes attained when enabling diffusion shows that we can save a factor of 10 in runtime by disabling diffusion. Another interesting behavior within our scheme is that, during the first few iterations, most time is spent in the tumor inversion, whereas as we reduce β_v, the registration does most of the work. This is to be expected, since the runtime of our scheme (more precisely, the condition number of the Hessian) for diffeomorphic registration, is not independent of β_v [Man15; Man17b].

Conclusion *We conclude that our Picard scheme is efficient and converges to a valid (local) minimum in the search space (we reduce the relative global gradient, the distance measure, and significantly increase the Dice coefficient). We obtain an excellent agreement between the data (patient tumor and geometry) and the predicted state (transported atlas geometry and predicted tumor) with a final Dice coefficient of* 9.47E−1 *and* 9.73E−1 *for the labels of the anatomy and tumor for reaction-only (ATAV-REAC) tumor-growth model and similar values for the reaction-diffusion (ATAV-DIF) case.*

The integration of a diffusion model into our inversion is very costly, at least for our current implementation. Designing a more efficient forward solver for the diffusion operator requires more work. We could demonstrate that our parametrization of the initial condition allows us to generate high-fidelity registration results, especially for the healthy anatomy, irrespective of the model that has been used to generate the data. These are clearly preliminary results, but they provide some evidence that reaction-only models might be sufficient for pure normal-to-abnormal image registration (something that is certainly not the case if we target parameter identification and patient specific tumor growth prediction [Hog08a] to aid clinical decision making; this requires more complex, high-fidelity models. The objective of tumor parameter estimation is targeted in §7.3 for the biophysically more meaningful moving-atlas formulation).

We can also observe that there are subtle differences in the reconstruction quality of the tumor if we use *the ''correct'' growth rate for the inversion for ATAV-DIF; these differences are more pronounced if we restrict the initial condition to a point source (see analysis and discussion in §7.3). Overall, we conclude that we (i) can neglect the diffusion model in the context of diffeomorphic registration and compensate the resulting loss in accuracy by a higher-dimensional Gaussian basis, (ii) might be able to identify appropriate growth rates if we run multiple inversions for different parameters with a sparsity constraint for the Gaussian basis (i.e., limiting the initial condition to a point source).*

7.2.3 Test Case ATRV: Analytic Tumor and Real Velocity

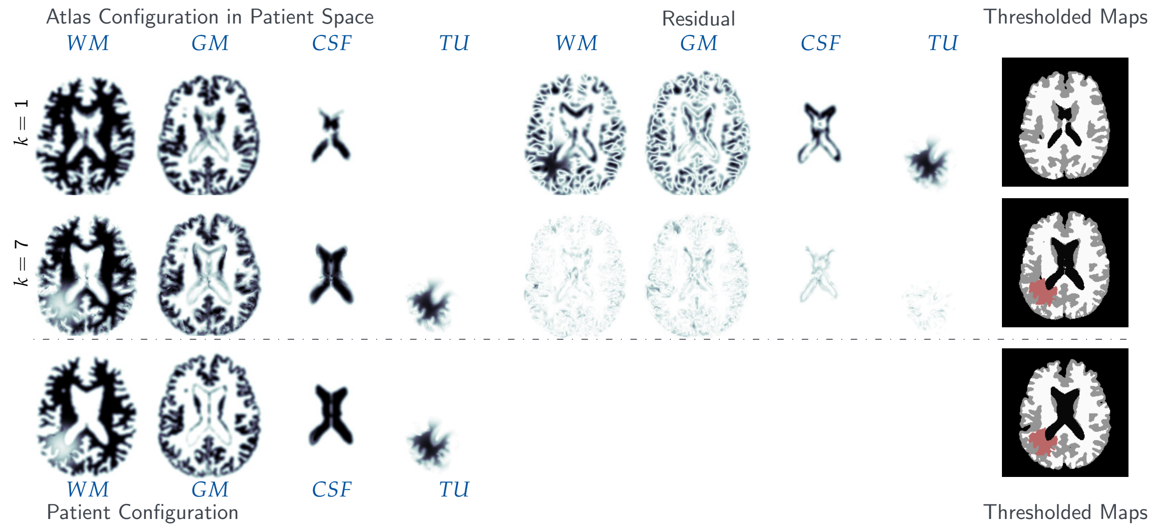
Purpose. We test our algorithm in the context of multi-subject registration for real brain data of two different individuals. We furthermore test the capabilities of our framework for the registration of an atlas to a patient's image that contains a multifocal tumor.

Setup. The patient data is no longer generated from the atlas image. We use an image of a different individual instead. Accordingly, we no longer know the ground truth velocity field; see also §7.1.1 for a more detailed description. For the specific experiments considered here, we (*i*) choose the same initial condition parameters, i.e., location and $\boldsymbol{p}^{\star}$, as in the previous ATAV case, but also (*ii*) consider multifocal tumors, i.e., tumors consisting of several separate components. For the latter, we choose the seed randomly, which in turn means that we have to choose the Gaussian basis functions adaptively in order to be able to recover all abnormal areas in the patient's image. We arrange the Gaussian basis functions as a regular grid around the center of mass of the tumor and choose their standard deviation σ automatically based on the tumor's bounding box and the prescribed number n_p of Gaussians; see §6.3.3. This is an extremely important step to make our solver applicable to real patient data, something we will explore in the next section. The initial conditions are generated using random tumor seeds activating a maximum of 3 Gaussians in the $x^3 = \text{height}/2-$plane. This creates a multifocal template tumor c_D. We solve the forward problem with diffusion disabled, i.e., reaction-only. We employ the tumor parameters used to generate this artificial tumor also in our inversion. To account for the possibility of tumor growth at far apart locations in the brain, we use a larger number of parameters $n_p = 343$, (see Tab. 7.2).

Results. We report qualitative and quantitative simulation results for the monofocal reaction-only case in Fig. 7.5. A summary of the experiments for the multifocal tumor cases (two runs) is given in Fig. 7.6.

Observations. The most important observations are that (*i*) we achieve an excellent registration quality with a Dice score of 9.34E−1 and 9.46E−1 for the hard segmentations of the anatomy and the tumor, respectively, and that (*ii*) our scheme seems to be still converging to a (local) minimum of the joint tumor inversion-registration problem in this more realistic test case scenario.

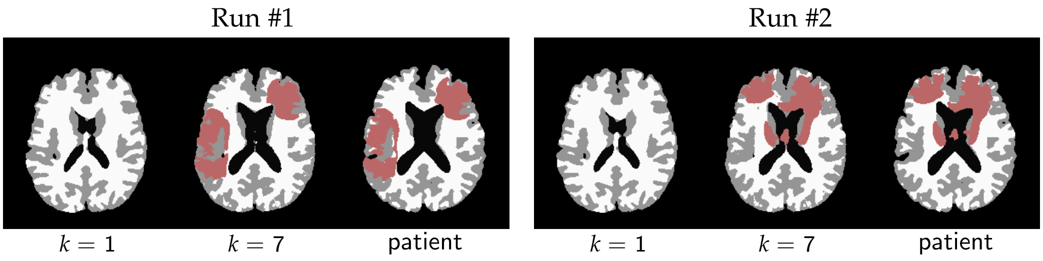
Overall, the results slightly deteriorate compared to the former test cases. We achieve a relative mismatch and Dice score for the anatomy that is about $\mathcal{O}(1\text{E}{-1})$ and $\mathcal{O}(1\text{E}{-2})$ worse, respectively, than in the former experiments. The reduction of the gradient of the coupled optimization problem of about two orders is comparable to the former test cases. Interestingly, there is a stronger imbalance between the time spent in the registration and the tumor inversion; the registration takes up more than 90% of the runtime already in the second iteration. This indicates that running our solver on real inter-subject problems at a higher resolution significantly increases the complexity of the registration problem.



iterations	β_v	μ_{B,L^2}	DICE$_B$	μ_{T,L^2}	DICE$_T$	$\lVert g \rVert_{\text{rel}}$	T^{it} [s]	T^{tu}_{inv} [%]	T^{reg}_{inv} [%]
initial	–	1.00	5.43E−1	1.00	0.00	1.00			
1	1	9.35E−1	5.84E−1	1.00	0.00	9.98E−1	6.03E+1	46.3	47.3
2	1E−1	7.12E−1	7.06E−1	2.48E−1	8.53E−1	8.47E−2	5.93E+1	6.8	92.5
3	1E−2	4.48E−1	8.38E−1	1.90E−1	9.11E−1	3.33E−2	1.51E+3	0.3	99.7
4	1E−3	2.96E−1	9.02E−1	1.54E−1	9.39E−1	1.36E−2	1.92E+3	0.1	99.8
5	1E−4	2.66E−1	9.13E−1	1.45E−1	9.48E−1	1.02E−2	4.31E+2	0.3	99.6
6	1E−4	2.66E−1	9.13E−1	1.44E−1	9.47E−1	1.27E−2	1.55E+2	0.8	98.9
7	1E−4	2.66E−1	9.13E−1	1.43E−1	9.46E−1	1.24E−2	1.51E+2	0.0	99.7
final	1E−4	2.66E−1	9.13E−1	1.43E−1	9.46E−1	1.24E−2	4.29E+3	4.11E+1	4.25E+3

FIGURE 7.5 *Results for the* ***analytic tumor with real velocity (ATRV)*** *test case, ground truth: ($\rho_f = 15$, $\rho_w = 1$, $\rho_g = 0$, $k_f = 0$, $\boldsymbol{p} = \boldsymbol{p}^\star$ (in patient domain); $\boldsymbol{v}$ N/A). The figure shows probability maps for the labels of the healthy atlas brain (top row) and the patient brain (with a synthetically generated tumor; bottom row). The patient's tumor is generated by solving the forward problem in the healthy patient data set with an initial condition parametrized by $\boldsymbol{p}^\star$. We show axial slice 128. We report the (summed) mismatch for the brain tissue probability maps (μ_{B,L^2}) and the tumor probability map (μ_{T,L^2}) in patient space, the mean Dice coefficient for the hard segmentation corresponding to the brain tissue (DICE$_B$) and the tumor DICE$_T$, respectively, as well as the relative norm of the gradient for the coupled problem (6.4) ($\lVert \boldsymbol{g} \rVert_{rel}$). We also show the run time per iteration in seconds (T^{it}), and the percentages (T^{tu}_{inv}) and (T^{reg}_{inv}) of this runtime spent in the tumor solver and the image registration solver, respectively. Note that the latter sums up to less than 100% as we do not explicitly measure time spent in additional coupling functionality and in the forward solvers. The last row shows the final values for mismatch and Dice as well as (accumulated) absolute run times for the sub-components in seconds.*

From the multifocal tumor cases, we observe that our moving-patient Picard iteration scheme with the adaptive parametrization of the initial condition is effective when it comes to the registration of a tumor-free atlas with a patient image that contains a multifocal tumor. We achieve almost the same Dice coefficient (up to between $\mathcal{O}(1\text{E}{-}2)$ and $\mathcal{O}(1\text{E}{-}3)$) for the hard segmentations associated with the probability maps for the tumor and brain anatomy. In particular, we can increase the Dice coefficient from 5.29E−1 to 9.12E−1 for run #1 and from 5.36E−1 to 9.08E−1 for run #2, respectively (see Fig. 7.6). The Dice coefficients for the tumor are 9.49E−1 and 9.52E−1. These values are comparable to what we achieved for ATRV. Moreover, we achieve a similar reduction of the relative norm of the coupled reduced gradient of about two orders of magnitude. We can also see that the overall runtime for the tumor inversion increases as we increase the number n_p of parameters, from 4.11E+1 seconds to 3.09E+2 or 1.17E+2 seconds (total tumor inversion runtimes in Fig. 7.6 and Fig. 7.5). However, this increase in runtime does not have a significant effect on the overall runtime of the total Picard scheme ($\mathcal{O}(5\text{E}{+}3)$ seconds), which is dominated by the registration.



run #		β_v	$\mu_{B,L2}$	DICE$_B$	$\mu_{T,L2}$	DICE$_T$	$\|g\|_{\text{rel}}$	T^{tot} [s]	T^{tu}_{inv} [s]	T^{reg}_{inv} [s]
#1	initial	–	1.00	5.29E−1	1.00	0.00	1.00	–	–	–
	final	1E−4	2.64E−1	9.12E−1	1.43E−1	9.49E−1	1.75E−2	3.29E+3	3.09E+2	2.99E+3
#2	initial	–	1.00	5.36E−1	1.00	0.00	1.00	–	–	–
	final	1E−4	2.64E−1	9.08E−1	1.46E−1	9.52E−1	1.40E−2	4.91E+3	1.17E+2	4.81E+3

FIGURE 7.6 *Results for the* ***analytic multi-focal tumor with real velocity (ATRV-MF)*** *test case (two runs), ground truth: ($\rho_f = 15$, $\rho_w = 1$, $\rho_g = 0$, $k_f = 0$, $\boldsymbol{p} = \boldsymbol{p}^\star$ (in patient domain), $\boldsymbol{v}$ N/A). We display the healthy atlas brain ($k = 1$), the reconstruction with tumor in patient space after the last Picard iteration ($k = 7$), and the given segmented patient image with tumors (axial-slice 128; top block). We report the (summed) mismatch for the brain tissue probability maps (μ_{B,L^2}) and the tumor probability map (μ_{T,L^2}) in patient space, the mean Dice coefficient for the hard segmentation corresponding to the brain tissue (DICE$_B$) and the tumor DICE$_T$, respectively, as well as the relative norm of the gradient for the coupled problem (6.4) ($\|g\|_{rel}$). We also show the run time until convergence of the Picard scheme in seconds (T^{tot}), and the (over all iterations accumulated) time (T^{tu}_{inv}) and (T^{reg}_{inv}) spent in the tumor solver and the image registration solver, respectively. Note that the latter sums up to less than the reported total runtime as we do not explicitly measure time spent in additional coupling functionality as well as in the forward solvers.*

Conclusion *We conclude that we can still achieve an excellent registration quality with a Dice score of* 9.34E−1 *and* 9.46E−1 *for the hard segmentations for the anatomy and the tumor, respectively, if we perform our joint inversion moving-patient scheme on real brains of different individuals representing a normal-to-abnormal registration scenario for the healthy atlas and pathologic patient. We can furthermore effectively deal with multifocal tumors, obtaining similar registration and reconstruction quality at only slightly higher computational cost (due to an increased number of Gaussian basis functions in the parametrization of the initial conditions).*

7.2.4 Test Case RTRV: Real Tumor and Real Velocity

Purpose. We test our approach on real data of patients diagnosed with glioma tumors and study the registration quality for a variety of parameter choices for the tumor growth model. The real data featuring large inter-subject anatomical variability poses significant challenges on the registration solver, and also the simplistic tumor model represents, in the best case, a rough description of the underlying tumor growth dynamics.

Setup. This test scenario consists of real patient brains with real tumors for which parameters are unknown. The patient data sets are the first proposal for a patient segmentation produced in the first iteration of GLISTR [Goo13]. We provide additional details in §7.1.1 and §7.1.2. Real data requires to identify the support of the domain spanned by the Gaussian basis functions for the tumor initial condition parametrization automatically. We consider the two automatic data-driven selection modes for the Gaussian basis function described in §6.3.3. For the tumor diameter (bounding-box) based selection mode, we define a spacing of $\delta = 1.5\sigma$ and choose the standard deviation σ for every unseen patient such that $n_p = 343$ basis functions are placed uniformly within the determined bounding box of the tumor. For the tumor shape- and intensity-based adaptive selection mode, we set $\delta = 2\sigma$ with $\sigma = {}^{2\pi}/_{128}$ and require a volume fraction of 0.99 of the σ-ball of each candidate Gaussian to be occupied by tumor cells in order to get selected. In contrast to the synthetic test cases ATAV and ATRV, we allow the tumor to also grow in gray matter instead of in white matter only, but with a 10 times smaller characteristic diffusivity, and a reaction parameter that is five times smaller than in white matter (see Tab. 7.2 for details). We use a variety of models and parameter settings in our Picard scheme for the two patients AAMH and AAAN to not only assess the performance of our method, but also to study its sensitivity towards parameter changes and model complexity. We vary the reaction parameter ρ_f between 0 and 15 and choose the diffusion coefficient k_f either as 0 or 1.00E$-$2. To foster a short time-to-solution, we set tighter maxit bounds on the employed Newton and Krylov solver, i.e., $\text{maxit}_R = (10, 20)$ and $\text{maxit}_T = (30, 30)$ for registration and tumor inversion, respectively. See §7.1.3 for additional details on the setup of the test case and the parameters.

Results. Fig. 7.7 – Fig. 7.9 show the healthy atlas ($k = 1$) in the top row and the corresponding patient image with tumor in the bottom row for axial, sagittal, and coronal orientations. The hard segmentations for the results computed with the proposed approach are shown in the middle row using the same orientations. In Tab. 7.7, we summarize the results for all patient data sets. We report the initial and final values for the mismatch and Dice coefficients associated with the probability maps for the tumor and the brain anatomy, as well as the relative norm of the gradient of the coupled problem in (6.4) for both data-driven selection modes of the Gaussian basis functions, described in §6.3.3. We also report timings for the entire inversion. Fig. 7.11 shows more detailed images of the probability maps for the patient AAMH (complex and large tumor). Results for varying reaction and diffusion model parameters for AAMH and AAAN are listed in Tab. 7.8. We furthermore study the impact of the number and placement of the Gaussian basis functions (as determined by the two available data-driven selection modes) on the reconstruction quality, convergence and runtime in Tab. 7.9 for the two data sets AAAN and AAMH. Note that all parameter choices refer to the model used for tumor reconstruction in the moving-patient Picard scheme. The true growth parameters of the tumors are unknown.

Observations. The most important observation is that we obtain very good registration results—qualitatively and quantitatively—in what is an extremely challenging normal-to-abnormal registration problem. From visual inspection of the data alone (Fig. 7.7 through Fig. 7.9) we can immediately see that there are significant anatomical differences between the atlas image, the patient images, and across patients. The tumors vary significantly in shape and size. Also, the actual clinical data has sharper edges than the brain data used for our synthetic cases. Overall, this poses considerable challenges to our framework. Notice also, that our biophysical model for tumor evolution is over-simplified and features various shortcomings. The results reported in Fig. 7.7 through Fig. 7.9 and in Fig. 7.11 clearly demonstrate that the deformed atlases are in very good agreement with the patient data for all six subjects. We reach Dice coefficients between 7.87E$-$1 to 8.44E$-$1 and 8.74E$-$1 to 9.75E$-$1 for the probability maps associated with the anatomy and the tumor if the diameter-based Φ-selection mode is employed with a total number of $n_{p=343}$ Gaussians. These results are slightly worse than those obtained for the artificially grown tumors in the former sections, but still are competitive. We also note that the initial Dice coefficients for the anatomy range between 2.92E$-$1 and 4.74E$-$1 for these data, which is drastically worse than what we have seen in our synthetic test cases. Increasing the number of Gaussian basis functions and allowing for tumor-shape-driven placement of the latter has the potential to significantly improve the visual tumor reconstruction[12];

[12] Note, that this improved reconstruction is due to a better representation of the initial condition, i.e., better interpolation of the observation, and not due to more accurate biophysical model parameters. Here we are mainly interested

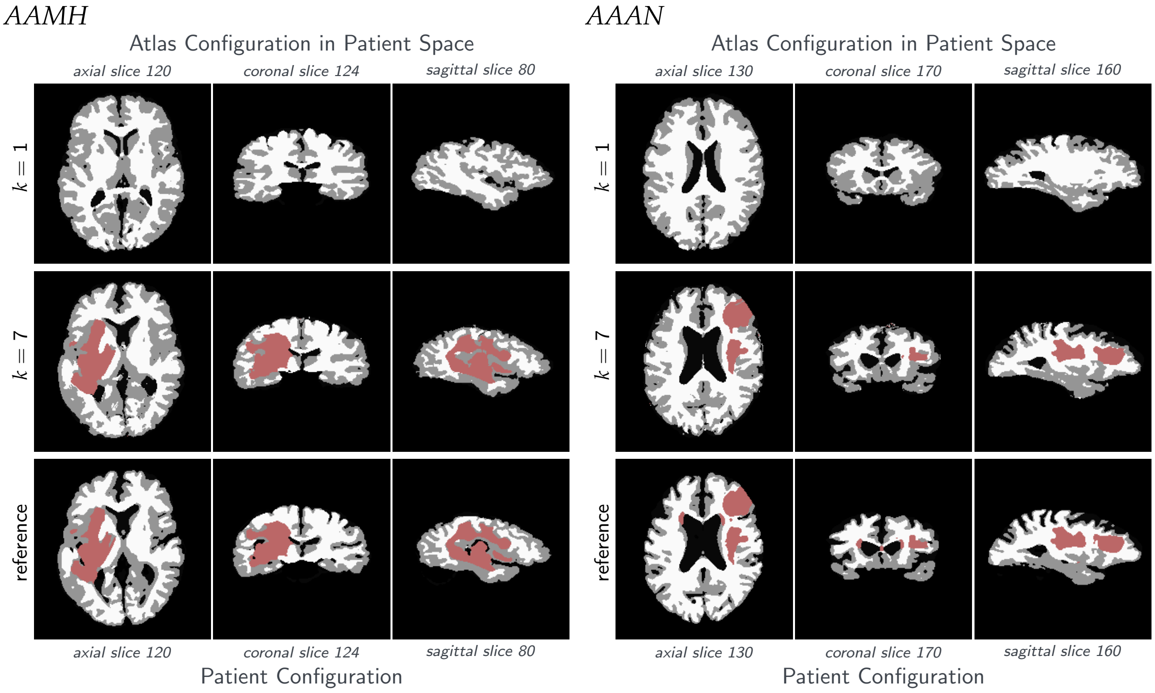


FIGURE 7.7 *Tumor and brain labels for the* ***real tumor / real velocity (RTRV)*** *test case; ground truth: (ρ N/A,* $\boldsymbol{k}$ *N/A,* $\boldsymbol{p}$ *N/A,* $\boldsymbol{v}$ *N/A); patients* ***AAMH, AAAN****. We set the parameters for the tumor solver to* $\rho_f = 15$, $k_f = 0$ *(reaction-only). We use* $n_p = 343$ *Gaussians for the inversion. The top row shows the original atlas image. The bottom row shows the patient image. The row in the middle shows the solution for our coupled scheme.*

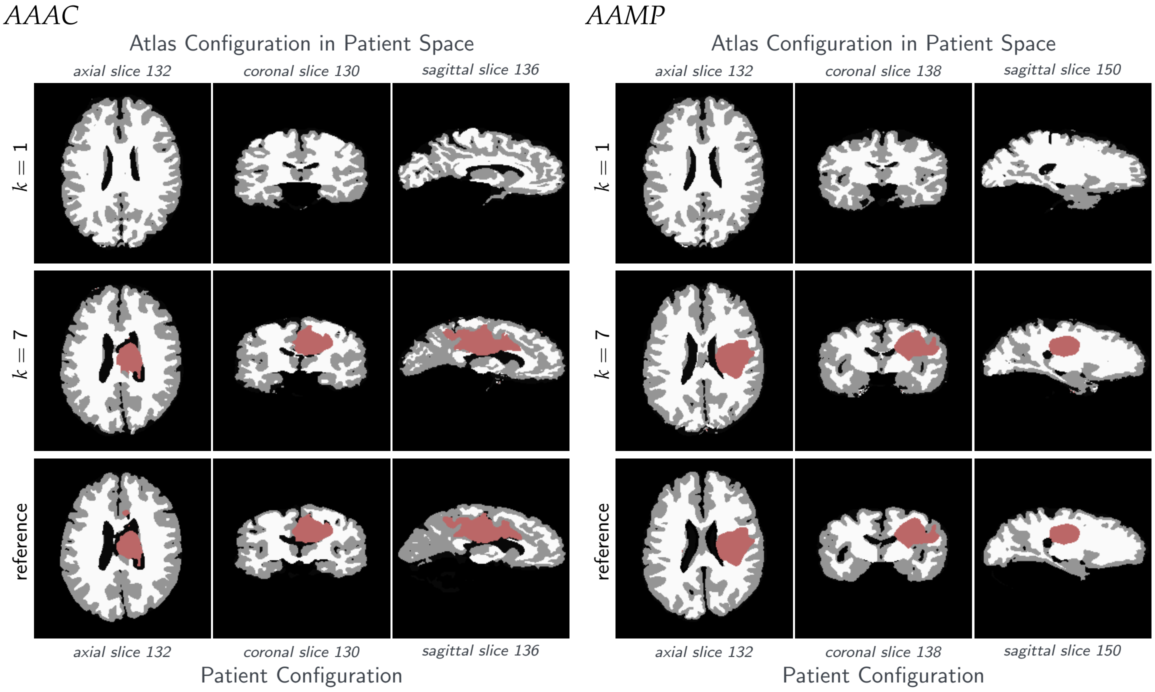


FIGURE 7.8 *Tumor and brain geometry for the* ***real tumor / real velocity (RTRV)*** *test case; ground truth: (ρ N/A,* $\boldsymbol{k}$ *N/A,* $\boldsymbol{p}$ *N/A,* $\boldsymbol{v}$ *N/A); patients* ***AAAC, AAMP****. We set the parameters for the tumor solver to* $\rho_f = 15$, $k_f = 0$ *(reaction-only). We use* $n_p = 343$ *Gaussians for the inversion. The top row shows the original atlas image. The bottom row shows the patient image. The row in the middle shows the solution for our coupled scheme.*

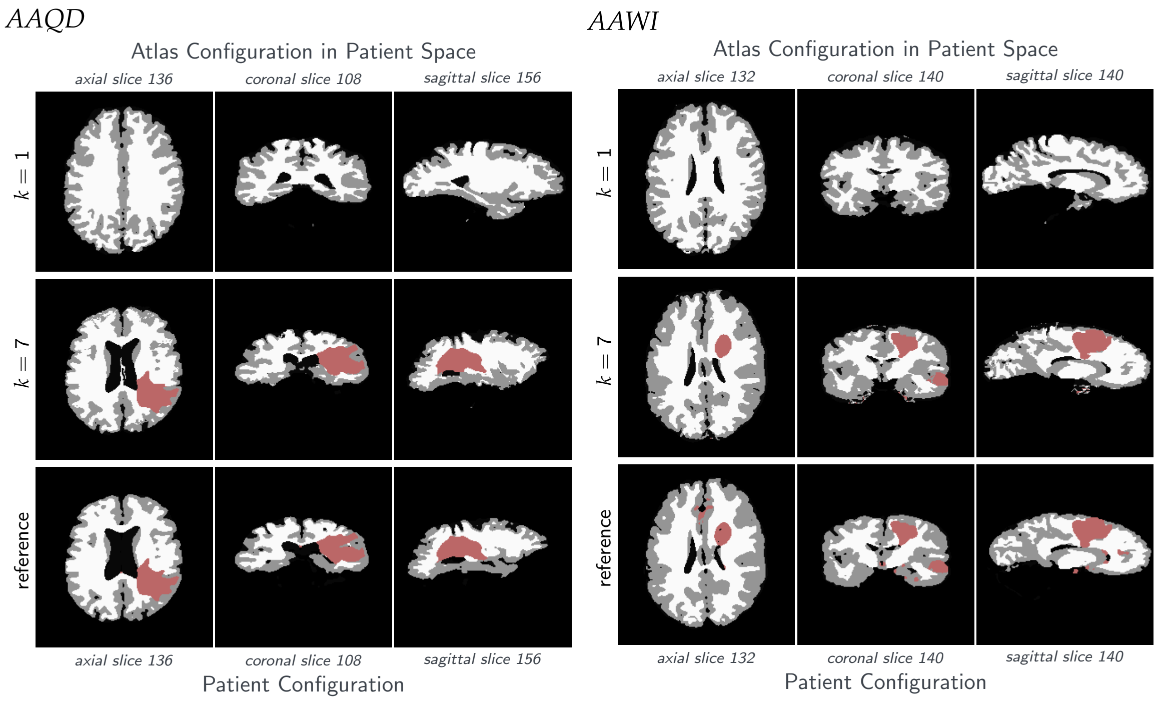


FIGURE 7.9 *Tumor and brain geometry for the **real tumor / real velocity (RTRV)** test case; ground truth: (ρ N/A, $\mathbf{k}$ N/A, $\mathbf{p}$ N/A, $\mathbf{v}$ N/A); patients **AAQD**, **AAWI**. We set the parameters for the tumor solver to $\rho_f = 15$, $k_f = 0$ (reaction-only). We use $n_p = 343$ Gaussians for the inversion. The top row shows the original atlas image. The bottom row shows the patient image. The row in the middle shows the solution for our coupled scheme.*

TABLE 7.7 *Summary of results for the **real tumor / real velocity (RTRV)** test case, ground truth: (ρ N/A, $\mathbf{k}$ N/A, $\mathbf{p}$ N/A, $\mathbf{v}$ N/A); based on real clinical data (taken from [Goo13]). We set the tumor parameters to $\rho_f = 15$, $k_f = 0$ (reaction-only). We consider two automatic data-driven approaches (see §6.3.3) for the selection of the Gaussian basis functions, (i) the diameter-based, regular grid selection mode (bbox) with $n_p = 343$ and $\delta = 1.5\sigma$, and (ii) the tumor shape- and intensity-driven, adaptive selection mode (adptv.) with $\sigma = 2\pi/128$, $\delta = 2\sigma$ and a required tumor volume fraction of 0.1. We report the (summed) mismatch for the brain tissue probability maps (μ_{B,L^2}) and tumor probability map (μ_{T,L^2}) in patient space, the mean Dice coefficient for the hard segmentation corresponding to the brain tissue (DICE$_B$) and the tumor (DICE$_T$), respectively, as well as the relative norm of the gradient for the coupled problem (6.4) ($\|g\|_{rel}$). We also report the total run time in seconds (T^{tot}), and the run time of the individual components of our Picard scheme, respectively (also in seconds; tumor solver: T^{tu}_{inv}; image registration: T^{reg}_{inv}). Note that the latter sums up to less than the reported total run time as we do not explicitly measure time spent in additional coupling functionality and forward solvers. We execute our code in parallel on 11 nodes using 256 MPI tasks on `HazelHen`.*

Patient		Φ-mode	n_p	β_v	μ_{B,L^2}	DICE$_B$	μ_{T,L^2}	DICE$_T$	$\|g\|_{\text{rel}}$	T^{tot} [s]	T^{tu}_{inv} [s]	T^{reg}_{inv} [s]
AAMH	initial			1	1.00	4.36E−1	1.00	0.00	1.00	–	–	–
	final	bbox	343	1E−4	3.45E−1	7.92E−1	1.95E−1	9.57E−1	3.92E−2	6.28E+2	1.95E+2	4.35E+2
	final	adptv.	2407	1E−4	3.37E−1	7.94E−1	8.65E−2	9.78E−1	2.16E−2	1.38E+3	9.71E+2	3.34E+2
AAAN	initial			1	1.00	4.37E−1	1.00	0.00	1.00	–	–	–
	final	bbox	343	1E−4	3.54E−1	8.35E−1	3.77E−1	8.91E−1	1.06E−1	6.34E+2	2.15E+2	4.17E+2
	final	adptv.	1412	1E−4	3.43E−1	8.37E−1	1.04E−1	9.72E−1	2.86E−2	1.01E+3	6.04E+2	3.39E+2
AAAC	initial			1	1.00	4.82E−1	1.00	0.00	1.00	–	–	–
	final	bbox	343	1E−4	3.36E−1	8.44E−1	2.45E−1	9.55E−1	4.38E−2	4.92E+2	1.59E+2	3.32E+2
	final	adptv.	1297	1E−4	3.25E−1	8.45E−1	9.49E−2	9.74E−1	2.55E−2	1.48E+3	9.12E+2	3.39E+2
AAMP	initial			1	1.00	4.74E−1	1.00	0.00	1.00	–	–	–
	final	bbox	343	1E−4	3.32E−1	8.05E−1	1.34E−1	9.75E−1	3.31E−2	6.48E+2	2.36E+2	4.13E+2
	final	adptv.	1000	1E−4	3.28E−1	8.07E−1	7.87E−2	9.85E−1	1.74E−2	6.70E+2	2.99E+2	3.40E+2
AAQD	initial			1.00	1.00	2.92E−1	1.00	0.00	1.00	–	–	–
	final	bbox	343	1E−4	3.24E−1	8.06E−1	2.55E−1	9.31E−1	9.88E−2	1.10E+3	2.19E+2	8.81E+2
	final	adptv.	1448	1E−4	3.10E−1	8.09E−1	1.11E−1	9.72E−1	2.48E−2	1.61E+3	8.84E+2	5.30E+2
AAWI	initial			1	1.00	4.53E−1	1.00	0.00	1.00	–	–	–
	final	bbox	343	1E−4	3.45E−1	7.87E−1	3.92E−1	8.74E−1	5.26E−2	6.84E+2	2.92E+2	3.90E+2
	final	adptv.	2609	1E−4	3.21E−1	7.93E−1	1.12E−1	9.72E−1	2.54E−2	1.95E+3	1.54E+3	3.38E+2

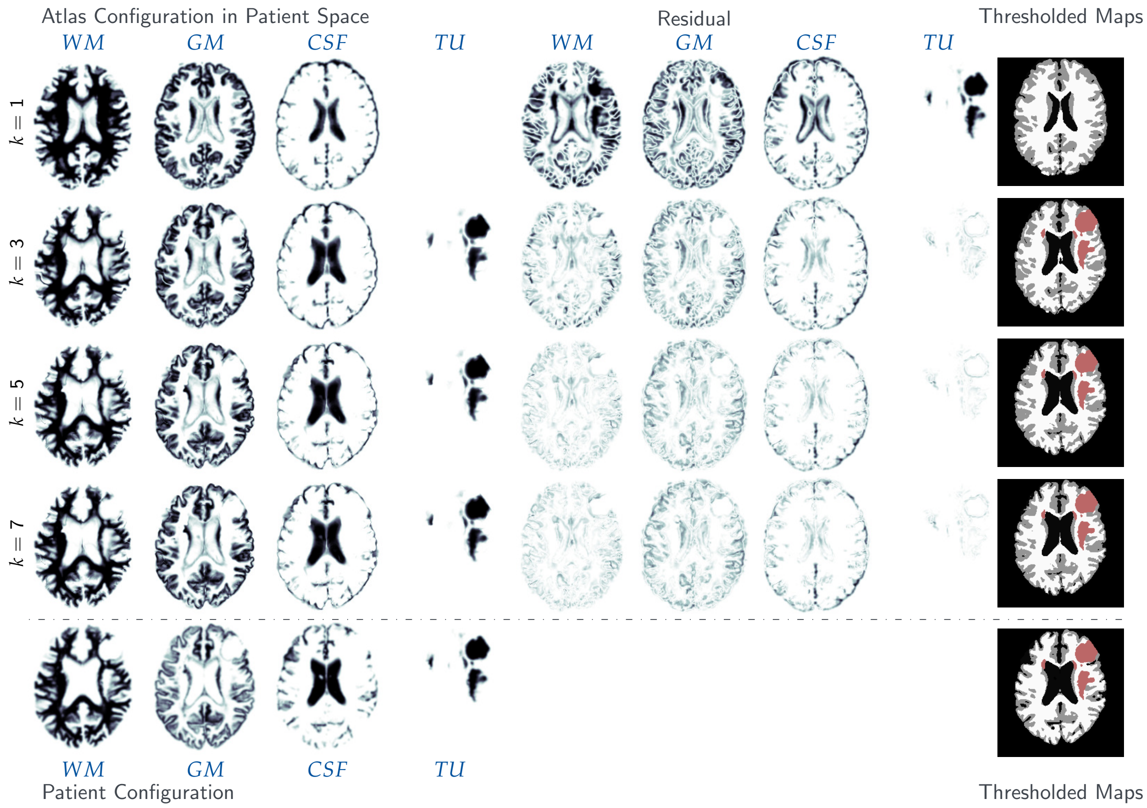
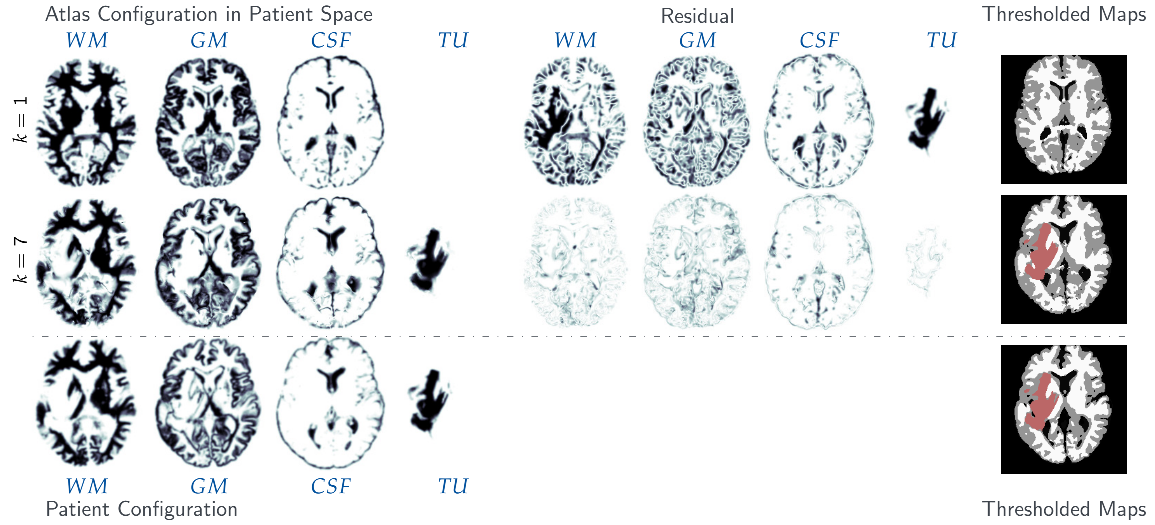


FIGURE 7.10 *Qualitative results for the* ***real tumor / real velocity (RTRV)*** *test case, ground truth: (ρ N/A,* ***k*** *N/A,* ***p*** *N/A,* ***v*** *N/A);* ***AAAN*** *patient. The figure shows probability maps for the labels of the healthy atlas brain ($k = 1$; top row) and the AAAN patient (target) brain probability maps with tumor (bottom row) along with the reconstructed probability maps throughout the Picard iterations ($k = 1, 3, 5, 7$) (axial-slice* 132*).*

TABLE 7.8 *Results for the* ***AAAN*** *and the* ***AAMH*** *patient* ***real tumor / real velocity (RTRV)*** *test case, ground truth: (ρ N/A,* ***k*** *N/A,* ***p*** *N/A,* ***v*** *N/A) using real clinical input data. We investigate the reconstruction quality with varying model parameters $\rho_f \in \{0, 5, 10, 15\}$ and $k_f \in \{0, 1\text{E}{-}2\}$. We consider the tumor diameter-based Gaussian selection mode with $n_{\boldsymbol{p}} = 343$ basis functions and a spacing of $\delta = 1.5\sigma$. The table shows the (summed) mismatch for the brain tissue probability maps (μ_{B,L^2}) and tumor probability map (μ_{T,L^2}) in patient space, the mean Dice coefficient for the hard segmentation corresponding to the brain tissue (DICE$_B$) and the tumor (DICE$_T$), respectively, for different values of the reaction scaling parameter ρ_f and of the diffusion coefficient scaling parameter k_f, as well as the relative norm of the gradient for the coupled problem (6.4) ($\|g\|_{rel}$). We also report the total run time in seconds (T^{tot}) and the run time of the individual components of our Picard scheme, respectively (also in seconds; tumor solver: T^{tu}_{inv}; image registration: T^{reg}_{inv}). Note that the latter sum up to less than the reported total run time as we do not explicitly measure time spent in additional coupling functionality and forward solvers. We execute our code in parallel on on* 11 *nodes using* 256 *MPI tasks on* `HazelHen`*.*

	ρ_f	k_f	n_p	μ_{B,L^2}	DICE$_B$	μ_{T,L^2}	DICE$_T$	$\|\boldsymbol{g}\|_{\text{rel}}$	T^{tot} [s]	T^{tu}_{inv} [s]	T^{reg}_{inv} [s]
ID AAAN	*initial config.*			1.00	4.37E−1	1.00	0.00	1.00	–	–	–
	0	0	343	3.42E−1	8.39E−1	2.56E−1	9.36E−1	2.19E−1	6.58E+2	2.02E+2	4.23E+2
	15	0	343	3.54E−1	8.35E−1	3.77E−1	8.91E−1	1.06E−1	6.34E+2	2.15E+2	4.17E+2
	5	1E−2	343	3.50E−1	8.35E−1	3.17E−1	9.06E−1	1.79E−1	1.82E+4	1.77E+4	3.33E+2
	10	1E−2	343	3.52E−1	8.35E−1	3.60E−1	8.85E−1	1.41E−1	1.88E+4	1.82E+4	3.41E+2
	15	1E−2	343	3.55E−1	8.34E−1	3.98E−1	8.62E−1	1.13E−1	1.85E+4	1.80E+4	3.28E+2
ID AAMH	*initial config.*			1.00	4.36E−1	1.00	0.00	1.00	–	–	–
	0	0	343	3.40E−1	7.92E−1	1.47E−1	9.69E−1	1.19E−1	6.09E+2	2.13E+2	3.63E+2
	15	0	343	3.45E−1	7.92E−1	1.95E−1	9.57E−1	3.92E−2	6.28E+2	1.95E+2	4.35E+2
	5	1E−2	343	3.44E−1	7.92E−1	1.88E−1	9.64E−1	7.68E−2	1.23E+4	1.18E+4	3.41E+2
	10	1E−2	343	3.46E−1	7.92E−1	2.07E−1	9.62E−1	5.16E−2	1.33E+4	1.28E+4	3.44E+2
	15	1E−2	343	3.48E−1	7.93E−1	2.35E−1	9.59E−1	3.47E−2	1.18E+4	1.13E+4	3.63E+2



iterations	β_v	μ_{B,L^2}	DICE$_B$	μ_{T,L^2}	DICE$_T$	$\|g\|_{\text{rel}}$	T^{it} [s]	T^{tu}_{inv} [%]	T^{reg}_{inv} [%]
initial	–	1.00	4.36E−1	1.00	0.00	1.00	–	–	–
1	1	9.59E−1	4.62E−1	1.00	0.00	9.99E−1	9.80E+1	74.6	15.0
2	1E−1	7.42E−1	5.60E−1	3.90E−1	8.43E−1	9.06E−2	4.55E+1	51.2	47.7
3	1E−2	5.05E−1	7.01E−1	3.13E−1	9.03E−1	4.61E−2	9.59E+1	25.8	73.7
4	1E−3	3.91E−1	7.66E−1	2.52E−1	9.39E−1	3.69E−2	1.18E+2	21.5	78.0
5	1E−4	3.60E−1	7.84E−1	2.20E−1	9.51E−1	3.62E−2	1.15E+2	21.2	78.3
6	1E−4	3.51E−1	7.89E−1	2.05E−1	9.55E−1	3.84E−2	8.99E+1	27.0	72.4
7	1E−4	3.45E−1	7.92E−1	1.95E−1	9.57E−1	3.92E−2	6.63E+1	0.0	99.3
final	1E−4	3.45E−1	7.92E−1	1.95E−1	9.57E−1	3.92E−2	6.28E+2	1.95E+2	4.20E+2

FIGURE 7.11 *Results for the **real tumor / real velocity (RTRV)** test case, ground truth: (ρ N/A, **k** N/A, **p** N/A, **v** N/A); **AAMH** patient. We consider the tumor diameter-based Gaussian selection mode with $n_{\mathbf{p}} = 343$ basis functions and a spacing of $\delta = 1.5\sigma$. The figure shows probability maps for the labels of the healthy atlas brain ($k = 1$; top row) and the AAMH patient (target) brain probability maps with tumor (bottom row) along with the reconstructed probability maps, i.e., the final result of our inversion algorithm ($k = 7$; middle row) (axial-slice 120). In the table, we report the (summed) mismatch for the brain tissue probability maps (μ_{B,L^2}) and tumor probability map (μ_{T,L^2}) in patient space, the mean Dice coefficient for the hard segmentation corresponding to the brain tissue (DICE$_B$) and the tumor DICE$_T$, respectively, as well as the relative norm of the gradient for the coupled problem (6.4) ($\|g\|_{rel}$). We also report the run time per iteration in seconds (T^{it}), and the percentages (T^{tu}_{inv}) and (T^{reg}_{inv}) of this runtime spent in the tumor solver and the image registration solver, respectively. Note that the latter sums up to less than 100% as we do not explicitly measure time spent in additional coupling functionality and in the forward solvers. The last row shows the final state and summed absolute timings for the respective solvers in seconds.*

we observe excellent Dice scores between 9.72E−1 and 9.85E−1 for the 6 considered data sets. The runtimes are comparable to our former experiment. We again achieve a reduction of the relative norm of the gradient for the coupled problem in (6.4) of about two orders of magnitude (slightly less than what we saw before).

The results of our study with varying tumor model parameters presented in Tab. 7.8 show that there are slight variances in the results depending on the parameter choices, but the Picard iteration scheme is successful in all cases. Furthermore, from the experiments studying the different selection modes for the Gaussian basis functions in Tab. 7.9, where the number and positioning of the basis functions is varied, we see, that the tumor shape- and intensity-based selection mode with more and sharper Gaussians yields better results in terms of L_2-data-misfit and Dice coefficients of the tumor reconstruction. The reaction-only runs show a slightly better reconstruction than the reaction-diffusion experiments. Both observations can be explained by the impact of the interpolation

in the normal-to-abnormal registration quality. For more biophysically meaningful results and patient specific tumor simulations, we focus on the moving-atlas scheme, studied in §7.3

TABLE 7.9 *Results for the **AAAN** and the **AAMH** patient **real tumor / real velocity (RTRV)** test case, ground truth: (ρ N/A, $\mathbf{k}$ N/A, $\mathbf{p}$ N/A, $\mathbf{v}$ N/A); using real clinical input data. We study the registration and reconstruction quality of our joint approach, depending on the fidelity of the parametrization, i.e., the number and placing of the Gaussian basis functions for reaction-only (REAC) and the reaction-diffusion (DIF) tumor model. In particular, we consider two automatic data-driven approaches (see §6.3.3): (i) the diameter-based, regular grid selection mode (bbox) with $n_p = 343$ and $\delta = 1.5\sigma$, and (ii) the tumor shape- and intensity-driven, adaptive selection mode (adptv.) with $\sigma = 2\pi/128$, $\delta = 2\sigma$ and a required tumor volume fraction τ_V. The table shows the (summed) mismatch for the brain tissue probability maps (μ_{B,L^2}) and tumor probability map (μ_{T,L^2}) in patient space, the mean Dice coefficient for the hard segmentation corresponding to the brain tissue (DICE$_B$) and the tumor (DICE$_T$), respectively, for different values of the reaction scaling parameter ρ_f and of the diffusion coefficient scaling parameter k_f, as well as the relative norm of the gradient for the coupled problem (6.4) ($\|g\|_{rel}$). We also report the total run time in seconds (T^{tot}) and the run time of the individual components of our Picard scheme (also in seconds; tumor solver: T^{tu}_{inv}; image registration: T^{reg}_{inv}). Note that the latter sums up to less than the reported total run time as we do not explicitly measure time spent in additional coupling functionality and forward solvers. We execute our code in parallel on on 11 nodes using 256 MPI tasks on `HazelHen`.*

		k_f	Φ-mode	σ	τ_V	n_p	μ_{B,L^2}	DICE$_B$	μ_{T,L^2}	DICE$_T$	$\|g\|_{\text{rel}}$	T^{tot} [s]	T^{tu}_{inv} [s]	T^{reg}_{inv} [s]
ID AAAN		*initial config.*					1.00	4.36E−1	1.00	0.00	1.00	–	–	–
	REAC	0	bbox	$2\pi/16$	–	125	3.67E−1	8.30E−1	5.71E−1	3.44E−1	1.28E−1	4.67E+2	8.38E+1	3.57E+2
		0	bbox	$2\pi/24$	–	343	3.54E−1	8.35E−1	3.77E−1	8.91E−1	1.06E−1	6.34E+2	2.15E+2	4.17E+2
		0	adptv.	$2\pi/32$	0.0	321	3.44E−1	8.37E−1	1.55E−1	9.53E−1	3.69E−2	5.56E+2	1.91E+2	3.53E+2
		0	adptv.	$2\pi/64$	0.1	485	3.44E−1	8.37E−1	1.31E−1	9.64E−1	2.96E−2	6.52E+2	2.96E+2	3.39E+2
		0	adptv.	$2\pi/128$	0.1	1412	3.43E−1	8.37E−1	1.04E−1	9.72E−1	2.86E−2	1.01E+3	6.04E+2	3.39E+2
		0	adptv.	$2\pi/256$	0.9	6267	3.43E−1	8.37E−1	9.99E−2	9.79E−1	2.28E−2	1.35E+3	8.28E+2	3.38E+2
	DIF	1E−2	bbox	$2\pi/24$	–	343	3.55E−1	8.34E−1	3.98E−1	8.62E−1	1.13E−1	1.85E+4	1.80E+4	3.28E+2
		1E−2	adptv.	$2\pi/32$	0.0	321	3.44E−1	8.37E−1	1.83E−1	9.41E−1	3.39E−2	1.16E+4	1.11E+4	3.52E+2
		1E−2	adptv.	$2\pi/64$	0.1	485	3.44E−1	8.37E−1	1.61E−1	9.47E−1	2.60E−2	1.17E+4	1.12E+4	3.58E+2
		1E−2	adptv.	$2\pi/128$	0.1	1412	3.42E−1	8.38E−1	1.45E−1	9.59E−1	2.33E−2	1.10E+4	1.05E+4	3.56E+2
		1E−2	adptv.	$2\pi/256$	0.9	6267	3.43E−1	8.37E−1	1.36E−1	9.65E−1	1.96E−2	1.02E+4	9.54E+3	3.39E+2
ID AAMH		*initial config.*					1.00	4.37E−1	1.00	0.00	1.00	–	–	–
	REAC	0	bbox	$2\pi/16$	–	125	3.50E−1	7.91E−1	2.36E−1	9.53E−1	4.83E−2	4.47E+2	7.20E+1	3.48E+2
		0	bbox	$2\pi/24$	–	343	3.45E−1	7.92E−1	1.95E−1	9.57E−1	3.92E−2	6.28E+2	1.95E+2	4.35E+2
		0	adptv.	$2\pi/32$	0.0	416	3.37E−1	7.94E−1	1.09E−1	9.74E−1	2.10E−2	5.57E+2	2.00E+2	3.41E+2
		0	adptv.	$2\pi/64$	0.1	765	3.36E−1	7.94E−1	9.86E−2	9.76E−1	1.98E−2	7.55E+2	3.86E+2	3.44E+2
		0	adptv.	$2\pi/128$	0.1	2407	3.37E−1	7.94E−1	8.65E−2	9.78E−1	2.16E−2	1.38E+3	9.71E+2	3.34E+2
	DIF	1E−2	bbox	$2\pi/24$	–	343	3.48E−1	7.93E−1	2.35E−1	9.59E−1	3.47E−2	1.18E+4	1.13E+4	3.63E+2
		1E−2	adptv.	$2\pi/32$	0.0	416	3.41E−1	7.94E−1	1.50E−1	9.76E−1	1.61E−2	1.38E+4	1.31E+4	4.02E+2
		1E−2	adptv.	$2\pi/64$	0.1	765	3.41E−1	7.93E−1	1.44E−1	9.76E−1	1.54E−2	1.62E+4	1.55E+4	4.23E+2
		1E−2	adptv.	$2\pi/128$	0.1	2407	3.41E−1	7.94E−1	1.37E−1	9.78E−1	1.65E−2	1.25E+4	1.18E+4	4.38E+2

error when trying to represent the observed target tumor by the initial condition as a superposition of Gaussian basis functions. On the one hand, the interpolation error is drastically reduced if more and sharper basis functions are used; on the other hand, switching off diffusion (and driving ρ to zero) in some sense resembles direct interpolation of the target tumor. If a sparsity constraint for the tumor inversion is employed, this is no longer valid as rich initial conditions (such as interpolations of the target tumor) are ruled out. We consider this for the moving-atlas scheme in §7.3 where we put emphasis on the biophysical model inversion. With a view of normal-to-abnormal registration as a target application, however, interpolation is a valid, cheap and efficient alternative. We note that the registration quality for the brain tissue labels WM, GM and CSF is only slightly influenced by the reconstruction quality of the tumor; for all settings in Tab. 7.9, we arrive at very similar Dice scores between 8.30E−1 and 8.38E−1 for AAAN and 7.91E−1 and 7.93E−1 for AAMH for the brain tissue labels, while the tumor reconstruction quality varies greatly (from Dice scores of 3.44E−1 to 9.79E−1 for AAAN and Dice scores from 9.53E−1 to 9.78E−1 for AAMH). As expected, in the reaction-only case, the runtime for the tumor inversion solver increases with increasing n_p (and increasing dimensionality of the Hessian system); for the reaction-diffusion run, the solution of the diffusion operator still dominates the cost and the larger system has no negative effect on the

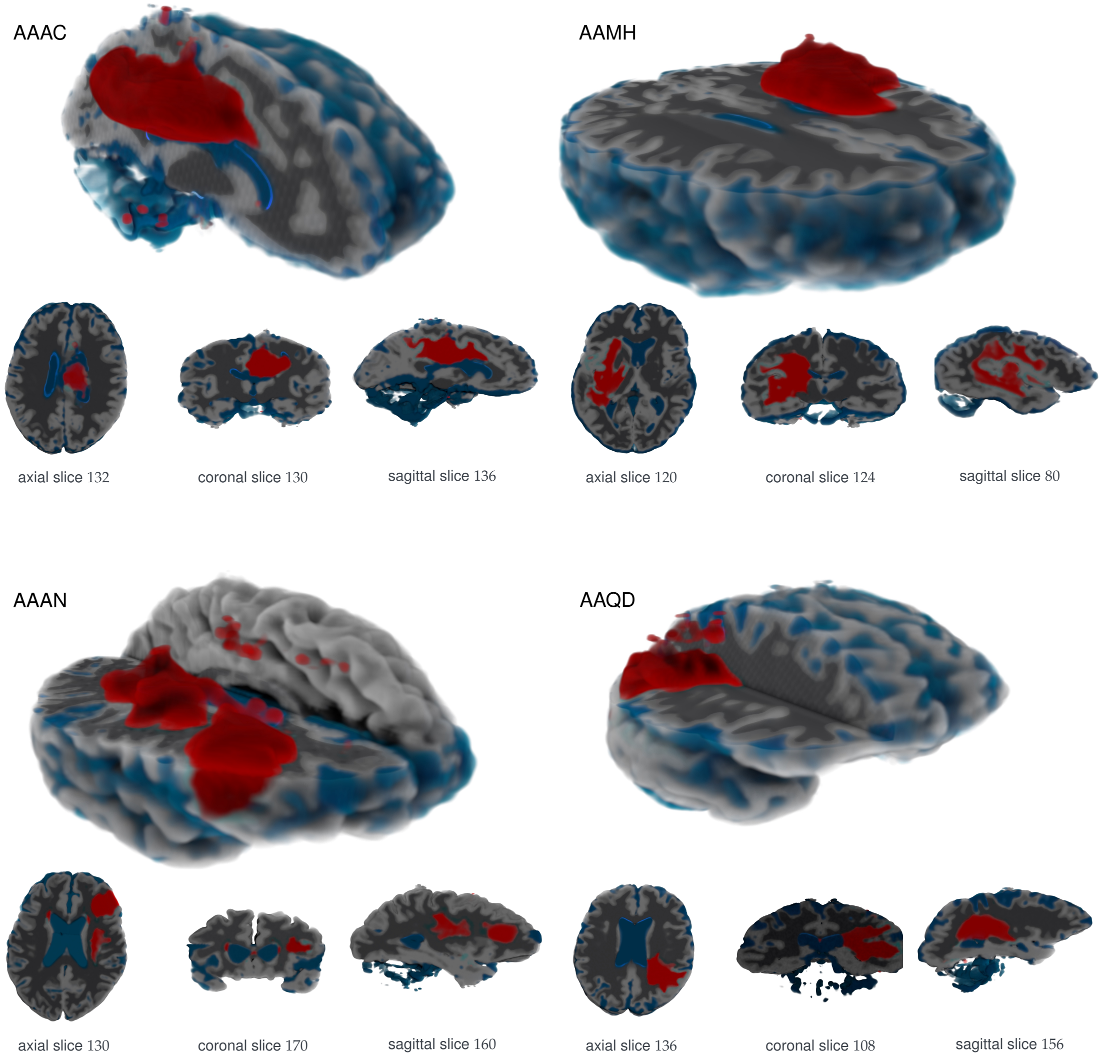


FIGURE 7.12 *Illustration of patient data. The figure shows two-dimensional slices and three-dimensional volume rendering of the segmented GLISTR patient data for ID's AAAC, AAAN, AAMH and AAQD.*

runtime[13].

Conclusion *We have tested the moving-patient formulation on real data that pose significant challenges due to large inter-subject anatomical variability and a strong variation in the appearance of the tumor, in shape, size, location and growth behavior. For most cases, we use a very simple model that only accounts for logistic growth. This, in combination with a flexible, high-dimensional parametrization of the initial condition allows us to overcome these challenges. Furthermore, the reconstruction results for the AAAN patient data show that we can reconstruct multifocal tumors with comparable quality and computational costs.*

We achieve extremely promising normal-to-abnormal registration accuracies with a Dice score of up to 8.44E$-$1 *and* 9.85E$-$1 *for the label maps associated with the probability maps of the brain anatomy and the tumor in what is an extremely challenging problem. Increasing the number of Gaussian basis functions and choosing their location adaptively, depending on shape and intensity of the tumor, has the potential to significantly reduce the final* L_2*-data-misfit and to increase the Dice coefficient for the tumor reconstruction.*

[13] The number of inner Krylov iterations is limited to 30.

The quality of the tumor model, however, has only a moderate to small influence on the normal-to-abnormal registration quality of the surrounding tissue labels; the sole existence of a tumor reconstruction module which is coupled to the registration is, however, inevitable (cf. §7.2.1).

For a large number of Gaussians with relatively small standard deviation, the interpolation quality increases and we arrive at rich initial conditions that are far from being biophysically meaningful. A sparsity constraint for the inversion remedies this problem; this is discussed in §7.3.

Conclusion: Normal-to-Abnormal Registration

We have presented a new method for the registration of images of patients diagnosed with mono- or multi-focal brain tumors to a common reference atlas. We simultaneously invert for a parametrization of the initial condition for the tumor model and a smooth velocity field to capture the inter-subject variability of brain anatomy. Application scenarios are biophysical model calibration and normal-to-abnormal image registration. Within this section, we focused on the latter and analyzed our moving-patient Picard iteration solution scheme, described in §6.1 and §6.3 for synthetic cases and actual clinical imaging data of real tumors. Here is what we have learned from our experiments on synthetic and real data:

(i) Despite the fact that our Picard iteration scheme neglects coupling terms that appear in the moving-patient formulation of the coupled optimization problem, we could experimentally show that it efficiently reduces the coupled gradient. A convergence proof of the Picard scheme to a local minimum is beyond the scope of this thesis and remains the subject of future work.

(ii) We could demonstrate that our parametrization of the initial condition (for a sufficient number and density of Gaussian basis functions) allows us to generate high-fidelity normal-to-abnormal registrations irrespective of the complexity of the data or the model used for the tumor simulations.

(iii) Overall, our numerical study, which includes real brain images with real tumors, shows that we can achieve high-fidelity results for the normal-to-abnormal registration application scenario with an overall low mismatch and high Dice scores (average Dice over all tissue labels), ranging from 7.94E−1 *to* 8.45E−1 *for real clinical cases. In particular, plain-vanilla registration without biophysical augmentation fails in the vicinity of and the area occupied by the pathology for the presented normal-to-abnormal registration problems. For the simulated and observed tumor, we achieve extremely high similarity of around 98% overlap for real complex-shaped tumors if sufficiently many Gaussian basis functions are used.*

(iv) In studies with various models from a pure interpolation with the basis functions that parametrize the initial condition (zero reaction and diffusion coefficient) over a reaction-only, to a full reaction-diffusion model, we could show in our synthetic cases, that we get the highest accuracy in tumor reconstruction, if we use the correct model. This indicates that our framework could eventually serve as a powerful tool for model selection. To obtain biophysically meaningful results from the joint inversion approach or identification of the "correct" tumor growth parameters, however, we require further enhancements such as a sparsity constraint for the tumor inversion solver (to fix the time horizon of the inversion process to the point of initial cell mutation from healthy to cancerous) and employ the improved moving-atlas formulation from §6.2 with its respective Picard solution scheme; see §6.3. This is what we investigate in the next section.

7.3 Evaluation of Biophysical Parameter Estimation Performance

In the previous section, we analyzed the quality and suitability of SIBIA (moving-patient formulation) for normal/abnormal inter-subject registration. In other words, emphasis was put on the visual similarity and overlap measures of the deformed template image and the observed data, rather then on the reconstruction of meaningful biophysical model parameters. This is what we investigate in this section. In §6.2, we presented a second formulation of the coupled problem which we denote as moving-atlas formulation. By construction, it remedies essential shortcomings of the moving-patient formulation when applied for model calibration purposes. As a proof of concept, we evaluate the moving-atlas Picard iteration scheme for synthetic tumors in real brain geometries with known ground truth, and compare the results to the moving-patient solution. Generally speaking, the tumor inversion problem has a non-unique solution. That means, different combinations of initial condition, reaction coefficient and characteristic diffusivity may predict the same final tumor state. In combining both schemes with a sparsity constraint, implemented as L_1-regularization, we intent to alleviate this problem by enforcing sparse initial conditions. We compare this feature to the previously used L_2-regularization.

The task of our numerical experiments is to assess (i) the convergence towards solutions with low mismatch both in the brain geometry $\boldsymbol{m}$ and the tumor c for both coupled formulations – the moving patient and the moving atlas; (ii) the reconstruction quality for the healthy patient brain – the latter is a direct output of the moving atlas formulation and can be generated by advecting the healthy atlas with the computed registration velocity $\boldsymbol{v}$ in the moving patient formulation[14]; (iii) the quality of inversion for the biophysical parameters, i.e., tumor growth initial conditions and diffusion coefficient.

7.3.1 Baseline: Tumor Standalone vs. Joint Inversion

In analogy to the previous section, we draw a baseline for the biophysical inversion by considering the tumor inversion solver as standalone, that means without coupling to inter-subject registration.

Purpose. Biophysical model estimation is a time dependent problem, and requires at least two snapshots in time. In practice, such data are typically not available, in particular imaging data of the healthy patient brain are in general not accessible. Our methodology artificially creates a second snapshot in time by integrating inter-subject registration. In this experiment, we address the loss in reconstruction quality if the inter-subject registration component is neglected.

Setup. We consider two instantiations of the synthetic test cases with artificial tumor and real velocity (ATRV) as described in §7.1.1. The two cases differ in the position of the tumor seed; a listing is given in Tab. 7.11. For this experiment, we consider the parameter setting #1 from Tab. 7.11 for both ATRV-C1 and ATRV-C2, respectively. The target data are generated from a forward simulation with $\rho = 15$ and $k_f = 1\text{E}{-1}$ from an initial condition with two Gaussians enabled, i.e., $p_{62} = p_{122} = 1$ (ATRV-C1), and $p_{62} = p_{122} = 2.8$ (ATRV-C2). Tumor growth is enabled only in white matter. All remaining solver settings are given in §7.1.1. For the tumor standalone solution, the inversion is carried out assuming a statistical atlas brain geometry as healthy patient brain. No inter-subject registration is applied. We invert for the tumor initial condition and the characteristic diffusivity k_f of white matter.

[14] Note that it is impossible to generate an estimate for a healthy patient from the moving patient formulation once we include the mass effect, i.e., brain tissue deformation as a result of tumor growth in our model.

TABLE 7.10 *Baseline for the single-snapshot tumor inversion problem. Comparison of the standalone tumor inversion solver and the joint inversion approaches developed in this work. We consider two different instantiations of the ATRV test case (ATRV-C1 and ATRV-C2, each with parameter setup #1 from Tab. 7.11). The moving-atlas strategy explicitly generates an approximation to the healthy patient geometry, which is typically unknown in practice. We report the L_2-mismatch and the Dice overlap measure between the predicted/reconstructed tumor and the target data. Furthermore, we report the L_2-error e_{c0,L^2} for the reconstruction of the initial condition with respect to the ground truth initial condition, and the estimated characteristic diffusivity k_f.*

		ATRV-C1				**ATRV-C2**			
Solver		μ_{T,L^2}	DICE$_T$	e_{c0,L^2}	k_f	μ_{T,L^2}	DICE$_T$	e_{c0,L^2}	k_f
Tumor (standalone)	L_2	2.01E−1	8.52E−1	1.36	1.04E−1	2.48E−1	7.45E−1	8.07E−1	8.48E−2
Moving-Atlas	L_2	6.58E−2	9.52E−1	3.17E−1	8.91E−2	7.40E−2	8.74E−1	4.44E−1	7.87E−2
Moving-Patient	L_2	1.35E−1	9.24E−1	1.15	8.95E−2	1.69E−1	8.17E−1	7.45E−1	5.27E−2
Tumor (standalone)	L_1wL_2	2.90E−1	7.92E−1	4.25E−1	1.15E−1	2.93E−1	5.54E−1	4.08E−1	1.48E−1
Moving-Atlas	L_1wL_2	6.49E−2	9.50E−1	7.13E−2	9.05E−2	6.11E−2	9.45E−1	1.19E−1	9.23E−2
Moving-Patient	L_1wL_2	1.44E−1	9.25E−1	3.25E−1	9.22E−2	1.90E−1	7.96E−1	4.70E−1	8.86E−2

Results. We report metrics for the quality of reconstruction in Tab. 7.10; For the tumor standalone solver, the moving-patient solver and the moving-atlas solver, we compare the data-mismatch μ_{T,L^2} and Dice coefficient DICE$_T$ between the predicted/reconstructed tumor and the target data as well as the L_2-error e_{c0,L^2} of the reconstructed initial condition $c^{(\cdot,0)}$ with respect to the ground truth $c^{(\cdot,0)^\star}$ and the reconstructed characteristic diffusivity k_f.

Observations. For all configurations, the tumor standalone solver results in a significantly lower reconstruction quality then our joint inversion approach. This is due to the error introduced by inversion in the "wrong healthy brain geometry". The moving-atlas solution outperforms the other strategies by far, both in terms of data-misfit and Dice coefficient, as well as in reconstructing the true initial condition. The difference in the estimation of the characteristic diffusivity in white matter k_f is not significant. For ATRV-C2, we observe a slightly better estimation of the characteristic diffusivity k_f if we use the more sophisticated moving-atlas strategy with sparsity constraint (L_1wL_2).

Conclusion *The coupling of the tumor inversion solver to inter-subject registration is inevitable to obtain meaningful results and useful parameter estimates from single-snapshot inversion. The error introduced when neglecting the registration depends on the discrepancy of the atlas and patient geometry and can be arbitrarily large.*

7.3.2 Test Case ATRV: Analytic Tumor and Real Velocity

The experiments conducted here serve as a proof of concept. We examine the moving-atlas Picard iteration solution scheme with an emphasis on model inversion and parameter estimation and address the scheme's eligibility to recover meaningful biophysical characteristics. Since, for actual clinical imaging data, the true biophysical parameters (and also the type of tumor progression model) are unknown, we solely consider synthetically grown tumors (using our reaction-diffusion tumor-growth model), but use real (healthy) brain imaging data to carry out the tumor progression simulations. We enhance the tumor inversion solver with a sparsity constraint to favor solution with sparse initial condition, i.e., a tumor initial condition which is closer (in time) to the actual tumor genesis.

Purpose. In §6.2, we developed and motivated the moving-atlas formulation to overcome significant shortcomings of the moving-patient formulation in terms of meaningful biophysical parameter

TABLE 7.11 *ATRV test case variations. Parameter choices for the* ***analytic tumor with real velocity and diffusion (ATRV)*** *test case; ground truth: ($\rho_f = 15$, $\rho_w = 1$, $\rho_g = 0$, $k_f = 1.00\mathrm{E}{-1}$, $k_w = 1$, $k_g = 0$, $\boldsymbol{p} = \boldsymbol{p}^\star$ (in patient domain), $\boldsymbol{v}$ N/A). For the initial condition parametrization, we use a regular grid of $n_{\boldsymbol{p}} = 125$ Gaussian basis functions with standard deviation σ as outlined below, and a spacing of $\delta = 1.5\,\sigma$. The grid is centered around the positions $\boldsymbol{x}_{C_1}$ and $\boldsymbol{x}_{C_2}$, respectively. For parameter setting #1 for ATRV-C1, we set $p_{62} = p_{122} = 1$, and $p_{62} = p_{122} = 2.8$ for all other parameter settings (cf. Fig. 7.13 and Fig. 7.15 for the location of tumor seed). The parameter setups differ in the time horizon T for the tumor evolution.*

	ATRV-C1 $\boldsymbol{x}_{C_1} = {}^{2\pi}\!/_{128}(39, 63, 64)$		**ATRV-C2** $\boldsymbol{x}_{C_2} = {}^{2\pi}\!/_{128}(61, 89, 64)$	
setting	*initial condition*	*tumor growth*	*initial condition*	*tumor growth*
#1	$\sigma = {}^{2\pi}\!/_{30}$	$T = 0.16$, $\rho_f = 15$, $k_f = 0.1$	$\sigma = {}^{2\pi}\!/_{64}$	$T = 0.16$, $\rho = 15$, $k = 0.1$
#2	$\sigma = {}^{2\pi}\!/_{64}$	$T = 0.32$, $\rho_f = 15$, $k_f = 0.1$	$\sigma = {}^{2\pi}\!/_{64}$	$T = 0.32$, $\rho = 15$, $k = 0.1$
#3	$\sigma = {}^{2\pi}\!/_{64}$	$T = 0.44$, $\rho_f = 15$, $k_f = 0.1$		

estimation. Following theoretical arguments, the moving-atlas solution strategy is expected to result in an improved approximation of the (initially unknown) healthy patient brain anatomy, and, thus, allows for more meaningful reconstruction of biophysical tumor-growth characteristics. For synthetic test cases with known biophysical model parameters, we examine and compare the solution of the advanced moving-atlas strategy to the solution obtained from the moving-patient scheme. Furthermore, we study the effect of a sparsity constraint added to both schemes to favor solutions closer to the actual tumor genesis. We are primarily interested in the quality of the healthy patient approximation (measured by the L_2-mismatch μ_{B_0,L^2} and Dice score DICE_{B_0}), the quality of the reconstruction of the shape and sparsity of the ground truth tumor initial condition (measured by the L_2-error e_{c0,L^2} and visual inspection of simulation data), and the quality of the resulting prediction of the grown tumor (measured by the L_2-mismatch μ_{T,L^2} and Dice score DICE_T). We assess convergence by monitoring the norm of the reduced gradient of the coupled formulation and report runtimes to address the efficiency of the scheme.

Setup. A general description of this synthetic tumor / real velocity (ATRV) test case is given in §7.1.1; we complete the description with some specific information. Note that only the tumor parameters are known for this example, whereas the registration velocity between healthy atlas and patient brain is unknown. We consider two different tumor locations ATRV-C1 and ATRV-C2 and varying time horizons for the tumor progression simulation. An overview is given in Tab. 7.11. For ATRV-C1 we also employ different fidelities of the parametrization of the tumor initial condition: For the parameter setting #1, we use a set of Gaussian basis functions with standard deviation $\sigma = {}^{2\pi}\!/_{30}$ and $p_{62} = p_{122} = 1$, whereas, for settings #2 and #3, we use smaller Gaussians with $\sigma = {}^{2\pi}\!/_{64}$ and $p_{62} = p_{122} = 2.8$. The target data are generated from a tumor forward simulation in the patient domain with cell proliferation rate $\rho_f = 15$ and characteristic diffusivity of $k_f = 1\mathrm{E}{-1}$ in white matter. All remaining solver settings are given in §7.1.3 and Tab. 7.2.

We conduct experiments for the moving-atlas and the moving-patient Picard iteration scheme. Both solution schemes are enhanced with a sparsity constraint for the tumor inversion. To this end, we first solve the tumor inversion problem employing a L_1-regularization to define the sparsity pattern of the initial condition, followed by an inversion with weighted L_2-regularization (short L_1wL_2). We compare these results to the L_2-regularized joint inversion. The inversion is done either for $\boldsymbol{p}$ and $\boldsymbol{v}$ only, or for $\boldsymbol{p}$, $\boldsymbol{v}$, and the diffusion parameter k_f.

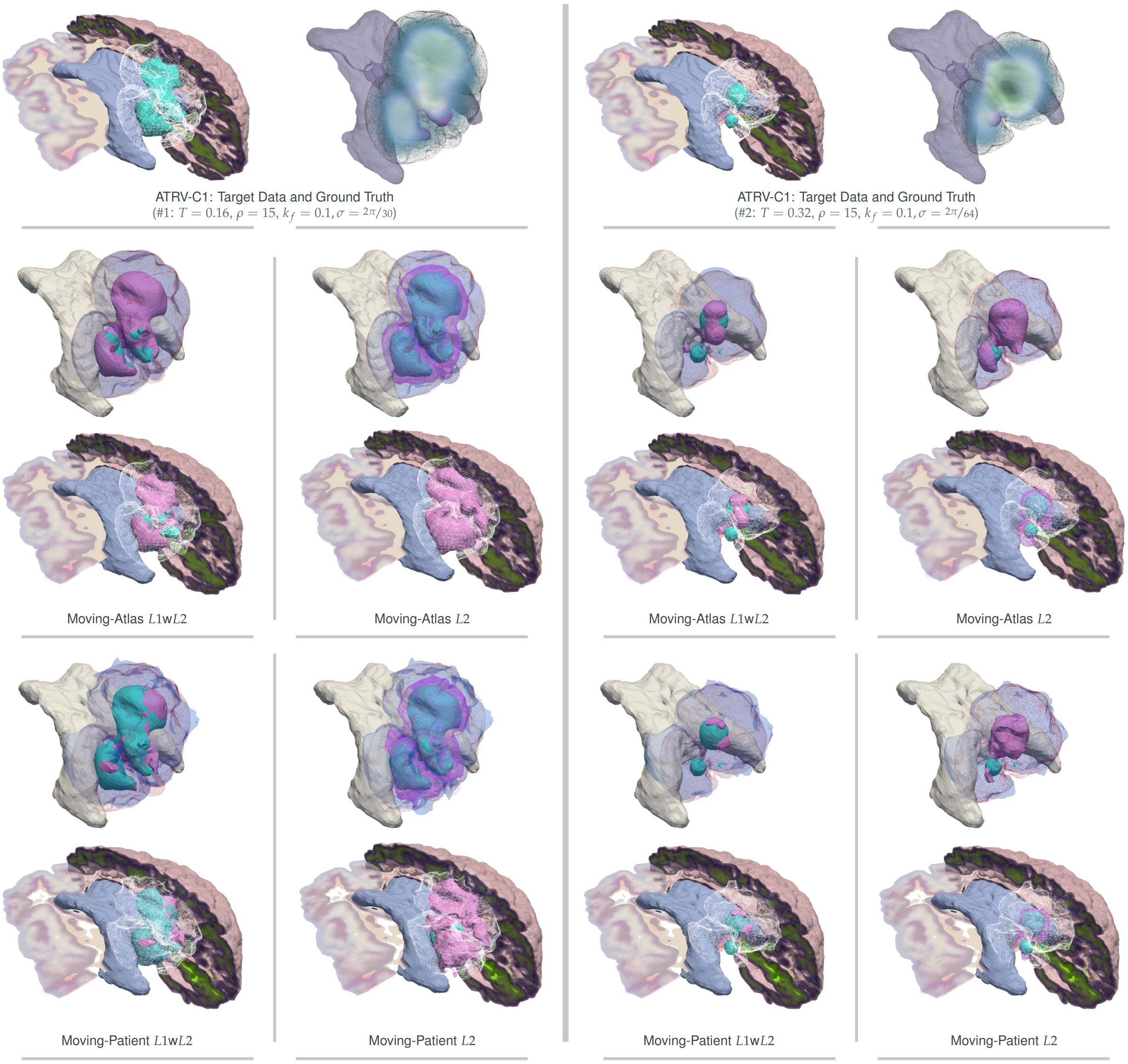
Results. For ATRV-C1, we report the L_2-mismatch and Dice overlay coefficients for the reconstruction of the (pathologic) brain anatomy (μ_{B,L^2} and DICE_B), the healthy patient anatomy (μ_{B_0,L^2} and DICE_{B_0}), and the reconstructed (grown) tumor (μ_{T,L^2} and DICE_T) for every Picard iteration of the moving-atlas solution strategy in Tab. 7.12. Furthermore, we monitor the L_2-error e_{c0,L^2} for the

TABLE 7.12 *Quantitative results for the **analytic tumor with real velocity and diffusion (ATRV-C1 #1)** test case with parameter setting #1 from Tab. 7.11; ground truth: ($\rho_f = 15$, $\rho_w = 1$, $\rho_g = 0$, $k_f = 1.00\text{E}{-1}$, $k_w = 1$, $k_g = 0$, $\boldsymbol{p} = \boldsymbol{p}^\star$ (in patient domain), $\boldsymbol{v}$ N/A). For the inversion, we employ the **moving-atlas** Picard iteration scheme with L_1wL_2 sparsity constraint and use the correct values $\rho_f = 15$ and $k_f = 1\text{E}{-1}$ for the characteristic proliferation rate and diffusivity in white matter. We report the average mismatch for the probability maps for the pathologic brain tissue labels μ_{B,L^2}, the healthy brain tissue labels μ_{B_0,L^2} and the tumor μ_{T,L^2}, the mean Dice coefficient for brain tissue DICE$_B$, healthy patient brain tissue DICE$_{B_0}$ and tumor DICE$_T$. Furthermore, we measure the L_2-error e_{c0,L^2} of the reconstructed tumor initial condition to the ground truth, and the relative norm $\|\boldsymbol{g}\|_{rel}$ of the reduced gradient of the coupled formulation. Timings are given for parallel execution using 64 MPI tasks on three nodes of* `HazelHen` *(see §7.1.5 for details). The last row shows the final values for mismatch and Dice as well as (accumulated) absolute run times for the sub-components in seconds.*

It	β_v	μ_{B,L^2}	DICE$_B$	μ_{B_0,L^2}	DICE$_{B_0}$	μ_{T,L^2}	DICE$_T$	$\|\boldsymbol{g}\|_{\text{rel}}$	e_{c0,L^2}	T^{it} [s]	T^{tu}_{inv} [%]	T^{reg}_{inv} [%]
1	1	1.00	5.45E−1	6.77E−1	5.51E−1	1.00	0.00	1.00	1.00	5.64E+2	58.9	1.7
2	1E−1	8.79E−1	5.80E−1	6.07E−1	5.93E−1	2.90E−1	7.92E−1	2.04E−2	4.25E−1	9.45E+2	35.5	2.5
3	1E−2	6.54E−1	7.29E−1	4.51E−1	7.38E−1	2.04E−1	8.24E−1	1.45E−2	3.51E−1	8.44E+2	24.0	10.0
4	1E−3	4.06E−1	8.66E−1	2.80E−1	8.72E−1	1.16E−1	8.80E−1	1.36E−2	2.08E−1	1.63E+3	14.4	53.2
5	1E−4	2.62E−1	9.32E−1	1.81E−1	9.37E−1	7.57E−2	9.48E−1	7.01E−3	9.55E−2	8.46E+2	26.2	10.9
6	1E−4	2.36E−1	9.42E−1	1.64E−1	9.46E−1	6.74E−2	9.51E−1	4.91E−2	5.75E−2	8.58E+2	30.4	7.3
7	1E−4	2.33E−1	9.43E−1	1.62E−1	9.47E−1	6.49E−2	9.50E−1	4.15E−2	7.13E−2	5.69E+3	1.59E+3	1.14E+3

TABLE 7.13 *Quantitative results for the **analytic tumor with real velocity and diffusion (ATRV-C1 #1) and (ATRV-C2 #1)** test case with centers $\boldsymbol{x}_{C_1}$ and $\boldsymbol{x}_{C_2}$, and parameter setting #1 from Tab. 7.11, respectively; ground truth: ($\rho_f = 15$, $\rho_w = 1$, $\rho_g = 0$, $k_f = 1.00\text{E}{-1}$, $k_w = 1$, $k_g = 0$, $\boldsymbol{p} = \boldsymbol{p}^\star$ (in patient domain), $\boldsymbol{v}$ N/A). We compare the **moving-atlas** Picard iteration scheme (MA) equipped with a sparsity constraint (L_1wL_2 solver) for the tumor initial condition to the previously analyzed **moving-patient** Picard iteration scheme (MP). Furthermore, we study the effect of the sparsity constraint compared to the original L_2-regularization. We either invert for the registration velocity $\boldsymbol{v}$ and tumor initial condition parametrization $\boldsymbol{p}$ only, or additionally invert for the characteristic diffusivity k_f (infiltration rate of malignant cells into surrounding healthy tissue) in white matter. In the first case, k_f is set to the true value $k_f = 1\text{E}{-1}$. We always use the correct proliferation rate $\rho_f = 15$. We report the average mismatch for the probability maps for the pathologic brain tissue labels μ_{B,L^2}, the healthy brain tissue labels μ_{B_0,L^2} and the tumor μ_{T,L^2}, the mean Dice coefficient for brain tissue DICE$_B$, for the healthy patient brain tissue DICE$_{B_0}$ and for the tumor DICE$_T$. Furthermore, we measure the L_2-error e_{c0,L^2} of the reconstructed tumor initial condition to the ground truth, and the relative norm $\|\boldsymbol{g}\|_{rel}$ of the reduced gradient of the coupled formulation. Timings are given for parallel execution using 64 MPI tasks on three nodes of* `HazelHen` *(see §7.1.5 for details).*

		solver	inv-k_f	μ_{B,L^2}	DICE$_B$	μ_{B_0,L^2}	DICE$_{B_0}$	μ_{T,L^2}	DICE$_T$	$\|\boldsymbol{g}\|_{\text{rel}}$	e_{c0,L^2}	T^{it} [s]	T^{tu}_{inv} [s]	T^{reg}_{inv} [s]
ATRV-C1 (#1)	*MP*	L_2	–	3.72E−1	8.73E−1	2.56E−1	8.79E−1	1.38E−1	9.14E−1	4.84E−2	1.74	4.46E+3	3.61E+3	7.40E+2
		L_2	8.63E−2	3.71E−1	8.73E−1	2.56E−1	8.79E−1	1.35E−1	9.24E−1	4.74E−2	1.15	2.26E+3	1.47E+3	7.10E+2
		L_1wL_2	–	3.72E−1	8.72E−1	2.56E−1	8.79E−1	1.45E−1	9.21E−1	4.72E−2	3.23E−1	5.85E+3	2.27E+3	7.46E+2
		L_1wL_2	9.22E−2	3.72E−1	8.72E−1	2.56E−1	8.79E−1	1.44E−1	9.25E−1	4.84E−2	3.25E−1	5.72E+3	1.99E+3	7.88E+2
	MA	L_2	–	2.31E−1	9.43E−1	1.60E−1	9.48E−1	6.95E−2	9.47E−1	3.55E−2	2.73E−1	4.72E+3	2.93E+3	1.75E+3
		L_2	8.80E−2	2.35E−1	9.43E−1	1.62E−1	9.47E−1	6.58E−2	9.52E−1	4.24E−2	3.17E−1	3.97E+3	3.00E+3	9.34E+2
		L_1wL_2	–	2.35E−1	9.43E−1	1.62E−1	9.48E−1	6.77E−2	9.56E−1	4.22E−2	8.02E−2	5.71E+3	1.65E+3	8.97E+2
		L_1wL_2	9.05E−2	2.33E−1	9.43E−1	1.62E−1	9.47E−1	6.49E−2	9.50E−1	4.15E−2	7.13E−2	5.69E+3	1.59E+3	1.14E+3
ATRV-C2 (#1)	*MP*	L_2	–	3.76E−1	8.79E−1	2.55E−1	8.80E−1	1.93E−1	6.28E−1	4.84E−2	6.82E−1	9.36E+3	8.49E+3	7.55E+2
		L_2	5.27E−2	3.76E−1	8.79E−1	2.55E−1	8.80E−1	1.69E−1	8.17E−1	4.66E−2	7.45E−1	3.62E+3	2.81E+3	7.20E+2
		L_1wL_2	–	3.76E−1	8.80E−1	2.55E−1	8.80E−1	1.93E−1	8.34E−1	4.82E−2	4.24E−1	5.31E+3	3.33E+3	7.48E+2
		L_1wL_2	8.86E−2	3.76E−1	8.79E−1	2.55E−1	8.80E−1	1.90E−1	7.96E−1	4.76E−2	4.70E−1	6.05E+3	2.50E+3	7.49E+2
	MA	L_2	–	2.36E−1	9.48E−1	1.60E−1	9.48E−1	8.76E−2	6.74E−1	4.15E−2	7.30E−1	1.35E+4	1.22E+4	1.30E+3
		L_2	7.69E−2	2.40E−1	9.46E−1	1.62E−1	9.47E−1	7.40E−2	8.74E−1	4.95E−2	4.44E−1	5.41E+3	3.98E+3	1.39E+3
		L_1wL_2	–	2.37E−1	9.48E−1	1.60E−1	9.48E−1	6.07E−2	9.70E−1	4.18E−2	6.33E−2	7.89E+3	4.40E+3	9.26E+2
		L_1wL_2	9.23E−2	2.39E−1	9.47E−1	1.62E−1	9.47E−1	6.11E−2	9.45E−1	3.86E−2	1.19E−1	7.12E+3	2.68E+3	1.29E+3



ATRV-C1: Target Data and Ground Truth
(#1: $T = 0.16$, $\rho = 15$, $k_f = 0.1, \sigma = 2\pi/30$)

ATRV-C1: Target Data and Ground Truth
(#2: $T = 0.32$, $\rho = 15$, $k_f = 0.1, \sigma = 2\pi/64$)

Moving-Atlas $L1$w$L2$

Moving-Atlas $L2$

Moving-Atlas $L1$w$L2$

Moving-Atlas $L2$

Moving-Patient $L1$w$L2$

Moving-Patient $L2$

Moving-Patient $L1$w$L2$

Moving-Patient $L2$

FIGURE 7.13 *Qualitative results for the* ***analytic tumor with real velocity and diffusion (ATRV-C1)*** *test case with parameter setting #1 from Tab. 7.11 (left), and parameter setting #2 from Tab. 7.11 (right). The image shows the reconstructed grown tumor and tumor initial condition for the moving-atlas and moving-patient solution scheme using L_2 and $L_1 wL_2$ regularization, respectively, for parameter settings #1 and #2 from Tab. 7.11, featuring different time horizons and sparsity of the initial condition. Illustrated are parts of the (approximated) patient brain geometry with the respective reconstructed tumor initial condition (magenta wireframe/volume) as compared to the ground truth initial condition (cyan volume). The grown tumor is indicated as white wireframe (3D cut image), and as blue semi-transparent volume compared to the target data given as red wireframe (close-up image). The top row shows the test case target data and initial condition ground truth for each set of parameters. The light blue/white area indicates the ventricles with CSF. We have two observations: (i) Enforcing the sparsity constraint employing the $L_1 wL_2$ regularization yields sparse initial conditions for both MA and MP that match with the ground truth, while L_2 regularization results in rich initial conditions; (ii) the MP results show non-smooth tumor- and initial condition-shapes that originate from the registration and advection, and furthermore yield less accurate target data reconstruction (compare blue semi-transparent volume overlaid with red wireframe).*

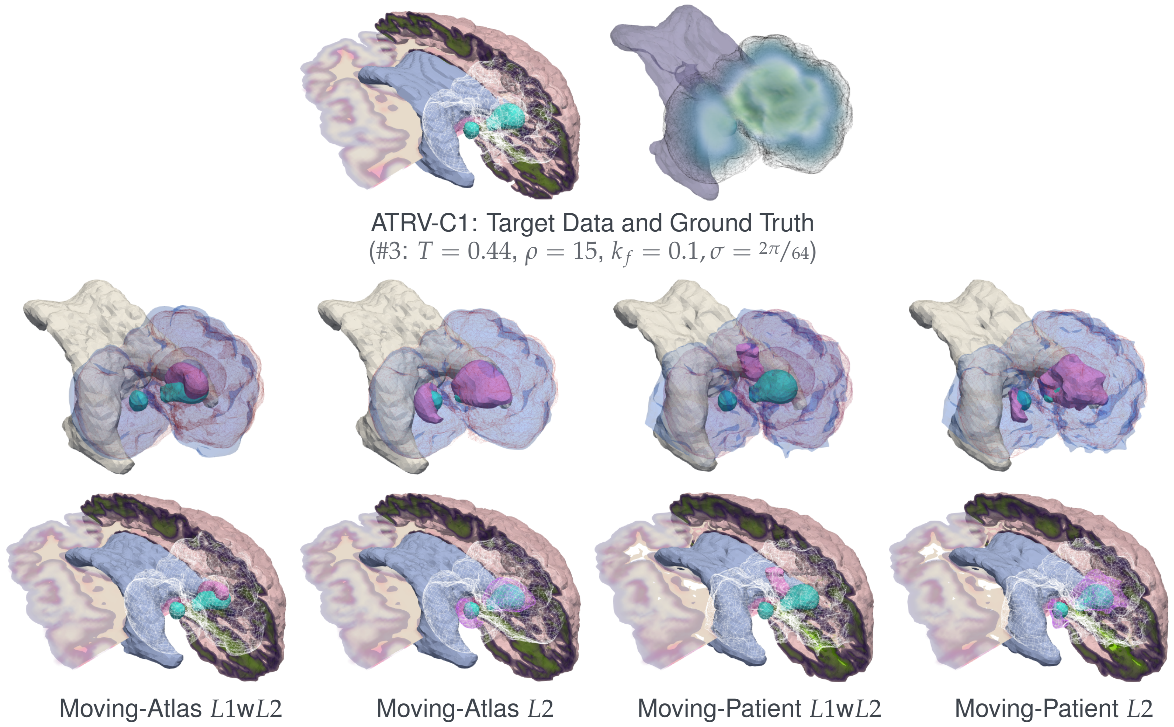
TABLE 7.14 *Quantitative results for the* ***analytic tumor with real velocity and diffusion (ATRV-C1 #1) and (ATRV-C2 #1)*** *test case with centers $\boldsymbol{x}_{C_1}$ and $\boldsymbol{x}_{C_2}$, and parameter setting #1 from Tab. 7.11, respectively; ground truth: ($\rho_f = 15$, $\rho_w = 1$, $\rho_g = 0$, $k_f = 1.00\text{E}{-1}$, $k_w = 1$, $k_g = 0$, $\boldsymbol{p} = \boldsymbol{p}^\star$ (in patient domain), $\boldsymbol{v}$ N/A). We analyze the sensitivity of the joint inversion with respect to perturbations in the characteristic diffusivity k_f. We compare results for the* ***moving-atlas (MA)*** *and* ***moving-patient (MP)*** *Picard iteration scheme with sparsity constraint (L_1wL_2) for different choices of k_f; bold numbers correspond to inversion with the ground truth value for k_f. We always use the correct proliferation rate $\rho_f = 15$. We report the average mismatch for the probability maps for the pathologic brain tissue labels μ_{B,L^2}, the healthy brain tissue labels μ_{B_0,L^2} and the tumor μ_{T,L^2}, the mean Dice coefficient for brain tissue DICE$_B$, for the healthy patient brain tissue DICE$_{B_0}$ and for the tumor DICE$_T$ (in cases without Dice score for the tumor reconstruction, the tumor probability map has values below 0.5 everywhere). Furthermore, we measure the L_2-error e_{c0,L^2} of the reconstructed tumor initial condition to the ground truth, and the relative norm $\|\boldsymbol{g}\|_{rel}$ of the reduced gradient of the coupled formulation.*

		k_f	μ_{B,L^2}	DICE$_B$	μ_{B_0,L^2}	DICE$_{B_0}$	μ_{T,L^2}	DICE$_T$	$\|\boldsymbol{g}\|_{\text{rel}}$	e_{c0,L^2}
ATRV-C1 (#1)	*MP*	5E−1	3.80E−1	8.62E−1	2.58E−1	8.79E−1	3.96E−1	–	4.89E−2	4.99E−1
		3E−1	3.76E−1	8.64E−1	2.57E−1	8.79E−1	2.88E−1	7.76E−1	1.75E−2	8.68E−1
		1E−1	**3.72E−1**	**8.72E−1**	**2.56E−1**	**8.79E−1**	**1.45E−1**	**9.21E−1**	**4.72E−2**	3.23E−1
		1E−3	3.73E−1	8.75E−1	2.59E−1	8.73E−1	2.49E−1	8.75E−1	4.90E−2	5.39E−1
	MA	5E−1	2.40E−1	9.46E−1	1.62E−1	9.47E−1	5.65E−1	–	3.86E−2	1.37
		3E−1	2.40E−1	9.37E−1	1.63E−1	9.48E−1	2.86E−1	3.66E−1	2.64E−2	9.41E−1
		1E−1	**2.35E−1**	**9.43E−1**	**1.62E−1**	**9.48E−1**	**6.77E−2**	**9.56E−1**	**4.22E−2**	**8.02E−2**
		1E−3	2.37E−1	9.48E−1	1.61E−1	9.48E−1	5.40E−1	–	4.06E−2	7.04E−1
ATRV-C2 (#1)	*MP*	3E−1	3.76E−1	8.79E−1	2.55E−1	8.80E−1	3.66E−1	–	4.78E−2	4.96E−1
		2E−1	3.76E−1	8.79E−1	2.55E−1	8.80E−1	2.64E−1	–	4.84E−2	5.25E−1
		1E−1	**3.76E−1**	**8.80E−1**	**2.55E−1**	**8.80E−1**	**1.93E−1**	**8.34E−1**	**4.82E−2**	**4.24E−1**
		1E−2	3.82E−1	8.75E−1	2.59E−1	8.75E−1	2.57E−1	6.86E−1	1.41E−2	6.94E−1
	MA	3E−1	2.41E−1	9.46E−1	1.63E−1	9.47E−1	4.20E−1	–	3.72E−2	5.99E−1
		2E−1	2.40E−1	9.46E−1	1.62E−1	9.47E−1	2.54E−1	–	3.94E−2	7.96E−1
		1E−1	**2.37E−1**	**9.48E−1**	**1.60E−1**	**9.48E−1**	**6.07E−2**	**9.70E−1**	**4.18E−2**	**6.33E−2**
		1E−2	2.40E−1	9.46E−1	1.62E−1	9.47E−1	3.39E−1	6.21E−1	3.88E−2	6.78E−1

reconstructed tumor initial condition with respect to the ground truth and the relative norm $\|\boldsymbol{g}\|_{\text{rel}}$ of the reduced gradient of the coupled formulation (6.10).

For both test case variants (ATRV-C1 and ATRV-C2), we compare results obtained using the moving-atlas strategy to the results for the moving-patient solution scheme. Furthermore, we study the effect of the sparsity constraint for the tumor initial condition (L_1wL_2 solver) compared to the original L_2-regularization. In this experiment, we either invert for the registration velocity $\boldsymbol{v}$ and tumor initial condition parametrization $\boldsymbol{p}$ only, or additionally invert for the characteristic diffusivity k_f (infiltration rate of malignant cells into surrounding healthy tissue) in white matter. In the first case, k_f is set to the true value of $k_f = 1\text{E}{-1}$. We always use the correct proliferation rate $\rho = 15$ and report timings with regards to parallel execution using 64 cores on three nodes of `HazelHen`. Qualitative results of the reconstructed tumor initial condition and the grown tumor compared to the target data and ground truth initial condition are given in Fig. 7.13 (for ATRV-C1) and Fig. 7.15 for (ATRV-C2). Fig. 7.13 furthermore illustrates and compares the reconstruction performance for the moving-atlas L_1wL_2 and MA-L_2 solver, and the moving-patient L_1wL_2 and MP-L_2 solver, respectively, for two different variations of ATRV-C1 (featuring different tumor-growth time horizons and different sparsities of the ground truth initial condition). A similar, comparison is drawn in Fig. 7.15 for ATRV-C2. We exemplarily show qualitative results for a longer time horizon of tumor evolution for the ATRV-C1 test case (with a very sparse initial condition) in Fig. 7.14.

We also study the sensitivity of the moving-atlas and moving-patient solution strategy with respect to perturbations in the tumor model parameters. In Tab. 7.14, we vary the value of the characteristic diffusivity in white matter from the ground truth and monitor the obtained reconstruction performance for both joint inversion strategies. For all runs, we employ a sparsity constraint for the tumor initial condition, i.e., we use the solvers moving-atlas-L_1wL_2 and moving-patient-L_1wL_2.



ATRV-C1: Target Data and Ground Truth
(#3: $T = 0.44$, $\rho = 15$, $k_f = 0.1, \sigma = {}^{2\pi}/_{64}$)

Moving-Atlas $L1$w$L2$ Moving-Atlas $L2$ Moving-Patient $L1$w$L2$ Moving-Patient $L2$

FIGURE 7.14 *Qualitative results for the* ***analytic tumor with real velocity and diffusion (ATRV-C1 #3)*** *test case with parameter setting #3 from Tab. 7.11. The image shows the reconstructed grown tumor and the tumor initial condition for the* ***moving-atlas*** *and* ***moving-patient*** *solution scheme using L_2 and L_1wL_2 regularization, respectively. The images show parts of the (approximated) patient brain geometry and illustrate the respective reconstructed tumor initial condition (magenta wireframe/volume) as compared to the ground truth initial condition (cyan volume). The grown tumor is indicated as white wireframe (bottom row), and as blue semi-transparent volume compared to the target data given as red wireframe (top row). The light blue/white area indicates the ventricles with CSF. The first row shows the test case target data and tumor initial condition ground truth. We have two observations: (i) Enforcing the sparsity constraint employing the L_1wL_2 regularization yields sparse initial conditions for both MA and MP that match with the ground truth, while L_2 regularization results in rich initial conditions; (ii) the MP results show non-smooth tumor- and initial condition-shapes that originate from the registration and advection, poor reconstruction of the grown tumor, and the MP L_1wL_2 reconstruction yields a wrong position of the initial condition.*

Observations. Reviewing the performance and reconstruction quality measures in Tab. 7.12 for the solution of the ATRV-C1 test case using the moving-atlas Picard iteration solution scheme with sparsity constraint for the tumor initial condition (L_1wL_2 solver), we observe excellent target data and ground truth reconstruction quality. In particular, the L_2-mismatch for the brain anatomy is reduced by 77% (compared to the initial error), and for the tumor reconstruction, we reach a reduction of 93% with respect to the initial error. The results furthermore attest a very good approximation of the healthy patient anatomy by the moving-atlas inversion scheme, improving the initial Dice score from 5.51E−1 to an excellent agreement of 9.47E−1. The direct approximation of the healthy patient anatomy is a new feature introduced by employing the moving-atlas strategy; the previously studied moving-patient formulation only indirectly allows for an approximation of the healthy patient by advection of the healthy atlas brain. Using the moving-atlas strategy in conjunction with the L_1wL_2 sparsity constraint for the tumor inversion solver, we are able to very accurately recover the initial condition used to generate the target data of the synthetic test case (with an error of 7%). An important fact to note is, that the reduced gradient for the coupled formulation decays over the course of the joint inversion, which indicates convergence to a local minimum.

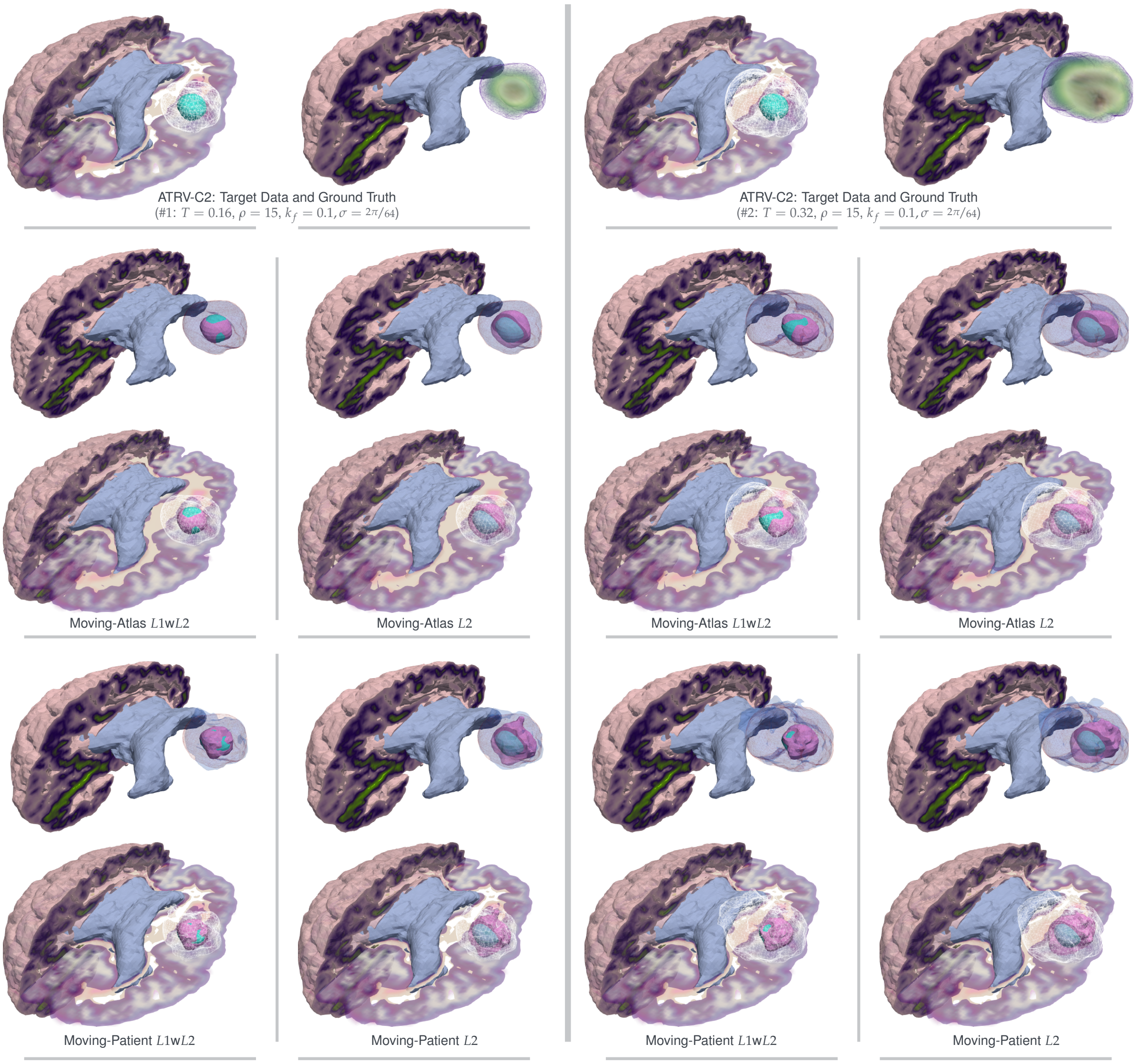

ATRV-C2: Target Data and Ground Truth
(#1: $T = 0.16$, $\rho = 15$, $k_f = 0.1, \sigma = 2\pi/64$)
ATRV-C2: Target Data and Ground Truth
(#2: $T = 0.32$, $\rho = 15$, $k_f = 0.1, \sigma = 2\pi/64$)
Moving-Atlas $L1$w$L2$
Moving-Atlas $L2$
Moving-Atlas $L1$w$L2$
Moving-Atlas $L2$
Moving-Patient $L1$w$L2$
Moving-Patient $L2$
Moving-Patient $L1$w$L2$
Moving-Patient $L2$

FIGURE 7.15 *Qualitative results for the* ***analytic tumor with real velocity and diffusion (ATRV-C2)*** *test case with parameter setting #1 from Tab. 7.11 (left image), and parameter setting #2 from Tab. 7.11 (right image). The image shows the reconstructed grown tumor and the tumor initial condition for the* ***moving-atlas*** *and* ***moving-patient*** *solution scheme using L_2 and L_1wL_2 regularization, featuring different time horizons for the tumor evolution. Illustrated are parts of the (approximated) patient brain geometry with the respective reconstructed tumor initial condition (magenta wireframe/volume) as compared to the ground truth initial condition (cyan volume). The grown tumor is indicated as white wireframe (bottom images), and as blue semi-transparent volume compared to the target data given as red wireframe (top images). The first row shows the test case target data and initial condition ground truth for each set of parameters.*

Moving-Atlas versus Moving-Patient. The moving-atlas Picard iteration solution strategy outperforms the moving-patient counterpart in various ways. For both considered test case variations in Tab. 7.13, the L_2-mismatch between the target and the reconstructed probability maps for the brain tissue labels as well as the L_2-error of the reconstruction of the target tumor are reduced significantly compared to the moving-patient results. For the moving-patient scheme, the overlap of the tissue labels of the target brain and reconstruction after joint inversion translates to a Dice score of 8.72E−1 compared to a Dice score of 9.43E−1 for the moving-atlas scheme. A similar improvement can be observed for the approximation of the probability maps for the healthy patient anatomy: The moving-patient inversion scheme results in an L_2-error of 25.6% (relative to the initial mismatch between atlas and ground truth healthy patient) and a Dice score of 8.79E−1 as opposed to a relative L_2-error of only 16.2% with Dice score of 9.47E−1 for the moving-atlas inversion. These results and the improved anatomy reconstruction are characteristic for the moving-atlas strategy; we observe similar trends for various solver and parameter configurations (compare also Tab. 7.14 with inversion under perturbed or wrong tumor model parameters).

Regardless of the employed regularization, the moving-atlas scheme furthermore results in improved reconstruction quality of the predicted grown tumor state and higher similarity to the target data. For ATRV-C1, the moving-atlas scheme results in a relative error of 6.4% compared to a relative error of 14.4% for the moving-patient solution of the same case. The numbers for ATRV-C2 are very similar with 6.0% relative error versus 19.0% relative error. A non-negligible fact is also that the predicted tumor is grown “naturally” (using the employed progression model) in the (approximated) patient brain anatomy when considering the moving-atlas scheme. As opposed to this, the reconstructed probability map for the tumor as obtained from the moving-patient solution scheme is substantially modified and deformed by the registration to ensure a good data-similarity. As a result, the tumor shape is non-smooth and the reconstruction may fall short to realistically represent the cell distribution of cancerous cells[15]. This can be seen from visual inspection of the simulation data in Figs. 7.13, 7.14 and 7.15.

The primary objective for the derivation of the moving-atlas scheme was to allow for a more informative inversion for biophysical model parameters. We primarily consider the estimation of the tumor initial condition, but also invert for the characteristic diffusivity k_f in white matter. The relative errors[16] e_{c0,L^2} (cf. (7.1d)) for the inversion of the ground truth tumor initial condition in Tab. 7.13 obtained for each scheme clearly attest the moving-atlas scheme to be more reliable and sound in recovering “biophysically meaningful”[17] results. Only considering the results obtained with enforced sparsity constraint (L_1wL_2 solver) from Tab. 7.13, we observe excellent reconstruction of the true initial condition with a relative error of only 7.1% (for ATRV-C1) and 6.3% (for ATRV-C2) using the moving-atlas scheme, as opposed to an error of 32.5% (for ATRV-C1) and 42.4% (for ATRV-C2), respectively, when using the moving-patient scheme instead. We explain the improved inversion properties of the moving-atlas scheme by its general idea to seek for a good approximation of the

[15] We do not claim that the currently employed moving-atlas strategy allows for such a realistic representation of the tumor cell distribution. However, if used with a more realistic tumor progression model, the moving-atlas strategy is clearly more promising than the moving-patient alternative.

[16] Note, that for the moving-atlas scheme, the reconstructed initial condition naturally “lives” in the patient space; for the moving-patient solution strategy, we invert for the tumor initial condition in the atlas space. For a fair comparison, the reconstructed initial condition for the moving-patient scheme is advected to the patient space before computing the relative error to the ground truth.

[17] Our analysis assumes that we do not have an error in the type of model which describes the tumor progression. In other words, we assume that tumor-growth is perfectly described by a reaction-diffusion model, which is quite certainly not true. When we say “biophysically meaningful” this is to be seen under the aforementioned assumption. To allow for a comparison of the solution strategies and as a proof of concept for our solution methodology, it is reasonable and valid to make such an assumption.

healthy patient brain anatomy first, and, thus, to carry out the inversion in the “correct” space. For the moving-patient scheme, the inversion in the “wrong” atlas space has the potential to induce large errors for the estimation of biophysical parameters, as the inter-subject deformation map (registration velocity) can aid matching the pathologic brains. In particular, there is no implication that this deformation map produces reasonable results when applied to the tumor initial condition (or other quantities that describe the time point of tumor genesis). This point is supported by visual inspection of our simulation results for longer time horizons. We observe unnaturally deformed shapes of the initial condition, and specifically in Fig. 7.14, the moving-patient solution recovers a wrong position of the tumor initial condition. A wrong position, wrong shape, or wrong sparsity of the initial condition can, in a second step, also cause a wrong estimation of the characteristic proliferation rate or tumor cell infiltration rate. For our experiments, the difference for the estimation of the characteristic diffusivity k_f is, however, not significant; both schemes yield comparable results. We assume that this is caused by the diffusion part of the gradient initially being steep but then flattening out compared to the part for the initial condition parametrization. The theoretical and numerical analysis of this effect is ongoing work.

On the other hand, if we do not invert for the characteristic diffusivity k_f but perturb its value from the ground truth for the inversion for the initial condition, we again observe improved qualities of the moving-atlas scheme over the moving-patient solution: From Tab. 7.14 we see, that the moving-atlas solution clearly detects wrong values for the diffusivity k_f by presenting a very poor similarity of target data and the predicted tumor in this case and a high μ_{T,L^2} tumor mismatch discrepancy. The moving-patient strategy does not show a comparable indication. In particular, the latter almost becomes oblivious of wrong model parameters if the fidelity of the parametrization for the tumor initial condition is increased (a larger number n_p of Gaussians with smaller standard deviation σ result in a reduced interpolation error). We have seen this in §7.2, e.g., in Tab. 7.6.

Lastly, we note that, with respect to time-to-solution and efficiency, both solution schemes are very comparable.

Sparsity Constraint (L_2 vs. L_1wL_2). We analyze the effect and the performance of the sparsity constraint for the tumor initial condition quantitatively in Tab. 7.13 and qualitatively in Figs. 7.13, 7.14, and 7.15. For both solution strategies, moving-atlas and moving-patient, we only achieve sparse initial conditions if the combination of L_1-regularization followed by a weighted L_2-regularization for the tumor inversion is used. Specifically for the moving-atlas scheme, this combination allows to very accurately reconstruct the true tumor initial condition used for the generation of the synthetic target data. This is not the case if we use the L_2-regularization for the tumor inversion. From visual inspection of our simulation results (Figs. 7.13, 7.14, and 7.15), we observe the reconstruction of too rich initial conditions for the inversion with an L_2-regularization functional.

Conclusion: Biophysical Parameter Estimation

We have analyzed and improved a more sophisticated formulation of the considered coupled multi-component problem using our derived Picard iteration type solution algorithm (see §6.3). For this so called moving-atlas scheme (see §6.2), we directly seek for an approximation of the healthy patient brain anatomy. This results in several improved properties with respect to the estimation of meaningful biophysical parameters, as compared to the previously studied moving-patient scheme (see §6.1). The conducted experiments are to be seen as a proof of concept and are clearly preliminary, yet indicate a very promising step towards a powerful model calibration and patient specific tumor analysis tool. Here is what we have learned from our experiments on synthetic and real data:

(i) *Our results show that the moving-atlas solution scheme clearly outperforms the moving-patient scheme (for the considered test cases) in terms of reconstruction quality for the brain anatomy, the healthy patient brain, the grown tumor, and most importantly for the reconstruction of the true initial condition used to generate the synthetic target tumor.*

(ii) *In particular, using the moving-atlas scheme, we were able to improve the Dice score for the brain anatomy from* 8.72E−1 *for the moving-patient strategy to* 9.43E−1*; the Dice score for the approximation of the healthy patient brain anatomy from* 8.79E−1 *for moving-patient to* 9.47E−1*; and the dice coefficient for the tumor reconstruction from* 8.34E−1 *for moving-patient to* 9.70E−1.

(iii) *The moving-atlas solution scheme in combination with a sparsity constraint for the tumor initial condition is capable of reconstructing the true tumor initial condition used for the generation of the synthetic target data with very high accuracy. We observe an excellent reconstruction of the true initial condition with a relative error of only* 7.1% *(for ATRV-C1) and* 6.3% *(for ATRV-C2) (compared to* 32.3% *and* 42.4% *for the moving-patient scheme) using the more sophisticated moving-atlas scheme. We further conclude, that the implementation of a sparsity constraint for the initial condition is essential if targeting biophysical parameter estimation from patient MRI. The previously used* L_2*-regularization results in rich initial conditions.*

(iv) *The reconstruction result obtained from the moving-atlas scheme appears to be more "natural" as the inversion is carried out in the "correct" brain anatomy of the (healthy) patient, and the registration does not act on the tumor probability map (as is the case for the moving-patient strategy).*

(v) *Lastly, the moving-atlas solution scheme has a higher capability of detecting wrong model parameters (such as cell proliferation rate and cell migration rate of brain tissue) than the moving-patient counterpart.*

7.4 Evaluation of Advanced Methods to Improve Performance and Runtime

This section studies various advanced methods for our joint inversion approach with the primary focus to either speedup the time-to-solution and increase performance of the scheme, or facilitate analysis and foster a better understanding of the scheme. With respect to the latter, we investigate the behavior of a corrected Picard formulation in §7.4.1, resembling a block-Newton iteration for the coupled formulation and target convergence of our scheme and the effect of different parameter-continuation schemes on the final solution in §7.4.2. In the remainder, we evaluate methods that ought to reduce the required time-to-solution of our method. In particular, we look at grid-continuation (a multilevel scheme) in §7.4.4 and replace the Gauß-Newton-Krylov for the tumor inversion solver by a LBFGS quasi-Newton alternative §7.4.5.

7.4.1 Block-Newton-Type Iterations – Corrected Objective

Recall that our Picard iteration solution schemes neglect parts of the fully coupled problem formulations in §6.4 (moving-patient) and §6.10 (moving-atlas). In particular, all numerical experiments considered so far use the simplified-objective formulation (6.16) and (6.19), considering only the data-misfit for the tumor probability map in the tumor inversion sub-problem. In this section, we study the corrected full-objective formulations (6.18) and (6.22) for the tumor inversion sub-problem and examine the impact on efficiency/convergence of the scheme and and quality of the respective solution.

Purpose. We study the performance of the full-objective block-Newton scheme for a series of different test cases, including synthetic and real patient data. In particular, we are interested in the reconstruction quality for probability maps of brain tissue labels and pathology, as well as convergence of the scheme and its overall runtime compared to the simpler scheme employed in large parts of this thesis, which only uses the misfit for the pathology reconstruction in the objective for the tumor inversion. We consider the block-Newton scheme for the moving-patient formulation and the moving-atlas formulation and in conjunction with different regularization schemes for the tumor inversion solver.

Setup. The full-objective block-Newton scheme uses the same mismatch terms in the objective function for both sub-solvers, registration and tumor inversion. As a result, the tumor inversion also sees the data-misfit for the probability maps of the healthy brain tissue labels and the latter occurs in the final condition of the tumor adjoint problem. We consider three different test cases: The synthetic cases ATAV and ATRV and the real data RTRV cases. The setup of the cases and solver settings remains unchanged from §7.1.1. We consider ATAV and RTRV for the moving-patient formulation and employ the moving-atlas Picard iteration scheme for ATRV experiments. For the latter, we also use the tumor inversion with sparsity constraint, i.e., an L_1-regularized inversion to extract the sparsity of the initial condition, followed by a weighted L_2-regularized tumor inversion, penalizing entries in $\boldsymbol{p}$ that have not been selected in the L_1-regularized inversion.

Results. We report quantitative results for synthetic cases (ATAV, ATRV) and clinical data of actual glioma patients (RTRV) in Tab. 7.15. We compare the full-objective block-Newton scheme to the simplified-objective scheme for the moving-patient and moving-atlas formulation applied to series of different test cases, respectively.

TABLE 7.15 *Results for the full-objective block-Newton scheme. We compare the full-objective block-Newton scheme to the simplified-objective solution scheme for various cases. We consider the moving-patient and moving-atlas formulation, as well as different regularization schemes for the tumor inversion to employ a sparsity constraint on the initial condition. We report the average mismatch for the probability maps for the brain tissue labels μ_{B,L^2} and the tumor μ_{T,L^2}, the mean Dice coefficient for brain tissue $DICE_B$ and tumor $DICE_T$, respectively. Timings for the overall time-to-solution, tumor inversion, and registration solve are given in seconds with respect to execution on* 3 *nodes/*64 *MPI tasks for the $n = 128^3$ resolution ATAV and ATRV cases, and execution on* 11 *nodes/*256 *MPI tasks for the $n = 256^3$ resolution RTRV cases.*

			Objective	Scheme	μ_{B,L^2}	DICE$_B$	μ_{T,L^2}	DICE$_T$	$\\|g\\|_{\text{rel}}$	T^{it} [s]	T^{tu}_{inv} [s]	T^{reg}_{inv} [s]
ATAV	REAC	#1	simple	MP-L_2	1.57E−1	9.31E−1	6.29E−2	9.73E−1	1.21E−2	4.45E+2	2.99E+1	4.13E+2
		#2	full	MP-L_2	1.66E−1	9.29E−1	1.08E−1	9.56E−1	8.23E−4	6.31E+2	9.39	6.16E+2
RTRV	AAAN	#3	simple	MP-L_2	3.54E−1	8.74E−1	3.77E−1	8.91E−1	1.06E−1	6.34E+2	2.15E+2	4.17E+2
		#4	full	MP-L_2	3.54E−1	8.75E−1	3.71E−1	9.13E−1	5.82E−3	6.38E+2	2.42E+2	3.61E+2
	AAMH	#5	simple	MP-L_2	3.45E−1	8.43E−1	1.95E−1	9.57E−1	3.92E−2	6.28E+2	1.95E+2	4.35E+2
		#6	full	MP-L_2	3.47E−1	8.43E−1	1.99E−1	9.64E−1	2.98E−3	5.54E+2	1.45E+2	3.73E+2
ATRV	L_2	#7	simple	MA-L_2	2.62E−1	9.29E−1	5.32E−2	9.65E−1	4.66E−2	1.00E+4	8.99E+3	1.00E+3
		#8	full	MA-L_2	2.62E−1	9.31E−1	1.35E−1	7.77E−1	8.53E−3	4.77E+3	3.74E+3	1.00E+3
	L_1wL_2	#9	simple	MA-L_1wL_2	2.63E−1	9.29E−1	5.15E−2	9.67E−1	4.82E−2	5.29E+3	4.25E+3	1.01E+3
		#10	full	MA-L_1wL_2	2.67E−1	9.32E−1	3.42E−1	1.69E−1	4.58E−3	1.38E+4	1.27E+4	1.07E+3

Observations. Qualitative and quantitative results for both schemes, the full-objective block-Newton scheme and the simplified-objective scheme, are similar, yet there are differences. While the final data-misfit for the brain tissue label probability maps is almost the same for all cases, we observe a tendency of less accurate tumor reconstruction results for the full-objective block-Newton scheme. For the ATAV test case, we see a final L_2-error for the tumor reconstruction of 6.29E−2 and 1.08E−1 (runs #1 and #2) for the simplified- and full-objective scheme, respectively. Similarly, for ATRV employing the moving-atlas Picard iteration, we observe final L_2-errors 5.32E−2 and 1.35E−1 (runs #7 and #8) for the tumor reconstruction, driven by the respective objective functions. The lower accuracy of tumor reconstruction for the block-Newton scheme can be explained by a smaller weighting of the tumor reconstruction misfit in the tumor inversion objective since L_2-errors for white matter, gray matter and CSF attenuate its driving force. Introducing appropriate weighting for the respective labels in the data-misfit term might be necessary. For the clinical data sets (RTRV; runs #3–#6), where we do not solve very accurately but allow only a maximum number of 30 Krylov and Newton iterations, we observe almost identical reconstruction accuracy for both schemes. The relative gradient norm of the reduced gradient for the respective coupled formulation is significantly smaller for the block-Newton scheme.

When analyzing the block-Newton scheme in more detail, we observe line-search issues for the tumor inversion solver (using the full-objective). In particular in conjunction with the sparsity constraint,[18] this causes the solver to break and we observe a drastic deterioration of reconstruction performance (3.42E−1 relative error of tumor reconstruction compared to 5.15E−2 relative error for the simplified-objective scheme; runs #9 and #10). Using the full-objective, the brain tissue data-misfit occurs in the final condition for the tumor adjoint equation. Line-search problems may arise due to significantly modified data by the registration, however, a more in-depth analysis is required to understand and remedy this problem.

With respect to runtime, the tumor inversion (and, as a result, the overall coupled inversion scheme) seems to be slightly faster for the block-Newton variation, with exception of the cases where an excessive number of line-search attempts slow down the tumor inversion solver. These results are,

[18] A constraint to enforce sparse initial conditions. To this end, we perform a tumor inversion with L_1-regularization to get the correct sparsity pattern, which is followed by a weighted L_2-regularization step, penalizing support of basis functions that have not been selected in the first run.

however, not significant.

Conclusion *Qualitative and quantitative results for the full-objective block-Newton scheme (see (6.18) for the moving-patient formulation and (6.22) for the moving-atlas formulation) are similar to the Picard scheme using a simplified tumor objective, which omits data-misfit for healthy brain tissue. Due to less emphasis on the tumor misfit term in the full-objective scheme, however, we observe a tendency for lower quality of the tumor reconstruction when compared to the simplified-objective scheme. The block-Newton scheme exhibits unresolved line-search issues at more frequent intervals, which is why we focus on the simplified-objective scheme. The runtime differences are insignificant with a marginal advantage for the block-Newton scheme.*

7.4.2 Different Parameter-Continuation Schemes

As described in §6.3.3, we use parameter-continuation schemes to convexify the solution process and foster convergence to a global minimum. For the moving-patient Picard scheme continuation on the regularization weight for the registration problem is also a methodological requirement.

***Purpose*.** We study the effect of different parameter-continuation schemes. In particular, we also examine parameter continuation on β_p (tumor regularization) and test different reduction-stepping/interlacing of the β_v- and β_p-continuation. Interesting questions to raise are (*i*) whether the final solution is independent of the continuation scheme, and (*ii*) if an additional continuation on β_p is beneficial (in terms of accuracy or runtime).

***Setup*.** We consider the fully analytical test case ATAV, as described in §7.1.1 and §7.2.2 with a reaction-diffusion tumor model ($\rho = 10$, $k_f = 1\text{E}{-2}$, $k_w = 1$, $k_g = 0$, i.e., growth in white matter only). With the exception of the settings for the parameter-continuation scheme (detailed below), we apply the same solver settings such as tolerances, number of iterations and regularization terms as in the above referenced description.

***Results*.** Results for four different combinations of parameter continuation schemes are given in Fig. 7.16. The figure indicates the reduction-stepping of β_v and β_p for the different schemes on the right and shows the (summed) L_2-error for the reconstruction of the probability maps of the brain tissue labels (left) and pathology (middle) plotted over the value of β_v in the continuation scheme. The black dots indicate Picard iterations. The table provides quantitative results for the presented schemes.

***Observations*.** The most important observation is that the joint inversion result depends on the employed continuation-scheme. For different β_p-continuation schemes (tumor regularization) we obtain fairly different L_2-error values, ranging from 1.66E−1 to 1.89E−1 relative error for brain tissue labels and from 1.08E−1 to 1.71E−1 relative error for the tumor reconstruction. We observe best results for a constant and small tumor regularization weight, i.e., without β_p-continuation and high fidelity tumor reconstruction from the very beginning of the Picard scheme. The overall reconstruction quality of the joint inversion decreases if we apply various β_p-continuation schemes and we do not observe a significant reduction in runtime.

In particular, the results from scheme IV in Fig. 7.16 exhibit a fundamental problem of the moving-patient formulation if used for biophysical inversion with the ultimate goal of employing patient specific tumor progression simulations: If the tumor inversion continues to yield a poor

Scheme	it_{sibia}	β_v	β_p	μ_{B,L^2}	DICE$_B$	μ_{T,L^2}	DICE$_T$	$\|g\|_{\text{rel}}$
	init	–	–	6.79E−1	1.00	0.00	1.00	
Scheme I	7	1—1E−4	1E−4	1.66E−1	9.48E−1	1.08E−1	9.56E−1	8.23E−4
Scheme II	7	1—1E−4	1E+1—1E−3	1.71E−1	9.48E−1	1.22E−1	9.58E−1	8.51E−4
Scheme III	7	1—1E−4	1E+1—1E−1	1.84E−1	9.44E−1	1.62E−1	9.47E−1	8.75E−4
Scheme IV	12	1—1E−4	1—1E−2	1.89E−1	9.42E−1	1.71E−1	9.44E−1	1.82E−3

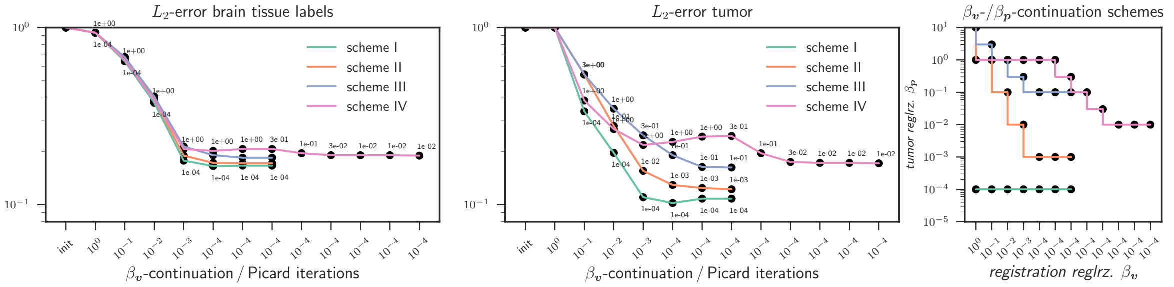


FIGURE 7.16 *Parameter-continuation schemes for* ***analytic tumor / analytic velocity with non-zero diffusion (ATAV-DIF)*** *test case; ground truth: ($\rho_f = 10$, $\rho_w = 1$, $\rho_g = 0$, $k_f = 1.00\text{E}{-2}$, $k_w = 1$, $k_g = 0$, $\boldsymbol{p} = \boldsymbol{p}^\star$, $\boldsymbol{v} = -\boldsymbol{v}^\star$). We report L$_2$-error for probability maps of brain tissue labels and tumor for different interlacing of β_v- and β_p-parameter continuation. The parameter stepping for β_v and β_p for every Picard iteration of the respective schemes is outlined on the right. Annotations in the L$_2$-error plots indicate values of the tumor regularization parameter β_v; values for β_v are given (every value corresponds to a Picard iteration in the solution scheme). The table shows L$_2$-error and Dice coefficients for the final iteration of each scheme for the probability maps of brain tumor tissue labels and tumor, respectively, as well as the relative norm of the coupled gradient (6.9).*

reconstruction[19] of the target pathology until the registration regularization is sufficiently small, the registration fits the tumor target data[20] to the poor reconstruction. This process is irreversible. Even if the biophysical model eventually becomes detailed/sophisticated/complex enough to perfectly reconstruct the pathology, the input data is already adjusted and the scheme converges to a non-optimal or wrong solution.

Conclusion *From this experiment, we conclude, that performing an additional parameter-continuation on the tumor regularization weight β_p is not beneficial. The solution of the joint inversion scheme depends on the employed (interlaced) continuation scheme and the reduction stepping for the regularization weights. In particular, for the moving-patient scheme, we observe that it is dangerous to maintain a rough/poor reconstruction of the pathology until the regularization weight for the registration hits regimes that allow for complex deformations (significant velocities $\boldsymbol{v} \neq 1$).*

7.4.3 Convergence of the Picard Iteration

Whereas we mainly present final results after convergence in §7.2 and §7.3, we examine the convergence behavior of our Picard iteration itself in this section. In particular, we are interested if the scheme keeps converging for smooth enough input data, or if it gets trapped in local minima.

Purpose. We study and verify the convergence of our scheme (moving-patient formulation). In particular, we investigate the dependence (of the convergence) of our scheme on the mesh size, i.e., the number of spatial unknowns n (reflected in unknowns for the velocity $\boldsymbol{v} \in \mathbf{R}^{3n}$), and the number of Gaussians n_p (reflected in unknowns for the tumor solver in $\boldsymbol{p} \in \mathbf{R}^{n_p}$ for the parametrization

[19] Here the poor reconstruction is due to a too large regularization parameter. In reality, a poor or too simplistic model is, however, a more realistic cause for poor reconstruction.

[20] Notice, that in the moving-patient formulation, the tumor inversion target data is the warped-to-atlas patient pathology.

of $c^{(\cdot,0)}$). For smooth enough data[21], Newton's method features "ideal" convergence properties in the sense that, near a solution it converges quadratically and maintains this convergence rate independently of the mesh (number of unknowns) [Hin09b]. We want to investigate our scheme towards these properties.

Furthermore, in a first proof-of-concept experiment, we investigate a new solution strategy based on ***inexact solves*** for each sub-problem. This means, we only perform a small number of Newton iterations per sub-component and alternate frequently between the two (detailed below). This emphasizes the coupling and possibly reduces oscillations. In particular, we want to study the effects of this alternative solution strategy on the convergence properties.

Setup. We consider a fully synthetic, smooth test case using sinusoidal functions for white matter and gray matter, defining the artificial brain. We use white matter for the atlas configuration as $m_{WM} := 1/2 cos(x_1)sin(x_2) + 1/2$ and gray matter as $m_{GM} := 1 - m_{WM}$; CSF is set to zero. Based on a smooth, sinusoidal velocity $\boldsymbol{v}^\star(\boldsymbol{x}) = (v_1^\star(\boldsymbol{x}), v_2^\star(\boldsymbol{x}), v_3^\star(\boldsymbol{x}))^T$ with $v_1^\star(\boldsymbol{x}) = sin(x_3)cos(x_2)sin(x_2)$, $v_2^\star(\boldsymbol{x}) = sin(x_3)cos(x_3)sin(x_3)$, and $v_3^\star(\boldsymbol{x}) = sin(x_2)cos(x_1)sin(x_1)$, we generate a patient configuration analogously to the ATAV test case by growing an artificial tumor in the atlas configuration and advecting the resulting abnormal brain with $\boldsymbol{v}^\star$ to the patient space.

We use the full-objective variant (6.18), which results in a block-Newton (or block-Gauß-Seidel) scheme for the modular Picard-type solution of the coupled problem. With regards the numerical solution, we consider two different strategies.

(a) ***A block-iterative Gauß-Seidel-type solution based on exact sub-component solves.*** This strategy reflects the solution method for most of the experiments conducted within this work. However, different from all other test cases and Tab. 7.2, we set the tolerance for the relative gradient in the registration[22] and in the tumor inversion solver to $\text{opttol}_R = \text{opttol}_T = 1\text{E}{-7}$ and allow $\text{maxit}_R = (100, 100)$ and $\text{maxit}_T = (100, 100)$ Newton and Krylov iterations, respectively. We enforce very accurate solutions (exact solves) to prevent the coupled solver from possible issues originating from early termination in the sub-components.

(b) ***An iterative solution based on inexact sub-component solves.*** This strategy follows an opposite approach: Instead of solving each sub-component "exactly" for the given input, we ask for fairly inexact sub-component solutions, but increase the frequency of alternating optimization in each direction. To achieve an overall acceptable accuracy, we significantly increase the number of sub-component calls (i.e., Picard iterations). In contrast to approach (a), this strategy might be helpful in order to prevent the solution process from descending in sub-optimal steps that emphasize the optimization of one components more than the other. This can hinder convergence due to significant oscillations, or lead to a local minimum. This short investigation is a proof-of-concept.

Concretely, in every Picard iteration, we perform two Newton iterations for the registration and tumor inversion solver, respectively. To compensate for the inexact solution, we perform a total of 10 Picard iterations before the regularization weight for the registration is reduced (as opposed to one iteration for scheme (a)). That means, within the parameter-continuation scheme, we perform a total of 20 Newton iterations per sub-component solver and regularization weight β_v. This is comparable to the number of Newton iterations performed for the above exact block-Newton scheme for a fixed regularization weight.

[21] all problem features need to be fully resolved already on the coarsest mesh

[22] the gradient-based convergence criterion (C1) is used instead of the (C2) criterion used in all the other test cases (cf. §6.3.3)

TABLE 7.16 *Results for the* ***sinusoidal analytic tumor / analytic velocity (SIN)*** *test case; ground truth: ($\rho_f = 15$, $\rho_w = 1$, $\rho_g = 0$, $k_f = 0$, $\boldsymbol{p} = \boldsymbol{p}^\star$, $\boldsymbol{v} = -\boldsymbol{v}^\star$). We report convergence of the Picard iteration scheme for the SIN test case for three runs with increasingly refined mesh. We increase the number of inversion variables by a factor of eight for both the registration and the tumor inversion solver; we increase the spatial resolution starting from $N_i = 64$ points per dimension, to $N_i = 128$ and $N_i = 256$, and, accordingly, we choose the number $n_{\boldsymbol{p}}$ of tumor parameters to be $8, 64$, and 512 respectively. After termination of the parameter-continuation scheme for the regularization parameter of the registration, we perform an additional number of 20 Picard iterations. We report the relative objective function value $\|\mathcal{J}\|_{rel}$ and the relative gradient norm $\|\boldsymbol{g}\|_{rel}$ of the coupled problem in (6.4) for every iteration. Additionally, the relative norm of the update $e_{c0,L^2,rel}$ and $e_{\boldsymbol{v},L^2,rel}$ of the inversion variables $\boldsymbol{p}$ and $\boldsymbol{v}$ respectively is given for every iteration. Results previously published in [Sch].*

		$N_i = 64$, $n_{\boldsymbol{p}} = 8$				$N_i = 128$, $n_{\boldsymbol{p}} = 64$				$N_i = 256$, $n_{\boldsymbol{p}} = 512$			
It	β_v	$\|\mathcal{J}\|_{\text{rel}}$	$\|\boldsymbol{g}\|_{\text{rel}}$	$e_{c0,L^2,rel}$	$e_{\boldsymbol{v},L^2,rel}$	$\|\mathcal{J}\|_{\text{rel}}$	$\|\boldsymbol{g}\|_{\text{rel}}$	$e_{c0,L^2,rel}$	$e_{\boldsymbol{v},L^2,rel}$	$\|\mathcal{J}\|_{\text{rel}}$	$\|\boldsymbol{g}\|_{\text{rel}}$	$e_{c0,L^2,rel}$	$e_{\boldsymbol{v},L^2,rel}$
init	–	1.00	1.00	–	–	1.00	1.00	–	–	1.00	1.00	–	–
1	1	6.67E−1	9.42E−3	–	–	6.73E−1	2.48E−3	–	–	6.67E−1	8.76E−4	–	–
2	1E−1	5.84E−1	5.37E−4	–	–	5.88E−1	2.27E−4	–	–	5.84E−1	7.58E−5	–	–
3	1E−2	3.08E−1	2.42E−4	8.08E−3	1.01E+1	3.08E−1	1.88E−4	2.66E−2	6.02	3.08E−1	6.97E−5	2.28E−2	4.97
4	1E−2	3.03E−1	1.77E−4	2.48E−3	1.51	3.04E−1	1.01E−4	2.35E−2	2.48	3.03E−1	4.37E−5	2.55E−2	2.94
5	1E−2	3.02E−1	1.44E−4	1.34E−3	2.60E−2	3.03E−1	6.20E−5	7.16E−3	1.33E−1	3.02E−1	2.62E−5	1.00E−2	1.48E−1
6	1E−2	3.01E−1	1.18E−4	8.72E−4	1.43E−2	3.02E−1	4.66E−5	2.51E−3	7.70E−2	3.01E−1	1.81E−5	3.61E−3	9.79E−2
7	1E−2	3.01E−1	9.64E−5	5.61E−4	7.25E−3	3.02E−1	3.78E−5	1.29E−3	4.61E−2	3.01E−1	1.41E−5	1.63E−3	6.43E−2
8	1E−2	3.00E−1	7.91E−5	3.62E−4	4.37E−3	3.02E−1	3.12E−5	8.04E−4	2.70E−2	3.00E−1	1.15E−5	9.29E−4	3.96E−2
9	1E−2	3.00E−1	6.50E−5	2.38E−4	3.30E−3	3.02E−1	2.59E−5	5.29E−4	1.61E−2	3.00E−1	9.51E−6	5.89E−4	2.48E−2
10	1E−2	3.00E−1	5.35E−5	1.57E−4	2.71E−3	3.02E−1	2.15E−5	3.54E−4	9.98E−3	3.00E−1	7.89E−6	3.89E−4	1.58E−2
11	1E−2	3.00E−1	4.40E−5	1.05E−4	2.31E−3	3.02E−1	1.79E−5	2.39E−4	6.42E−3	3.00E−1	6.56E−6	2.62E−4	1.02E−2
12	1E−2	3.00E−1	3.63E−5	6.99E−5	1.90E−3	3.02E−1	1.49E−5	1.63E−4	4.22E−3	3.00E−1	5.47E−6	1.78E−4	6.75E−3
13	1E−2	3.00E−1	2.99E−5	4.69E−5	1.58E−3	3.02E−1	1.24E−5	1.12E−4	2.09E−3	3.00E−1	4.55E−6	1.21E−4	4.54E−3
14	1E−2	3.00E−1	2.46E−5	3.16E−5	1.30E−3	3.02E−1	1.04E−5	7.63E−5	3.09E−3	3.00E−1	3.80E−6	8.33E−5	3.10E−3
15	1E−2	3.00E−1	2.03E−5	2.13E−5	1.07E−3	3.02E−1	8.64E−6	5.26E−5	1.35E−3	3.00E−1	3.17E−6	5.74E−5	2.19E−3
16	1E−2	3.00E−1	1.67E−5	1.44E−5	8.85E−4	3.02E−1	7.21E−6	3.63E−5	1.71E−3	3.00E−1	2.64E−6	3.96E−5	1.60E−3
17	1E−2	3.00E−1	1.38E−5	9.73E−6	7.29E−4	3.02E−1	6.05E−6	2.51E−5	9.31E−4	3.00E−1	2.21E−6	2.74E−5	1.20E−3
18	1E−2	3.00E−1	1.14E−5	6.59E−6	6.01E−4	3.02E−1	5.08E−6	1.59E−5	7.94E−4	3.00E−1	1.86E−6	1.89E−5	8.04E−4
19	1E−2	3.00E−1	9.37E−6	4.47E−6	4.95E−4	3.02E−1	4.30E−6	1.12E−5	6.62E−4	3.00E−1	1.57E−6	1.16E−5	6.50E−4
20	1E−2	3.00E−1	7.73E−6	3.03E−6	4.08E−4	3.02E−1	3.64E−6	7.61E−6	5.22E−4	3.00E−1	1.34E−6	8.04E−6	5.29E−4
21	1E−2	3.00E−1	6.37E−6	2.06E−6	3.36E−4	3.02E−1	3.09E−6	5.23E−6	4.27E−4	3.00E−1	1.14E−6	5.69E−6	4.42E−4
22	1E−2	3.00E−1	5.26E−6	1.40E−6	2.77E−4	3.02E−1	2.62E−6	3.71E−6	3.59E−4	3.00E−1	9.63E−7	4.08E−6	3.74E−4
23	1E−2	3.00E−1	4.35E−6	6.45E−7	2.29E−4	3.02E−1	2.23E−6	1.91E−6	3.05E−4	3.00E−1	8.17E−7	2.10E−6	3.17E−4

For both solution strategies, we use the following, common setting. The target value for the regularization parameter β_v in the continuation scheme is set to 1E−2. This choice differs from the real brain cases and from Tab. 7.1. The higher regularization for $\boldsymbol{v}$ is justified due to few geometric structures in the smooth SIN test case input data. For the mesh-independence study, we consider three mesh resolutions in which we increase the resolution for both $\boldsymbol{v}$ and $\boldsymbol{p}$ by a factor of eight. In space, we use 64, 128, and 256 unknowns per dimension. For the parametrization of the tumor initial condition we employ a Gaussian grid of size 8, 64, and 512, respectively. For the latter, we use a spacing of $\delta = 2\sigma$.

Results. We report results for the convergence of the full-objective moving-patient Picard scheme in Tab. 7.16 and Fig. 7.17. We monitor the relative (to the initial value) reduction in the objective function value $\|\mathcal{J}\|_{\text{rel}}$, and the trend of the relative (to the initial value) reduced gradient $\|\boldsymbol{g}\|_{\text{rel}}$ of the coupled problem formulation (6.4). We furthermore report the relative norm $e_{c0,L^2,rel}$ and $e_{\boldsymbol{v},L^2,rel}$, respectively of the update for the inversion variables. In Fig. 7.17 we compare the trend of $\|\boldsymbol{g}\|_{\text{rel}}$ for the three different mesh resolutions. In particular, we also compare the convergence properties and gradient trends for the exact block-Newton scheme (a) with the solution strategy based on inexact solves (b). We report the convergence histories for the registration and tumor inversion sub-component solvers, respectively for both solution strategies in Fig. 7.18.

Observations. The most important observation from Tab. 7.16 and Fig. 7.17 is that our method is *mesh-independent*. For example, we observe that for all resolutions the inversion variables $\boldsymbol{p}$ and $\boldsymbol{v}$ have essentially converged after seven iterations in Tab. 7.16. For the block-Newton Picard iteration

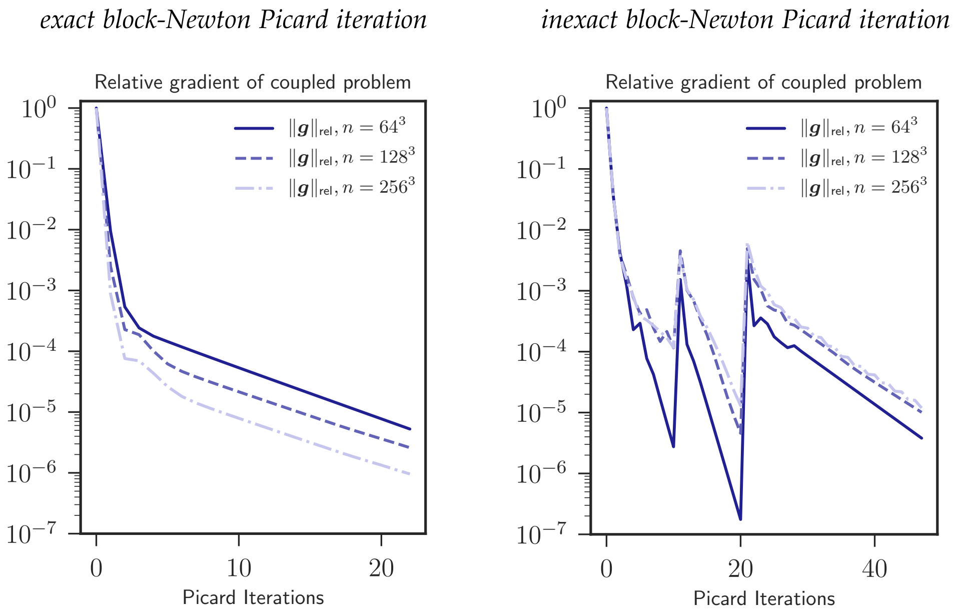


FIGURE 7.17 *Sinusoidal test case. We report the trend of the reduced gradient of the coupled problem formulation (6.4) as observed for the solution of the* ***sinusoidal analytic tumor / analytic velocity (SIN)*** *test case, using the (i) exact block-Newton Picard iteration (left), and the (ii) Picard iteration based on inexact solves (right), for three different mesh sizes. For the block-Newton scheme, exact numbers are reported in Tab. 7.16. The tree different curves correspond to the three different spatial resolutions for the velocity and tumor parameters. We use* $n \in \{64^3, 128^3, 256^3\}$ *points per dimension for the velocity and analogously* $n_p \in \{8, 64, 512\}$*. For both solution strategies, 20 Picard iterations are preformed after termination of the parameter-continuation scheme for the regularization parameter of the registration solver.*

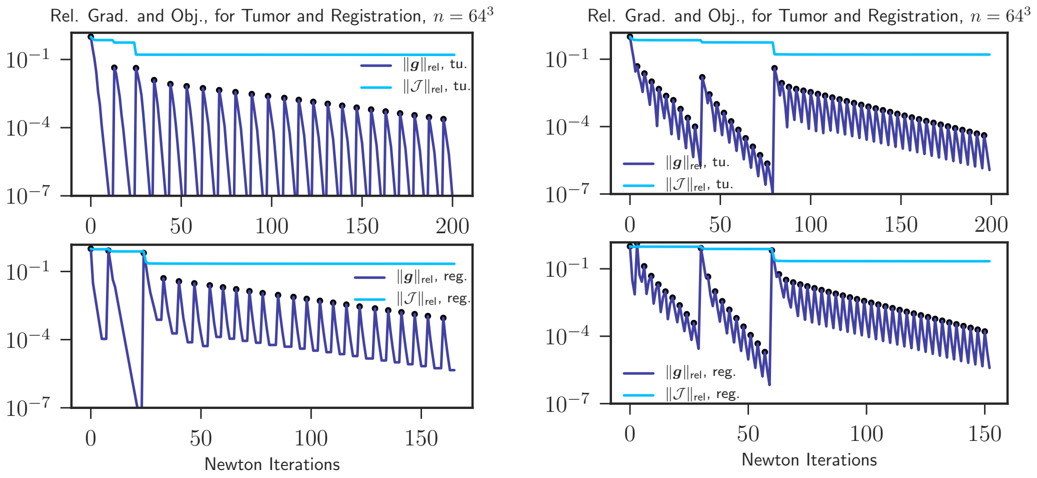


FIGURE 7.18 *Sinusoidal test case. We report the trend of the (relative) reduced gradients and (relative) objective function values of the sub-component solvers – tumor inversion (top) and registration (bottom) – as a function of Newton iterations for the coupled moving-patient Picard solution of the* ***sinusoidal analytic tumor / analytic velocity (SIN)*** *test case. We employ the (i) exact block-Newton Picard iteration (left), and the (ii) Picard iteration based on inexact solves (right). The black dots indicate Picard iterations; the large jumps in the block on the right hand side reflect the reduction of the regularization weight for the registration within the parameter-continuation scheme. We report results for the coarsest mesh using a spatial resolution of* $n = 64^3$ *and* $n_p = 8$ *Gaussian basis functions. The trends for the meshes are very similar. For both solution strategies, 20 Picard iterations are preformed after termination of the parameter-continuation scheme for the regularization parameter of the registration solver.*

scheme with exact sub-component solves, we observe rapid convergence until a relative gradient reduction of approximately five orders of magnitude, and convergence does not slow down if the number of unknowns is increased, as would be the case for a steepest descent method. After this point, convergence slows down, yet does not stagnate. A Newton method would maintain rapid convergence. Note, however, that for the second order Newton convergence rate the discretized gradient and the discretized objective function need to be consistent. Due to our optimize-then-discretize approach and the semi-Lagrangian time stepping for the pure advection steps, this is not the case.

A second important observation is, that the alternative solution strategy based on inexact solves is capable of reducing the reduced-gradient of the coupled scheme even further than the block-Newton based on exact solves, and, in particular, reduces oscillations. This can be observed from inspection of Fig. 7.18: The mutually induced jumps in the gradient norms of the sub-component solvers after every Picard iteration (black dots) are much less pronounced for the inexact scheme on the right than for the exact scheme on the left. We explain this feature by the stronger coupling of the sub-components, and more frequent exchange of information. Note also, that for the given setting, both solution strategies perform the same number of Newton iterations.

Conclusion *We conclude that our modular Picard iteration scheme features mesh-independent linear convergence. Although it does not perform as well as Newton's method, we observe rapid convergence rates until we reach gradient values that are significantly lower then what is required and feasible (due to discretization and modeling errors) in real clinical cases.*

Furthermore, the preliminary convergence considerations for the solution strategy based on inexact sub-component solutions seem to be very promising and worth further investigation. In particular, it reduces the mutually influenced oscillations in the gradients of the sub-components after every Picard iteration significantly compared to the block-Newton Picard iteration with an "exact" or more accurate solution of the sub-components.

7.4.4 Grid-Continuation

Using grid-continuation as described in §6.3.3, we start our optimization with a very coarse spatial resolution and refine this gradually until we reach the target resolution.

Purpose. We study the grid-continuation scheme with respect to (*i*) its convexifying properties and quality of the obtained final result, and (*ii*) the reduction in runtime observed when applying the scheme as opposed the full-resolution Picard-scheme . In particular, we want to find out if the grid-continuation scheme is able to retain full-resolution quality of the final solution, and, what the influence of the particular instantiation of the scheme on the obtained solution is.

Setup. We consider three different schemes, outlined in Tab. 7.17. We present results for the RTRV test case using actual clinical data; details are given in §7.1.1. We use the same tolerances, solver settings and regularization schemes as in §7.2.4 on all levels. We re-sample a hierarchy of different (increasingly coarse) image resolutions from the full-resolution representation ($n = 256^3$) of the observation patient data and the atlas brain. Starting from a coarse spatial representation ($n = 64^3$) of the data, we consecutively solve registration and tumor inversion on increasingly refined resolutions. The respective solvers are warm-started with solutions $P\boldsymbol{v}_{\ell-1}$ and $\boldsymbol{p}_{\ell-1}$ from the previous level (where P is a prolongation operator). We consider two alternatives, (*i*) warm-start only for the

TABLE 7.17 *Grid-continuation schemes. We consider three different schemes, intertwining grid-continuation and parameter-constinuation. Scheme 1 and Scheme 2 only differ in the number j_{final} of Picard iterations on the finest resolution with smallest regularization parameter.*

		grid-cont-I				grid-cont-II				grid-cont-III			
Level	n	β_v^0	β_v^{lo}	β_p	j_{final}	β_v^0	β_v^{lo}	β_p	j_{final}	β_v^0	β_v^{lo}	β_p	j_{final}
$L1$	64^3	1	1E−2	2.50E−3	0	1	1E−2	2.50E−3	0	1	1E−1	2.50E−3	0
$L2$	128^3	1E−2	1E−4	2.50E−4	0	1E−2	1E−4	2.50E−4	0	1E−1	1E−3	2.50E−4	0
$L3$	256^3	1E−4	1E−4	2.50E−5	1	1E−4	1E−4	2.50E−5	3	1E−3	1E−4	2.50E−5	1

registration velocity $\boldsymbol{v}$,[23] and (*ii*) warm-start for both inversion variables, i.e., use $\boldsymbol{v}_{\ell-1}$ and $\boldsymbol{p}_{\ell-1}$ from the previous level . We use bi-cubic interpolation for the prolongation P of $\boldsymbol{v}_{\ell-1}$ to the finer level. For this experiment we only consider the tumor diameter-based, regular-grid selection mode (bbox mode) for the Gaussian basis functions. The adaptive approach is also conceivable. However, in order to allow for warm-starts of the tumor inversion solver using solutions $\boldsymbol{p}_{\ell-1}$ from coarser levels, the same set of basis functions needs to be selected for all levels.

The different spatial resolutions allow for different degrees of parallelization. We execute the moving-patient Picard iteration scheme separately[24] on each level, steered by the input parameters β_v^0 and β_v^{lo} for the embedded parameter-continuation scheme (see §6.3.3). To ensure the inherently sequential character of the Picard iteration algorithm, dependencies between the separate execution runs are defined. This approach allows us to exploit the full degree of parallelism on each spatial level without having idling processes. More precisely, for the presented experiments we use three levels $n \in \{64^4, 128^3, 256^3\}$ and solve the coupled Picard iteration scheme using 16 MPI tasks (one node) for level one with resolution $n_{\ell_1} = 64^3$, 64 MPI tasks (on three nodes) for level 2 ($n_{\ell_2} = 128^3$), and 256 MPI tasks (on 11 nodes) for the original resolution $n_{\ell_3} = 256^3$ (level 3).

Results. We present quantitative results for the grid-continuation scheme on real data (RTRV test case, patient ID's AAMH, AAAC) in Tab. 7.18 and Tab. 7.19 and compare them to the full-resolution scheme with respect to runtime and quality of the registration and tumor reconstruction. We analyze runtime and quality of the final solution for different continuation schemes from Tab. 7.17 in Tab. 7.19. Qualitative results for the multi-level solution of the two considered patient ID's are given in Fig. 7.19.

Observations. Comparing full-resolution scheme results and grid-continuation scheme-I (see Tab. 7.17) for patient ID AAMH (Tab. 7.19), we observe a significant speedup of a factor of 5 for the overall consumed cpu-h (i.e., elapsed wall-time times number of occupied MPI tasks), subdivided into a factor of 10.4 speedup for the tumor solver and a factor of 2.5 speedup for the registration solver, resulting from warm-starts for the respective solvers using the solutions $\boldsymbol{p}_{\ell-1}$ and $P\boldsymbol{v}_{\ell-1}$ from the previous level. The obtained registration and reconstruction quality of the target data is slightly worse for the multi-level scheme; we observe a relative L_2-error of 3.82E−1 for the data mismatch of the brain tissue labels compared to an error of 3.45E−1 for the full-resolution scheme. The quality of the tumor reconstruction is very similar with relative L_2-error values of 2.09E−1 and 1.95E−1 for the grid-continuation and full-resolution scheme, respectively. The slightly worse similarity of the probability maps for the brain tissue labels is due to the fact that we only perform one registration solve on the finest level with the original target data resolution; differences are manifested by details hidden on coarser representations of the target data. From Tab. 7.19 we observe, that the data-similarity is further improved if more registration solves are performed on the finest

[23] In this case, tumor inversion is solved in an initialization phase to obtain an initial guess for $\boldsymbol{p}$

[24] For each level, a different job is submitted, exploiting the maximal degree of parallelism on the respective level. Dependencies between jobs are defined to ensure the inherently sequential character of the algorithm.

TABLE 7.18 *Grid-continuation. Results for the* ***real tumor / real velocity (RTRV)*** *test case, patient ID* ***AAMH****, ground truth: (ρ N/A,* $\mathbf{k}$ *N/A,* $\mathbf{p}$ *N/A,* $\mathbf{v}$ *N/A); using real clinical input data. We report quantitative results per Picard iteration for the grid-continuation scheme I outlined in Tab. 7.17 using* $\mathbf{v}$*- and* $\mathbf{p}$*-warm-start mode and compare it to the full-resolution scheme solution. The table shows the (summed) L_2-mismatch of brain tissue labels μ_{B,L^2} and tumor μ_{T,L^2} and the mean Dice coefficient for brain tissue $DICE_B$ and tumor $DICE_T$. Timings for the Picard iteration scheme T^{tot}, the tumor inversion T^{tu}_{inv} and the registration T^{reg}_{inv} (given in seconds and cpu-h) are with respect to parallel execution on* 11 *nodes using* 256 *MPI tasks on the finest level with $n_{\ell_3} = 256^3$,* 3 *nodes using* 64 *MPI tasks on level 2 with $n_{\ell_2} = 128^3$, and* 1 *node using* 16 *MPI ranks on the coarsest level with $n_{\ell_1} = 64^3$.*

			grid-cont-I v-/p-warm-start					full-resolution				
	it_{sibia}	β_v	μ_{B,L^2}	DICE$_B$	μ_{T,L^2}	DICE$_T$	T^{it} [s]	Itr.	β_v	μ_{B,L^2}	μ_{T,L^2}	T^{it} [s]
Level 1	init	–	1.00	5.73E−1	1.00	0.00	–	init	–	5.75E−1	1.00	–
	1	1	9.37E−1	5.87E−1	1.00	0.00	1.59E+1	1	1	9.59E−1	1.00	9.80E+1
	2	1E−1	7.12E−1	6.53E−1	3.78E−1	8.46E−1	6.33	2	1E−1	7.42E−1	3.90E−1	4.55E+1
	3	1E−2	5.23E−1	7.43E−1	3.13E−1	9.07E−1	8.22	3	1E−2	5.05E−1	3.13E−1	9.59E+1
	4	1E−2	5.13E−1	7.47E−1	3.03E−1	9.10E−1	2.85	4	1E−3	3.91E−1	2.52E−1	1.18E+2
Level 2	init	–	1.00	5.77E−1	4.64E−1	0.00	–	5	1E−4	3.60E−1	2.20E−1	1.15E+2
	1	1E−2	4.80E−1	7.76E−1	3.12E−1	9.04E−1	2.56E+1	6	1E−4	3.51E−1	2.05E−1	8.99E+1
	2	1E−3	4.09E−1	8.10E−1	2.54E−1	9.35E−1	2.27E+1	7	1E−4	3.45E−1	1.95E−1	6.63E+1
	3	1E−4	3.86E−1	8.20E−1	2.32E−1	9.44E−1	2.21E+1					
	4	1E−4	3.78E−1	8.24E−1	2.22E−1	9.47E−1	8.06					
Level 3	init	–	1.00	5.75E−1	5.15E−1	0.00	–					
	1	1E−4	4.06E−1	8.15E−1	2.25E−1	9.42E−1	7.11E+1					
	2	1E−4	3.82E−1	8.26E−1	2.09E−1	9.48E−1	3.42E+1					
	TOTAL	1E−4	3.82E−1	8.26E−1	2.09E−1	9.48E−1	5.42E+2	TOTAL	1E−4	3.45E−1	1.95E−1	2.68E+3

	T^{it} [cpu-h]	T^{tu}_{inv} [cpu-h]	T^{reg}_{inv} [cpu-h]	T^{it} [cpu-h]	T^{tu}_{inv} [cpu-h]	T^{reg}_{inv} [cpu-h]
Acc. time in cpu-h	5.42E+2	1.72E+2	3.35E+2	2.68E+3	1.79E+3	8.32E+2

TABLE 7.19 *Quantitative results for grid-continuation. Results for the* ***AAAC*** *and the* ***AAMH*** *patient* ***real tumor / real velocity (RTRV)*** *test case, ground truth: (ρ N/A,* $\mathbf{k}$ *N/A,* $\mathbf{p}$ *N/A,* $\mathbf{v}$ *N/A); using real clinical input data. We compare different grid-continuation schemes (cf. Tab. 7.17 with* $\mathbf{v}$*-warm-start and* $\mathbf{v}$*-/*$\mathbf{p}$*-warm-start) against the full-resolution result with respect to runtime (for the entire Picard iteration T^{tot}, the tumor inversion T^{tu}_{inv}, and the registration T^{reg}_{inv}, respectively; in cpu-h, i.e., elapsed wall-time in [h] multiplied by the number of occupied MPI tasks, accumulated over all levels) and reconstruction quality μ_{B,L^2} and μ_{T,L^2} as the L_2-mismatch for the probability maps of brain tissue labels and tumor, respectively. Timings are with respect to parallel execution on* 11 *nodes using* 256 *MPI tasks on the finest level with $n_{\ell_3} = 256^3$,* 3 *nodes using* 64 *MPI tasks on level 2 with $n_{\ell_2} = 128^3$, and* 1 *node using* 16 *MPI ranks on the coarsest level with $n_{\ell_1} = 64^3$. We give the speedup of the grid-continuation schemes with respect to the full-resolution execution of the coupled Picard scheme S^{tot}, the tumor inversion S^{tu}, and the registration solve S^{reg}.*

		Scheme	warm-start	μ_{B,L^2}	μ_{T,L^2}	T^{tot} [cpu-h]	T^{tu}_{inv} [cpu-h]	T^{reg}_{inv} [cpu-h]	S^{tot}	S^{tu}	S^{reg}
AAAC	#1	**full-resolution**	–	3.36E−1	2.45E−1	2.10E+3	1.38E+3	6.79E+2	1.00 ×	1.00 ×	1.00 ×
	#2	**grid-cont-I**	v	3.87E−1	3.08E−1	7.64E+2	3.48E+2	3.79E+2	2.75 ×	3.95 ×	1.79 ×
	#3	**grid-cont-I**	v/p	3.80E−1	2.73E−1	4.98E+2	1.34E+2	3.29E+2	4.22 ×	10.30 ×	2.06 ×
	#4	**grid-cont-II**	v/p	3.60E−1	2.53E−1	1.15E+3	4.60E+2	6.47E+2	1.83 ×	3.00 ×	1.05 ×
	#5	**grid-cont-III**	v/p	3.70E−1	2.58E−1	9.86E+2	3.46E+2	5.99E+2	2.13 ×	3.98 ×	1.13 ×
AAMH	#6	**full-resolution**	–	3.45E−1	1.95E−1	2.68E+3	1.79E+3	8.32E+2	1.00 ×	1.00 ×	1.00 ×
	#7	**grid-cont-I**	v	3.91E−1	2.19E−1	9.02E+2	5.64E+2	3.02E+2	2.97 ×	3.18 ×	2.75 ×
	#8	**grid-cont-I**	v/p	3.82E−1	2.09E−1	5.42E+2	1.72E+2	3.35E+2	4.94 ×	10.39 ×	2.48 ×

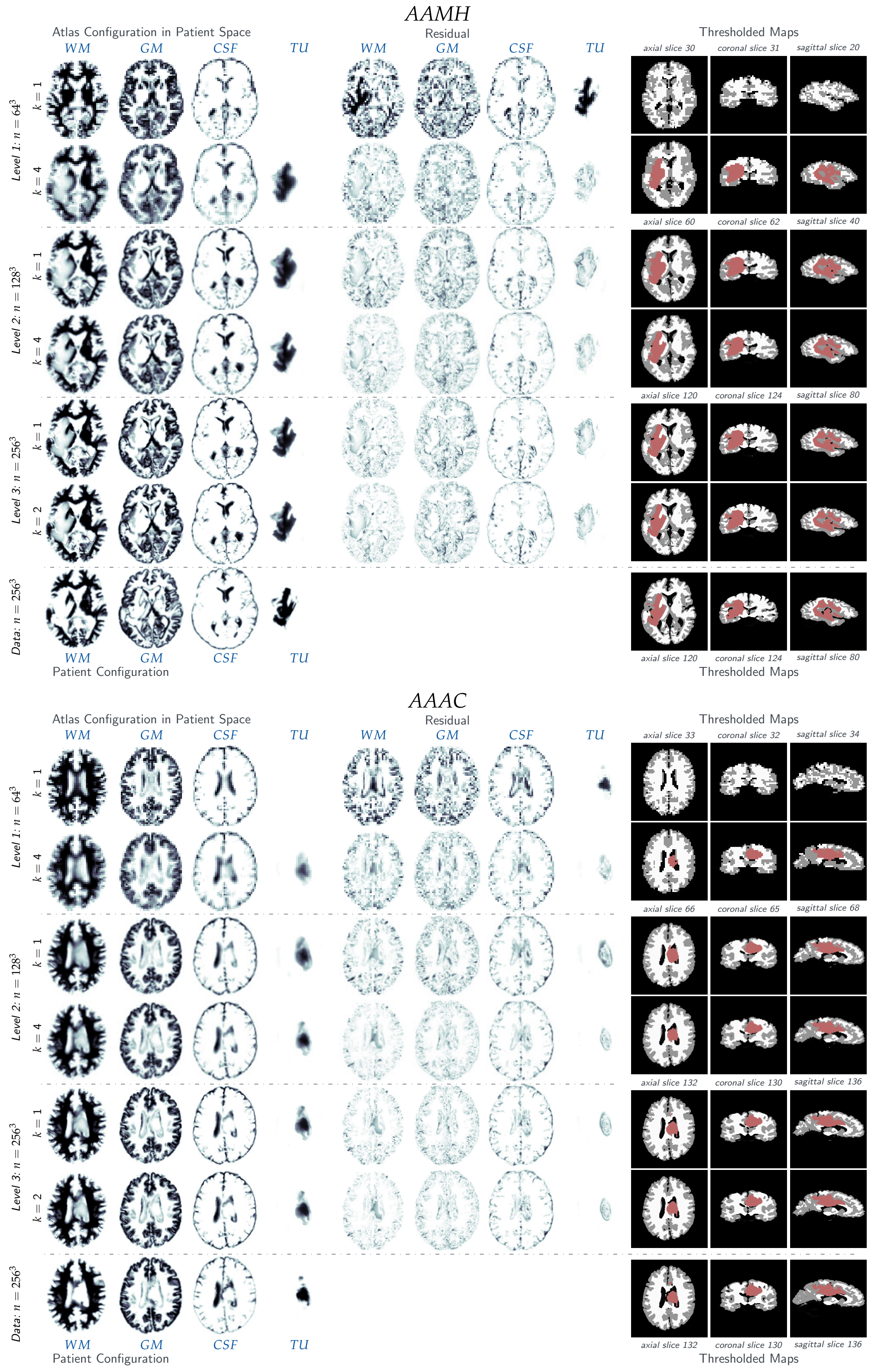


FIGURE 7.19 *Qualitative results for grid-continuation (scheme I) with warm-starts for* $\mathbf{p}$ *and* $\mathbf{v}$ *for the* ***real tumor / real velocity (RTRV)*** *test case, ground truth: (ρ N/A,* $\mathbf{k}$ *N/A,* $\mathbf{p}$ *N/A,* $\mathbf{v}$ *N/A); patient ID's* ***AAMH*** *and* ***AAAC****.*

level (compare run #2, grid-cont-I with run #3, grid-cont-II for AAAC patient). Performing additional 3 registration solves on full-resolution (scheme-II, Tab. 7.17) yields comparable data similarity, with a resulting speedup factor of roughly 2 (compared to 4.2 for scheme-I). Further, using solutions from coarser levels as initial guess for both, $\boldsymbol{v}$ and $\boldsymbol{p}$ not only speeds up the overall runtime by a factor of 2 compared to only using warm-starts for the registration velocity $\boldsymbol{v}$, but also improves the accuracy of the final solution (compare runs #2 and #3, and runs #7 and #8, respectively). Looking at runs #5, #4 and #3, we observe that using different continuation schemes, i.e., different interlacing/stepping of grid- and parameter-continuation affects the quality of the final result. The variations are, however, marginally.

Conclusion *From these experiments, we conclude that interlacing grid-continuation with parameter-continuation has the potential to significantly speedup the overall time to solution: We achieve speedup factors of* 4.94 *and* 4.22 *for ID AAMH and AAAC, respectively. The scheme converges to a similar solution as the full-resolution scheme, exhibiting slightly worse data similarity for the registration result. Performing more expensive full-resolution Picard iterations (registration solves) improves the reconstruction quality as details and image features might be hidden on coarser resolutions.*

Generally speaking, runtime and solution quality depend on the employed grid-continuation scheme and its interlacing with the parameter-continuation scheme. For our experiments, we obtain relatively small variations/discrepancies in the final solution and speedup factors between 2 *and* 5 *when compared to the full-resolution scheme. Visual inspection of qualitative results in Fig. 7.19 reassures overall good registration and pathology reconstruction quality. Determination of an optimal and robust scheme requires more thorough numerical analysis and application to further individual (real data) cases.*

7.4.5 Performance of Biophysical Inversion Solver: Gauß-Newton versus Quasi-Newton

In this section we study the potential of quasi-Newton LBFGS as described in §6.4.5 as a faster alternative for Gauß-Newton-Krylov for various settings: (*i*) stand-alone tumor inversion solver, (*ii*) joint inversion approach for synthetic tumors (ATAV), and (*iii*) joint inversion approach for real clinical imaging data of glioma patients. The low dimensionality of the Hessian matrix motivates to use quasi-Newton methods over Gauß-Newton-Krylov methods, which have significantly less computational cost per iteration and, thus, might outperform GNK despite the slower convergence. Critical for quasi-Newton (esp. LBFGS) is a low-rank property of the system Hessian as well as a good initial guess for the approximation of the inverse Hessian.

Performance of Quasi-Newton LBFGS method for Standalone Tumor Solver.

Purpose. We investigate whether quasi-Newton is able to outperform GNK in terms of overall runtime, despite excellent convergence properties of the latter for the considered problem (convergence within 6 iterations). Further, we study the LBFGS performance and convergence properties for our specific tumor problem; in particular, we wish to investigate (*i*) the influence of the number μ of used difference vectors for the LBFGS update, and (*ii*) the effect of different choices for the initial guess of the inverse Hessian approximation, in particular the generic, built-in PETSc/TAO approaches, compare §2.3.1.

Setup. We consider a purely synthetic test case for the tumor inversion sub-component. We assume the true healthy patient to be given, i.e., no registration is involved. The target tumor is

TABLE 7.20 *Performance of LBFGS(m) vs. Gauß-Newton-Krylov for a synthetic stand-alone tumor problem for different numbers of stored vectors $\mu \in \{6, 10, 14, 18, 22, 26, 30\}$ for the limited-memory BFGS update. We use PETSc/TAO's Broyden scaling of the identity matrix as an initial guess for the inverse Hessian approximation. An additional scalar re-scaling is performed, using up to $r \in \{5, 10, 15, 30\}$ difference vectors to solve for the scaling weight ω in a least-squares sense.*

Method	r	it_N	$\mathcal{J}$	$\|g_p\|_{\text{rel}}$	μ_{T,L^2}	T^{tu}_{inv} [s]	S^{tu}
GNK		6	8.80E−2	8.33E−4	1.31E−2	4.50E+1	1.00×
LBFGS(6)	5	41	7.90E−2	9.67E−4	1.06E−2	4.78E+1	0.94×
LBFGS(10)	5	35	7.85E−2	9.75E−4	1.05E−2	4.09E+1	1.01×
LBFGS(14)	5	36	**7.33E−2**	**6.71E−4**	**8.63E−3**	4.08E+1	1.10×
LBFGS(18)	5	32	7.95E−2	8.85E−4	1.08E−2	3.70E+1	1.22×
LBFGS(22)	5	31	7.75E−2	9.06E−4	1.01E−2	3.61E+1	1.25×
LBFGS(26)	5	**28**	8.11E−2	9.22E−4	1.12E−2	**3.26E+1**	**1.38** ×
LBFGS(30)	5	28	8.10E−2	8.98E−4	1.12E−2	3.26E+1	1.37×
LBFGS(30)	10	34	8.51E−2	9.42E−4	1.24E−2	3.99E+1	1.12×
LBFGS(30)	15	39	7.75E−2	7.56E−4	1.01E−2	4.54E+1	0.99×
LBFGS(30)	30	50	7.55E−2	8.43E−4	9.48E−3	5.81E+1	0.77×

artificially grown in the healthy patient geometry, using our reaction-diffusion model with the parameters $\rho_f = 10$, $\rho_w = 1$, $\rho_g = 0$, $k_f = 1.00\text{E}{-2}$, $k_w = 1$, $k_g = 0$, $n_{\boldsymbol{p}} = 125$, $\boldsymbol{p} = \boldsymbol{p}^\star$ with $p^\star_{62} = 0.9$ and $p^\star_{61} = 0.2$, $\boldsymbol{v} = \boldsymbol{0}$ $\sigma = {}^{2\pi}/_{10}$, $\delta = 1.5\sigma$. We use GNK and LBFGS to solve the non-linear system without bounding the maximum number of Newton and Krylov iterations. We use PETSc/Tao's built-in Broyden scaling with applied re-scaling for the initial inverse Hessian approximation, using up to 5 vectors in the multi-secant equation to compute the Broyden factors and the re-scaling factor, respectively. In addition, we examine the influence of the number of vectors r used in the re-scaling for the Broyden scaling approach in three runs with higher $r \in \{10, 15, 30\}$.

Results. Numerical results for solving a fully synthetic, tumor only inverse problem using Gauß-Newton-Krylov and LBFGS quasi-Newton variants to solve the resulting non-linear optimality system are given in Tab. 7.20.

Observation. For the considered problem, the GNK solver shows excellent convergence, reducing the gradient by three orders of magnitude within 6 non-linear iterations. Despite requiring a much larger number of iterations, the LBFGS quasi-Newton solver achieves a comparable or even reduced runtime (speedup by a factor of roughly 1.4). The performance of the quasi-Newton method, however, strongly depends on the number of stored vectors utilized in the LBFGS Hessian update formula—a problem dependent tuning parameter. We have studied this effect for similar methods in the context of transient FSI simulations in part I of this thesis. Unlike for the transient case considered there, the optimal number of stored vectors here is solely determined by the increasing memory footprint and computational cost to compute the update. The method's convergence rate does not deteriorate if more vectors are stored as for the given optimization problem (tumor standalone) there is no outdated or contradicting information.

We furthermore studied different choices for the initial guess of the inverse Hessian, as described in §2.3.1 in Chapter 2 and §6.4.5 in Chapter 6. We briefly summarize our findings. The simplest approach of using either $\boldsymbol{M}^k_{prev} = \gamma \boldsymbol{I}$ for the Hessian or $\boldsymbol{M}^{-1,k}_{prev} = \gamma \boldsymbol{I}$ with $\gamma = 1$ for the inverse Hessian initial guess results in line-search failure and divergence after the first Newton-iteration. Solving a multi-secant least-squares problem (2.48) for a scaling parameter γ significantly improves the quality of the initial guess and the LBFGS solver converges for the considered test problem. Using PETSc/Tao's built-in Broyden diagonal scaling of the identity further improved the convergence properties and showed best overall performance. Tab. 7.20 shows numerical results obtained using the Broyden diagonal scaling method with an additional re-scaling with the parameter ω, obtained from solving (2.49) using up to r difference vectors in the least-squares system; small numbers for r showed superior performance. Lastly, using $\boldsymbol{M}_{prev}^{-1,k} = \beta_{\boldsymbol{p}} \Phi^T \Phi$ as initial guess for the Hessian[25], i.e., incorporating complementary a priori information of the problem, and solve $\boldsymbol{M}_{prev}^{k} \hat{\boldsymbol{q}}_k = \boldsymbol{g}^k$ by PCG iterations also resulted in divergence due to line-search failure. From this we conclude, that using the constant part of the Hessian only, is not sufficient. Using a low-rank approximation of the non-constant part of the Hessian by sampling from $\boldsymbol{U}^{-1}\Phi$ as described in (6.43) is likely to resolve these issues and improve convergence; however, this is computationally more expensive and still involves the solution of a linear system (i.e., the method's complexity per iteration becomes similar to using GNK).

Conclusion *From this first simplistic experiment we conclude, that, if done right, the quasi-Newton LBFGS method has the potential to outperform the established, rapidly converging GNK solver in terms of overall runtime and computational cost. The number of used columns in the LBFGS update and the initial guess for the inverse Hessian are the most dominating convergence influencing factors. For the considered PDE constrained optimization problem, it is beneficial to store as many vectors as tenable with respect to memory consumption and computational cost per LBFGS update.*[26] *A good choice is the number of maximally allowed Newton iterations.*

Further, a good initial guess for the inverse Hessian is crucial and appropriate scaling is necessary for convergence. TAO's diagonal Broyden scaling with additional scalar re-scaling outperforms all other considered variants including approximations of the Hessian that incorporate a priori problem specific information. In particular using a small number of $r \leqslant 5$ difference vectors in the least squares-system to compute the Broyden factors and re-scaling parameter ω performs best. Replacing the GNK solver by the LBFGS quasi-Newton method for the considered problem yields only very insignificant speedups. For this fully synthetic toy problem GNK features excellent near-quadratic convergence; for more complicated cases, actual clinical data and the joint registration and biophysical inversion, the convergence rate for GNK drops. [27]*Under these considerations, the above LBFGS results are quite promising and the method is potentially more robust with respect to issues arising from more complicated use cases, in particular in the context of our joint inversion scheme. This is what we consider next.*

Performance of Quasi-Newton LBFGS for Joint Inversion using Synthetic Test Cases.

Purpose. Low rank approximability of the system Hessian as well as effective initial guesses are highly problem dependent. We study the suitability of quasi-Newton LBFGS for solving the tumor inversion problem for a synthetically grown target tumor using our model. Our main objective is the reduction of the overall time-to-solution for our joint inversion scheme. We study the influence of the number μ of used vectors in the LBFGS update and different approximations for the initial guess of the inverse Hessian. Preliminary tests with standalone tumor inversion and analysis of the Hessian matrix for perturbed observation data and initial guess $\boldsymbol{p}^0$ indicate that the Hessian changes moderately with these perturbations and information from inverse Hessian approximation from previous Picard iterations might be beneficial. Thus, in particular, we consider recycling of difference

[25] Note that $\beta_{\boldsymbol{p}} \Phi^T \Phi$ represents the invariant part of the Hessian if the regularizer $\|\Phi \boldsymbol{p}\|_{L_2(\Omega)}$ is employed. The full Hessian reads $\boldsymbol{H} = \beta_{\boldsymbol{p}} \Phi^T \Phi - \Phi^T \alpha(0)$, i.e., the missing part $\Phi^T \alpha(0)$ is dominated by the solution of the adjoint final-value problem and depends on the data.

[26] In particular, filtering techniques that maintain good conditioning of the least-squares system are not necessary

[27] This is caused by the appearance of non-physical negative values in the probability map for tumor cell distribution throughout the inversion process. The reason for this is the parametrization of the initial condition as superposition of Gaussian basis functions. Clipping of these values leads to a discrepancy between Hessian, gradient and objective function and is reflected in a degradation of the convergence speed for the Gauß-Newton-Krylov solver.

vectors from tumor inversion solves in previous Picard iterations (with varying target data), i.e., retaining vectors from converged solves from previous Picard iterations. We analyze convergence, cost, runtime and robustness of the methods.

Setup. For all conducted experiments, we use the moving-patient Picard iteration scheme described in §6.3. We furthermore use the fully synthetic (analytic tumor / analytic velocity (**ATAV**)) test case based on real brain geometries as described in §7.1.1. We use a resolution of $n_i = 128$, and ground truth parameters $\rho_f = 10$, $\rho_w = 1$, $\rho_g = 0$, $k_f = 1.00\text{E}{-2}$, $k_w = 1$, $k_g = 0$, $n_{\boldsymbol{p}} = 125$, $\boldsymbol{p} = \boldsymbol{p}^\star$ with $p^\star_{62} = 0.9$ and $p^\star_{61} = 0.3$, $\boldsymbol{v} = -\boldsymbol{v}^\star$, $\sigma = {}^{2\pi}/_{10}$, $\delta = 1.5\sigma$. For the non-linear solvers, we limit the number of Newton- and Krylov-iterations to $\text{maxit}_R = (50, 80)$ for the registration solver and $\text{maxit}_T = (50, 100)$ for the tumor inversion solver, respectively.

We consider the limited-memory BFGS (LBFGS) method for the tumor inversion solver and compare it against the Gauß-Newton-Krylov method. In particular, for LBFGS we investigate different numbers $\mu \in \{10, 30, 50, 70, 100\}$ of stored vectors and either (*i*) gather vectors only within each tumor inversion solve and reset the memory after convergence, referred to as $LBFGS(\text{maxit}_{T,N}, \mu, \text{ls-type})$, or (*ii*) retain (up to μ) stored vectors from previous tumor inversion solves across Picard iterations. The latter is denoted as $\text{LBFGS-r}(\text{maxit}_{T,N}, m, \text{ls-type})$. For quasi-Newton methods, Armijo line-search which guarantees sufficient decrease in the objective function is in general not sufficient; our numerical experiments confirm this. We therefore use a more expensive Moré-Thuente (mt) line-search method enforcing the Wolfe conditions. We use PETSc/TAO's built-in diagonal Broyden scaling of the identity matrix with an additional scalar re-scaling for the initial guess of the inverse Hessian, i.e., we use $\boldsymbol{M}^{-1}_{prev} = \omega\gamma\boldsymbol{I}$. We use up to 5 vectors in the leas-squares system to compute the scaling factors. For details, see §2.3.1 and [Ben03]. We run our experiments on 64 MPI ranks on 3 nodes of `HazelHen` using an Intel compiled `petsc-3.9.0` build.[28]

Results. We report results for the joint inversion using a limited-memory BFGS quasi-Newton solver for the tumor inversion in Tab. 7.21. Convergence histories for GNK and LBFGS are given in Fig. 7.21 in terms of gradient- and objective function value reduction plotted over wall clock time and the number of Newton interactions. Picard iterations of the joint solution scheme are indicated by black dots. Fig. 7.21 shows a similar plot of convergence histories for different number of stored vectors μ for the LBFGS update. We present more detailed monitoring of the tumor inversion solver such as required Newton iterations per inversion solve as well as gradient reduction, number of state and adjoint solves and the elapsed runtime across all Picard iterations for various settings in Tab. 7.22.

Observations. We focus on studying two factors that greatly influence the method's convergence properties: the number μ of vectors used for the LBFGS update, and if recycling of vectors for solves in Picard iterations $k \geqslant 2$ from previous tumor inversion solves is beneficial. The latter can be seen as an informed initial guess of the inverse Hessian. Consequently, we consider an LBFGS recycle (LBFGS-r) alternative, retaining $\mu - j_k \geqslant 0$ difference vectors from previous Picard iterations ($j_k \geqslant 0$ number of Newton-iterations for current solve). As can be seen from runs #2, #6 and #7, the recycle option reduces the accumulated time spent in the tumor inversion solver by about a factor of $2 - 3$ for solving the ATAV test case with our joint inversion scheme. Comparing LBFGS runs with reset and recycle strategy (runs #2 and #6), we observe a similar number of Newton iterations, however,

[28] The LBFGS recycle option has been added to PETSc/TAO with the help of Alp Dehner and is available in `petsc-3.9.0`. As of today there is no Cray optimized build available for this version of PETSc, which is why the absolute timings in this section are considerably higher than for the other experiments (where we used the Cray optimized `petsc-3.7.6.2` build).

TABLE 7.21 *Performance of Gauß-Newton-Krylov and quasi-Newton for the tumor inversion solver for* ***analytic tumor / analytic velocity with non-zero diffusion (ATAV-DIF)*** *test case; ground truth: ($\rho_f = 10$, $\rho_w = 1$, $\rho_g = 0$, $k_f = 1.00\text{E}{-2}$, $k_w = 1$, $k_g = 0$, $\boldsymbol{p} = \boldsymbol{p}^\star$, $\boldsymbol{v} = -\boldsymbol{v}^\star$). We report the final data L_2-mismatch after the joint inversion (moving-patient) for brain tissue labels μ_{B,L^2} and tumor probability maps μ_{T,L^2} as well as the (summed) number of Newton iterations it_N, the (summed) number of state and adjoint evaluations,and the (summed) timings for the joint inversion T^{tot}, tumor inversion T^{tu}_{inv}, and registration T^{reg}_{inv}, respectively in seconds. We give the speedup of the execution of tumor inversion S^{tu} and overall time-to-solution S^{tot} with respect to the Gauß-Newton-Krylov method. For Gauß-Newton-Krylov, we limit the number of Newton and Krylov iterations to $maxit_{N,T} = 50$ and $maxit_{K,T} = 100$, respectively, and use the cheaper Armijo line-search method. For LBFGS we investigate different numbers $\mu \in \{10, 30, 50, 70, 100\}$ of stored vectors and either (i) gather vectors only within each tumor inversion solve and reset the memory after convergence, or (ii) retain (up to μ) stored vectors from previous tumor inversion solves across Picard iterations. The latter is denoted as LBFGS-r(50,μ,ls). We further use a (more expensive) Moré-Thuente (mt) line-search method enforcing the Wolfe conditions as required for quasi-Newton methods. For the initial inverse Hessian approximation, we use PETSc/TAO's built-in Broyden scaling of the identity matrix using up to 5 vectors to compute the diagonal scaling factors, along with a re-scaling also using up to 5 vectors; for details see §2.3.1 and [Ben03].*

	Method(maxit$_N$, H, ls)	μ_{B,L^2}	μ_{T,L^2}	it_N	#state	#adj	T^{tot} [s]	T^{tu}_{inv} [s]	T^{reg}_{inv} [s]	S^{tu}	S^{tot}
#1	GNK(50, 100, armijo)	1.61E−1	7.01E−2	39	2077	1703	1.86E+4	1.79E+4	5.44E+2	1.00×	(1.00)×
#2	LBFGS(50, 50, mt)	1.59E−1	5.71E−2	125	205	205	1.73E+3	1.22E+3	4.56E+2	14.72×	(10.73)×
#3	LBFGS(50, 50, armijo)	1.59E−1	5.71E−2	128	435	138	2.00E+3	1.50E+3	4.46E+2	11.95×	(9.27)×
#4	LBFGS-r(50, 10, mt)	1.58E−1	5.55E−2	142	306	306	2.19E+3	1.69E+3	4.43E+2	10.62×	(8.50)×
#5	LBFGS-r(50, 30, mt)	1.59E−1	5.58E−2	132	243	243	1.89E+3	1.38E+3	4.54E+2	9.46×	(9.82)×
#6	LBFGS-r(50, 50, mt)	1.58E−1	5.61E−2	129	144	144	1.20E+3	6.98E+2	4.40E+2	25.68×	(15.55)×
#7	LBFGS-r(50, 70, mt)	1.59E−1	5.66E−2	85	108	108	1.13E+3	6.17E+2	4.52E+2	29.03×	(16.50)×
#8	LBFGS-r(50, 100, mt)	1.59E−1	5.51E−2	129	198	198	1.67E+3	1.19E+3	4.22E+2	15.12×	(11.16)×
#9	LBFGS-r(50, 50, armijo)	1.59E−1	5.57E−2	124	437	134	3.05E+3	2.48E+3	4.94E+2	7.22×	(6.10)×

TABLE 7.22 *Analysis of computational cost and convergence for Gauß-Newton-Krylov and quasi-Newton for the tumor inversion solver for* ***analytic tumor / analytic velocity with non-zero diffusion (ATAV-DIF)*** *test case. We consider the same test case setting and a subset of the solver variations from Tab. 7.21; for more details, see above. We report the objective function value and relative gradient for the tumor inversion sub-problem for every Picard iteration of the joint inversion scheme. Furthermore, the number of Newton iterations, the number of objective function and gradient evaluations and the number of state and adjoint solves is given, along with the reason for termination of the sub-component optimization in every Picard iteration.*

it_{sibia}	it_N	$\mathcal{J}_T$	$\Vert\boldsymbol{g_p}\Vert_{\text{rel}}$	#state	#adj	T^{tu}_{inv}	term.	it_N	$\mathcal{J}_T$	$\Vert\boldsymbol{g_p}\Vert_{\text{rel}}$	#state	#adj	T^{tu}_{inv}	term.
	GNK(50, 50, armijo) reference							initial	1.63E+3	1				
1	27	1.53E+2	9.97E−4	1572	1540	1.37E+4	conv.							
2	4	9.86E+1	2.17E−3	115	46	9.65E+2	ls failed							
3	4	3.58E+1	3.62E−3	124	51	1.07E+3	ls failed							
4	2	1.06E+1	4.20E−3	97	29	8.01E+2	ls failed							
5	2	6.86	3.41E−3	96	29	8.26E+2	ls failed							
6	0	6.84	3.41E−3	73	8	5.19E+2	ls failed							
	LBFGS(50, 50, mt) reset mode							LBFGS-r(50, 50, mt) recycle mode						
1	51	1.60E+2	1.53E−3	53	53	2.84E+2	maxit	51	1.60E+2	1.53E−3	53	53	2.86E+2	maxit
2	14	1.03E+2	3.38E−3	67	67	3.77E+2	ls failed	23	1.02E+2	9.80E−4	29	29	1.63E+2	conv.
3	26	3.65E+1	7.75E−4	38	38	2.19E+2	conv.	14	3.531	9.31E−4	18	18	1.01E+2	conv.
4	23	7.55	9.59E−4	28	28	1.63E+2	conv.	10	7.15	8.46E−4	15	15	8.48E+1	conv.
5	10	4.00	9.04E−4	15	15	9.07E+1	conv.	7	3.51	8.16E−4	9	9	5.28E+1	conv.
6	1	3.99	6.25E−4	4	4	2.52E+1	conv.	1	3.49	6.23E−4	2	2	1.25E+1	conv.
	LBFGS(50, 50, armijo) reset mode							LBFGS-r(50, 50, armijo) recycle mode						
1	51	1.60E+2	1.53E−3	104	53	3.96E+2	maxit	51	1.60E+2	1.53E−3	104	53	3.97E+2	maxit
2	14	1.04E+2	1.88E−3	127	15	3.45E+2	ls failed	16	1.03E+2	1.75E−3	134	17	3.68E+2	ls failed
3	26	3.60E+1	8.85E−4	106	28	3.63E+2	conv.	30	3.53E+1	8.65E−4	122	32	4.26E+2	conv.
4	24	7.30	8.48E−4	56	26	2.44E+2	conv.	16	7.12	8.75E−4	45	18	1.87E+2	conv.
5	11	3.82	9.87E−4	29	13	1.31E+2	conv.	9	3.61	9.83E−4	27	11	1.18E+2	conv.
6	2	3.81	7.80E−4	13	3	4.85E+1	conv.	2	3.55	7.48E−4	5	3	2.73E+1	conv.

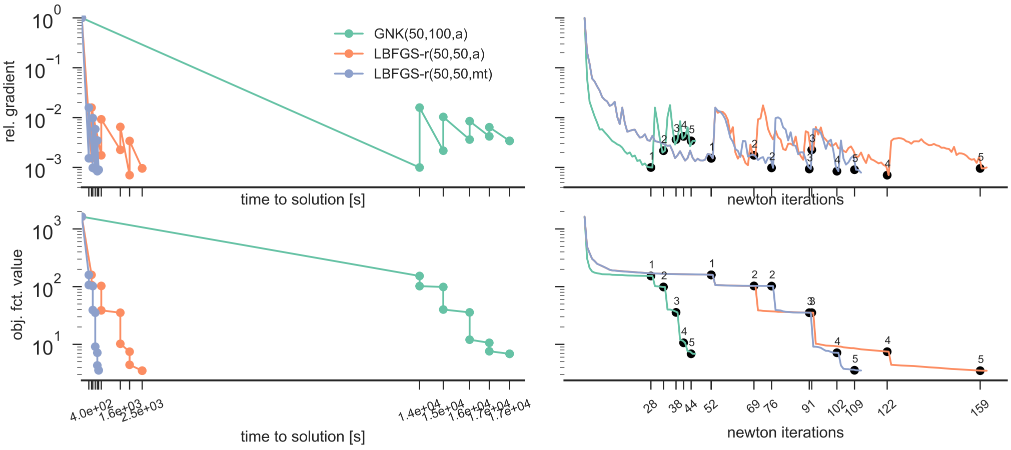


FIGURE 7.20 *Gauß-Newton-Krylov vs. Quasi-Newton results for* ***analytic tumor / analytic velocity with non-zero diffusion (ATAV)*** *test case. Convergence histories and numerical properties for Gauß-Newton-Krylov solver compared with limited memory BFGS (LBFGS) quasi-Newton Method (Armijo and Moré-Thuente line-search). The black labels indicate completion of Picard iterations of the joint inversion scheme.*

significantly less line-search attempts for the recycle version. This can be seen from the reduced number of state and adjoint solves and is ultimately reflected in the runtime. The number μ of stored vectors used for the LBFGS update has a great influence on the method's performance, especially if recycling is considered; see Fig. 7.21 for an illustration. For the problem at hand, we observe best convergence for $\mu = 70$ stored vectors, resulting in an accumulated number of 85 Newton iterations across all Picard iterations and a roughly 3 times faster accumulated runtime for the tumor inversion compared to using $\mu = 30$ vectors in the LBFGS update[29]. When using the recycle variant of the LBFGS method, the parameter μ of stored and used previously seen difference vectors becomes a tuning parameter. While restricting μ to a small number results in too limited information to build a sufficient approximation of the inverse Hessian matrix, allowing too many (and probably outdated) vectors in the LBFGS update may as well deteriorate the method's convergence rate (from the first throughout the last Picard iteration, the Hessian of the tumor inversion optimality system may change significantly due to modified observation data). The increased number of line-search attempts for run #8 compared to run #7 indicates that the approximation of the inverse Hessian is corrupted due to outdated secant-information from an early stage of the joint inversion process.

Following our predominating objective of runtime reduction, we reach a speedup of a factor of 29 for the accumulated time spent in the tumor inversion solver when comparing the best LBFGS variant from run #7 with the joint inversion run using a Gauß-Newton-Krylov solver using an upper bound of $\text{maxit}_T = (50, 100)$ for allowed Newton- and Krylov-iterations, respectively. The overall achieved accuracy for the joint inversion process with respect to the L_2-mismatch for brain tissue label and tumor probability maps is in fact slightly better for the runs using LBFGS. Looking at Tab. 7.22, we further observe, that for Picard iterations $k \geqslant 2$ the tumor inversion using the quasi-Newton solver converges up to the prescribed gradient reduction, whereas the Gauß-Newton-Krylov solver leads to termination due to line-search failure in these cases.[30] The best configuration for the quasi-Newton

[29] Note that storing only $\mu = 30$ vectors with a maximum number of 50 of allowed Newton iterations hardly allows for retaining old vectors from previous Picard iterations.

[30] Due to our parametrization as superposition of Gaussian basis functions and their filtering with the brain geometry, the tumor inversion process may generate negative probability for tumor, or, probability larger than one. This behavior is not physical, and thus corrected (i.e., the initial condition is clipped to have values between zero and one) in the diffusion solver. As a result, gradient and objective function do no longer correlate exactly, and, the GNK solver my produce

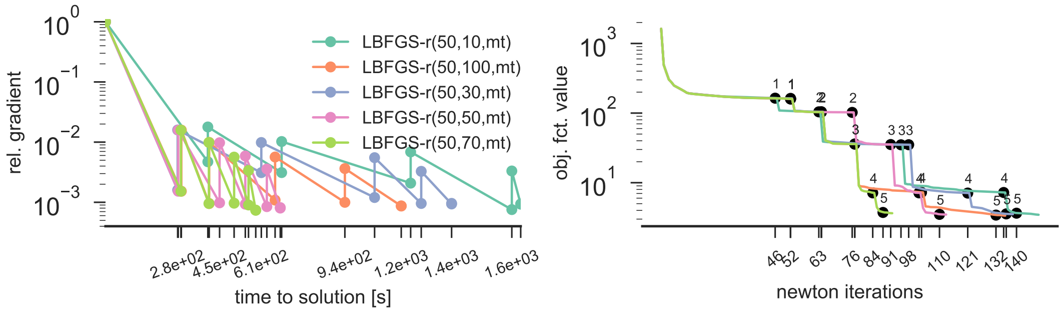


FIGURE 7.21 *Convergence histories for LBFGS quasi-Newton Method for the **analytic tumor / analytic velocity with non-zero diffusion (ATAV)** test case and varying number μ of retained difference vectors in the least-squares system. The black labels indicate completion of Picard iterations of the joint inversion scheme.*

approach only requires about twice the number of Newton iterations if compared to the GNK solver. A comparison and illustration of the solver convergence histories is given in Fig. 7.20. Lastly, we studied the effect of the line-search method on the quasi-Newton performance. Our numerical results confirm that using an Armijo line-search, which only enforces a sufficient decrease in the objective function, is not sufficient for quasi-Newton LBFGS. We see a drastic increase in line-search attempts (significantly more state equation PDE solves than adjoint equation PDE solves) for the Armijo line-search runs #3 (LBFGS(50) reset) and #9 (LBFGS(50) recycle); compare also Tab. 7.22. For these runs, steps of poor quality are taken and automatically incorporated in the Hessian approximation; the self-correction property of the BFGS update formula, as well as its preservation of positive definiteness is no longer given. Frequent line-search failures, degradation of convergence speed and a longer overall runtime are the consequence (compare Fig. 7.20).

Conclusion *For the ATAV test case, the quasi-Newton LBFGS solver clearly outperforms Gauß-Newton-Krylov despite its slower convergence rate, due to significantly lower computational cost per Newton update, if employed for the tumor inversion solver within the joint inversion scheme. We gain remarkably speedups of factors between* 10 *and* 29 *(given that enough vectors are used for the LBFGS update). Recycling difference vectors from converged solves of previous Picard iterations additionally fosters convergence of the method, but renders μ to be a tuning parameter. We observe a speedup factor of* 3.

Overall, the results suggest that the system Hessian for this problem can be effectively represented by a low-rank approximation. Thus, different strategies[31] *for retaining previous information and, in particular, subspace tracking methods that maintain a low-rank approximation of the inverse Hessian by periodically updating a truncated SVD in between Picard iterations, might be promising. A brief description of these approaches is given in §6.4.5; their implementation and numerical analysis, however, remains subject to future work. In a similar way it might be worthwhile to investigate whether a single rank-m update ($m \leqslant \mu$ number of collected vectors) has an advantage over performing m sequential rank-*1 *updates.*

Performance of Quasi-Newton LBFGS for Joint Inversion using Clinical Data.

Purpose. Real patient clinical data poses significant challenges on the tumor inversion solver. In contrast to the synthetic cases, our simplistic biophysical tumor growth model now is at best a crude approximation of the observations. Furthermore, all model parameters are unknown and suggested

slightly worse search directions. This highly depends on the given case.

[31] For example using only the first 5 difference vectors of every Picard iteration.

TABLE 7.23 *Performance of Gauß-Newton-Krylov and quasi-Newton for the tumor inversion solver for* ***real tumor / real velocity (RTRV)*** *test case; ground truth (ρ N/A, $\mathbf{k}$ N/A, $\mathbf{p}$ N/A, $\mathbf{v}$ N/A); based on real clinical data (taken from [Goo13]). We set the tumor parameters to $\rho_f = 15$, and $k_f = 1\text{E}{-2}$ (DIF) or $k_f = 0$ (REAC), respectively for reaction-diffusion and reaction-only tumor inversion runs. We report the final data L_2-mismatch after the joint inversion (moving-patient) for brain tissue labels μ_{B,L^2} and tumor probability maps μ_{T,L^2} as well as (summed) number of Newton iterations it_N, (summed) number of state and adjoint evaluations, as well as (summed) timings for the joint inversion T^{tot}, tumor inversion T^{tu}_{inv}, and registration T^{reg}_{inv}, respectively in seconds. We give the speedup of the execution of the overall time-to-solution S^{tot} with respect to the Gauß-Newton-Krylov method. For Gauß-Newton-Krylov, we limit the number of Newton and Krylov iterations to $\text{maxit}_T = (30, 60)$ and use the cheaper armijo line-search method. For LBFGS we investigate different numbers $\mu \in \{10, 30, 50, 70\}$ of stored vectors and also retain (up to μ) stored vectors from previous tumor inversion solves across Picard iterations (recycle mode). The latter is denoted as LBFGS-r(50,μ,ls). We further use a Moré-Thuente (mt) line-search method enforcing the Wolfe conditions, as required for quasi-Newton methods. For the initial inverse Hessian approximation, we use TAO's built-in Broyden scaling of the identity matrix with an additional re-scaling; for details see §6.4.5 and [Ben03].*

			Method(maxit$_N$, H, ls)	μ_{B,L^2}	μ_{T,L^2}	it_N	#state	#adj	T^{tot} [s]	T^{tu}_{inv} [s]	T^{reg}_{inv} [s]	S^{tot}
AAAN	DIF	#10	GNK(30, 60, armijo)	3.51E−1	3.55E−1	123	1119	776	1.87E+4	1.81E+4	3.53E+2	1.00×
		#11	LBFGS-r(30, 10, mt)	3.46E−1	2.74E−1	59	238	238	5.26E+3	4.72E+3	3.51E+2	3.56×
		#12	LBFGS-r(30, 50, mt)	3.47E−1	2.83E−1	51	241	241	5.23E+3	4.71E+3	3.43E+2	3.56×
	REAC	#13	GNK(30, 60, armijo)	3.49E−1	3.20E−1	186	1442	1226	9.06E+2	5.44E+2	3.44E+2	1.00×
		#14	LBFGS-r(30, 10, mt)	3.49E−1	3.08E−1	53	225	225	4.20E+2	4.76E+1	3.56E+2	2.15×
		#15	LBFGS-r(30, 30, mt)	3.48E−1	3.01E−1	68	259	259	4.11E+2	5.84E+1	3.38E+2	2.20×
AAMH	DIF	#16	GNK(30, 60, armijo)	3.47E−1	2.22E−1	49	1224	827	2.04E+4	1.98E+4	3.47E+2	1.00×
		#17	LBFGS-r(30, 30, mt)	3.45E−1	2.03E−1	47	284	284	6.28E+3	5.74E+3	3.52E+2	3.25×
	REAC	#18	GNK(30, 60, armijo)	3.40E−1	1.65E−1	177	1777	1565	1.10E+3	7.43E+2	3.37E+2	1.00×
		#19	LBFGS-r(30, 30, mt)	3.37E−1	1.44E−1	61	79	79	3.85E+2	2.13E+1	3.49E+2	2.85×

values might be arbitrarily wrong. As a result, more stress is put on the tumor inversion solver. We want to study the performance of the LBFGS solver for actual clinical data and compare against experiments obtained using the GNK solver.

Setup. We consider actual clinical data, i.e., a real tumor / real velocity (RTRV) test case based on real clinical data (taken from [Goo13]) as described in §7.1.1. We employ the moving-patient formulation, set the tumor parameters to $\rho_f = 15$, and $k_f = 1\text{E}{-2}$ (DIF) or $k_f = 0$ (REAC), respectively for reaction-diffusion or reaction-only tumor inversion runs. We use a regular Gaussian grid determined by uniformly distributing $n_p = 343$ basis functions within the radius of the tumor. We limit the number of iterations to $\text{maxit}_R = (10, 20)$ Newton- and Krylov-iterations for the registration, and $\text{maxit}_T = (30, 60)$ Newton- and Krylov-iterations for the tumor inversion, respectively.

Results. We report numerical results comparing LBFGS quasi-Newton and GNK for the tumor inversion component of our joint inversion scheme for actual clinical data for two patients in Tab. 7.23.

Observations. Similar to the ATAV case, we observe better performance, both in terms of final L_2-mismatch for the predicted tumor probability map as well as the overall time-to-solution, when using the LBFGS solver as opposed to the GNK solver. For patient ID AAAN, GNK (run #12) requires about 5× more state equation solves and about 3× more adjoint solves than the quasi-Newton LBFGS solver (run #13), which results in a speedup of a factor 3.5 for the overall time-to-solution of the joint inversion process. At the same time the final reconstruction error for LBFGS is much smaller (35% relative error for GNK compared to 27% relative error for LBFGS). Very similar observations can be made from the reaction-only run as well as for the second patient ID AAMH. For all conducted experiments, we arrive at a notably smaller final data-mismatch (i.e., at an improved solution of the coupled multi-component problem) while significantly reducing the time-to-solution for the joint inversion process (speedup factors of $2 - -4$). For the reaction-only runs of patient ID AAMH (runs

#21 – #22) the quasi-Newton alternative reduces the number of state and adjoint solves by a factor of over 20, and, if only considering the accumulated time spent in the biophysical inversion step, this translates to a speedup of a factor of 35. For reaction-only runs, however, the computational cost for the tumor inversion is dominated by the registration and small fluctuations in the computational cost of the latter possibly overshadow higher speedups for the overall runtime.

Conclusion *Considering quasi-Newton LBFGS as alternative to GNK to solve the non-linear tumor inversion problem yields excellent results for synthetic cases where the true model and true parameters are known, as well as for actual clinical imaging data of glioblastoma patients, where our simplistic tumor growth model is at best a crude approximation of the true underlying dynamics. Using LBFGS, we observe speedups of factors between* 4 *and* 16 *for the time-to-solution of the joint inversion process and in some cases much higher speedups for the accumulated time spent in the tumor inversion solve. LBFGS converges to the prescribed relative gradient tolerance more frequently than the GNK counterpart and less often terminates due to line-search failure. The achieved overall accuracy of the final solution is typically higher when LBFGS is employed—for the real patient cases this is more pronounced.*

Conclusion: Advanced Methods

We have investigated various methods that enhance our joint inversion scheme, increase its performance, significantly reduce its required time-to-solution, or facilitate a better understanding and a deeper analysis of the scheme. Here is what we have learned:

(i) *We significantly reduce the time-to-solution for our scheme, using a grid-continuation (multi-level) approach and replacing the GNK solver for the tumor inversion by an advanced quasi-Newton LBFGS solver. We reach excellent speedups of up to a factor of* 5 *for grid-continuation and an additional speedup factor of up to* 16 *for the overall time-to-solution of our inversion scheme (ATAV test case), if replacing GNK by the cheaper LBFGS solver for the tumor sub-component. We observe speedup factors of the tumor inversion solver alone of up to* 30*. For clinical patient data quasi-Newton LBFGS still reduces the overall time-to-solution of the joint inversion by a factor of* 4.[a]

(ii) *The initial guess for the approximation of the inverse Hessian and the number of used vectors for the LBFGS update are critical factors influencing the convergence of the LBFGS method. For the initial Hessian approximation, we found a sophisticated Broyden scaling of the identity to perform best. Convergence is improved if many vectors are used for the Hessian update. In particular, we found that recycling vectors from previous inversion solves (with slightly modified input data) has the potential to speedup the runtime by a factor of up to* 3*. In general, our experiments show that quasi-Newton is a good alternative which outperforms GNK for our application problem and solution scheme. The results further suggest, that the inverse Hessian may be efficiently approximated by a low-rank representation and other strategies to retain Hessian information from previous inversion solves, such as an updated SVD subspace tracking method, may be beneficial.*

(iii) *Enhancing the tumor inversion solver objective function by including the brain data-misfit to resemble a block-Newton scheme of the coupled formulation did not show any advantage over the simplified method employed in large parts in this chapter.*

(iv) *Lastly, convergence and final solution of our joint inversion scheme depend on the particular parameter-continuation scheme employed for the regularization weights. Specifically, we found that additional continuation on the tumor regularization weight* β_p *is not recommendable and using an optimal (small) regularization weight from the beginning of our joint inversion iteration yields best results.*

[a] Notice, that the smaller speedup factor is due to the limited number of Newton and Krylov iterations for the real data test cases and occasionally occurring line search failures.

8 Conclusion

Under the heading of numerical assessment and simulation of interacting multi-component systems, we have targeted two distinct strongly coupled application problems:

I. ***Surface-coupled multi-physics simulation***, in particular, ***fluid-structure interaction***

II. ***Brain tumor biophysical parameter inversion coupled with medical image registration***

Both problems feature strong mutual interactions, are inherently hard to solve and result in formidable systems of coupled PDEs, requiring advanced and tailored solution approaches. In this thesis, we developed highly efficient and accurate modular algorithms to tackle both problems. Primary design goals included

- ***high accuracy*** to realistically capture coupled effects and phenomena and meet the applications' requirements;
- ***high efficiency*** and ***parallel scalability*** to enable solving the challenging systems with extremely large data-sets and billions of unknowns in tolerable runtime, which is crucial in particular in clinical settings, but also in engineering applications.
- ***robustness*** to cope with varying input data and problem settings;
- ***modularity*** to foster flexibility and adaptability to different problem settings, and reduce the development time.

We focus on the development of solution methodologies that combine efficient sub-component solvers in a modular way to solve the strongly coupled multi-component problems. In doing so, we are the first to solve the joint registration and biophysical inversion problem by employing gradient based optimization and we lift robustness and parallel scalability of quasi-Newton methods for partitioned surface-coupled multi-physics simulations to a new level. Inexact-Newton and quasi-Newton methods catch attention within this work as they are an essential ingredient for the solution approaches of both application problems.

In what follows, we summarize the main contributions and conclusions obtained within this work. We discuss the practical relevance of the developed methodologies and tools, address limitations, and give recommendations for future work.

8.1 Part I. Partitioned Coupling of Multi-Physics Simulation

The first part targets general purpose black-box coupling functionalities and emphasizes on advanced quasi-Newton coupling schemes to establish surface-coupling for partitioned multi-physics simulations. Main challenges were to develop robust and scalable methods that cope with strong instabilities and hidden solver internals by only requiring input/output relations of the solvers. Quasi-Newton methods have been shown to effectively solve this task; their convergence, however, is very sensitive to certain problem dependent parameters and parallel scalability is often limited. The derivation of ***robust*** and ***efficient*** quasi-Newton alternatives that are feasible in a large-scale, parallel setting, was a challenging task.

8.1.1 Contributions and Practical Relevance

Part I: Contributions

The most important contributions are:

- *To obtain improved parallel efficiency, we switch from a serial Gauß-Seidel-type execution order to a parallel Jacobi-type scheme [Lin15; Sch15].*
- *We developed coupling schemes for partitioned fluid-structure interaction simulations. In particular, we derived highly robust and efficient advanced quasi-Newton methods to accelerate the non-linear fixed-point equation coupling. Based on previous work [Sch15], we propose two robust and scalable new quasi-Newton methods: (i) a matrix-free Multi-Vector-Update (MV) restart alternative (RS-SVD) based on a sophisticated subspace-tracking technique to retain information across restart borders [Sch17], and (ii) a Least-Squares (LS) method with powerful filtering of quasi-Newton input vectors. Both allow to effectively reuse information from previous time steps in a robust way without parameter tuning, feature linear complexity in the number of unknowns and a highly scalable parallelization up to* 2, 048 *tasks. We extensively evaluate the numerical properties and convergence of the herein developed advanced quasi-Newton methods. We analyzed the sensitivity of a large variety of quasi-Newton methods with respect to the amount of retained information from the past, and the effect of stabilizing filtering techniques to eliminate linear dependencies [Hae15].*
- *The discussed quasi-Newton coupling schemes have been realized on distributed memory and integrated within the general purpose black-box coupling library* preCICE *[Bun16b; Sch17; Bun16a].*

In the following, we give more details and share the most crucial conclusions alongside with some practical implications.

The commonly used Least-Squares (LS) method benefits greatly from explicit reuse of past information in time-dependent simulations, and can be realized very efficiently in a matrix-free way. The optimal amount of reused information, however, is highly problem dependent and requires costly tuning. Its convergence properties furthermore sensitively depend on a powerful filtering technique for input information to maintain stability. The MV method developed in [Sch15] alleviates these drawbacks by implicit reuse of past information, but features quadratic storage and runtime complexity. This renders the original method infeasible for large-scale simulations. Based on an efficient (decomposed) representation of the inverse Jacobian in conjunction with periodic restarts, we reduced the computational complexity from a quadratic to a linear one. To maintain the good MV convergence properties, Jacobian information is retained across restarts via a sophisticated SVD-based subspace tracking method (RS-SVD). The SVD updating step has cubic complexity in the rank $\overline{\kappa}$ of the truncated SVD representation of the Jacobian and requires a rank substantially smaller than – and in particular independent of – the number of unknowns. We observed small ranks for all conducted experiments, independent of the mesh-resolution. This results in mesh-independent convergence and $\mathcal{O}(N)$ overall complexity and allows for large-scale simulations (where MV is not feasible). We reported numerical results to compare different restart alternatives for the MV method with respect to convergence speed and efficiency, and concluded that the RS-SVD method achieves good MV convergence and outperforms all other approaches.

We conducted an extensive numerical analysis of LS(ξ) (with reuse of information from ξ previous time steps) and the MV RS-SVD approach with respect to their ***convergence***, their overall ***efficiency***, their dependency and sensitivity to an efficient ***filtering*** method as well as to their ***parallel scalability*** and ***runtime***. We observed that the MV RS-SVD method is highly robust without the need for additional filtering or tuning of problem dependent parameters, and is powerful enough to allow for small restart periods (memory efficiency). In terms of runtime and efficiency, we achieved a reduction in runtime per iteration of 87% when using the RS-SVD method as opposed to the MV method for the three-dimensional flexible tube FSI scenario with $9,600$ unknowns[1]. Due to the linear complexity of RS-SVD and its excellent parallel efficiency, we can produce arbitrarily high speedups by increasing the number of unknowns. By choosing a large reuse $\xi \to \infty$ in conjunction with the QR2 filter method, we were able to showcase excellent robustness for the LS method. The large number of vectors in the least-squares system (and triggered re-computations of the QR-decomposition), however, drastically increases the computational complexity per iteration and renders the method inferior to MV RS-SVD.

Practical Relevance. The general purpose coupling library preCICE in which we implemented the new quasi-Newton methods is of great relevance in setting up parallel multi-physics simulations with minimal development effort by re-using existing, highly sophisticated solvers. The software is widely used and has a growing number of users. The terms ***generic*** and ***black-box*** are crucial here. The coupling numerics are a main building block, and hidden solver internals of the black-box approach render an efficient coupling challenging. The herein developed methods and advanced features provide a powerful tool for generic black-box coupling. They have been implemented in parallel, and allow for highly scalable, massively parallel execution. The developed methods within this work enhance preCICE to establish a consistent coupled solution of even strongly interacting, yet partitioned physical fields within a small number of coupling iterations. The excellent parallel scalability of the coupling numerics allow to exploit the full parallel potential of sub-component solvers.

One might argue that the costs for coupling numerics at a lower-dimensional surface are always neglectable and not worth the effort invested in our work. This statement, however, is wrong for two reasons: (*i*) The number of required coupling iterations is absolutely critical as every extra iteration corresponds to solving a full implicit time step in the involved single-physics sub-components, and trial-and-error parameter tuning is not an option. (*ii*) When going to large-scale simulations and massively parallel execution (more than a few thousand cores), the time spent in the coupling component is no longer neglectable (compared to the runtime of the single-physics solvers) [Uek16]. The linear complexity of the herein proposed methods and their good parallel scalability is therefore essential for large-scale simulations.

[1] The original MV methods becomes infeasible beyond this point. For the MV RS-SVD method, we scaled up to $307,200$ interface unknowns for this scenario.

8.1.2 Limitations and Recommendations for Future Work

Limitations. We have evaluated our methods and demonstrated their performance for a number of challenging, strongly coupled FSI simulations that feature a lot of the typical characteristics observed in partitioned multi-physics. Thereby, we observed some dependencies of our methods on the utilized sub-solvers, e.g., the quality of the employed mesh-movement. Therefore, the proposed methods need to be thoroughly evaluated in different settings, i.e., different multi-physics applications and different employed sub-component solvers, in order to claim their excellent ***generic performance***. In particular, the RS-SVD method is only efficient if the problem can be effectively approximated by a low rank model, and in particular the rank of the truncated SVD remains small and mesh-independent. Although this holds true for the here considered FSI scenarios, this property needs to be evaluated for a broader family of multi-physics problems.

Recommendations for Future Work. The above described limitations directly offer possibilities for future research. Another very promising aspect is the investigation of a multi-level approach in conjunction with the presented quasi-Newton methods. The manifold mapping method [Blo14a; Blo15a; Blo14b] realizes such a multi-level concept by relying on solutions of a surrogate (coarse model) of the high fidelity model to update the latter in a defect-correction manner. A manifold mapping method which employs the herein developed advanced quasi-Newton methods as coarse model solvers has been implemented within this work into preCICE with promising preliminary results [Blo15b]. The investigation of different solver combinations, effects of reuse of past information and filtering is very promising to further speedup the coupling numerics for partitioned multi-physics simulations.

8.2 Part II. Coupling of Bio-Physical Brain-Tumor Models with Medical Image Registration

The second part targets MR image analysis of glioblastoma multiforme pathologies and patient specific simulation of brain tumor progression. We approach this highly involved and sophisticated physiological phenomenon using a joint medical image registration and biophysical inversion strategy. We focus on two main application goals, (*i*) ***biophysics aided normal-to-abnormal*** registration, and (*ii*) parameter estimation and ***biophysical model calibration***. Both target at facilitating diagnosis, aiding and supporting surgical planning, and improving the efficacy of brain tumor therapy. Our problem formulation results in a large-scale, highly non-linear and non-convex PDE-constrained optimization problem, which we decomposed into two tightly coupled inverse problems. The joint biophysical inversion and registration approach is novel. We simultaneously invert for a parametrization of the initial condition for the tumor model and a smooth velocity field to capture the inter-subject variability of brain anatomy. With this approach, we are the first to solve the inverse tumor-growth problem based on a single patient snapshot with a gradient-based approach.

8.2.1 Contributions and Practical Relevance

Part II: Contributions

The most important contributions are:

- *We developed the SIBIA framework for integration of biophysical inversion and medical image registration to be applied in brain tumor MR image analysis. It features reliable, highly efficient, and highly scalable solvers to tackle the formidable coupled multi-component optimization problem of joint tumor inversion and medical image registration. Such solvers can be used either to transfer spatial (structural or functional) information from a statistical atlas brain to the specific patient brain (where the grown tumor alters the brain topology), or for tumor-growth inversion for model calibration based on a single snapshot in time. We are the first to solve this problem employing a gradient based approach [Sch].*
- *We deduced two formulations for the coupled multi-component problem, focusing on the two application goals – (i) the quality in* ***normal-to-abnormal*** *registration, and (ii) the biophysical parameter inversion and* ***model calibration*** *for* ***patient specific simulations*** *and prediction.*
- *We derived a* ***modular Picard iteration-type*** *solution strategy for each formulation, based on the two sub-components inverse tumor-growth simulation and image registration. We showed (mesh-independent)* ***convergence*** *of both schemes by monitoring the reduced gradient of the coupled problem formulations. The modular approach allows for great* ***flexibility*** *in the employed biophysical model and registration approach, but also the concrete realization of the underlying solvers.*
- *We demonstrated the* ***validity*** *and* ***efficiency*** *of our approach by thorough* ***numerical analysis*** *on synthetic and actual* ***clinical data sets****. We compared both schemes with regards to quality and suitability for normal-to-abnormal registration and the potential suitability for the reconstruction of clinically relevant or diagnostically meaningful information.*
- *We optimize the parallel, distributed memory solvers and algorithms to achieve excellent* ***parallel scalability*** *and significantly reduce the* ***time-to-solution****. Here, we reach a speedup factor of 10-20 for the time-to-solution by employing quasi-Newton methods and a multi-level strategy with gradually improving coarse level solutions for the non-linear optimization.*

In what follows, we go into more detail for some of the aforementioned points, outline the most crucial conclusions, and discuss practical relevance.

Our work improves the approach in [Bak15; Kwo14; Hog08a; Goo13] in terms of formulation of the problem, employed algorithms and solvers, scalability, and performance, and overcomes some of the shortcomings of the existing approaches. We employ and design very efficient algorithms that make use of gradient-based optimization and are scalable to hundreds of thousands of cores [Man16b; Gho17c]. Globalized inexact Newton methods such as the Gauß-Newton-Krylov method and quasi-Newton methods are a core ingredient of our solver. In particular, our methodology integrates one of the most advanced state-of-the-art algorithms for constrained large deformation diffeomorphic image registration [Man18a].

The formulation of the complex problem is not straightforward. We propose two formulations, (*i*) the so-called ***moving-patient*** formulation intended for normal-to-abnormal registration, and (*ii*) the so-called ***moving-atlas*** formulation, meant to improve the reconstruction of biophysical parameters and help to extract clinically meaningful information. A Newton-type monolithic solver for the coupled formulation would have promised rapid convergence, but its development and efficient realization is highly involved. To account for ever changing models for registration and tumor progression and foster a short development time, we thus deduced modular Picard iteration-type solution strategies and gain great flexibility by exploiting sophisticated sub-component solvers. The solution of our application, however, requires a tight integration of both components, adaption of solvers, and integration of tailored features. We enhanced the registration to integrated information of the tumor solver within the solution process. To account for the non-linearity and non-convexity of the problem, we integrated a parameter-continuation scheme for the regularization parameter of the registration and a grid-continuation scheme into the Picard iteration-type methodology.

Our scheme is capable of automatically processing (segmented) actual clinical imaging data of arbitrary mono- and multi-focal brain tumor malignancies without manual seeding. This is enabled by a parametrization of the tumor initial condition and automatized, data-driven selection of basis functions. This also drastically reduced the number of inversion variables. To improve the reconstruction of clinically meaningful and relevant biophysical parameters, the moving-atlas scheme is used in combination with a sparsity constraint (L1-regularization) for the unknown initial condition of the tumor. By employing the sparsity constraint, the time horizon of the tumor growth can be fixed to the time-point of initial cell mutation (tumor genesis).

By thorough numerical analysis, we demonstrated the quality, validity and efficiency of our solution schemes. Empirically, we showed convergence of the reduced gradient of the respective fully coupled formulation and, thus, convergence to a local minimum. We furthermore attested a mesh-independent convergence rate of our scheme. We concluded that our moving-patient Picard iteration solution scheme is a very powerful tool for inter-subject normal-to-abnormal registration as used in automated segmentation tools. Our numerical study, which includes actual clinical data with real tumors, showed that we can achieve high-fidelity results for the normal-to-abnormal registration application scenario with an overall low mismatch and high Dice scores, ranging from 7.94E−1 to 8.45E−1 for real clinical cases. In particular, plain-vanilla registration without biophysical augmentation failed in the vicinity of and the area occupied by the pathology for the presented problems. For the simulated and observed tumor, we achieved extremely high similarity of around 98% overlap for real complex-shaped tumors if sufficiently many Gaussian basis functions are used. The simplistic reaction-diffusion tumor-growth model is sufficient if we are primarily interested in good visual reconstruction and data similarity. This is quite certainly not true if we target parameter identification and patient specific tumor growth prediction to aid clinical decision making.

With a view towards diagnostic relevance, we analyzed the quality of the results obtained from the moving-atlas formulation. This modified scheme clearly showed improvements in terms of biophysical parameter estimation compared to the moving-patient scheme. For artificial test cases where the ground truth is known[2], we showcased improved reconstruction of the brain anatomy, and, in particular, the anatomy of the (unknown) healthy patient brain. Most importantly, we observed improved tumor reconstruction and the scheme exhibits convergence of the reconstructed initial condition towards the actual ground truth initial condition. That means, that as opposed to the moving-patient scheme, this improved formulation is capable of reconstructing the actual biophysical parameters if the correct type of model (i.e., reaction-diffusion, etc.) is known. We want to emphasize that these are preliminary results that serve as a proof-of-concept and by no means clinically relevant. However, the results demonstrated the validity of our developed solution strategy. The integration

[2] By this we mean that no modeling error for the tumor growth simulation is present, i.e., in case tumor-growth was accurately and truly described by a reaction-diffusion process

of more complex tumor progression models contains great potential.

In clinical practice, high accuracy and time-to-solution are critical. We therefore opt for highly efficient and scalable algorithms that allow for a short time-to-solution to process high resolution data. We presented efficiency and scalability results for the registration solver and the tumor inversion solver for up to 16 thousand cores and 200 billion unknowns (problem size 64 times larger than state-of-the-art). For clinically relevant test cases, SIBIA is up to eight times faster than the state-of-the-art. In an attempt to further reduce the time-to-solution, we replaced the GNK solver for the tumor inversion by a quasi-Newton LBFGS alternative. Inspired by the findings for quasi-Newton methods used to accelerate fixed-point iterations, we recycled information from previous Picard iterations (i.e., previous tumor inversion solutions with slightly modified data)[3]. Together with a grid-continuation multi-level approach, we reach speed-up factors of $10 - 20$.

Practical Relevance. Implications for the practical relevance of the developed methodology emerge from the stated contributions and limitations. For clarity, we repeat the most important points. The developed joint inversion scheme is capable to automatically analyze mono- and multi-focal brain tumor MR imaging data. The excellent performance for biophysically aided normal-to-abnormal registration is valuable for automated segmentation and the mapping of structural and functional information from expert-labeled atlas brains to specific patient brains. This finds application in surgery planning. The capabilities for biophysical inversion are still limited and the employed model is too simplistic to be clinically meaningful or predictive. However, SIBIA has been used in first tests and cross-validation with clinical observations at the radiology institute of the University of Pennsylvania. The methodology exhibits high potential if used with a more complex tumor model. We solve the challenging problem on 128 cores in about an hour. The response time as well as the required compute system is feasible for clinical application.

8.2.2 Limitations and Recommendations for Future Work

Limitations. First and foremost, we want to emphasize that within this work we developed a new methodology for a previously unsolved problem. The results are clearly preliminary and our methodology is not yet sophisticated enough to be predictive or applicable in clinical practice. This work primarily has to be seen as a proof-of-concept and foundation towards a powerful tool in computational oncology. In the sequel, we summarize the main limitations. (*i*) Our tumor model is very simplistic and clearly not predictive. It does not model edema, necrosis, angiogenesis and chemotaxis and is lacking a description of tumor mass-effect. This is an essential restriction and primary cause for limited clinical relevance. More complex models that account for such phenomena are expected to enhance our scheme to be predictive. (*ii*) Further, multimodal input data from different MRI modalities cannot be processed as of now. The input is assumed to be a set of probability distributions for each brain tissue and actual tumor concentration values. (*iii*) It is (in general) not possible to simultaneously invert for the cell proliferation and net migration of malignant cells into the surrounded healthy tissue. Nonetheless, for our methodologies we need to somehow estimate the correct proliferation rate (reflected by the reaction coefficient ρ) of the given tumor (while inverting for the net migration rate, reflected by the characteristic diffusivity k_f). (*iv*) Lastly, we exclusively consider a deterministic problem formulation. In reality, however, all quantities and models, i.e., the measurement of the observed data, the mathematical model for the description of the underlying process, the model parameters, and the inversion algorithm itself

[3] This new recycle feature is implemented in the latest release of PETSc's TAO package.

are subject to uncertainties due to noise, modeling error or numerical errors. Thus, instead of point estimates for our quantity of interest, confidence intervals are required, in order to allow for an uncertainty quantification and propagation from the input to the quantity of interest.

With respect to our solution strategy and the employed Picard iteration-type scheme, we note that we do not have a theoretical proof for the convergence of our Picard iteration-type solution scheme, but provide empirical evidence for convergence. Our block-Newton scheme does not fully inherit the Newton-like rapid quadratic convergence, however converges fast to the practically required accuracy in a mesh-independent way. We have not compared our scheme to other approaches, e.g., a monolithic Newton solver, as such solvers do not exist for our application problem. An inherent problem in non-linear and non-convex optimization is, that we can only guarantee convergence to a local minimum. However, we reduced the chance of getting trapped in local minima by applying of parameter- and grid-continuation schemes.

Recommendations for Future Work. The above described limiting factors of our joint registration and biophysical inversion approach directly suggest specific improvements and promising ideas for future work. As motivated above, a more sophisticated tumor-growth model is absolutely critical. In particular, the tumor-growth induced deformation of brain parenchyma (mass-effect) needs to be addressed, alongside with a model to describe edema, necrosis, angiogenesis and chemotaxis. Further, support for multimodal input data is recommended to maximize the amount of usable input information. To account for uncertainties, our deterministic inversion approach can be combined with statistical inference methods, such as Bayesian posterior sampling [Mar12; Pet14]. Our method inverts for the initial condition of the tumor and the characteristic net migration of malignant cells into surrounding healthy tissue. Simultaneously inverting for the proliferation rate results in an ill-posed problem with non-unique solution. We propose a continuation scheme to recover the correct proliferation rate ρ: Starting with $\rho = 1$, we perform a series of joint inversion runs, successively doubling the proliferation rate ρ until the maximum value of the initial condition is less than 1[4]. Thereafter, a binary search is applied to find the value for ρ which results in the lowest data-mismatch.

Furthermore, several directions for further investigation and room for improvement arise in terms of our joint inversion solution strategy. To back up theory, a rigorous proof for convergence of our Picard iteration-type solver to a solution of the coupled formulation is desirable. We discovered that, instead of accurate sub-component solutions in every Picard iteration, inexact solutions potentially improve convergence and yield an overall more efficient solver. In particular, "over-solving" the current state in a sub-component or "over-fitting" could be avoided. Preliminary results show, that, by performing only two Newton iterations per sub-component solver and Picard iteration, sub-component gradient oscillations can be damped effectively and convergence is improved. A thorough analysis of a scheme using inexactness in the sub-component solvers is highly recommended. Also acceleration methods such as used in FSI (part I) used to speed-up the convergence of the Picard scheme might be worthwhile to investigate. For the sake of comparison (of convergence and obtained solution), a monolithic Newton solver which tackles the coupled formulation and the thereof resulting optimality system of non-linear, strongly coupled PDEs at once, is of increased interest.

For the currently employed scheme, modification (advection, re-selection) of the Gaussian basis functions for the tumor parametrization according to changed input data (for moving-patient) or modified simulation geometry (for moving-atlas) should be considered. We effectively employed

[4] A too low value for the proliferation rate ρ causes the tumor inversion solver to create initial conditions with unreasonable values that are much larger than 1.

an LBFGS solver for the inverse tumor-growth problem, which recycles information from previous inversions (of preceding Picard iterations) with slightly different data. We believe that the Hessian matrix can be approximated by a low-rank matrix representation; an SVD subspace-tracking approach as developed in Chapter 2 might be beneficial to further speed-up convergence.

8.3 Quasi-Newton Schemes for Fixed-Point Eq. and Optimization

In Chapter 2, we furthermore looked at different quasi-Newton methods applied in (*i*) accelerating fixed-point iterations for surface coupling of partitioned multi-physics applications, and (*ii*) non-linear, PDE-constrained optimization.

8.3.1 Contributions

Quasi-Newton: Contributions

The most important contribution is:

- *We gave a lucid overview of, and compare the methodological components and characteristics of quasi-Newton methods utilized for acceleration of non-linear fixed-point problems and in PDE-constrained optimization. In our discussion, we aligned the notation from two different application fields, relate the LBFGS method to the Multi-Vector Update (MV) method, and outlined their commonalities and differences. This allows for the development of new methods and strategies that allow both worlds to benefit from each other.*

Quasi-Newton methods are of great importance in both accelerating non-linear fixed-point iterations and non-linear optimization. Yet, both fields use fairly different formulations and method names. We align the notation, analyze the methodological components and outline similarities and differences. The LBFGS method can be seen as a symmetrized MV update, but instead of performing one rank-k update, LBFGS performs k successive rank-1 updates. We showed that this is not equivalent. This very fact also explains why LBFGS copes without the need of a stabilizing filtering technique. Primary factors that influence convergence for LS and MV is the amount η of ***retained previous information*** and the quality of the ***filtering technique***. For LBFGS (when applied to optimization problems), the ***initial inverse Hessian approximation*** is critical; while in the fixed-point acceleration context, MV and LS work fine with a trivial initial guess for the inverse Jacobian matrix (i.e., $\boldsymbol{M}^{-1} = \boldsymbol{0}$), LBFGS for non-linear optimization fails unless a good initial guess can be provided. Our work allows to easily access similarities and differences of established methods in the respective fields and, in particular, lays the basis for strategies of how the different application fields may benefit from each other. This is what we discuss next.

8.3.2 Resulting Ideas for Future Directions

Answering the question of how established quasi-Newton methods for fixed-point acceleration of partitioned multi-physics application could benefit from the "optimization world", the most obvious point is the investigation of better initial guesses for the approximation of the inverse Jacobian. Although methods like LS and MV work fine with a zero initial guess, considering some of the more advanced techniques used for LBFGS in PDE-constrained optimization might entail a significant convergence improvement for the former, due to a better, informed initial guess. Problem dependent as well as the generic Broyden-scaling approaches are worth further investigation.

On the other hand, LBFGS can be modified in several ways to potentially make better use of past information. First, modifying the method to perform a single rank-k update rather than applying k successive rank-1 updates, might improve the method's robustness. We have learned that, in doing so, the MV method achieves much better robustness. In this case, filtering might become necessary to ensure linearly independence of information. Furthermore, if used in a data assimilation context as given in part II of this thesis, LBFGS might benefit from different reuse strategies. In particular, retaining information from previous (similar) optimization solutions can speed-up the convergence and allow for informed, good initial inverse Hessian approximations. We verified this for the tumor inversion problem. In this context, a subspace tracking method similar to the restart-strategy applied for the MV RS-SVD method can help to retain the most important information across optimization cycles (Picard iterations in the SIBIA context) in an implicit way.

9 Bibliography

[A15] **A.**, G. et al.: *Accfft: A library for distributed-memory FFT on CPU and GPU architectures*. CoRR abs/1506.07933 (2015).

[Akc02] **Akcelik**, V.; **Biros**, G.; **Ghattas**, O.: *Parallel multiscale Gauss-Newton-Krylov methods for inverse wave propagation*. Proc ACM/IEEE Conference on Supercomputing. 2002, pp. 1–15.

[Akc06] **Akcelik**, V. et al.: *Parallel algorithms for PDE constrained optimization*. Ed. by **Heroux**, M. A.; **Raghavan**, P.; **Simon**, H. D. Vol. 20. Parallel Processing for Scientific Computing. Philadelphia, Pennsylvania, US: SIAM, 2006. Chap. 16, pp. 291–322.

[Ale07] **Alexander**, A. L. et al.: *Diffusion tensor imaging of the brain*. Neurotherapeutics : the journal of the American Society for Experimental NeuroTherapeutics 4.3 (2007), pp. 316–329. 10.1016/j.nurt.2007.05.011.

[And17] **Anderson**, D. G.: *Comments on "anderson acceleration, mixing and extrapolation"*. Numerical Algorithms (2017), pp. 1–100.

[And65] **Anderson**, D. G.: *Iterative procedures for nonlinear integral equations*. Journal of the ACM 12.4 (1965), pp. 547–560.

[Ang07] **Angelini**, E. D. et al.: *Glioma dynamics and computational models: A review of segmentation, registation, in silico growth algorithms and their clinical applications*. Curr Med Imaging Rev 3.4 (2007), pp. 262–276.

[Ara04] **Araujo**, R. P.; **McElwain**, D. S.: *A history of the study of solid tumour growth: the contribution of mathematical modelling*. Bulletin of mathematical biology 66.5 (2004), pp. 1039–1091.

[Ara05] **Araujo**, R. P.; **McElwain**, D. S.: *A mixture theory for the genesis of residual stresses in growing tissues i: a general formulation*. SIAM Journal on Applied Mathematics 65.4 (2005), pp. 1261–1284.

[Ars06] **Arsigny**, V. et al.: *A Log-Euclidean framework for statistics on diffeomorphisms*. Proc Medical Image Computing and Computer-Assisted Intervention. Vol. LNCS 4190. 2006, pp. 924–931.

[Ash07] **Ashburner**, J.: *A fast diffeomorphic image registration algorithm*. NeuroImage 38.1 (2007), pp. 95–113.

[Bab01] **Babuska**, I. et al.: *The finite element method and its reliability*. Oxford university press, 2001.

[Bak15] **Bakas**, S. et al.: *GLISTRboost: Combining multimodal MRI segmentation, registration, and biophysical tumor growth modeling with gradient boosting machines for glioma segmentation*. Brainlesion 9556 (2015), pp. 144–155.

[Bal16a] **Balay**, S. et al.: *PETSc Users Manual*. Tech. rep. ANL-95/11 - Revision 3.7. Argonne National Laboratory, 2016.

[Bal16b] **Balzani**, D. et al.: *Numerical modeling of fluid–structure interaction in arteries with anisotropic polyconvex hyperelastic and anisotropic viscoelastic material models at finite strains*. International journal for numerical methods in biomedical engineering 32.10 (2016), e02756.

[Ban89] **Banks**, H. T.; **Kunisch**, K.: *Estimation techniques for distributed parameter systems*. Birkhäuser, Boston, MA, 1989.

[Bar16] **Barbu**, V.; **Marinoschi**, G.: *An optimal control approach to the optical flow problem*. Systems & Control Letters 87 (2016), pp. 1–9.

[Bas94] **Basser**, P.; **Mattiello**, J.; **LeBihan**, D.: *Mr diffusion tensor spectroscopy and imaging*. Biophysical Journal 66.1 (1994), pp. 259–267. issn: 0006-3495. https://doi.org/10.1016/S0006-3495(94)80775-1.

[Bat07] **Bathe**, K.-J.; **Ledezma**, G. A.: *Benchmark problems for incompressible fluid flows with structural interactions*. Computers & Structures 85.11-14 (2007), pp. 628–644. issn: 00457949. 10.1016/j.compstruc.2007.01.025.

[Bat95] **Bathe**, K.; **Nitikitpaiboon**, C.; **Wang**, X.: *A mixed displacement-based finite element formulation for acoustic fluid-structure interaction*. Computers & Structures 56.2-3 (1995), pp. 225–237.

[Bau13] **Bauer**, S. et al.: *A survey of MRI-based medical image analysis for brain tumor studies*. Physics in Medicine and Biology 58.13 (2013), R97.

[Baz11] **Bazilevs**, Y. et al.: *3d simulation of wind turbine rotors at full scale. part ii: fluid–structure interaction modeling with composite blades*. International Journal for Numerical Methods in Fluids 65.1-3 (2011), pp. 236–253.

[Beg05] **Beg**, M. F. et al.: *Computing large deformation metric mappings via geodesic flows of diffeomorphisms*. International Journal of Computer Vision 61.2 (2005), pp. 139–157.

[Ben03] **Benson**, S. et al.: *TAO users manual*. Tech. rep. Argonne National Lab., IL (US), 2003.

[Ben11] **Benzi**, M.; **Haber**, E.; **Taralli**, L.: *A preconditioning technique for a class of PDE-constrained optimization problems*. Advances in Computational Mathematics 35.2-4 (2011), pp. 149–173.

[Bie03] **Biegler**, L. T. et al.: *Large-scale PDE-constrained optimization*. Springer, 2003.

[Bir05a] **Biros**, G.; **Ghattas**, O.: *Parallel Lagrange-Newton-Krylov-Schur methods for PDE-constrained optimization—Part I: The Krylov-Schur solver*. SIAM Journal on Scientific Computing 27.2 (2005), pp. 687–713.

[Bir05b] **Biros**, G.; **Ghattas**, O.: *Parallel Lagrange-Newton-Krylov-Schur methods for PDE-constrained optimization—Part II: The Lagrange-Newton solver and its application to optimal control of steady viscous flows*. SIAM Journal on Scientific Computing 27.2 (2005), pp. 714–739.

[Bir13] **Birken**, P. et al.: *Fast solvers for thermal fluid structure interaction*. ECCOMAS Marine V (2013), p. 2.

[Bir99] **Biros**, G.; **Ghattas**, O.: *Parallel Newton-Krylov methods for PDE-constrained optimization*. Proc ACM/IEEE Conference on Supercomputing. 1999, pp. 28–40.

[Blo14a] **Blom**, D. S.; **Zuijlen**, A. H. van; **Bijl**, H.: *Acceleration of strongly coupled fluid-structure interaction with manifold mapping*. Proceedings of the 11th. World Congress on Computational Mechanics, 5th. European Congress on Computational Mechanics, 6th. European Congress on Computational Fluid Dynamics. Ed. by **Oñate**, E.; **Oliver**, X.; **Huerta**, A. 2014, pp. 4484–4495.

[Blo14b] **Blom**, D. S.; **Zuijlen**, A. H. van; **Bijl**, H.: *Acceleration of strongly coupled fluid-structure interaction with manifold mapping*. Proceedings of the 11th. World Congress on Computational Mechanics, 5th. European Congress on Computational Mechanics, 6th. European Congress on Computational Fluid Dynamics. Ed. by **Oñate**, E.; **Oliver**, X.; **Huerta**, A. 2014, pp. 4484–4495.

[Blo15a] **Blom**, D. S.; **Zuijlen**, A. H. van; **Bijl**, H.: *Multi-level acceleration with manifold mapping of strongly coupled partitioned fluid-structure interaction*. Journal of Computational Physics 296 (2015), pp. 211–231.

[Blo15b] **Blom**, D. et al.: *A Review on Fast Quasi-Newton and Accelerated Fixed Point Iterations for Partitioned Fluid-Structure Interaction Simulation*. Englisch. Modeling and Simulation in Science, Engineering and Technology. Springer International Publishing, 2015, pp. 1–12.

[Boe10] **Boer**, A. de; **van Zuijlen**, A. H.; **Bijl**, H.: *Radial Basis Functions for Interface Interpolation and Mesh deformation*. Advanced Computational Methods in Science and Engineering. Vol. 71. Lecture Notes in Computational Science and Engineering. Springer Berlin Heidelberg, 2010. Chap. 6, pp. 143–178. 978-3-642-03343-8.

[Bog12] **Bogaers**, A. E.; **Kok**, S.; **Franz**, T.: *Strongly coupled partitioned fsi using proper orthogonal decomposition* (2012).

[Bog14] **Bogaers**, A. E. J. et al.: *Quasi-Newton methods for implicit black-box FSI coupling*. Computer Methods in Applied Mechanics and Engineering 279 (2014), pp. 113–132.

[Bor02] **Borzì**, A.; **Ito**, K.; **Kunisch**, K.: *Optimal control formulation for determining optical flow*. SIAM Journal on Scientific Computing 24.3 (2002), pp. 818–847.

[Bor12] **Borzì**, A.; **Schulz**, V.: *Computational optimization of systems governed by partial differential equations*. Philadelphia, Pennsylvania, US: SIAM, 2012.

[Bra06] **Brand**, M.: *Fast low-rank modifications of the thin singular value decomposition*. Linear Algebra and its Applications 415.1 (2006), pp. 20–30. issn: 00243795. 10.1016/j.laa.2005.07.021.

[Bra07] **Braess**, D.: *Finite elements: Theory, fast solvers, and applications in solid mechanics*. Cambridge University Press, 2007.

[Bre01] **Brett**, M. et al.: *Spatial normalization of brain images with focal lesions using cost function masking*. NeuroImage 14.2 (2001), pp. 486–500.

[Bre12] **Brekhovskikh**, L. M.; **Goncharov**, V.: *Mechanics of continua and wave dynamics*. Vol. 1. Springer Science & Business Media, 2012.

[Bro14] **Brown**, R. W. et al.: *Magnetic resonance imaging: physical principles and sequence design*. John Wiley & Sons, 2014.

[Bro65] **Broyden**, C. G.: *A class of methods for solving nonlinear simultaneous equations*. Mathematics of computation (1965), pp. 577–593.

[Bro67] **Broyden**, C. G.: *Quasi-newton methods and their application to function minimisation*. Mathematics of Computation 21.99 (1967), pp. 368–381.

[Bro69] **Broyden**, C.: *A new double-rank minimisation algorithm. preliminary report*. Notices of the American Mathematical Society. Vol. 16. 4. AMER MATHEMATICAL SOC 201 CHARLES ST, PROVIDENCE, RI 02940-2213. 1969, p. 670.

[Bro81] **Broit**, C.: *Optimal registration of deformed images*. PhD thesis. Philadelphia, Pennsylvania, US: Computer and Information Science, University of Pennsylvania, 1981.

[Bun16a] **Bungartz**, H.-J. et al.: *Precice – a fully parallel library for multi-physics surface coupling*. Computers and Fluids (2016).

[Bun16b] **Bungartz**, H.-j. et al.: *Partitioned fluid-structure-acoustics interaction on distributed data - coupling via preCICE*. Software for Exascale Computing - SPPEXA 2013-2015. Lecture Notes on Computational Science and Engineering (LNCSE). Springer, Berlin Heidelberg New York, 2016, pp. 239–266.

[Buz08] **Buzug**, T. M.: *Computed tomography: from photon statistics to modern cone-beam CT*. Springer Science & Business Media, 2008.

[Byr03] **Byrne**, H. M. et al.: *A two-phase model of solid tumour growth*. Applied Mathematics Letters 16.4 (2003), pp. 567–573.

[Car98] **Carlson**, N. N.; **Miller**, K.: *Design and application of a gradient-weighted moving finite element code i: in one dimension*. SIAM Journal on Scientific Computing 19.3 (1998), pp. 728–765.

[Cau05] **Causin**, P.; **Gerbeau**, J.-F.; **Nobile**, F.: *Added-mass effect in the design of partitioned algorithms for fluid–structure problems*. Computer methods in applied mechanics and engineering 194.42-44 (2005), pp. 4506–4527.

[Cav07] **Cavagna**, L.; **Quaranta**, G.; **Mantegazza**, P.: *Application of navier–stokes simulations for aeroelastic stability assessment in transonic regime*. Computers & Structures 85.11-14 (2007), pp. 818–832.

[Ceb97] **Cebral**, J. R.; **Lohner**, R.: *Conservative load projection and tracking for fluid-structure problems*. AIAA journal 35.4 (1997), pp. 687–692.

[Che11a] **Chen**, K.: *Optimal control based image sequence interpolation*. PhD thesis. University of Bremen, 2011.

[Che11b] **Chen**, K.; **Lorenz**, D. A.: *Image sequence interpolation using optimal control*. Journal of Mathematical Imaging and Vision 41 (2011), pp. 222–238.

[Che12] **Chen**, K.; **Lorenz**, D. A.: *Image sequence interpolation based on optical flow, segmentation and optimal control*. Image Processing, IEEE Transactions on 21.3 (2012), pp. 1020–1030.

[Chr06] **Christensen**, G. E. et al.: *Introduction to the non-rigid image registration evaluation project*. Proc Biomedical Image Registration. Vol. LNCS 4057. 2006, pp. 128–135.

[Chr94] **Christensen**, G. E.; **Rabbitt**, R. D.; **Miller**, M. I.: *3D brain mapping using a deformable neuroanatomy*. Physics in Medicine and Biology 39.3 (1994), pp. 609–618.

[Chr96] **Christensen**, G. E.; **Rabbitt**, R. D.; **Miller**, M. I.: *Deformable templates using large deformation kinematics*. Image Processing, IEEE Transactions on 5.10 (1996), pp. 1435–1447.

[Chu13] **Chung**, K.; **Deisseroth**, K.: *CLARITY for mapping the nverous system*. Nature Methods 10 (2013), pp. 508–513.

[Cla05] **Clatz**, O. et al.: *Realistic simulation of the 3D growth of brain tumors in MR images coupling diffusion with biomechanical deformation*. IEEE Transactions on Medical Imaging 24.10 (2005), pp. 1334–1346.

[Col14] **Colin**, T. et al.: *An inverse problem for the recovery of the vascularization of a tumor*. Journal of Inverse and Ill-posed Problems 22.6 (2014), pp. 759–786.

[Col98] **Collins**, V.: *Gliomas*. Cancer Survey, vol. 32, pp. 37–51 (1998).

[Cos01] **Costa**, K.; **Holmes**, J.; **McCulloch**, A.: *Modeling cardiac mechanical properties in three dimensions*. Philosophical Transactions of the Royal Society A 359 (2001), pp. 1233–1250.

[Cot99] **Cotin, S. and Delingette, H. and Ayache, N.**: *Real-time elastic deformations of soft tissues for surgery simulation*. IEEE Transactions On Visualization And Computer Graphics 5.1 (1999), pp. 62–73.

[Cri03] **Cristini**, V.; **Lowengrub**, J.; **Nie**, Q.: *Nonlinear simulation of tumor growth*. Journal of mathematical biology 46.3 (2003), pp. 191–224.

[Dan76] **Daniel**, J. W. et al.: *Reorthogonalization and stable algorithms for updating the gram-schmidt factorization*. Mathematics of Computation 30.136 (1976), pp. 772–795.

[Dar09] **Darwish**, M.; **Sraj**, I.; **Moukalled**, F.: *A coupled finite volume solver for the solution of incompressible flows on unstructured grids*. Journal of Computational Physics 228.1 (2009), pp. 180–201. issn: 0021-9991. 10.1016/j.jcp.2008.08.027.

[De 07] **De Boer**, A.; **Van der Schoot**, M.; **Bijl**, H.: *Mesh deformation based on radial basis function interpolation*. Computers & structures 85.11-14 (2007), pp. 784–795.

[Deg08a] **Degroote**, J. et al.: *Stability of a coupling technique for partitioned solvers in FSI applications*. Computers and Structures 86 (2008), pp. 2224–2234.

[Deg08b] **Degroote**, J. et al.: *Stability of a coupling technique for partitioned solvers in FSI applications*. Computers & Structures 86.23-24 (2008), pp. 2224–2234. issn: 00457949. 10.1016/j.compstruc.2008.05.005.

[Deg09] **Degroote**, J.; **Bathe**, K.-J.; **Vierendeels**, J.: *Performance of a new partitioned procedure versus a monolithic procedure in fluid-structure interaction*. Computers and Structures 87.11-12 (2009), pp. 793–801.

[Deg10] **Degroote**, J.: *Development of algorithms for the partitioned simulation of strongly coupled fluid-structure interaction problems*. PhD thesis. Ghent University, 2010.

[Deg13a] **Degroote**, J.: *Partitioned simulation of fluid-structure interaction*. Archives of Computational Methods in Engineering 20.3 (2013), pp. 185–238.

[Deg13b] **Degroote**, J. et al.: *Partitioned solution of an unsteady adjoint for strongly coupled fluid-structure interactions and application to parameter identification of a one-dimensional problem*. Structural and Multidisciplinary Optimization 47.1 (2013), pp. 77–94.

[Del03] **Del Pino**, S.; **Pironneau**, O.: *A fictitious domain based general pde solver*. Numerical methods for scientific computing variational problems and applications, Barcelona (2003).

[Del11] **Delfour**, M. C.; **Zolâsio**, J.-P.: *Shapes and geometries: metrics, analysis, differential calculus, and optimization*. Vol. 22. Siam, 2011.

[Dem82] **Dembo**, R. S.; **Eisenstat**, S. C.; **Steihaug**, T.: *Inexact newton methods*. SIAM Journal on Numerical Analysis 19.2 (1982), pp. 400–408. 10.1137/0719025.

[Den68] **Dennis** Jr., J. E.: *On newton-like methods*. Numer. Math. 11.4 (1968), pp. 324–330. issn: 0029-599X. 10.1007/BF02166685.

[Den77] **Dennis** Jr, J. E.; **Moré**, J. J.: *Quasi-newton methods, motivation and theory*. SIAM review 19.1 (1977), pp. 46–89.

[Dep06] **Deparis**, S. et al.: *Fluid–structure algorithms based on steklov–poincaré operators*. Computer Methods in Applied Mechanics and Engineering 195.41-43 (2006), pp. 5797–5812.

[Din06] **Diniz dos Santos**, N.; **Gerbeau**, J.; **Bourgat**, J.: *Partitioned fsi strategy for simulations of a thin elastic valve*. Proceedings of ECCOMAS CFD, The Netherlands, September 5-8, 2006. 2006.

[Dok89] **Dokainish**, M.; **Subbaraj**, K.: *A survey of direct time-integration methods in computational structural dynamics—i. explicit methods*. Computers & Structures 32.6 (1989), pp. 1371–1386.

[Dum07] **Dumont**, K. et al.: *Comparison of the hemodynamic and thrombogenic performance of two bileaflet mechanical heart valves using a cfd/fsi model*. Journal of biomechanical engineering 129.4 (2007), pp. 558–565.

[Dup98] **Dupuis**, P.; **Gernander**, U.; **Miller**, M. I.: *Variational problems on flows of diffeomorphisms for image matching*. Quarterly of Applied Mathematics 56.3 (1998), pp. 587–600.

[Ech05] **Echeverria**, D.; **Hemker**, P. W.: *Space mapping and defect correction*. Computational Methods in Applied Mathematics Comput. Methods Appl. Math. 5.2 (2005), pp. 107–136.

[Eis96] **Eisentat**, S. C.; **Walker**, H. F.: *Choosing the forcing terms in an inexact Newton method*. SIAM Journal on Scientific Computing 17.1 (1996), pp. 16–32.

[Eng15] **Engwer**, C. et al.: *Glioma follow white matter tracts: a multiscale dti-based model*. Journal of mathematical biology 71.3 (2015), pp. 551–582.

[Eng96] **Engl**, H.; **Hanke**, M.; **Neubauer**, A.: *Regularization of inverse problems*. Dordrecht, NL: Kluwer Academic Publishers, 1996.

[Ewi01] **Ewing**, R. E.; **Wong**, H.: *A summary of numerical methods for time-dependent advection-dominated partial differential equations*. Journal of Computational and Applied Mathematics 128.1-2 (2001), pp. 423–445.

[Fan08] **Fang**, H.-R.; **Saad**, Y.: *Two classes of multisecant methods for nonlinear acceleration*. Numerical Linear Algebra with Applications 16 (3 2008), pp. 197–221. 10.1002/nla.617.

[Far00] **Farhat**, C.; **Lesoinne**, M.: *Two efficient staggered algorithms for the serial and parallel solution of three-dimensional nonlinear transient aeroelastic problems*. Computer methods in applied mechanics and engineering 182.3-4 (2000), pp. 499–515.

[Far03] **Farhat**, C.; **Geuzaine**, P.; **Brown**, G.: *Application of a three-field nonlinear fluid–structure formulation to the prediction of the aeroelastic parameters of an f-16 fighter*. Computers & Fluids 32.1 (2003), pp. 3–29.

[Far06] **Farhat**, C.; **Van der Zee**, K. G.; **Geuzaine**, P.: *Provably second-order time-accurate loosely-coupled solution algorithms for transient nonlinear computational aeroelasticity*. Computer methods in applied mechanics and engineering 195.17-18 (2006), pp. 1973–2001.

[Fel01] **Felippa**, C. A.; **Park**, K.; **Farhat**, C.: *Partitioned analysis of coupled mechanical systems*. Computer methods in applied mechanics and engineering 190.24-25 (2001), pp. 3247–3270.

[Fer01] **Ferrant, M. and Nabavi, A. and Macq, B. and Jolesz, F.A. and Kikinis, R. and Warfield, S.K.**: *Registration of 3-D intraoperative MR images of the brain using a finite-element biomechanical model*. IEEE Transactions On Medical Imaging 20.12 (2001), pp. 1384–1397.

[Fer05] **Fernández**, M. A.; **Moubachir**, M.: *A Newton method using exact jacobians for solving fluid-structure coupling*. Computers and Structures 83 (2005), pp. 127–142. issn: 00457949. 10.1016/j.compstruc.2004.04.021.

[Fis02] **Fischer**, B.; **Modersitzki**, J.: *Fast diffusion registration*. Contemporary Mathematics 313 (2002), pp. 117–129.

[Fis03] **Fischer**, B.; **Modersitzki**, J.: *Curvature based image registration*. Journal of Mathematical Imaging and Vision 18.1 (2003), pp. 81–85.

[Fis08] **Fischer**, B.; **Modersitzki**, J.: *Ill-posed medicine – an introduction to image registration*. Inverse Problems 24.3 (2008), pp. 1–16.

[Fle12] **Fletcher**, C. A.: *Computational techniques for fluid dynamics 2: Specific techniques for different flow categories*. Springer Science & Business Media, 2012.

[Fle63] **Fletcher**, R.; **Powell**, M. J. D.: *A rapidly convergent descent method for minimization*. The Computer Journal 6.2 (1963), pp. 163–168. 10.1093/comjnl/6.2.163.

[Fle70] **Fletcher**, R.: *A new approach to variable metric algorithms*. The computer journal 13.3 (1970), pp. 317–322.

[Fle91] **Fletcher**, C. A.: *Computational techniques for fluid dynamics 1: fundamental and general techniques*. NY: Springer (1991).

[For01] **Formaggia**, L. et al.: *On the coupling of 3d and 1d navier–stokes equations for flow problems in compliant vessels*. Computer methods in applied mechanics and engineering 191.6-7 (2001), pp. 561–582.

[Fri05] **Frigo**, M.; **Johnson**, S. G.: *The design and implementation of FFTW3*. Proc of the IEEE. vol. 93. 2. 2005, pp. 216–231.

[Fri17] **Frigo**, M.; **Johnson**, S. G.: *FFTW home page*. 2017. http://www.fftw.org.

[G68] **G.**, S.: *On the construction and comparison of difference schemes*. SIAM Journal on Numerical Analysis 5(3):506–517 (1968).

[Gan13] **Ganine**, V.; **Hills**, N.; **Lapworth**, B.: *Nonlinear acceleration of coupled fluid–structure transient thermal problems by anderson mixing*. International Journal for Numerical Methods in Fluids 71.8 (2013), pp. 939–959.

[Gar15] **Gardner**, D. et al.: *Implicit integration methods for dislocation dynamics*. Modelling and Simulation in Materials Science and Engineering 23.2 (2015), p. 025006.

[Gas92] **Gaspar**, L. E. et al.: *Supratentorial malignant glioma: patterns of recurrence and implications for external beam local treatment*. International Journal of Radiation Oncology* Biology* Physics 24.1 (1992), pp. 55–57.

[Gat14] **Gatzhammer**, B.: *Efficient and Flexible Partitioned Simulation of Fluid-Structure Interactions*. PhD thesis. Technische Universität München, 2014, p. 261.

[Ger01] **Gering**, D. et al.: *An integrated visualization system for surgical planning and guidance using image fusion and an open MR*. Journal of Magnetic Resonance Imaging 13.6 (2001), pp. 967–975.

[Gho16a] **Gholami**, A.; **Mang**, A.; **Biros**, G.: *An inverse problem formulation for parameter estimation of a reaction-diffusion model of low grade gliomas*. Journal of Mathematical Biology 72.1 (2016), pp. 409–433.

[Gho16b] **Gholami**, A. et al.: *AccFFT: A library for distributed-memory FFT on CPU and GPU architectures*. arXiv e-prints (2016).

[Gho17a] **Gholami**, A.: *Fast Algorithms for Biophysically-Constrained Inverse Problems in Medical imaging*. PhD thesis. The University of Texas at Austin, 2017.

[Gho17b] **Gholami**, A.; **Biros**, G.: *AccFFT home page*. 2017. http://www.accfft.org.

[Gho17c] **Gholami**, A. et al.: *A framework for scalable biophysics-based image analysis*. Proc ACM/IEEE Conference on Supercomputing. 19. 2017, 19:1–19:13.

[Gie03] **Giese**, A. et al.: *Cost of migration: invasion of malignant gliomas and implications for treatment*. Journal of Clinical Oncology 21.8 (2003), pp. 1624–1636. 10.1200/JCO.2003.05.063.

[Gie96] **Giesexs**, A.; **Westphal**, M.: *Glioma invasion in the central nervous system*. Neurosurgery 39.2 (1996), pp. 235–252.

[Gil81] **Gill**, P. E.; **Murray**, W.; **Wright**, M. H.: *Practical optimization*. Waltham, Massachusetts, US: Academic Press, 1981.

[Gil89] **Gilbert**, J. C.; **Lemaréchal**, C.: *Some numerical experiments with variable-storage quasi-newton algorithms*. Mathematical Programming 45.1 (1989), pp. 407–435. issn: 1436-4646. 10.1007/BF01589113.

[Glü01] **Glück**, M. et al.: *Computation of fluid–structure interaction on lightweight structures*. Journal of Wind Engineering and Industrial Aerodynamics 89.14-15 (2001), pp. 1351–1368.

[Gol70] **Goldfarb**, D.: *A family of variable-metric methods derived by variational means*. Mathematics of computation 24.109 (1970), pp. 23–26.

[Gon13] **Gong**, P. et al.: *A general iterative shrinkage and thresholding algorithm for non-convex regularized optimization problems*. CoRR abs/1303.4434 (2013).

[Goo11] **Gooya**, A. et al.: *Joint segmentation and deformable registration of brain scans guided by a tumor growth model*. Medical Image Computing and Computer-Assisted Intervention – MICCAI 2011. Vol. 6892. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2011, pp. 532–540.

[Goo13] **Gooya**, A. et al.: *GLISTR: Glioma image segmentation and registration*. Medical Imaging, IEEE Transactions on 31.10 (2013), pp. 1941–1954.

[Gra03] **Grama**, A. et al.: *An Introduction to parallel computing: Design and analysis of algorithms*. Second. Addison Wesley, 2003.

[Gra08] **Graczyk**, M.; **Moan**, T.: *A probabilistic assessment of design sloshing pressure time histories in lng tanks*. Ocean Engineering 35.8-9 (2008), pp. 834–855.

[Gre98a] **Grenander**, U.; **Miller**, M. I.: *Computational anatomy: An emerging discipline*. Quarterly of Applied Mathematics 56.4 (1998), pp. 617–694.

[Gre98b] **Gresho**, P. M.; **Sani**, R. L.: *Incompressible flow and the finite element method. volume 1: advection-diffusion and isothermal laminar flow* (1998).

[Gri92] **Griewank**, A.: *Achieving logarithmic growth of temporal and spatial complexity in teverse automatic differentiation*. Optimization Methods and Software 1 (1992), pp. 35–54.

[Gri97] **Griebel**, M.; **Dornseifer**, T.; **Neunhoeffer**, T.: *Numerical simulation in fluid dynamics: a practical introduction*. Vol. 3. Siam, 1997.

[Gu12] **Gu**, S. et al.: *Applying a patient-specific bio-mathematical model of glioma growth to develop virtual [18F]-FMISO-PET images*. Mathematical Medicine and biology 29.1 (2012), pp. 31–48.

[Gun03] **Gunzburger**, M. D.: *Perspectives in flow control and optimization*. Philadelphia, Pennsylvania, US: SIAM, 2003.

[Hab03] **Habib**, S.; **Molina-Paris**, C.; **Deisboeck**, T. S.: *Complex dynamics of tumors: modeling an emerging brain tumor system with coupled reaction–diffusion equations*. Physica A: Statistical Mechanics and its Applications 327.3-4 (2003), pp. 501–524.

[Had02] **Hadamard**, J.: *Sur les problemes aux derive espartielles et leur signification physique*. Bulletin of Princeton University 13 (1902), pp. 1–20.

[Hae] **Haelterman**, R. et al.: *Quasi-newton methods for the acceleration of multi-physics codes* ().

[Hae09a] **Haelterman**, R.: *Analytical study of the Least Squares Quasi-Newton method for interaction problems*. PhD thesis. Ghent University, 2009.

[Hae09b] **Haelterman**, R. et al.: *The Quasi-Newton Least Squares Method: A New and Fast Secant Method Analyzed for Linear Systems*. SIAM Journal on Numerical Analysis 47.3 (2009), pp. 2347–2368. issn: 0036-1429. 10.1137/070710469.

[Hae10] **Haelterman**, R. et al.: *On the similarities between the Quasi-Newton inverse least squares method and GMRES*. SIAM Journal on Numerical Analysis 47.6 (2010), pp. 4660–4679.

[Hae15] **Haelterman**, R. et al.: *Improving the performance of the partitioned qn-ils procedure for fluid-structure interaction problems: filtering*. Computers and Structures (2015).

[Hag00] **Hager**, W. W.: *Runge-Kutta methods in optimal control and the transformed adjoint system*. Numerische Mathematik 87 (2000), pp. 247–282.

[Hai93] **Hairer**, E.; **Nørsett**, S. P.; **Wanner**, G.: *Solving ordinary differential equations. i, volume 8 of springer series in computational mathematics*. 1993.

[Hai96] **Hairer**, E.; **Wanner**, G.: *Solving ordinary differential equations ii: stiff and differential-algebraic problems second revised edition with 137 figures*. Springer Series in Computational Mathematics 14 (1996).

[Har07] **Harpold**, H. L. P.; **Alvord**, E. C.; **Swanson**, K. R.: *The evolution of mathematical modeling of glioma proliferation and invasion*. Journal of Neuropathology & Experimental Neurology 66.1 (2007), pp. 1–9.

[Har09] **Hart**, G. L.; **Zach**, C.; **Niethammer**, M.: *An optimal control approach for deformable registration*. Proc IEEE Conference on Computer Vision and Pattern Recognition. 2009, pp. 9–16.

[Haw13] **Hawkins-Daarud**, A. et al.: *Modeling tumor-associated edema in gliomas during anti-angiogenic therapy and its impact on imageable tumor*. Frontiers in Oncology 3.66 (2013), pp. 1–12.

[Hei05] **Heinkenschloss**, M.: *A time-domain decomposition iterative method for the solution of distributed linear quadratic optimal control problems*. Journal of Computational and Applied Mathematics 1.1 (2005), pp. 169–198.

[Hen04] **Henn**, S.; **Hömke**, L.; **Witsch**, K.: *Lesion preserving image registration with application to human brains*. Lect Notes Comput Sc. Vol. 3175. Lect Notes Comput Sc. 2004, pp. 496–503.

[Hen13a] **Hennig**, P.: *Fast probabilistic optimization from noisy gradients*. International conference on machine learning. 2013, pp. 62–70.

[Hen13b] **Hennig**, P.; **Kiefel**, M.: *Quasi-newton method: a new direction*. Journal of Machine Learning Research 14.Mar (2013), pp. 843–865.

[Her09] **Hernandez**, M.; **Bossa**, M. N.; **Olmos**, S.: *Registration of anatomical images using paths of diffeomorphisms parameterized with stationary vector field flows*. International Journal of Computer Vision 85.3 (2009), pp. 291–306.

[Her10] **Herzog**, R.; **Kunisch**, K.: *Algorithms for PDE-constrained optimization*. GAMM Mitteilungen 33.2 (2010), pp. 163–176.

[Her18] **Herzog**, R.; **Pearson**, J. W.; **Stoll**, M.: *Fast iterative solvers for an optimal transport problem*. ArXiv e-prints (2018). "arXiv": 1801.04172.

[Hin09a] **Hinkle**, J. et al.: *4D MAP image reconstruction incorporating organ motion*. Proc Information Processing in Medical Imaging. LNCS 5636. 2009, pp. 676–687.

[Hin09b] **Hinze**, M. et al.: *Optimization with PDE constraints*. Berlin, DE: Springer, 2009.

[Hog06] **Hogea**, C. et al.: *A framework for soft tissue simulations with applications to modeling brain tumor mass-effect in 3D images*. Medical Image Computing and Computer-Assisted Intervention Workshop on Biomechanics. Copenhagen, 2006.

[Hog07a] **Hogea**, C. S. et al.: *A robust framework for soft tissue simulations with application to modeling brain tumor mass effect in 3D MR images*. Physics in Medicine and Biology 52.23 (2007), p. 6893.

[Hog07b] **Hogea**, C.; **Davatzikos**, C.; **Biros**, G.: *Modeling glioma growth and mass effect in 3D MR images of the brain*. Lect Notes Comput Sc. Vol. 4791. 2007, pp. 642–650.

[Hog08a] **Hogea**, C.; **Davatzikos**, C.; **Biros**, G.: *Brain-tumor interaction biophysical models for medical image registration*. SIAM Journal on Imaging Sciences 30.6 (2008), pp. 3050–3072.

[Hog08b] **Hogea**, C.; **Davatzikos**, C.; **Biros**, G.: *An image-driven parameter estimation problem for a reaction-diffusion glioma growth model with mass effects*. Journal of Mathematical Biology 56.6 (2008), pp. 793–825.

[Hol01] **Holland**, E. C.: *Progenitor cells and glioma formation*. Current opinion in neurology 14.6 (2001), pp. 683–688.

[Hur08] **Hurley**, N. P.; **Rickard**, S. T.: *Comparing measures of sparsity*. CoRR abs/0811.4706 (2008).

[Jac15] **Jackson**, P. R. et al.: *Patient-specific mathematical neuro-oncology: Using a simple proliferation and invasion tumor model to inform clinical practice*. Bulletin of Mathematical Biology 77.5 (2015), pp. 846–856.

[Kak15] **Kakinuma**, R. et al.: *Ultra-high-resolution computed tomography of the lung: image quality of a prototype scanner*. PloS one 10.9 (2015), e0137165.

[Kam14] **Kamensky**, D. et al.: *A variational immersed boundary framework for fluid-structure interaction: isogeometric implementation and application to bioprosthetic heart valves*. Computer Methods in Applied Mechanics and Engineering (2014), p. 2.

[Kan00] **Kansal**, A. R. et al.: *Simulated brain tumor growth dynamics using a three-dimensional cellular automaton*. Journal of theoretical biology 203.4 (2000), pp. 367–382.

[Kel87a] **Kelley**, C.; **Sachs**, E.: *Quasi-newton methods and unconstrained optimal control problems*. SIAM Journal on Control and Optimization 25.6 (1987), pp. 1503–1516.

[Kel87b] **Kelly**, P. J. et al.: *Imaging-based stereotaxic serial biopsies in untreated intracranial glial neoplasms*. Journal of neurosurgery 66.6 (1987), pp. 865–874.

[Kel87c] **Kelly**, P. J. et al.: *Stereotactic histologic correlations of computed tomography-and magnetic resonance imaging-defined abnormalities in patients with glial neoplasms*. Mayo Clinic Proceedings. Vol. 62. 6. Elsevier. 1987, pp. 450–459.

[Kno04] **Knoll**, D. A.; **Keyes**, D. E.: *Jacobian-free newton–krylov methods: a survey of approaches and applications*. Journal of Computational Physics 193.2 (2004), pp. 357–397.

[Kno13] **Knopoff**, D. A. et al.: *Adjoint method for a tumor growth PDE-constrained optimization problem*. Computers & Mathematics with Applications 66.6 (2013), pp. 1104–1119.

[Kno17] **Knopoff**, D. et al.: *A mathematical method for parameter estimation in a tumor growth model*. Computational and Applied Mathematics 36.1 (2017), pp. 733–748.

[Kon09] **Konukoglu**, E.: *Modeling Glioma Growth and Personalizing Growth Models in Medical Images*. Theses. Université Nice Sophia Antipolis, 2009.

[Kon10a] **Konukoglu**, E. et al.: *Extrapolating glioma invasion margin in brain magnetic resonance images: Suggesting new irradiation margins*. Medical Image Analysis 14.2 (2010), pp. 111–125.

[Kon10b] **Konukoglu**, E. et al.: *Image guided personalization of reaction-diffusion type tumor growth models using modified anisotropic eikonal equations*. Medical Imaging, IEEE Transactions on 29.1 (2010), pp. 77–95.

[Küt08] **Küttler**, U.; **Wall**, W. A.: *Fixed-point fluid–structure interaction solvers with dynamic relaxation*. Computational Mechanics 43.1 (2008), pp. 61–72.

[Kut16] **Kutten**, K. S. et al.: *A diffeomorphic approach to multimodal registration with mutual information: Applications to CLARITY mouse brain images*. ArXiv e-prints (2016).

[Kut17] **Kutten**, K. S. et al.: *A diffeomorphic approach to multimodal registration with mutual information: Applications to CLARITY mouse brain images*. Proc Medical Image Computing and Computer-Assisted Intervention. Vol. LNCS 10433. 2017, pp. 275–282.

[Kwo14] **Kwon**, D. et al.: *PORTR: Pre-operative and post-recurrence brain tumor registration*. Medical Imaging, IEEE Transactions on 33.3 (2014), pp. 651–667.

[Le 01] **Le Bihan**, D. et al.: *Diffusion tensor imaging: concepts and applications*. Journal of magnetic resonance imaging 13.4 (2001), pp. 534–546.

[Le 05] **Le Tallec**, P. et al.: *Fluid structure interaction problems in large deformation*. Comptes Rendus Mecanique 333.12 (2005), pp. 910–922.

[Le16] **Le**, M. et al.: *MRI based Bayesian personalization of a tumor growth model*. IEEE Transactions on Medical Imaging 35.10 (2016), pp. 2329–2339.

[Lee10] **Lee**, E.; **Gunzburger**, M.: *An optimal control formulation of an image registration problem*. Journal of Mathematical Imaging and Vision 36.1 (2010), pp. 69–80.

[Lee11] **Lee**, E.; **Gunzburger**, M.: *Analysis of finite element discretization of an optimal control formulation of the image registration problem*. SIAM Journal on Numerical Analysis 49.4 (2011), pp. 1321–1349.

[Les98] **Lesoinne**, M.; **Farhat**, C.: *Higher-order subiteration-free staggered algorithm for nonlinear transient aeroelastic problems*. AIAA journal 36.9 (1998), pp. 1754–1757.

[LeV02] **LeVeque**, R. J.: *Finite volume methods for hyperbolic problems*. Vol. 31. Cambridge university press, 2002.

[Li10] **Li**, X.: *Registration of images with varying topology using embedded maps*. PhD thesis. Blacksburg, Virginia, US: Department of Electrical & Computer Engineering, Virginia Polytechnic Institute and State University, 2010.

[Li12] **Li**, X. et al.: *Registration of images with varying topology using embedded maps*. IEEE T Med Imaging 31.3 (2012), pp. 749–765.

[Lia98] **Liang**, B. C.; **Weil**, M.: *Locoregional approaches to therapy with gliomas as the paradigm*. Current opinion in oncology 10.3 (1998), pp. 201–206.

[Lim16] **Lima**, E. A. B. F. et al.: *Selection, calibration, and validation of models of tumor growth*. Mathematical Models and Methods in Applied Sciences 26.12 (2016), pp. 2341–2368.

[Lin15] **Lindner**, F. et al.: *A comparison of various quasi-Newton schemes for partitioned fluid-structure interaction*. Proceedings of 6th International Conference on Computational Methods for Coupled Problems in Science and Engineering. Venice, 2015, pp. 1–12.

[Lio72] **Lions**, J.-L.: *Some aspects of the optimal control of distributed parameter systems*. Philadelphia, Pennsylvania, US: SIAM, 1972.

[Liu14] **Liu**, Y. et al.: *Patient specific tumor growth prediction using multimodal images*. Medical Image Analysis 18.3 (2014), pp. 555–566.

[Lof15] **Loffeld**, J.; **Woodward**, C. S.: *Considerations on the implementation and use of Anderson acceleration on distributed memory and GPU-based parallel computers*. Tech. rep. Technical Report LLNLPROC-675918, Lawrence Livermore National Laboratory, 2015.

[Lom13] **Lombardi**, M.; **Parolini**, N., et al.: *Unsteady fsi simulations of downwind sails*. V International Conference on Computational Methods in Marine Engineering, MARINE 2013. International Center for Numerical Methods in Engineering (CIMNE). 2013, pp. 266–277.

[Loo06] **Loon**, R. van; **Anderson**, P. D.; **Vosse**, F. N. van de: *A fluid–structure interaction method with solid-rigid contact for heart valve dynamics*. Journal of computational physics 217.2 (2006), pp. 806–823.

[Lot12] **Lott**, P. et al.: *An accelerated Picard method for nonlinear systems related to variably saturated flow*. Advances in Water Resources 38 (2012), pp. 92–101. issn: 03091708. 10.1016/j.advwatres.2011.12.013.

[Lot13] **Lothode**, C. et al.: *Fluid-structure interaction analysis of an hydrofoil*. ECCOMAS Marine V (2013), p. 2.

[Lou07] **Louis**, D. N. et al.: *The 2007 who classification of tumours of the central nervous system*. Acta Neuropathologica 114.2 (2007), pp. 97–109. 10.1007/s00401-007-0243-4.

[Lüs13] **Lüsebrink**, F.; **Wollrab**, A.; **Speck**, O.: *Cortical thickness determination of the human brain using high resolution 3T and 7T MRI data*. Neuroimage 70 (2013), pp. 122–131.

[Lyn80] **Lynch**, D. R.; **O'Neill**, K.: *Elastic grid deformation for moving boundary problems in two space dimensions*. Finite elements in water resources 2 (1980).

[M M98] **M. Miga**; **K. Paulsen**; **al.**: *Initial in-vivo analysis of 3d heterogeneous brain computations for model-updated image-guided neurosurgery*. LNCS 1496 (1998), pp. 743–752.

[Ma13] **Ma**, X. et al.: *Image-based fluid–structure interaction model of the human mitral valve*. Computers & Fluids 71 (2013), pp. 417–425.

[Mam95] **Maman**, N.; **Farhat**, C.: *Matching fluid and structure meshes for aeroelastic computations: a parallel approach*. Computers & Structures 54.4 (1995), pp. 779–785.

[Man06] **Mandonnet**, E.; **Capelle**, L.; **Duffau**, H.: *Extension of paralimbic low grade gliomas: toward an anatomical classification based on white matter invasion patterns*. Journal of Neuro-Oncology 78.2 (2006), pp. 179–185. issn: 1573-7373. 10.1007/s11060-005-9084-y.

[Man12] **Mang**, A. et al.: *Biophysical modeling of brain tumor progression: from unconditionally stable explicit time integration to an inverse problem with parabolic PDE constraints for model calibration*. Medical Physics 39.7 (2012), pp. 4444–4459.

[Man14] **Mang**, A.: *Methoden zur numerischen Simulation der Progression von Gliomen: Modellentwicklung, Numerik und Parameteridentifikation [in german; Methods for the numerical simulation of the progression of glioma: Model design, numerics and parameter identification]*. Springer Fachmedien Wiesbaden, 2014.

[Man15] **Mang**, A.; **Biros**, G.: *An inexact Newton–Krylov algorithm for constrained diffeomorphic image registration*. SIAM Journal on Imaging Sciences 8.2 (2015), pp. 1030–1069.

[Man16a] **Mang**, A.; **Biros**, G.: *Constrained H^1-regularization schemes for diffeomorphic image registration*. SIAM Journal on Imaging Sciences 9.3 (2016), pp. 1154–1194.

[Man16b] **Mang**, A.; **Gholami**, A.; **Biros**, G.: *Distributed-memory large-deformation diffeomorphic 3D image registration*. Proc ACM/IEEE Conference on Supercomputing. 2016.

[Man17a] **Mang**, A.; **Biros**, G.: *A semi-Lagrangian two-level preconditioned Newton–Krylov solver for constrained diffeomorphic image registration*. SIAM Journal on Scientific Computing 39.6 (2017), B1064–B1101.

[Man17b] **Mang**, A.; **Ruthotto**, L.: *A Lagrangian Gauss–Newton–Krylov solver for mass- and intensity-preserving diffeomorphic image registration*. SIAM Journal on Scientific Computing 39.5 (2017), B860–B885.

[Man17c] **Mang**, A. et al.: *SIBIA-GlS: Scalable Biophysics-Based Image Analysis for Glioma Segmentation*. Englisch. The multimodal brain tumor image segmentation benchmark (BRATS), MICCAI. -, 2017, pp. 202–209.

[Man17d] **Mang**, A. et al.: *PDE constrained optimization in medical image analysis*. Optimization and Engineering (2017). in review.

[Man18a] **Mang**, A. et al.: *Claire: a distributed-memory solver for constrained large deformation diffeomorphic image registration*. ArXiv e-prints (2018). "arXiv": 1808.04487.

[Man18b] **Mang**, A. et al.: *Pde-constrained optimization in medical image analysis*. ArXiv e-prints (2018). "arXiv": 1803.00058.

[Mar05] **Markert**, J. et al.: *Gliobalastoma multiforme*. Burlington, MA: Jones Bartlett (2005).

[Mar08] **Marks**, L.; **Luke**, D.: *Robust mixing for ab initio quantum mechanical calculations*. Physical Review B 78.7 (2008), p. 075114.

[Mar12] **Martin**, J. et al.: *A stochastic Newton MCMC method for large-scale statistical inverse problems with application to seismic inversion*. SIAM Journal on Scientific Computing 34.3 (2012), A1460–A1487.

[Men11] **Menze**, B. H. et al.: *A generative approach for image-based modeling of tumor growth*. Proc Information Processing in Medical Imaging. Vol. 22. 2011, pp. 735–747.

[Men15] **Menze**, B. H. et al.: *The multimodal brain tumor image segmentation benchmark (BRATS)*. IEEE Transactions on Medical Imaging 34.10 (2015), pp. 1993–2024.

[Mic04] **Michler**, C.: *An interface Newton–Krylov solver for fluid–structure interaction*. International journal for numerical methods in fluids 47.10-11 (2004), pp. 1189–1195. issn: 02712091. 10.1002/fld.850.

[Mic11] **Michler**, C.; **Van Brummelen**, H.; **De Borst**, R.: *An investigation of Interface-GMRES(R) for fluid-structure interaction problems with flutter and divergence*. Computational Mechanics 47 (2011), pp. 17–29. issn: 01787675. 10.1007/s00466-010-0519-8.

[Mil01] **Miller**, M. I.; **Younes**, L.: *Group actions, homeomorphism, and matching: A general framework*. International Journal of Computer Vision 41.1/2 (2001), pp. 61–81.

[Mil05] **Miller**, K.: *Nonlinear krylov and moving nodes in the method of lines*. Journal of computational and applied mathematics 183.2 (2005), pp. 275–287.

[Mir16] **Mira**, D. et al.: *Heat transfer effects on a fully premixed methane impinging flame*. Flow, Turbulence and Combustion 97.1 (2016), pp. 339–361.

[Mit05] **Mittal**, R.; **Iaccarino**, G.: *Immersed boundary methods*. Annu. Rev. Fluid Mech. 37 (2005), pp. 239–261.

[Mod04] **Modersitzki**, J.: *Numerical methods for image registration*. New York: Oxford University Press, 2004.

[Mod09] **Modersitzki**, J.: *FAIR: Flexible algorithms for image registration*. Philadelphia, Pennsylvania, US: SIAM, 2009.

[Moh06] **Mohamed**, A.; **Shen**, D.; **Davatzikos**, C.: *Deformable registration of brain tumor images via a statistical model of tumor-induced deformation*. Med Image Anal 10 (2006), pp. 752–763.

[Mor08] **Mori**, S. et al.: *Stereotaxic white matter atlas based on diffusion tensor imaging in an ICBM template*. NeuroImage 40.2 (2008), pp. 570–582.

[Mor94] **Moré**, J. J.; **Thuente**, D. J.: *Line search algorithms with guaranteed sufficient decrease*. ACM Trans. Math. Softw. 20.3 (1994), pp. 286–307. issn: 0098-3500. 10.1145/192115.192132.

[Mos12] **Mosayebi**, P. et al.: *Tumor invasion margin on the Riemannian space of brain fibers*. Medical Image Analysis 16.2 (2012), pp. 361–373.

[Mun14] **Munson**, T. et al.: *Toolkit for Advanced Optimization (TAO) Users Manual*. Tech. rep. ANL/MCS-TM-322 - Revision 3.5. Argonne National Laboratory, 2014. http://www.mcs.anl.gov/tao.

[Mun15] **Munson**, T. et al.: *TAO 3.6 users manual*. Argonne National Laboratory, Mathematics and Computer Science Division. 2015.

[Mur89] **Murray**, J. D.: *Mathematical biology*. Springer-Verlag, New York, 1989.

[Mus09] **Museyko**, O. et al.: *On the application of the Monge-Kantorovich problem to image registration*. SIAM Journal on Imaging Sciences 2.4 (2009), pp. 1068–1097.

[Ngu15] **Nguyen**, V.-T.; **Gatzhammer**, B.: *A fluid structure interactions partitioned approach for simulations of explosive impacts on deformable structures*. International Journal of Impact Engineering 80 (2015), pp. 65–75.

[Ni09] **Ni**, P.: *Anderson acceleration of fixed-point iteration with applications to electronic structure computations*. PhD thesis. Worcester Polytechnic Institute, 2009.

[Noc06] **Nocedal**, J.; **Wright**, S. J.: *Numerical Optimization*. New York, New York, US: Springer, 2006.

[Noc80] **Nocedal**, J.: *Updating quasi-newton matrices with limited storage*. Mathematics of computation 35.151 (1980), pp. 773–782.

[Oos00] **Oosterlee**, C. W.; **Washio**, T.: *Krylov subspace acceleration of nonlinear multigrid with application to recirculating flows*. SIAM Journal on Scientific Computing 21.5 (2000), pp. 1670–1690.

[Pai13] **Painter**, K.; **Hillen**, T.: *Mathematical modelling of glioma growth: the use of diffusion tensor imaging (dti) data to predict the anisotropic pathways of cancer invasion*. Journal of theoretical biology 323 (2013), pp. 25–39.

[Par14] **Parisot**, S. et al.: *Concurrent tumor segmentation and registration with uncertainty-based sparse non-uniform graphs*. Medical Image Analysis 18.4 (2014), pp. 647–659.

[Pet11] **Petra**, N.; **Stadler**, G.: *Model variational inverse problems governed by partial differential equations*. Tech. rep. Texas University at Austin Institute for Computational Engineering and Sciences, 2011.

[Pet14] **Petra**, N. et al.: *A computational framework for infinite-dimensional Bayesian inverse problems Part II: Stochastic Newton MCMC with application to ice sheet flow inverse problems*. SIAM Journal on Scientific Computing 36.4 (2014), A1525–A1555.

[Pip01] **Piperno**, S.; **Farhat**, C.: *Partitioned procedures for the transient solution of coupled aeroelastic problems–part ii: energy transfer analysis and three-dimensional applications*. Computer methods in applied mechanics and engineering 190.24-25 (2001), pp. 3147–3170.

[Pip95] **Piperno**, S.; **Farhat**, C.; **Larrouturou**, B.: *Partitioned procedures for the transient solution of coupled aroelastic problems part i: model problem, theory and two-dimensional application*. Computer methods in applied mechanics and engineering 124.1-2 (1995), pp. 79–112.

[Pir12] **Pironneau**, O.: *Optimal shape design for elliptic systems*. Springer Science & Business Media, 2012.

[Pra09] **Prastawa**, M.; **Buillitt**, E.; **Gerig**, G.: *Simulation of brain tumors in MR images for evaluation of segmentation efficacy*. Medical Image Analysis 13 (2009), pp. 297–311.

[Pri04] **Price**, S. J. et al.: *Tissue signature characterisation of diffusion tensor abnormalities in cerebral gliomas*. European Radiology 14.10 (2004), pp. 1909–1917. 10.1007/s00330-004-2381-6.

[Pri07] **Price**, S. J. et al.: *Predicting patterns of glioma recurrence using diffusion tensor imaging*. European radiology 17.7 (2007), pp. 1675–1684.

[Pul80] **Pulay**, P.: *Convergence acceleration of iterative sequences. the case of scf iteration*. Chemical Physics Letters 73.2 (1980), pp. 393–398.

[Qua15] **Quarteroni**, A.: *Modeling the heart and the circulatory system*. Vol. 14. Springer, 2015.

[Qui15] **Quiroga**, A. A. I. et al.: *Adjoint method for a tumor invasion PDE-constrained optimization problem in 2D using adaptive finite element method*. Applied Mathematics and Computation 270 (2015), pp. 358–368.

[Qui16] **Quiroga**, A. A. I. et al.: *Nonlinear optimization for a tumor invasion PDE model*. Computational and Applied Mathematics (2016), pp. 1–15.

[Qui94] **Quirk**, J. J.: *An alternative to unstructured grids for computing gas dynamic flows around arbitrarily complex two-dimensional bodies*. Computers & fluids 23.1 (1994), pp. 125–142.

[Rah17] **Rahman**, M. M. et al.: *A fully coupled space-time multiscale modeling framework for predicting tumor growth*. Computational Methods in Applied Mechanics and Engineering 320 (2017), pp. 261–286.

[Rek13] **Rekik**, I. et al.: *Tumor growth parameters estimation and source localization from a unique time point: Application to low-grade gliomas*. Computer Vision and Image Understanding 117.3 (2013), pp. 238–249.

[Rie98] **Riemslagh**, K.; **Vierendeels**, J.; **Dick**, E.: *Coupling of a navier-stokes solver and an elastic boundary solver for unsteady problems.* 4th European Computational Fluid Dynamics conference. Wiley. 1998, pp. 1040–1045.

[Rik10] **Riklin-Raviv**, T. et al.: *Segmentation of image ensembles via latent atlases.* Medical image analysis 14.5 (2010), p. 654.

[Roc10] **Rockne**, R. et al.: *Predicting the efficacy of radiotherapy in individual glioblastoma patients in vivo: a mathematical modeling approach.* Physics in Medicine and Biology 55.12 (2010), p. 3271.

[Roh11] **Rohwedder**, T.; **Schneider**, R.: *An analysis for the diis acceleration method used in quantum chemistry calculations.* Journal of mathematical chemistry 49.9 (2011), p. 1889.

[S05] **S.**, J. et al.: *Simulation of anisotropic growth of low-grade gliomas using diffusion tensor imaging.* Magnetic Resonance in Medicine 54(3):616–624 (2005).

[Sar01] **Sarrate**, J.; **Huerta**, A.; **Donea**, J.: *Arbitrary lagrangian–eulerian formulation for fluid–rigid body interaction.* Computer Methods in Applied Mechanics and Engineering 190.24-25 (2001), pp. 3171–3188.

[Sar13] **Saratoon**, T. et al.: *A gradient-based method for quantitative photoacoustic tomography using the radiative transfer equation.* Inverse Problems 29.7 (2013), p. 075006.

[Sat07] **Sathe**, S. et al.: *Fluid–structure interaction modeling of complex parachute designs with the space–time finite element techniques.* Computers & Fluids 36.1 (2007), pp. 127–135.

[Sch] **Scheufele**, K. et al.: *Coupling brain-tumor biophysical models and diffeomorphic image registration.* Computer Methods in Applied Mechanics and Engineering (), pp. 533–567. issn: 0045-7825. https://doi.org/10.1016/j.cma.2018.12.008.

[Sch10] **Schäfer**, F. et al.: *Fluid-structure-acoustic interaction of the flow past a thin flexible structure.* AIAA journal 48.4 (2010), pp. 738–748.

[Sch15] **Scheufele**, K.: *Robust Quasi-Newton Methods for Partitioned Fluid-Structure Simulations.* Master thesis, University of Stuttgart. 2015.

[Sch17] **Scheufele**, K.; **Mehl**, M.: *Robust multi-secant quasi-newton variants for parallel fluid-structure simulations – and other multi-physics applications.* SIAM SISC Copper Mountain Special Issue (2017).

[Sci13] **Sciumè**, G. et al.: *A multiphase model for three-dimensional tumor growth.* New journal of physics 15.1 (2013), p. 015005.

[Ser06a] **Sermesant**, M.; **Delingette**, H.; **Ayache**, N.: *An electromechanical model of the heart for image analysis and simulation.* Medical Imaging, IEEE Transactions on 25.5 (2006), pp. 612–625.

[Ser06b] **Sermesant**, M. et al.: *Cardiac function estimation from MRI using a heart model and data assimilation: Advances and difficulties.* Medical Image Analysis 10.4 (2006), pp. 642–656.

[Sha70] **Shanno**, D. F.: *Conditioning of quasi-newton methods for function minimization.* Mathematics of computation 24.111 (1970), pp. 647–656.

[Shu15] **Shukaev**, A. K.: *Computational Science and Engineering (Int. Master's Program).* PhD thesis. Technische Universität München, 2015.

[Sil97] **Silbergeld**, D. L.; **Chicoine**, M. R.: *Isolation and characterization of human malignant glioma cells from histologically normal brain.* Journal of neurosurgery 86.3 (1997), pp. 525–531.

[Sim12] **Simoncini**, V.: *Reduced order solution of structured linear systems arising in certain PDE-constrained optimization problems.* Computational Optimization and Applications 53.2 (2012), pp. 591–617.

[Sir87] **Sirovich**, L.: *Turbulence and the dynamics of coherent structures. i. coherent structures.* Quarterly of applied mathematics 45.3 (1987), pp. 561–571.

[Sot13] **Sotiras**, A.; **Davatzikos**, C.; **Paragios**, N.: *Deformable medical image registration: A survey.* Medical Imaging, IEEE Transactions on 32.7 (2013), pp. 1153–1190.

[Sta15] **Starosolski**, Z. et al.: *Ultra high-resolution in vivo computed tomography imaging of mouse cerebrovasculature using a long circulating blood pool contrast agent.* Scientific Reports 5.10178 (2015).

[Ste04] **Stefanescu**, R. et al.: *Non-rigid atlas to subject registration with pathologies for conformal brain radiotherapy.* Lect Notes Comput Sc. Vol. 3216. 2004, pp. 704–711.

[Ste05] **Stein**, K. R. et al.: *Simulation of parachute descent and maneuvers*. Conference on Computation of Shell and Spatial Structures. 2005, p. 1.

[Ste97] **Stein**, K. et al.: *Parallel computation of parachute fluid-structure interactions*. 14th Aerodynamic Decelerator Systems Technology Conference. 1997, p. 1505.

[Sun09] **Sundar**, H.; **Davatzikos**, C.; **Biros**, G.: *Biomechanically constrained 4D estimation of mycardial motion*. Proc Medical Image Computing and Computer-Assisted Intervention. Vol. LNCS 5762. 2009, pp. 257–265.

[Swa00] **Swanson**, K. R.; **Alvord**, E. C.; **Murray**, J. D.: *A quantitative model for differential motility of gliomas in grey and white matter*. Cell Proliferation 33.5 (2000), pp. 317–330.

[Swa02] **Swanson**, K. R.; **Alvord**, E. C.; **Murray**, J. D.: *Virtual brain tumours (gliomas) enhance the reality of medical imaging and highlight inadequacies of current therapy*. British Journal of Cancer 86.1 (2002), pp. 14–18.

[Swa08] **Swanson**, K. R.; **Rostomily**, R. C.; **Alvord**, E. C.: *A mathematical modelling tool for predicting survival of individual patients following resection of glioblastoma: A proof of principle*. British Journal of Cancer 98.1 (2008), pp. 113–119.

[Tez07] **Tezduyar**, T. E. et al.: *Modelling of fluid–structure interactions with the space–time finite elements: arterial fluid mechanics*. International Journal for Numerical Methods in Fluids 54.6-8 (2007), pp. 901–922.

[Tom14] **Tomer**, R. et al.: *Advanced CLARITY for rapid and high-resolution imaging of intact tissues*. Nature protocols 9.7 (2014), pp. 1682–1697.

[Tom94] **Tome**, M. F.; **McKee**, S.: *Gensmac: a computational marker and cell method for*. Journal of Computational Physics 110.17 (1994), pp. 1–186.

[Tos06] **Toselli**, A.; **Widlund**, O.: *Domain decomposition methods-algorithms and theory*. Vol. 34. Springer Science & Business Media, 2006.

[Tot15] **Toth**, A.; **Kelley**, C. T.: *Convergence analysis for Anderson acceleration*. SIAM Journal on Numerical Analysis 53.2 (2015), pp. 805–819. issn: 00361429. 10.1137/130919398.

[Tot17] **Toth**, A. et al.: *Local improvement results for anderson acceleration with inaccurate function evaluations*. SIAM Journal on Scientific Computing 39.5 (2017), S47–S65.

[Tov93] **Tovi**, M.: *Mr imaging in cerebral gliomas analysis of tumour tissue components*. Acta radiologica. Supplementum 384 (1993), pp. 1–24.

[Tra99] **Tracqui**, P.; **Mendjeli**, M.: *Modelling three-dimensional growth of brain tumours from time series of scans*. Mathematical Models and Methods in Applied Sciences 9.04 (1999), pp. 581–598.

[Tro98] **Trouvé**, A.: *Diffeomorphism groups and pattern matching in image analysis*. International Journal of Computer Vision 28.3 (1998), pp. 213–221.

[Tur06] **Turek**, S.; **Hron**, J.: *Proposal for numerical benchmarking of fluid-structure interaction between an elastic object and laminar incompressible flow*. Springer, 2006.

[Tur11] **Turek**, S. et al.: *Numerical simulation and benchmarking of a monolithic multigrid solver for fluid-structure interaction problems with application to hemodynamics*. Fluid Structure Interaction II. Springer, 2011, pp. 193–220.

[Uek13a] **Uekermann**, B. et al.: *A parallel, black-box coupling for fluid-structure interaction*. Computational Methods for Coupled Problems in Science and Engineering, COUPLED PROBLEMS 2013. Stanta Eulalia, Ibiza, Spain, 2013.

[Uek13b] **Uekermann**, B. et al.: *A parallel, black-box coupling algorithm for fluid-structure interaction*. Proceedings of 5th International Conference on Computational Methods for Coupled Problems in Science and Engineering. 2013, pp. 1–12.

[Uek16] **Uekermann**, B.: *Partitioned Fluid-Structure Interaction/Multi-Physics on Massively Parallel Systems*. PhD thesis. Technische Universität München, 2016.

[Uni] **University**, N.; **Laboratory**, A. N.: *PnetCDF*. https://trac.mcs.anl.gov/projects/parallel-netcdf.

[Van09a] **Van Brummelen**, E.: *Added mass effects of compressible and incompressible flows in fluid-structure interaction*. Journal of Applied mechanics 76.2 (2009), p. 021206.

[Van09b] **Van Brummelen**, E.: *Added mass effects of compressible and incompressible flows in fluid-structure interaction*. Journal of Applied mechanics 76.2 (2009), p. 021206.

[Via12] **Vialard**, F.-X. et al.: *Diffeomorphic 3D image registration via geodesic shooting using an efficient adjoint calculation*. International Journal of Computer Vision 97 (2012), pp. 229–241.

[Vie07] **Vierendeels**, J. et al.: *Implicit coupling of partitioned fluid–structure interaction problems with reduced order models*. Computers & structures 85.11-14 (2007), pp. 970–976.

[Wal11] **Walker**, H. F.; **Ni**, P.: *Anderson acceleration for fixed-point iterations*. SIAM J. Numer. Anal. 49.4 (2011), pp. 1715–1735. issn: 0036-1429. 10.1137/10078356X.

[Wal99] **Wall**, W. A.: *Fluid-struktur-interaktion mit stabilisierten finiten elementen* (1999).

[Wan91] **Wanner**, G.; **Hairer**, E.: *Solving ordinary differential equations ii*. Stiff and Differential-Algebraic Problems (1991).

[War03] **Warfield, S.K. and Talos, F. and Kemper, C. and O'Donnell, L. and Westin ,C.F. and Wells W.M. and Black, P. Mc L. and Jolesz, F.A. and Kikinis, R.**: *Capturing brain deformation*. Surgery Simulation and Soft Tissue Modeling. International Symposium, IS4TM 2003. Proceedings (Lecture Notes in Comput. Sci. Vol.2673). Juan-Les-Pins, France: Springer-Verlag, 2003, p. 203.

[Was97] **Washio**, T.; **Oosterlee**, C. W.: *Krylov subspace acceleration for nonlinear multigrid schemes*. Electronic Transactions on Numerical Analysis 6.271-290 (1997), pp. 3–1.

[Wei12] **Weizman**, L. et al.: *Automatic segmentation, internal classification, and follow-up of optic pathway gliomas in mri*. Med. Image Anal., vol. 16, pp. 177–188 (2012).

[Wil15] **Wilcox**, L. C. et al.: *Discretely exact derivatives for hyperbolic PDE-constrained optimization problems discretized by the discontinuous Galerkin method*. Journal of Scientific Computing 63.1 (2015), pp. 138–162.

[Wol96] **Wolf**, K.; **Brakkee**, E.: *Coupling fluids and structures codes on mpi*. MPI Developer's Conference, 1996. Proceedings., Second. IEEE. 1996, pp. 130–137.

[Won07] **Wong**, K. et al.: *Integrating functional and structural images for simultaneous cardiac segmentation and deformation recovery*. Medical Image Computing and Computer-Assisted Intervention – MICCAI 2007. Ed. by **Ayache**, N.; **Ourselin**, S.; **Maeder**, A. Vol. 4791. LNCS. Springer, 2007, pp. 270–277.

[Won15] **Wong**, K. C. L. et al.: *Pancreatic tumor growth prediction with multiplicative growth and image-derived motion*. Proc Information Processing in Medical Imaging. Vol. LNCS 9123. 2015, pp. 501–513.

[Woo50] **Woodbury**, M.: *Inverting modified matrices. statistical research group, memo. rep. no. 42* (1950).

[Wüc06] **Wüchner**, R.: *Computational mechanics of form finding and fluid-structure interaction of membrane structures*. PhD thesis. PhD thesis, Technischer Universit/" at M/" unchen, 2006.(Cited on page 1.), 2006.

[Yan13] **Yankeelov**, T. et al.: *Clinically relevant modeling of tumor growth and treatment response*. Science translational medicine 5.187 (2013), 187ps9–187ps9.

[You07] **Younes**, L.: *Jacobi fields in groups of diffeomorphisms and applications*. Quarterly of Applied Mathematics 650.1 (2007), pp. 113–134.

[You10] **Younes**, L.: *Shapes and diffeomorphisms*. Springer, 2010.

[Yu05] **Yu**, Z.: *A dlm/fd method for fluid/flexible-body interactions*. Journal of computational physics 207.1 (2005), pp. 1–27.

[Zac08a] **Zacharaki**, E. I. et al.: *A comparative study of biomechanical simulators in deformable registration of brain tumor images*. IEEE Transactions on Biomedical Engineering 55.3 (2008), pp. 1233–1236.

[Zac08b] **Zacharaki**, E. I. et al.: *Parallel optimization of tumor model parameters for fast registration of brain tumor images*. Proc SPIE Medical Imaging: Image Processing. 2008, 69140K1–69140K10.

[Zac09] **Zacharaki**, E. I. et al.: *Non-diffeomorphic registration of brain tumor images by simulating tissue loss and tumor growth*. NeuroImage 46.3 (2009), pp. 762–774.

[Zie77] **Zienkiewicz**, O. C. et al.: *The finite element method*. Vol. 36. McGraw-hill London, 1977.

All URLs have last been checked on December 4, 2018.

Declaration of Authorship

I, Klaudius Scheufele, declare that this thesis titled, "Coupling Schemes and Inexact Newton for Multi-Physics and Coupled Optimization Problems" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at the University of Stuttgart.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Stuttgart, December 4, 2018

Klaudius Scheufele