Institute of Parallel and Distributed Systems
University of Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Master's Thesis Nr. 14

Robust Quasi-Newton Methods for Partitioned Fluid-Structure Simulations

Klaudius Scheufele

Course of Study:	Computer Science
Examiner:	Prof. Dr. rer. nat. Miriam Mehl
Supervisor:	Prof. Dr. rer. nat. Miriam Mehl
Commenced:	November 03, 2014
Completed:	May 05, 2015
CR-Classification:	G.1.0, G.1.3, G.1.5, G.1.8, G.4

Abstract

In recent years, quasi-Newton schemes have proven to be a robust and efficient way for the coupling of partitioned multi-physics simulations in particular for fluid-structure interaction. The focus of this work is put on the coupling of partitioned fluid-structure interaction, where minimal interface requirements are assumed for the respective field solvers, thus treated as black box solvers. The coupling is done through communication of boundary values between the solvers. In this thesis a new quasi-Newton variant (IQN-IMVJ) based on a multi-vector update is investigated in combination with serial and parallel coupling systems. Due to implicit incorporation of passed information within the Jacobian update it renders the problem dependent parameter of retained previous time steps unnecessary. Besides, a whole range of coupling schemes are categorized and compared comprehensively with respect to robustness, convergence behaviour and complexity. Those coupling algorithms differ in the structure of the coupling, i. e., serial or parallel execution of the field solvers and the used quasi-Newton methods. A more in-depth analysis for a choice of coupling schemes is conducted for a set of strongly coupled FSI benchmark problems, using the in-house coupling library preCICE. The superior convergence behaviour and robust nature of the IQN-IMVJ method compared to well known state of the art methods such as the IQN-ILS method, is demonstrated here. It is confirmed that the multi-vector method works optimal without the need of tuning problem dependent parameters in advance. Furthermore, it appears to be especially suitable in conjunction with the parallel coupling system, in that it yields fairly similar results for parallel and serial coupling. Although we focus on FSI simulation, the considered coupling schemes are supposed to be equally applicable to various kinds of different volume- or surface-coupled problems.

Kurzfassung

In den letzten Jahren haben sich quasi-Newton Verfahren als robuste und effiziente Methode für die Kopplung partitionierter Multiphysik-Simulationen herausgestellt, insbesondere im Bereich der Kopplung von Fluid-Struktur Interaktion. Der Fokus dieser Arbeit liegt auf der Kopplung partitionierten Fluid-Struktur Interaktion mit minimalen Schnittstellenanforderungen an die betreffenden Fluid- oder Struktur-Löser, die daher als black-box Löser angesehen werden können. Die Kopplung an sich wird durch Kommunikation von Randwerten zwischen den Lösern realisiert. Im Rahmen dieser Arbeit wird eine neue quasi-Newton Methode in Kombination mit einem seriellen und parallelen Kopplungssystem untersucht. Diese Variante ist im Gegensatz zu bisherigen Methoden weitestgehend frei von problemabhängigen Parametern wie z. B. der optimalen Menge an wiederverwendeter Information aus vergangenen Zeitschritten, indem alle bisher bekannte Information in einer norm-minimalen und impliziten Art und Weise für die Aufdatierung der Jacobi Matrix verwendet wird. Darüberhinaus betrachten wir in dieser Arbeit ein ganzes Spektrum an Kopplungs-Schemata, die verschiedene quasi-Newton Varianten mit seriellen und parallellen Kopplungsansätzen kombinieren. Eine sorgfältige Klassifizierung sowie ein umfassender Vergleich der Verfahren bezüglich Robustheit, Komplexität und des Konvergenzverhaltens, verschaffen einen guten Überblick. Für eine Auswahl der besten Schemata wird eine eigehendere Analyse anhand einer Reihe anspruchsvoller FSI Anwendungen mit Hilfe der Kopplungs-Bibliothek preCICE durchgeführt. In diesem Zusammenhang gehen wir auf die robuste sowie effiziente Implementierung der Kopplungs-Algorithmen ein. Im Zuge der Experimente zeigt die IQN-IMVJ quasi-Newton Methode ein überlegenes und weitaus robusteres Konvergenzverhalten im Vergleich zu bisherigen Varianten wie beispielsweise die IQN-ILS Methode und arbeitet optimal ohne vorherige Justierung problemabhängiger Parameter. Darüber hinaus ist sie hervorragend für einen parallelen Kopplungsansatz geeeignet und ermöglicht eine effiziente und massiv-parallele Simulation. Die betrachteten Kopplungs-Schemata sind nicht auf die Kopplung von Fluid-Struktur Interaktion Simulation beschränkt, sondern sind vielmehr für alle Arten von oberflächen- oder volumengekoppelten Problemen geeignet.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

With increasing computational power and intensified research for more sophisticated numerical methods a large class of applications involving several physical effects and different physical models, so-called multi-physics applications, have become feasible for numerical simulation. Fluid-structure interaction (FSI) simulation is part of this class of applications, where the simulation comprises the physical model for the fluid as well as for the solid. Naturally, real world scenarios are way better reflected by multi-physics models than single-physics models which allows for a higher accuracy of the respective numerical simulation. A lot of applications require very high precision and accuracy and thus render multi-physics simulations indispensable.

A very prominent example for FSI is the blood flow in the human cardiovascular system [40, 50, 12, 59], where the risk for aneurysms and calcification can only be predicted when fluid and solid dynamics are simulated in combination at a very high accuracy. A further example is the simulation of blood flow in artificial heart valves [20, 58, 21]. In aeronautics, FSI is used to assess flutter [27, 13] in connection with weight minimization of airplanes as flow induced vibrations of structure have a significant impact on the stability and durability of an aircraft [24, 25]. Another challenging problem is the simulation of opening and descending parachutes [6, 43, 40] due to the thin, highly flexible structure and its interaction with free surface flow.

An FSI problem consists of a fluid domain and a structure domain with the fluid-structure interface in between. The resulting multi-physics problem with adjacent, non-overlapping domains can be simulated in a so-called monolithic or in a partitioned way. For the monolithic approach the flow equations and the structure equations as well as their coupling equations are solved simultaneously, which requires dedicated simulation codes for certain FSI scenarios. On the other hand, the partitioned approach solves the fluid and solid equations separately and preserves software modularity and high flexibility. This raises the possibility to use existing, highly sophisticated software codes for the involved single-physics phenomena in a modular way and allows for the easy exchange of solvers depending on the specific needs of the considered scenario. Thus, the partitioned approach is especially attractive if it is either uncertain which effects should be considered within the physical model or if it is unknown which solvers have to be chosen for the single-physics phenomena. Moreover, this approach is advisable if a new multi-physics simulation is to be established within a short development time.

In order to make the most of the modularity and flexibility that comes with the partitioned approach we are restricted to use limited information, i. e., solely boundary information at the interface for the coupling and cannot assume information about solver internals such as discretization details, which is quite realistic if commercial solvers are used. The single-physics solvers are therefore assumed to be black-box solvers delivering certain physical quantities such as velocities, pressures, forces and temperatures. Unsurprisingly, these assumptions render the coupling of fluid and structure within the partitioned approach more challenging. This thesis focuses on the development and evaluation of robust and sophisticated coupling algorithms for the partitioned approach to FSI simulation.

As FSI simulation is a very challenging type of multi-physics simulation and in particular for incompressible fluids it tends to be ill-conditioned and instable, the pure passing of boundary values amongst the single-physics solvers in between time steps is not feasible. In fact, a stable and converged configuration has to be found within each time step which raises the need of an implicit coupling iteration. There are several ways how to realize the respective iterations that differ in terms of the calling order of the single-physics solvers and in terms of the method used to stabilize and accelerate the coupling iteration. In terms of calling order mainly two versions are in use that result in different fixed-point equations, namely a staggered execution of the respective field solvers and a variation that allows for a simultaneous execution of the single-physics solvers. Within this work, we focus on cases in which severe instabilities are present. Thus, a pure fixed-point iteration, i e., a Gauß-Seidel- or Jacobi-type iteration, that has been the common approach for a long time (cf. [26]) is not feasible. Consequently, more sophisticated accelerating methods have to be applied. Regarding the accelerating post-processing methods, we focus on different quasi-Newton methods that also yield the best results.

For stabilization, newly coupling schemes, i. e., the combination of different fixed-point equations for the partitioned coupling with additional accelerating and stabilizing post-processing quasi-Newton methods for the respective iterative solution, are introduced and analyzed. Both, existing and novel coupling algorithms are intelligibly categorized and compared in detail. Most existing coupling methods require the tuning of a set of problem dependent parameters beforehand, e. g., the optimal amount of retained secant information from the past which is a costly and time consuming step. In this thesis a newly and more sophisticated quasi-Newton method, the IQN-IMVJ method that utilizes available secant information from passed time steps in an implicit way is introduced to accelerate the coupling iteration and appears to be optimal without the need of tuning additional parameters. In that it outperforms even state of the art coupling algorithms such as the IQN-ILS quasi-Newton method [17] and shows improved robustness properties, however, at the cost of storing an explicit representation of the Jacobian approximation.
Several quasi-Newton variants are analyzed for different coupling systems and comprehensively compared for a one-dimensional test scenario. The most promising combinations have been implemented and integrated into the in-house partitioned coupling library preCICE. A more in-depth comparison of the IQN-IMVJ quasi-Newton method with existing state of the art coupling algorithms is conducted for a series of more realistic FSI applications and benchmarks.

Outline

The thesis is structured as follows:

Chapter 2 – Physics and Modelling: This chapter comprises an overview of the foundations of (partitioned) FSI simulation needed in this work, i. e., the mathematical models and governing equations for fluid and solid as well as the necessary coupling conditions at the fluid-structure interface. Essentials concerning the spatial and temporal discretization of the mathematical models are added.

Chapter 3 – Fixed-Point Equations for Partitioned Fluid-Structure Coupling: In this chapter, the partitioned coupling approach is explained focusing on some basic ingredients that are crucial for the understanding of the remainder of the thesis. Different fixed-point formulations for the partitioned problem are introduced and several assets and drawbacks are outlined. Basic iterative solvers for the respective fixed-point coupling system such as pure fixed-point iteration and underrelaxation are presented at the close of the chapter.

Chapter 4 – Quasi-Newton Post-Processing Methods: A comprehensive and structured categorization of both, existing and newly quasi-Newton post-processing methods for the iterative solution of the respective fixed-point equations is given in this chapter. This includes a detailed presentation of the different quasi-Newton methods with linkage to relevant literature. At the end, all the considered coupling schemes with respective quasi-Newton methods are summarized.

Chapter 5 – Software and Implementation: After a short introduction to the utilized software and simulation tools, this chapter considers the implementation details for the integration of the newly coupling algorithms into the partitioned coupling library preCICE. Emphasis is put on a robust implementation.

Chapter 6 – Comparison and Evaluation of Implicit Coupling Schemes: All the considered coupling schemes are comprehensively evaluated and compared for a simple one-dimensional test problem. Assets and drawbacks of methods and variants are worked out to identify the most promising, i. e., the most robust and fast combinations for further investigation.

Chapter 7 – Benchmarks and Applications: A subset of the most qualified coupling schemes is evaluated and compared for three realistic FSI applications and benchmarks. In particular, the newly coupling schemes are compared to state of the art methods.

Chapter 8 – Summary and Outlook: A summary of the contributions and findings of this thesis conclude this work, together with some ideas and impulses for further research and improvement.

Chapter 2

Physics and Modelling

Every numerical simulation is based on a mathematical model that tries to describe the underlying physics as good as possible. In fluid-structure interaction this involves the description of the behaviour of fluid dynamics, structural mechanics and the coupling itself. This chapter summarizes the mathematical models of fluid dynamics in Section 2.1 and structural mechanics in Section 2.2, and states the essential coupling conditions in Section 2.3. An overview of the discretization possibilities available to solve for a approximate solution to the mathematical models is given in Section 2.4.
Since the focus of this work is not on modelling, this is ought to be a rough summary of the essential equations and concepts that are frequently used in the following, hence derivations and detailed explanations are omitted. A short paragraph listing the different points of view, i. e., formulation of the models for an observer moving with the simulated material or an observer at a fixed position in time, is given at the beginning for better understanding of physical modelling.

For more information on modelling and discretization details the reader is referred to the given literature. The following is based on the elaboration in [31].

Points of View

The considered point of view when observing phenomena may heavily influence the form of the observations made. While the formulation of observations may be correct in any point of view, certain observations or characteristics appear to have a rather complicated or even simple description, if the point of view is changed. Hence, not all the results have an equal description or are equally simple to interpret in each point of view. There are three important points of view when considering continuum mechanics which are the *Lagrangian Observer*, the *Eulerian-Observer* and the *Lagrangian-Eulerian Observer* which is a mixture of the two preceding ones. For further details see [31, 37].

The **Lagrangian Observer** takes the initial material domain as reference frame and follows its movements and deformations. This is also referred to as the *material description*, since we follow each point in the initial material configuration and measure its properties. This point of view is advantageous when it comes to the description of moving boundaries and it is easy

to access the property history of a material point. However, its application is limited to rather small deformations an thus often applied in structural mechanics.

As opposed to this, the **Eulerian Observer** considers the moving spatial domain as reference domain and measures material properties at fixed positions in the spatial domain, i. e, the observer considers the phenomena from the exterior. With this point of view it is possible to handle arbitrary deformations, however, it lacks a proper description of the boundaries of the observed material which raises the need of special methods to deal with that. The Eulerian Observer is very popular in fluid dynamics, due to usually large deformations.

The so called **Arbitrary Lagrangian-Eulerian Observer** (ALE) is a combination of the approaches mentioned above, trying to combine the advantages of both observers, i e. to be suitable for arbitrary displacements and provide a proper description of moving boundaries. However, this is only possible to some extend when it comes to discretization for practical realization.

The different points of view explained above can be translated into one another. In the face of multi-physics simulations it is possible to have different descriptions for the different physics in the respective domain. The different observers are of major importance when it comes to re-meshing due to large deformation or to ensure a valid mesh at the common interface between two material domains.

In the following, we summarize the necessary fundamentals of fluid dynamics and structural mechanics and state the conditions that have do be met in order to obtain a stable coupling at the interface between fluid and structure. The following is based on the elaboration in [31]. For more details, the reader is referred upon this and the references therein.

2.1 Fluid Dynamics

In this section we briefly consider the physical and mathematical models of fluid dynamics. The field of fluid dynamics is discriminated by different types of flow, that also change the behaviour of the governing equations. Important categories of flow are incompressible or compressible flows, viscous or inviscid flow and laminar or turbulent flow.
We speak of incompressible flow if the density of the respective fluid is constant (or nearly constant over space) or if the speed of sound is large compared to the velocity of the fluid which is the case for fluids with small Mach number. In this thesis incompressible flow is of major importance as it often leads to stability problems when coupled with structural mechanics simulations. Nonetheless, all the coupling variants described later on are equally applicable to compressible flow.

The viscosity determines the rate of deformation of a fluid, given a certain amount of shear force. Inviscid flow is characterized by a very low viscosity and thus nearly vanishing shear forces. Turbulences are highly irregular flow patterns in space and time that occur, if certain flow parameters, summarized by the so called Reynolds number Re, exceed a certain threshold. Here, we merely focus on laminar flow to limit the overall complexity. For further details, the reader is referred to standard text books, e g., [29].

The Mathematical Model: The Navier-Stokes Equations

The flow of a fluid in a domain $\Omega_F \in \mathbb{R}^N$, $N \in \mathbb{N}$ throughout time $t \in [t_0; t_{end}]$ is characterized by the following quantities:

$$\begin{aligned} \boldsymbol{v} &\colon \Omega_F \times [t_0; t_{end}] \to \mathbb{R}^N \quad \text{velocity field,} \\ p &\colon \Omega_F \times [t_0; t_{end}] \to \mathbb{R} \quad \text{pressure,} \\ \rho &\colon \Omega_F \times [t_0; t_{end}] \to \mathbb{R} \quad \text{density.} \end{aligned}$$

For incompressible flow, it is given that changes in density are negligible over space and time and thus $\rho(x,t) = \rho_\infty = \text{const}$. The fundamental conservation laws of mass, momentum and energy[1] finally lead to a system of partial differential equations, called the *Navier-Stokes Equations*

$$\rho \left(\frac{\partial \boldsymbol{v}}{\partial t} + (\boldsymbol{v} \cdot \nabla)\boldsymbol{v} \right) = -\nabla p + \mu \Delta \boldsymbol{v} + \rho \boldsymbol{f} \tag{2.1}$$

$$\nabla \cdot \boldsymbol{v} = 0 \tag{2.2}$$

Here, μ is the shear viscosity of the fluid and $\boldsymbol{f}$ denotes the distributed volume forces such as gravity.
The first equation (2.1) stems from the conservation of momentum, while the second equation (2.2) ensures the conservation of mass, also referred to as continuum equation. A derivation of the above equations is given in [32].

Boundary Conditions

In order to complete the set of equations and set up a well defined initial value problem (IVP), we need to define proper boundary conditions and initial values. To this end, the boundary Γ is usually decomposed in a Dirichlet part Γ_D and a Neumann part Γ_N with $\Gamma = \Gamma_D \cup \Gamma_N$ and $\Gamma_D \cap \Gamma_N = \emptyset$.

The so called Dirichlet boundary conditions fix the velocity to a prescribed value u_D on the Dirichlet-boundary while the Neumann or natural boundary conditions prescribe dynamic variables, reading

$$\boldsymbol{v} = \boldsymbol{v}_D, \quad \text{on } \Gamma_D, \tag{2.3}$$

$$\boldsymbol{\sigma} \cdot \boldsymbol{n} = \boldsymbol{f}_N, \quad \text{on } \Gamma_N. \tag{2.4}$$

Here, $\boldsymbol{f}_N$ denotes the prescribed force and $\boldsymbol{\sigma} = -pI + \boldsymbol{\tau}$ is the so called Cauchy stress tensor, composed of a unidirectional pressure p and a deviatoric stress tensor $\boldsymbol{\tau}$; $\boldsymbol{n}$ denotes the unit normal of the boundary.

Together with the initial conditions for the velocity $\boldsymbol{v}$

$$\boldsymbol{v}(t_0, \cdot) = \boldsymbol{v}_0, \quad \text{in } \Omega_F, \tag{2.5}$$

the set of initial and boundary conditions is complete.

[1] The conservation of energy is not considered here, as compressible flow is not in the focus of this work.

2.2 Structural Mechanics

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., it defines a corresponding stress and strain tensor. Multiple material deformation models exist, such as elastic material[2], anelastic material[3], viscoelastic[4], plastic[5] or hyperelastic[6] material, that describe different deformation properties.

In most practical applications, we are restricted to comparatively small strains due to the risk of material failure like for example cracks. Hence, we have to deal with rather small displacements (compared to fluid dynamics) which allows for the lagrangian point of view to describe the kinematics. However, in FSI applications, the displacements are usually too large to allow for a proper description using the linear-elasticity model. Therefore, we stick to a non-linear geometric description and furthermore assume the continuum to be homogeneous and isotropic.
A mathematical description of these assumptions is given by the Saint-Venant-Kirchhoff material model, that is applied in the following. We state the governing equation of motion using the above material model and the essential model parameters. For details of the derivation and constitutive models, the reader is kindly referred to standard text books [9, 1]. A short derivation is given in [31].

The Mathematical Model

The deformation of a solid $\Omega_S \in \mathbb{R}^N$ throughout time $t \in [t_0; t_{end}]$ is characterized by the displacement field

$$\boldsymbol{u} \colon \Omega_S \times [t_0; t_{end}] \to \mathbb{R}^N.$$

The dynamics of a structure describe its deformation under the influence of a external force. In structural mechanics, a equation of motion can be derived similar to eq. (2.1), based on an equilibrium of forces and Newton's second law $F = m \cdot a$, reading in differential form

$$\rho \left(\frac{\partial^2 \boldsymbol{u}}{\partial t^2} \right) = \nabla \cdot \boldsymbol{S} + \rho \boldsymbol{f} \tag{2.6}$$

Here, $\boldsymbol{S}$ is the 2nd Piola-Kirchhoff stress tensor, which models surface forces and entails the constitutive material model. The density of the solid is depicted by ρ and $\boldsymbol{f}$ represents the

[2] Elastic material satisfies Hooke's Law
[3] Anelastic material almost satisfies Hooke's law, but the applied force additionally induces time-dependent resistive forces.
[4] For viscoelastic material the time-dependent resistive contributions are large and cannot be neglected. The material does not satisfy Hooke's law (elastic hysteresis)
[5] A material is called plastic, if the applied force induces non-linear displacements, i. e., force and displacements are not proportional.
[6] For an hyperelastic material, the force induces displacements in the material following a strain energy density function.

distributed volume force.
The tensor $\boldsymbol{S}$ relates strains and stresses for the hyperelastic Saint-Venant-Kirchhoff model, that realises non-linear deformations for linear elasticity behaviour, and is computed as

$$\boldsymbol{S} = \lambda \cdot tr(\boldsymbol{E})I + 2\mu\boldsymbol{E}. \tag{2.7}$$

The Lagrangian Green strain tensor models the kinematics in terms of non-linear deformation and is given by $\boldsymbol{E} = \frac{1}{2}\left(\boldsymbol{H} + \boldsymbol{H}^T + \boldsymbol{H}^T\boldsymbol{H}\right)$ with the deformation gradient $\boldsymbol{H}$.

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, i. e.,

$$E = \frac{\mu(3\lambda + 2\mu)}{\lambda + \mu} \qquad \nu = \frac{\lambda}{2(\lambda + \mu)} \tag{2.8}$$

Boundary Conditions

Appropriate initial and boundary conditions enhance eq. (2.6) to a well posed initial value problem (IVP). As for fluid dynamics, the boundary is partitioned into Dirichlet and Neumann boundary $\Gamma = \Gamma_D \dot{\cup} \Gamma_N$ where the following boundary conditions are applied

$$\boldsymbol{u} = \boldsymbol{u}_D, \quad \text{on } \Gamma_D, \tag{2.9}$$
$$\boldsymbol{S} \cdot \boldsymbol{n} = \boldsymbol{f}_N, \quad \text{on } \Gamma_N. \tag{2.10}$$

with the fixed displacement $\boldsymbol{u}_D$, the prescribed force vector $\boldsymbol{f}_D$ and the boundary unit normal $\boldsymbol{n}$. Initial conditions for the displacements and the initial velocity

$$\boldsymbol{u}(t_0, \cdot) = \boldsymbol{u}_0, \quad \text{in } \Omega_S, \tag{2.11}$$
$$\frac{\partial \boldsymbol{u}(t_0, \cdot)}{\partial t} = \boldsymbol{v}_0, \quad \text{in } \Omega_S. \tag{2.12}$$

need to be defined for time-dependent problems with the initial displacement $\boldsymbol{u}_0$ and the initial velocity $\boldsymbol{v}_0$.

2.3 Coupling of Fluids and Structures

For a valid coupling of the fluid domain Ω_F and the structure domain Ω_S, we need to employ special boundary conditions at the common interface

$$\Gamma_{FS} = \Gamma_F \cap \Gamma_S,$$

also called *wet-surface*. The following kinematic and dynamic conditions have to be satisfied at the fluid-structure interface Γ_{FS} to set up a proper coupling.

In continuum mechanics it is assumed, that the corresponding fluid or solid entirely fills its domain and that neither gaps nor overlapping of material can occur. From this, one can

conclude the equality of displacements at the fluid-structure interface. Furthermore, it is assumed that fluid molecules next to the solid boundary are bound to it by molecular attractive forces. Consequently, the equality of velocities is enforced at the interface.

Summarized, the *kinematic interface conditions* read as

$$\boldsymbol{x}_F = \boldsymbol{u}_S \quad \text{at } \Gamma_{FS} \quad \text{and} \tag{2.13}$$

$$\boldsymbol{v}_F = \frac{\partial \boldsymbol{u}_S}{\partial t} \quad \text{at } \Gamma_{FS}, \tag{2.14}$$

where $\boldsymbol{x}_F$ denotes the discretized positions of the fluid at the fluid-structure interface. Furthermore, a balance of forces is required at the interface which is expressed point-wise by the *dynamic interface condition* in terms of surface stresses

$$\boldsymbol{\sigma}_F \cdot \boldsymbol{n}_F = -\boldsymbol{\sigma}_S \cdot \boldsymbol{n}_S \tag{2.15}$$

with the interface surface normals $\boldsymbol{n}_F = -\boldsymbol{n}_S$. Note, that $\boldsymbol{\sigma}_S$ corresponds to the 2nd Piola-Kirchhoff stress tensor $\boldsymbol{S}$ from eq. (2.7).

2.4 Discretization of the Mathematical Models

In order to allow the computer to search for a solution of the partial differential equations (PDEs) that describe the behaviour of fluids and structures, they need to be transformed into algebraic equations which is called discretization. During this discretization process a so-called discretization error is introduced. A proper discretization tries to keep the discretization error at a tolerable size while controlling the overall computational effort.

The discretization process is usually split in two stages, the *spatial* and *temporal* discretization. The spatial discretization transforms the PDEs into ordinary differential equations which are then discretized in time and solved by a numerical integration scheme.

2.4.1 Spatial Discretization

The spatial discretization refers to the transformation process of the continuous representation of space to a discrete representation with a finite number of degrees of freedom (DoFs). Although the discretization process is crucial for a computer to find a solution, the discrete solution differs from the ideal (often unknown) continuous solution. In order to limit that error a suitable discretization has to be chosen.
There are a lot of available discretizations that differ in the overall computational effort required and the resulting accuracy of the solution. In the succeeding, we state the most popular discretization techniques, though suppress a detailed explanation or concrete discretization in terms of formulas.

The most popular methods are the *finite element method* (FEM), the *finite volume method* (FVM) and the *finite differences method* (FDM). While FEM and FVM are realized through a

partitioning of space in cells, the finite differences method uses a point-wise representation. There are a lot of detailed elaborations about spatial discretization of PDEs as for example [54] that deals with the finite difference method, [3] for the finite element method and [42] for the finite volume method. An easy to understand discretization and computation of the Navier-Stokes equations using the FDM is given in [32].

For some of the methods stated above, there exist mesh-free approaches like for example the *particle finite-element method* (PFEM) [46] or the *partition of unity method* [52]. In this work we focus on mesh-based methods which are especially well understood an developed.

A mesh consists of cells, faces and nodes. One distinguishes between cartesian meshes, structured meshes or unstructured meshes, having regular or arbitrary cells. The unknowns from the partial differential equations are associated to cells faces or nodes. Typically the distribution of unknowns to mesh elements is driven by stability considerations. Figrues 2.1 (a)-(b) show a few conceivable distributions of unknowns on mesh elements.

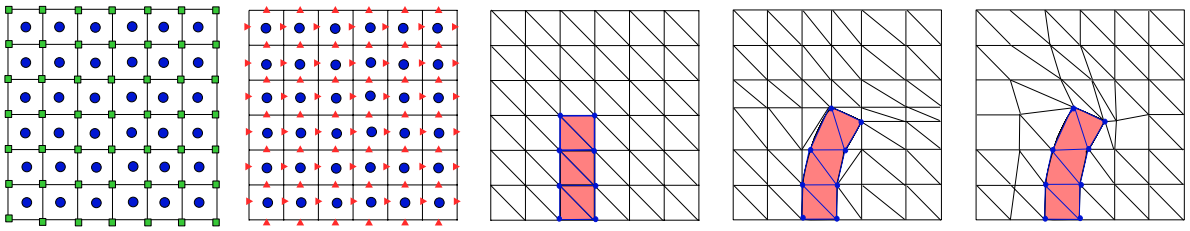


Figure 2.1: Distribution of unknowns on mesh elements and an example of mesh motion and remeshing. (a) Either the blue (cell based) or the green (node based) distribution of unknowns on cell elements or a combination of both is conceivable. (b) Different unknowns are distributed on different mesh elements due to stability considerations. For example for the Navier-Stokes equations this fully staggered approach is conceivable, where the pressure is located at the cell center and x- and y-velocity on the indicated mesh faces (triangles). A similar picture is found in [31].
(c)-(e) show an example of dynamic mesh behaviour. (c) Initial configuration, black denotes fluid mesh, blue solid mesh. The red domain is a solid beam in cross flow. (b) Example of remeshing, where new nodes, cells and faces are generated. (e) Example of mesh-motion. The nodes remain connected but undergo a transformation.

A very important point with regards to partitioned FSI coupling is the coupling of the spatial discretization. With separate solvers for the fluid and structural part, it is very likely to have different, non matching, spatial discretization or even mathematical models that are described in different points of view. In order to ensure the coupling conditions (2.13) and (2.15) a mapping of unknowns from probably different discretizations is needed. Furthermore, the generated mesh cannot be kept constant throughout the entire simulation, but has to be adapted according to the moving simulation domain, especially in case of fluid simulation. While for Lagrangian approaches, the mesh is deformed analogously to the simulation domain, the mesh-motion process is somewhat more involved for ALE approaches. Here, only the grid points at the boundary are deformed in conformity with the moving domain, while interior grid points are found by means of adequate interpolation and smoothing operators such as *radial-basis function interpolation*. In case of large displacements, mere mesh-motion may be not sufficient. In such cases, either renoding or remeshing techniques has to be applied. An example of both is given in Figure 2.1 (c)-(e). In contrast to remeshing, renoding only

changes the mesh cells and faces but leaves the mesh nodes untouched. Both techniques are computationally expensive and induce oscillations in the numerical solving process, as unknowns are modified, created or destroyed and should thus be avoided whenever possible.

2.4.2 Temporal Discretization

After discretizing the partial differential equations in space, the resulting ordinary differential equations with respect to time need to be discretized and solved. One also speaks of semi-discrete equations, that can be stated as follows

$$\frac{\partial \boldsymbol{x}}{\partial t} - \boldsymbol{\mathcal{D}}_x(\boldsymbol{x}, t) = 0, \tag{2.16}$$

where $\boldsymbol{\mathcal{D}}_x$ indicates the spatial discretized terms and $\boldsymbol{x}$ is the vector of unknowns in space. As for the spatial part, the continuous time is split into a finite number of discrete time steps Δt, where a solution for (2.16) is sought for each of the discrete points in time. As opposed to the solving step of the spatial part, where the solution to all the unknowns is computed at once, here the solution is obtained in a sequential manner using a time integration scheme.

A time integration scheme I computes subsequent solutions x^n at discrete time instances t^n with $n \in \mathbb{N}$. Depending on the input data, the integration schemes are divided into two classes, the explicit time integration schemes and the implicit time integration schemes. While the explicit schemes (2.17) solely use available information from the past time steps, the latter (2.18) additionally use the future (unknown) solution x^{n+1} implicitly.

$$\boldsymbol{x}^{n+1} = \boldsymbol{I}(\Delta t^n, \boldsymbol{x}^n, \cdots, \boldsymbol{x}^{n-m}, \boldsymbol{\mathcal{D}}_x^n, \cdots, \boldsymbol{\mathcal{D}}_x^{n-m}) \tag{2.17}$$

$$\boldsymbol{x}^{n+1} = \boldsymbol{I}(\Delta t^n, \boldsymbol{x}^{n+1}, \boldsymbol{x}^n \cdots, \boldsymbol{x}^{n-m}, \boldsymbol{\mathcal{D}}_x^{n+1}, \boldsymbol{\mathcal{D}}_x^n, \cdots, \boldsymbol{\mathcal{D}}_x^{n-m}) \tag{2.18}$$

Here, m denotes the number of past time steps that are used for integration. As for the implicit integration scheme, the future solution $\boldsymbol{x}^{n+1}$ occurs on the right hand side. A system of equations has to be solved in each time step. For fluid-structure interaction applications, this system of equation is non-linear due to the governing non-linear equations that describe the physics and can be solved using a Newton solver.
Although, the implicit scheme is computationally more expensive than the explicit scheme, it provides better stability and often outweighs this drawback by being capable of choosing larger time step sizes. Some of the more sophisticated integration schemes even choose the time step size adaptively, for example by means of an error estimator. Explicit schemes for their part, often raise severe instability problems and have to be limited to very small time step sizes.
Since both, the spatial and the temporal discretization error determine the overall error of the approximation to the continuous solution, the accuracy for spatial and temporal methods should be chosen similarly.

For a detailed elaboration of solving ordinary differential equations, the reader is kindly referred to the standard text books [34, 35].

Chapter 3

Fixed-Point Equations for Partitioned Fluid-Structure Coupling

Fixed-Point Equations for Partitioned Fluid-Structure Coupling is a large class of applications that involve several physical effects, described by different mathematical models so called multi-physics simulations. A brief summary of the physical modelling for fluid dynamics and structural mechanics as well as the needed discretization methods has been given in Chapter 2. Here, we focus on the coupling aspect between the different physics, i. e., the installation and abidance of discretized coupling conditions at the interface as well as the choice of suitable iterative solvers for the overall system of equations. In the following, a coupling of fluid and structure is assumed but most of the concepts can be translated to the coupling of arbitrary multi-physics simulations.

The coupling occurs at the fluid-structure interface and can be embedded within one global system of equations, including the discrete flow variables $\boldsymbol{x}_F$, the discrete structure variables $\boldsymbol{x}_S$ and their coupling, reading

$$\boldsymbol{A}(\boldsymbol{x}_F, \boldsymbol{x}_S) = 0, \tag{3.1}$$

which refers to as the monolithic approach.

Although there are plenty of variations, we assume that one discretization method is used for both, the fluid and structural field. The above system is solved simultaneously for all unknowns. If $\boldsymbol{A}$ is linear, the structure of the matrix is given as

$$\begin{pmatrix} \boldsymbol{A}_F & \boldsymbol{A}_{FS} \\ \boldsymbol{A}_{SF} & \boldsymbol{A}_S \end{pmatrix} = \begin{pmatrix} \boldsymbol{x}_F \\ \boldsymbol{x}_S \end{pmatrix} = \begin{pmatrix} \boldsymbol{b}_F \\ \boldsymbol{b}_S \end{pmatrix},$$

for an appropriate reordering of the unknowns. The block matrices A_F and A_S on the main diagonal comprise the discretized governing equations for the fluid and structural internal variables while the off-diagonal blocks A_{FS} and A_{SF} ensure the coupling of fluid and structure at the interface. Usually the system (3.1) is non-linear and it is solved by means of Newton iterations [56].

Monolithic solvers for FSI coupling are often dedicated to a certain FSI application or a class of FSI problems, due to their design choices. Its range of applications is limited to a subset that can be described and treated properly by the internal physical models and the utilized discretization method. Hence, a new solver has to be developed in case of a substantially different FSI problem. However, as the coupling conditions are directly implemented and embedded on a very low level, they can be ensured during the solution step of fluid and structural unknowns and thus the monolithic approach is very robust and provides pretty accurate solutions for difficult problems.

A second possibility is to separate the solution of the fluid and structural equations into different single physics solvers and enforce the coupling externally in between solver calls. This is called the partitioned approach and is of major interest in this thesis and introduced in detail in the following sections. Apart from that, there are a lot of hybrid forms of the above extremes [4] that are not considered here, though.

In comparison to previous coupling methods, a newly and sophisticated coupling method has been introduced and implemented in this thesis, that provides the combination of various coupling systems for the partitioned approach with a multi-vector quasi-Newton method that accelerates and stabilizes the coupling. This method appears to be more robust than the existing methods and renders the tuning of several problem dependent parameters unnecessary. Section 3.1 presents some fundamentals of the partitioned coupling approach. Different coupling systems and some basic iterative solvers are discussed in Section 3.2. A much deeper elaboration of advanced solvers is given in Chapter 4, including the newly introduced concepts.

3.1 Ingredients of Partitioned Coupling

The partitioned coupling approach is driven by the idea of high flexibility and modularity and is based on the ideas of domain decomposition methods with non-overlapping subdomains. Here, the coupling of physics is done on the highest level possible. When applying the concepts from domain decomposition to FSI-coupling, the simulation domain is subdivided into a disjoint partition of fluid domain and structure domain (also called component or field). For a mathematical description, we introduce the interface-operators $\boldsymbol{F}$ and $\boldsymbol{S}$ that represent the fluid and structural field solver, respectively.

Fluid-structure-interaction is a surface-coupled problem, which means that the respective field solvers are coupled by their kinematic and dynamic boundary values at the fluid-structure interface, in such way that the coupling conditions (2.13) and (2.15) are fulfilled. Following the *Dirichlet-Neumann method* for the iterative solution of a decomposed problem with non-overlapping subdomains, this leads to the succeeding definition for the interface-operators

$$\begin{aligned} \boldsymbol{F}&: \boldsymbol{x}_d \mapsto \boldsymbol{x_f} \\ \boldsymbol{S}&: \boldsymbol{x}_f \mapsto \boldsymbol{x_d}. \end{aligned} \tag{3.2}$$

Here, $\boldsymbol{x}_d$ are the kinematic boundary values at the fluid-structure interface, i. e., displacements or velocities and $\boldsymbol{x}_f$ denotes the dynamic boundary values in terms of stresses or forces that

model the impact of the fluid on the structure at the interface. The coupling is based on the exchange of boundary values, i. e., both field solvers use the proceeding output values from the complementary component as their input values and *assume them to be fixed during their solution*. The boundary values are either instantly passed on to the complementary component or advanced by some predictor rule as for example time extrapolation.

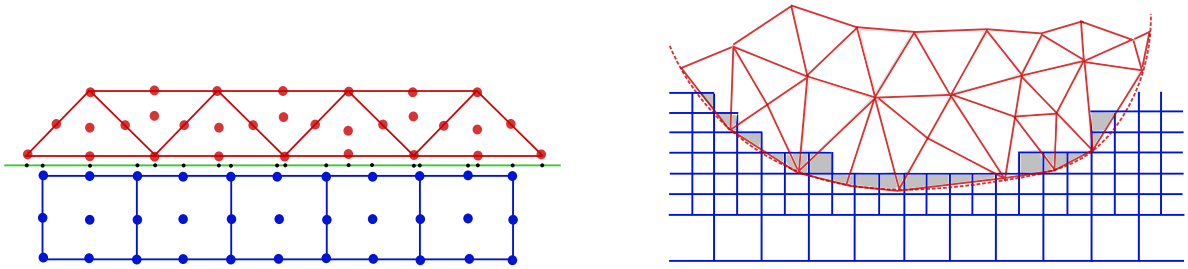
Flexibility and Modularity. The partitioned coupling approach requires minimal information for the solver interface and is highly independent of the field solver internals, i. e., applied physical models and utilized numerical schemes. The field solvers can therefore be seen as black-box solvers that do solely communicate input and output values at the common fluid-structure interface. This offers a great amount of flexibility and modularity for the partitioned coupling approach and allows to re-use existing solver software. The latter is eminently beneficial when highly sophisticated commercial or closed software solvers are used, where nearly no information is available concerning solver internals. As a consequence, one can benefit from experiences that have been made in every field and the scalability of the respective field solvers on massive-parallel machines is well known. Furthermore, the coupling process can be implemented in a separated coupling tool for multi-physics simulations and fluid or structural component solvers can be substituted in a plug and play manner. This allows for a wide range of problems to be solved as the solvers can be tuned in a better way to particular characteristics of each physical system.

3.1.1 Data Mapping

As described above, one of the most important use cases for partitioned FSI coupling is when existing highly sophisticated solvers are used as black box solvers. In this case, it is very unlikely that a matching surface discretization is given at the interface. The meshes can be simply non-conforming, which means that the locations of nodes or unknowns do not match at the interface, cf. Figure 3.1 (a). It might even be that overlapping or gaps between the meshes occur at the interface, cf. Figure 3.1 (b). This gives raise to the crucial step of *data mapping* between the surface representations of the fluid and structural components. Here, it is of great importance to avoid introducing significant errors into the solution so that conservation laws for mass, momentum and energy as well as preservation of rigid body motion are not violated. Not least, the partitioned design choice together with the data mapping module bear the advantage that for each field solver the optimal mesh can be used for discretization. Although this is a very important part of partitioned FSI coupling, it is of no consequence for the investigations done in this work and is therefore not discussed any further. A detailed elaboration can be found in [31].

3.1.2 Partitioned Equation Coupling

The overall goal of partitioned coupling is to re-establish the solution of the monolithic approach. To that end, there are plenty of different equation coupling schemes, that are divided into explicit and implicit coupling schemes. While explicit coupling schemes are suitable for weakly



(a) (b)

Figure 3.1: The fluid-structure interface is discretized by two partitioned, independent solvers. A non-conforming mesh appears at the interface due to different discretization and individual meshing. (a) Both meshes share the same interface, however, the unknowns are not aligned, i. e., the combination is non-conforming (dots on green line) and raise the need of data mapping. (b) A more complicated fluid-structure interface is approximated by the field solvers using different mesh types, i. e., adaptive cartesian (blue mesh) and triangular (red mesh). Here, the meshes are not simply non-conforming, but additionally overlapping occurs. A similar picture can be found in [31].

coupled FSI problems, where the interaction between fluid and structure is small, strong interactions or stability issues raise the need of implicit coupling schemes.
In this work, we focus on implicit coupling schemes, especially the more sophisticated ones that are advanced with a quasi-Newton post-processing scheme, presented in Chapter 4.

Explicit Coupling Schemes

Explicit coupling schemes approximate the solution of the monolithic system (3.1) by executing the fluid and structural field solvers along with the exchange of boundary values for a *fixed number of times* per time step. Hence, the equilibrium of displacements and stresses is only enforced up to a certain error order which depends on the discrete time step length. This introduces an error in every time step and is only applicable for weakly coupled FSI problems. Different explicit coupling schemes are explained in [31], i. e., the *conventional serial staggered* (CSS) and the *conventional parallel staggered* (CPS) together with improved variants and generalizations. Some of the coupling schemes have been tested for compressible flow FSI applications in the field of aerodynamical simulation in [48, 41, 47, 28].

Implicit Coupling Schemes

Implicit coupling schemes approximate the solution of the monolithic system (3.1) in an iterative process, which means that they repeatedly solve for a solution of the structure and fluid solvers involving the communication of boundary values, until convergence is achieved within one time step. Convergence means, that the equality of displacements and stresses is guaranteed up to a certain (arbitrary) precision. The use of implicit coupling schemes is motivated by the fact that they really solve for an approximation of the monolithic solution and are more robust in case of instabilities due to the added-mass effect (see next paragraph).

Additionally, implicit schemes can be enhanced by a further post-processing scheme that improves the convergence behavior (see remainder of this chapter).

In the following, we denote the fluid and structural interface variables of time step $n+1$ as $\boldsymbol{x}_{f_{n+1}}$ and $\boldsymbol{x}_{dn+1}$, respectively. In order to indicate the k-th iterate of the implicit coupling scheme at the current time step we use ${\boldsymbol{x}_d}^k$ and ${\boldsymbol{x}_f}^k$. If not stated explicitly, the current time step is assumed to be $n+1$ and the index is omitted, i. e., ${\boldsymbol{x}_d}^k := \boldsymbol{x}_{dn+1}^k$.

Instabilities and the Added-Mass Effect

Although explicit schemes work well for a lot of FSI applications they fail to converge in cases when instabilities are present. Such instabilities are created by the partitioning of equations and are therefore inherent to the partitioned coupling approach. The instabilities occur due to the staggered solution of fluid and structure component, where each solver has to use boundary values from the other that are assumed to be *constant* for the entire time step.
For compressible flow, this effect can be circumscribed by an upper limit to the time step size, while for FSI simulations with incompressible flow this raises the need of more advanced coupling schemes, i. e., implicit coupling schemes with an additional post-processing method (cf. Section 3.2 and Chapter 4).

Theoretical investigations of the so-called added-mass effect in terms of stability analysis has been made for simplified and idealized FSI examples [12, 18, 30]. The results that have been learned can be summarized as follows (see also [31, p. 68]). The added-mass instabilities increase when

- the density ratio of fluid and structure is close to one
- the structural stiffness decreases or the fluid viscosity increases
- the time step size decreases for implicit coupling schemes
- the geometric aspect ratio of the fluid-structure interface increases

Subcycling. Another useful feature of partitioned FSI coupling is that of *subcycling*. This means that a single time step can be subdivided into m smaller parts which allows for a decoupling of the time scales used for the different components. Usually, the fluid field solver requires smaller time steps than the structure solver. No data is communicated between the solvers when subcycling is deployed but input data from the other component can be extrapolated over time.

3.2 Fixed-Point Formulations for the Partitioned Problem

The term coupling scheme is used to indicate the combination of a specific coupling system with an appropriate fixed-point equation solver or post-processing method, that solves the fixed-point equation induced by the coupling system. In this work, we focus on three implicit coupling systems, presented in Section 3.2.1, namely a serial or staggered implicit coupling system a vectorial or parallel implicit coupling system and a block-iterative serial coupling system. Each of them offers some advantages and drawbacks which will be discussed later on.

The canonical implicit coupling is represented by the serial or staggered system which is quite efficient due to its rather low computational effort. However, it offers some drawbacks concerning parallelization. Here, the parallel implicit system is superior while having comparable convergence results. The block-iterative serial coupling system is slightly more robust with respect to instabilities and allows, due to overlapping post-processing, for marginal better parallel efficiency than the pure serial system. Still, its overall performance on massively parallel machines is however insufficient.

The induced fixed-point equations need to be solved using an appropriate method. Such a method naturally consists of a base frame fixed-point iteration, enhanced by a so called post-processing method that further improves the convergence of the coupling. The quality of a partitioned coupling algorithm can be measured by the extend to which it fulfils the requirements to be both, efficient and robust for a wide range of FSI coupling scenarios. In the context of black-box partitioning, one assumes that the largest cost with respect to the coupling iterations in a given time step is compounded by the computational effort of the black-box field solvers. For an efficient coupling it is therefore important to minimize the number of solver calls, in other words the required numbers of coupling iterations. Furthermore, it is desirable for a coupling scheme to be robust, that means that it shows comparable performance, for a wide range of scenarios without the need for tuning a set of problem dependent parameters beforehand. The coupling methods considered in this thesis, especially those with a quasi-Newton post-processing, will be judged by the extend they meet the above requirements (see Chapter 6 and 7).

In Sect. 3.3 we give a brief overview of the plain fixed-point iteration which is basically used as base frame iteration for an additional post-processing as it provides poor or no converge for severe instabilities.

In the succeeding, we present supporting post-processing methods that improve convergence of the coupling, beginning with the rather simple constant and adaptive Aitken's underrelaxation, but focus on the more sophisticated quasi-Newton methods in Chapter 4. The latter are of great importance for FSI scenarios that entail stability issues due to the added mass effect. Furthermore, only quasi-Newton methods – that also showed the best results for staggered schemes [17, 31, 7] – accelerate the parallel implicit coupling scheme enough to make it competitive to the serial implicit coupling scheme, compare [57].

3.2.1 Coupling Variants and Fixed Point Equations

In this section, three different implicit coupling systems and the respective fixed-point equations are presented, namely the serial-implicit coupling system, the vectorial- or parallel-implicit coupling system and the serial-implicit block coupling system.

(A) Serial Implicit Coupling System (S-System)

Using the serial implicit coupling system, the field solvers are executed in a staggered way. This is the standard approach to couple a black box fluid solver with a black box solid solver in a partitioned setting [19, 26, 7, 28]. The flow solver computes the current time step, resulting in stresses or forces exerted by the fluid on the structure. The structural solver takes this forces as input and computes the resulting displacements of the structure.
If the result of such a coupled iteration is the same as the input displacement, up to a certain precision, the iteration has reached a fixed-point and the coupling is stable. The displacements at the fluid-structure interface fulfill the sequential fixed-point equation

$$\boldsymbol{x}_d \overset{!}{=} \boldsymbol{S} \circ \boldsymbol{F}(\boldsymbol{x}_d). \tag{3.3}$$

Note, that multiplication of $\boldsymbol{F}$ from the left ensures equality of forces as well.

However, the inherent serial calling order of the field solvers leads to performance issues for massively parallel simulations due to bad load balancing. A better adjusted, parallel coupling scheme is introduced in the following and thus a few words concerning efficient parallelization in FSI simulations are added in the following.

Efficient Parallelization. Parallel efficiency is of great importance for highly accurate simulations. In order to be beneficial compared to single physics simulations, multi-physics simulations do have to provide a very high accuracy. Otherwise, the additional information obtained through multi-physics simulation would be of no consequence, compared to the overall error due to low accuracy.
On the other hand, solving for a high accuracy solution entails very high computational costs and in most practical cases a massive parallel computation is indispensable. In the field of partitioned FSI coupling, massive parallel computations often raise problems and are rather inefficient due to poor load balancing.

Due to different complexity, the fluid solver comprises the vast majority of compute resources and/or computational time. There are very efficient parallel explicit coupling schemes [26], that need exactly one call to the fluid and solid solver per time step. However, for most applications that include incompressible flow, implicit schemes are required to achieve a stable coupling. The parallel efficient approaches cannot be generalized to implicit schemes, as they usually call for the field solvers in a serial way, see (3.3), which leads to a mismatch of work balance and in the worst case results in a parallel efficiency of 50 %.

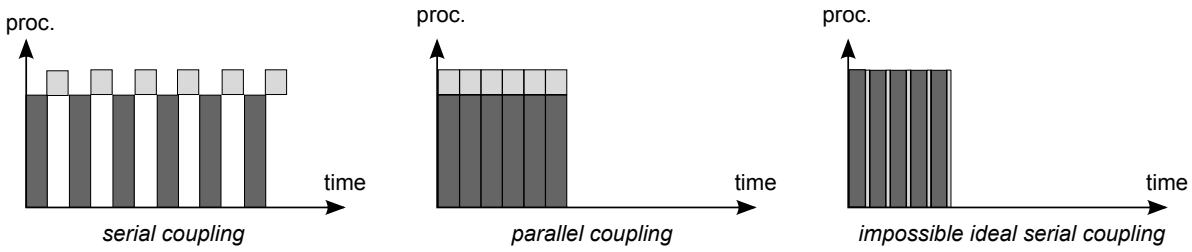


Figure 3.2: The number of busy processors is plotted over time. There is a substantial mismatch of work load for the serial coupling scheme. While the fluid solver (dark grey) entails a good scalability and keeps a lo of processors busy, the solid solver (light grey) usually has low computational cost and cannot keep as many processors busy. Hence, the computational time for both solvers is similar which results in a parallel efficiency of 50 %. **Left:** Inefficient parallelization for a serial execution order of fluid and solid solvers. **Middle:** Efficient use of compute resources using a parallel coupling scheme. This, however, requires a rethinking of the coupling numerics. **Right:** Ideal work load distribution which is however not possible as the structure solver does not scale with as high number of processors than the fluid solver.

Figure 3.2 shows scalability sketches for difference coupling approaches. While the fluid solver can be scaled on a very large number of processors P, the fluid solver runs on a small number of processors p, having the same runtime T_r. As can be seen in the left figure, P processors cannot be kept busy during the solution step of the solid solver. Thus, the ideal load balancing, depicted in the right figure, is not possible and the serial coupling system has poor performance on massively parallel machines.

A newly parallel coupling system, see (3.4), that allows for simultaneous computation of fluid and solid solver is presented in the following and provides the possibility to a perfect load balancing, as can be seen in the left diagram of Fig. 3.2. However, applying this system, the coupling numerics change and need to be reconsidered.

(B) Parallel Implicit Coupling System (V-System)

As we have seen, the above serial implicit coupling system offers some drawbacks with regard to efficient parallelization due to a substantial mismatch of work load between the structure and the fluid field solver. The only way to overcome this limitations is to execute the fluid and structure solver in parallel. To this end, the V-system uses the original input/output relation for both solvers but the boundary values are exchanged after each iteration. This leads to the vectorial fixed-point equation

$$\begin{pmatrix} \boldsymbol{x}_f \\ \boldsymbol{x}_d \end{pmatrix} \stackrel{!}{=} \begin{pmatrix} \boldsymbol{F}(\boldsymbol{x}_d) \\ \boldsymbol{S}(\boldsymbol{x}_f) \end{pmatrix} = \begin{pmatrix} 0 & \boldsymbol{F} \\ \boldsymbol{S} & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{x}_f \\ \boldsymbol{x}_d \end{pmatrix}. \tag{3.4}$$

However, the above system results in two independent instances of the S-system that need to be coupled by a quasi-Newton step after each iteration. This coupling turns out to be strong enough so that one iteration of the V-system is comparable to one iteration of the S-system (cf. [11, 57]). Hence, the V-system is quite competitive, as the benefit from the good parallel scalability clearly outweighes fractional losses in convergence speed.

(C) Serial Implicit Block System (B-System)

This coupling system was developed as an advancement of the serial-implicit coupling system (A). It is supposed to be more stable and robust as it provides intermediate results that are passed along within a block-iterative scheme. Furthermore, it offers improved modularity as one directly approximates for the Jacobians of the single-physics solvers. Thus, additional information can be utilized if available. Although it looks quite similar to the vectorial coupling system[1] (B) the calling order of the fluid and solid field solvers remains serial, i. e., a parallel execution of the black-box solvers is not possible. Using this coupling system, the fixed-point equation

$$\begin{pmatrix} \boldsymbol{x}_f^F \\ \boldsymbol{x}_d^S \end{pmatrix} \overset{!}{=} \begin{pmatrix} \boldsymbol{F} \circ \boldsymbol{S}(\boldsymbol{x}_f^F) \\ \boldsymbol{S} \circ \boldsymbol{F}(\boldsymbol{x}_d^S) \end{pmatrix} = \begin{pmatrix} \boldsymbol{F} \circ \boldsymbol{S} & 0 \\ 0 & \boldsymbol{S} \circ \boldsymbol{F} \end{pmatrix} \begin{pmatrix} \boldsymbol{x}_f^F \\ \boldsymbol{x}_d^S \end{pmatrix}. \tag{3.5}$$

is solved in a block-iterative fashion which means that the equations are solved successively and the output from the first equation is used as input for the second. The first equation can be seen as the serial fixed-point equation for the fluid solver (superscript F) while the second states the equality of displacements for the solid solver (superscript S). Obviously, if one of the fixed-point equations is fulfilled the other holds true as well.

The application of the block-iterative coupling system is somewhat more involved and differs from the application of the two coupling systems (3.3) and (3.4) explained earlier. A generic combination of this coupling system with a quasi-Newton method is depicted later on in Section 4.4.

[1] In fact, we obtain the fixed-point equation (3.5) if we square the parallel-implicit coupling system from (3.4)

$$\begin{pmatrix} 0 & \boldsymbol{F} \\ \boldsymbol{S} & 0 \end{pmatrix}^2 \begin{pmatrix} \boldsymbol{x}_f \\ \boldsymbol{x}_d \end{pmatrix} = \begin{pmatrix} \boldsymbol{F} \circ \boldsymbol{S} & 0 \\ 0 & \boldsymbol{S} \circ \boldsymbol{F} \end{pmatrix} \begin{pmatrix} \boldsymbol{x}_f \\ \boldsymbol{x}_d \end{pmatrix}.$$

However, as we want to solve this fixed-point equation in a block-iterative way the coupling numerics appear to be quite different than for parallel-implicit coupling system, compare Section 4.4.

3.3 Basic Iterative Solvers for the Fixed-Point Equations

In the previous section we presented three fixed-point equations that differ in the execution order of the fluid and structure solver but also provide different coupling numerics. These fixed-point equations need to be solved in each FSI coupling iteration. In this section we present a couple of fixed-point equation solvers. For the sake of completeness we firstly summarize the rather simple and well known iterative solvers such as ordinary Schwarz iteration and constant and dynamic Aitken's underrelaxation. A more detailed elaboration of different quasi-Newton variants is given in Chapter 4. This is a class of root finding algorithms based on Newton's method that solve for a set of non-linear equations, but rather than computing the exact Jacobian they try to find a suitable approximation.

For a unified description of these solvers, we introduce the notation

$$\boldsymbol{x} \in \left\{ \begin{array}{cc} \text{S-system} & \text{V/B-system} \\ \boldsymbol{x}_d, & \begin{pmatrix} \boldsymbol{x}_f \\ \boldsymbol{x}_d \end{pmatrix} \end{array} \right\} \quad \text{and}$$

$$\boldsymbol{H} \in \left\{ \begin{array}{ccc} \text{S-system} & \text{B-system} & \text{V-system} \\ \boldsymbol{S} \circ \boldsymbol{F}, & \begin{pmatrix} \boldsymbol{F} \circ \boldsymbol{S} & 0 \\ 0 & \boldsymbol{S} \circ \boldsymbol{F} \end{pmatrix}, & \begin{pmatrix} 0 & \boldsymbol{F} \\ \boldsymbol{S} & 0 \end{pmatrix} \end{array} \right\}$$

For the S-system (3.3), the V-system (3.4) and the B-system (3.5), respectively.
Thus, we have to solve a fixed point equation

$$\boldsymbol{H}(\boldsymbol{x}) = \boldsymbol{x} \;\Leftrightarrow\; \boldsymbol{R}(\boldsymbol{x}) = \boldsymbol{H}(\boldsymbol{x}) - \boldsymbol{x} = 0. \tag{3.6}$$

Note that the unified notation for the B-system is mentioned merely for demonstrative purposes. The hereinafter presented fixed-point equation solvers, especially the quasi-Newton systems, can be clearly combined with the B-system however in a slightly different way. The following elaboration is only applicable to the S-system and the V-system. As opposed to the latter two, the B-system is solved in a block-iterative fashion and hence needs approximations for the Jacobians of the fluid and solid interface operators $\boldsymbol{F}$ and $\boldsymbol{S}$, rather than for the fixed-point operator $\boldsymbol{H}$ or the residual operator $\boldsymbol{R}$, or their inverse. The ideas and concepts can however be transferred to the approximation of the fluid and solid solver Jacobians.

For convenience, we add a brief summary of further notation that is frequently used hereinafter.

- $\tilde{\boldsymbol{x}}^k$ .. result of one fixed-point iteration, i. e., $\tilde{\boldsymbol{x}}^k = \boldsymbol{H}(\boldsymbol{x}^k)$, intermediate result
- $\boldsymbol{x}^{k+1}$ .. next iterate of coupling scheme, usually composed of intermediate result $\tilde{\boldsymbol{x}}^k$ and a correction from a post-processing method, i. e., $\tilde{\boldsymbol{x}}^k \overset{(PP)}{\rightsquigarrow} \boldsymbol{x}^{k+1}$
- $\boldsymbol{r}^k$ .. residual of the fixed-point equation, i. e., $\boldsymbol{R}(\boldsymbol{x}^k) = \boldsymbol{r}^k = \boldsymbol{H}(\boldsymbol{x}^k) - \boldsymbol{x}^k = \tilde{\boldsymbol{x}}^k - \boldsymbol{x}^k$. Usually the residual operator is denoted by $\boldsymbol{R}(\boldsymbol{x}^k)$ while the residual vector is written as $\boldsymbol{r}^k$.
- $\boldsymbol{J}_\square$ .. Jacobian of the operator $\square \in \{\boldsymbol{R}, \boldsymbol{H}, \boldsymbol{F}, \boldsymbol{S}\}$
- $\widehat{\boldsymbol{J}_\square}$ .. approximation of the Jacobian of the respective operator.

3.3.1 Pure Fixed-Point Iterations

Given the abstract fixed point equation (3.6), the canonical fixed-point iteration

$$\tilde{\boldsymbol{x}}^k = \boldsymbol{H}(\boldsymbol{x}^k), \qquad \boldsymbol{x}^{k+1} = \tilde{\boldsymbol{x}}^k \tag{3.7}$$

can be seen as the plainest fixed point solver for implicit coupling schemes. Here, we refer $\tilde{\boldsymbol{x}}^k = \boldsymbol{H}(\boldsymbol{x}^k)$ to be the result of one fixed-point iteration to differentiate is from the later introduced result of the post-processing, which is denoted by $\boldsymbol{x}^{k+1}$. Depending on whether the serial (3.3) or parallel fixed-point equation (3.4) is solved, one speaks of the *multiplicative* or *additive Schwarz procedure* [51, 53], also known as block *Gauß-Seidel method* and *block Jacobi method*.

Convergence Criteria. We want convergence towards the monolithic solution in every time step. Since the latter is not available we have to use a suitable convergence criterion to decide whether we want to proceed with the iteration or terminate. In general, different convergence criteria will lead to different iteration numbers. Here we define that the iteration continues until the l_2-norm of the residual $\boldsymbol{r}^k := \boldsymbol{R}(\boldsymbol{x}^k) = \boldsymbol{H}(\boldsymbol{x}^k) - \boldsymbol{x}^k$ falls below a certain absolute or relative convergence criterion, i. e.,

$$\|\boldsymbol{r}^k\|_{L^2} < \epsilon_{abs}, \qquad \frac{\|\boldsymbol{r}^k\|_{L^2}}{\|\tilde{\boldsymbol{x}}^k\|_{L^2}} < \epsilon_{rel} \tag{3.8}$$

Since the norm of the residual can change by orders of magnitude throughout one simulation an absolute convergence criterion is not feasible for all situations. Setting the residual in relation to the current iterate solves this problem in terms of a relative convergence measure. However, a relative convergence measure involves instabilities and fails to work properly if the current iterates are close to zero. Therefore it is advisable to use a combination of both, relative and absolute convergence measure. Not least, the number of coupling iterations depends on the internal convergence measures of the black box field solvers that should demand for a higher accuracy than the coupling criterion above.

The plain fixed-point iteration does only converge if the fixed-point operator $\boldsymbol{H}$ is a contraction mapping. If instabilities are present, in general, this is not the case and the Schwarz procedures fail to converge and are therefore not suitable to stabilize the coupling (added mass effect). In order to guarantee for the convergence of the fixed-point iteration, additional efforts need to be made. The multiplicative or additive Schwarz iteration can be seen as a base frame or template for all following fixed-point equation solvers which is enhanced by a *post-processing method* to improve the convergence of the coupling scheme.

A rough pseudo code representation of the fixed-point iteration as base frame iteration for all coupling schemes, is given in Algorithm 1. To gain a clearer understanding of the internals, especially the temporal order in which the black box solvers are called, the different coupling schemes are depicted in terms of the fluid and structural interface operators $\boldsymbol{F}$ and $\boldsymbol{S}$. The result of the application of the fixed point operator is seen as a intermediate result $\tilde{\boldsymbol{x}}^k$ that is further modified by a post-processing method. In the following, but especially in

Input: initial value $\boldsymbol{x}^0 = (\boldsymbol{x}_f^0, \boldsymbol{x}_d^0)^T = \text{extr}(\boldsymbol{x}_n^*, \boldsymbol{x}_{n-1}^*, \boldsymbol{x}_{n-2}^*)$ time extrapolation from previous time steps

Output: $\boldsymbol{x}_{n+1}^*$, converged configuration of current time step.

S-system($\boldsymbol{x}^0$)

for $k = 0 \ldots$ **do**
- $\tilde{\boldsymbol{x}}_f^k = \boldsymbol{F}(\boldsymbol{x}_d^k)$
- $\tilde{\boldsymbol{x}}_d^k = \boldsymbol{S}(\tilde{\boldsymbol{x}}_f^k)$
- $\tilde{\boldsymbol{x}}_d^k \overset{(PP)}{\rightsquigarrow} \boldsymbol{x}_d^{k+1}$ PP

end

V-system($\boldsymbol{x}^0$)

for $k = 0 \ldots$ **do**
- $\tilde{\boldsymbol{x}}_f^k = \boldsymbol{F}(\boldsymbol{x}_d^k)$
- $\tilde{\boldsymbol{x}}_d^k = \boldsymbol{S}(\boldsymbol{x}_f^k)$
- $\tilde{\boldsymbol{x}}_d^k \overset{(PP)}{\rightsquigarrow} \boldsymbol{x}_d^{k+1}$ PP
- $\tilde{\boldsymbol{x}}_f^k \overset{(PP)}{\rightsquigarrow} \boldsymbol{x}_f^{k+1}$ PP

end

B-system($\boldsymbol{x}^0$)

for $k = 0 \ldots$ **do**
- $\tilde{\boldsymbol{x}}_f^k = \boldsymbol{F}(\boldsymbol{x}_d^k)$
- $\tilde{\boldsymbol{x}}_f^k \overset{(PP)}{\rightsquigarrow} \boldsymbol{x}_f^{k+1}$ PP
- $\tilde{\boldsymbol{x}}_d^k = \boldsymbol{S}(\boldsymbol{x}_f^{k+1})$
- $\tilde{\boldsymbol{x}}_d^k \overset{(PP)}{\rightsquigarrow} \boldsymbol{x}_d^{k+1}$ PP

end

Algorithm 1: Representation of the fixed-point iteration as base frame or template iteration for all coupling schemes. The result of one fixed-point iteration is treated as an intermediate value $\tilde{\boldsymbol{x}} = \boldsymbol{H}(\boldsymbol{x})$, that undergoes further modification in the post-processing stage (PP). The post-processing can be either constant or dynamic underrelaxation or one of the quasi-Newton methods, considered here. The for-loop iterates until a certain convergence criterion is fulfilled. For the sake of clarity, the base frame fixed-point iterations are depicted in terms of the fluid and structure interface operators $\boldsymbol{F}$ ad $\boldsymbol{S}$. **Left:** Fixed-point base frame iteration for the serial-implicit fixed-point equation (S-system), also called additive Schwarz method. **Middle:** Fixed-point base frame iteration for the parallel-implicit fixed-point equation (V-system), also called multiplicative Schwarz method. **Right:** Fixed-point base frame iteration for the serial-implicit block-iterative fixed-point equation (B-system). Here, the post-processing is done block-wise in between the solver calls.

Chapter 4 multiple alternative fixed-point equation solvers are discussed, that can operate as post-processing methods. They differ in terms of computational effort and efficiency with respect to required coupling iterations, i. e., the rate of convergence.

3.3.2 Constant and Aitken's Underrelaxation

One of the most straightforward approaches to stabilize the plain fixed-point iteration is constant underrelaxation, i. e.,

$$\begin{aligned} \tilde{\boldsymbol{x}}^k &= \boldsymbol{H}(\boldsymbol{x}^k), \\ \boldsymbol{x}^{k+1} &= (1-\omega)\boldsymbol{x}^k + \omega\tilde{\boldsymbol{x}}^k = \boldsymbol{x}^k + \omega\boldsymbol{r}^k \end{aligned} \tag{3.9}$$

Here, the constant underrelaxation factor ω has to be chosen in the range of $0 < \omega < 1$ in order to stabilize the iteration. However, for a lot of scenarios ω has to be chosen close to zero in order to achieve convergence at all which leads to a unmanageable computational cost for severe instabilities.

The convergence of the coupling scheme can be accelerated and the above drawback can be limited by choosing the underrelaxation parameter ω dynamically in each iteration. This

method is called Aitken's dynamic relaxation. Here, the dynamic relaxation factor is modified at the beginning of each iteration, based on the information of two previous fixed-point iterates. According to the Secant method for root finding problems, the previous iterates are linearly extrapolated to find the position of zero residual $\boldsymbol{r}_k = \boldsymbol{R}(\boldsymbol{x}^k) = 0$. The relaxation factor can then be computed as

$$\omega_k = -\omega_{k-1} \frac{(\boldsymbol{r}_{k-1})^T (\boldsymbol{r}_k - \boldsymbol{r}_{k-1})}{\|\boldsymbol{r}_k - \boldsymbol{r}_{k-1}\|_2^2} \tag{3.10}$$

and the fixed-point iteration is enhanced by the underrelaxation step

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \omega_k (\boldsymbol{H}(\boldsymbol{x}^k) - \boldsymbol{x}^k), \tag{3.11}$$

This method is widely used in FSI coupling as it shows quite good convergence for a lot of applications while being very efficient with respect to computational cost and memory requirements. The performance and suitability of Aitken's relaxation for FSI coupling is investigated and compared to other methods in [57, 8].
Aitken's relaxation is briefly revisited in the next chapter about quasi-Newton solvers to illustrate that it is in fact the simplest form of a quasi-Newton method available.

Chapter 4

Quasi-Newton Post-Processing Methods

The fixed-point equation solvers presented in Chapter 3, especially the pure fixed-point iteration (3.7) and the constant underrelaxation (3.9) tend to be unstable and are in particular up to two times slower for the parallel fixed-point equation compared to the staggered one (compare [57]). Therefore, using an additional method that stabilizes and accelerates the coupling is often essential.

Solving any of the presented fixed-point equations originating from different coupling systems in the field of FSI coupling, we are given a problem with the following characteristics, compare [23]:

i) The dimensionality of the problem is potentially large (although its lower dimensional than overall simulation due to surface coupling)

ii) An analytic form of the derivative of the interface operators $\boldsymbol{F}$ and $\boldsymbol{S}$ is not available, neither for the fixed-point operator $\boldsymbol{H}$, nor for the residual operator $\boldsymbol{R}$

iii) The cost of evaluating $\boldsymbol{R}(\boldsymbol{x})$, resp. $\boldsymbol{H}(\boldsymbol{x})$, $\boldsymbol{F}(\boldsymbol{x})$ and $\boldsymbol{S}(\boldsymbol{x})$ is very high

iv) The problem is noisy, i. e., evaluated function values $\boldsymbol{R}(\boldsymbol{x})$ and $\boldsymbol{H}(\boldsymbol{x})$ usually contain errors (as field solvers solve up to a certain precision)

Several existing methods are not applicable due to the listed characteristics. For example, characteristic i) suggests to use matrix-free, limited-memory algorithms while ii) prevents us to use standard methods such as Newton iteration. Line-searching algorithms are not advisable because of characteristic iii) and all methods based on finite-differences, such as matrix-free Newton-Krylov solvers, are impractical due to characteristic iii) and iv).

Therefore, algorithms that rely on secant equations using input and output data to estimate an approximation of the system Jacobian have attracted a lot of interest in the past years. Important examples of so called *quasi-Newton methods* are Broyden's method [10, 44, 45] and its generalized versions [23, 22, 38], also known as Anderson acceleration for fixed-point iterations.

In the following, we consider different variants of quasi-Newton methods, beginning with an overview of existing methods in literature, including their classification and correlation to the considered methods in this thesis. In the course of the presentation of considered quasi-Newton methods, a newly post-processing method called IQN-IMVJ is introduced in combination with various coupling systems. It is expected to render an additional tuning of problem dependent parameters unnecessary. This costly pre-processing step in turn, is essential for all previously existing quasi-Newton methods.

For the sake of clarity, we start with the formulation of the Newton iteration, from which the quasi-Newton methods are motivated. For further stabilization, after computing one step of the fixed-point iteration,

$$\tilde{\boldsymbol{x}}^k = \boldsymbol{H}(\boldsymbol{x}^k),$$

we approximate a Newton iteration

$$\textbf{solve} \quad \left[I - \boldsymbol{J}_{H^{-1}}(\tilde{\boldsymbol{x}}^k)\right] \Delta\tilde{\boldsymbol{x}}^k = \boldsymbol{J}_{\tilde{R}}(\tilde{\boldsymbol{x}}^k)\Delta\tilde{\boldsymbol{x}}^k = -\tilde{\boldsymbol{R}}(\tilde{\boldsymbol{x}}^k), \tag{4.1}$$

$$\textbf{set} \quad \boldsymbol{x}^{k+1} = \tilde{\boldsymbol{x}}^k + \Delta\tilde{\boldsymbol{x}}^k \tag{4.2}$$

for the fixed-point equation (3.6) that is equivalent to the inverse form

$$\tilde{\boldsymbol{R}}(\tilde{\boldsymbol{x}}) = \tilde{\boldsymbol{x}} - \boldsymbol{H}^{-1}(\tilde{\boldsymbol{x}}) \stackrel{!}{=} 0 \tag{4.3}$$

as $\boldsymbol{H}(\boldsymbol{x}) - \boldsymbol{x} = \tilde{\boldsymbol{x}} - \boldsymbol{H}^{-1}(\tilde{\boldsymbol{x}})$ with $\tilde{\boldsymbol{x}} := \boldsymbol{H}(\boldsymbol{x})$.

As the exact Jacobian

$$\boldsymbol{J}_{\tilde{R}}(\boldsymbol{x}^k) := I - \boldsymbol{J}_{H^{-1}}(\tilde{\boldsymbol{x}}^k)$$

is not accessible for black-box solvers, we work with an approximation $\widehat{\boldsymbol{J}_{\tilde{R}}}(\tilde{\boldsymbol{x}}^k)$. Quasi-Newton methods approximate $\boldsymbol{J}_{\tilde{R}}(\boldsymbol{x}^k)$ or $\boldsymbol{J}_{\tilde{R}}^{-1}(\boldsymbol{x}^k)$ and obtain the current approximation by adding a low-rank matrix to a previous Jacobian approximation or initial Jacobian in each iteration. They use input and output values of the fixed point operator for the low-rank updates.

Broyden's first method [10] refers to a rank-one update that uses the information from one input-output pair. The resulting Jacobian approximation is the unique matrix that minimizes the change in the Frobenius norm and fulfills the secant equation. The Jacobian or its approximation are in general dense matrices. Although, the linear equation system in (4.1) usually is rather small compared to the overall degrees of freedom of the simulation, finding a solution for the dense system often entails a computationally bottleneck. In order to minimize the computational costs, it is of great benefit to directly approximate for the inverse of the Jacobian $\widehat{\boldsymbol{J}_{\tilde{R}}}^{-1}(\tilde{\boldsymbol{x}}^k)$. This approach is referred to as Broyden's second method [39] which received a lot of attention in the field of partitioned FSI coupling [17, 57, 7]. The resulting approximation of the inverse Jacobian is the unique matrix that minimizes the change in the Frobenius norm in terms of the inverse approximation and fulfills a modified secant equation.

Both variants can easily be enhanced to be rank-m updates of the estimated matrix by using m previous input-output relations to set up a system of secant equations. The minimization process now ranges over the last m iterates and minimizes the change with respect to a initial

guess. In the field of partitioned FSI coupling, a natural choice for m is the number of coupling iterations k that have already been performed within the current time step. In this vein, Broyden's second generalized rank-m update method corresponds to the IQN-IMVJ method, presented in Section 4.3, that minimizes the change in the Jacobian approximation within one time step. Thus, in every time step, the initial guess for the inverse Jacobian approximation is taken over from the previous time step. Likewise, the IQN-ILS method in Section 4.1 is a generalized rank-m update method for the inverse Jacobian, however, minimizing the difference of the current approximation to the zero-matrix as initial guess in every time step. For convenience, we assume that information form $k+1$ coupling iterations is readily available.

Vierendeels et al. [59] applied a block-iterative quasi-Newton least squares method using multiple input-output pairs to FSI coupling. This method was further developed by Degroote et al. [17] to the well known interface quasi-Newton least squares (IQN-ILS) method mentioned above (see Sect. 4.1), which approximates the inverse Jacobian of the residual operator by adding a rank-m update to a zero initial guess in every time step. There are several advancements of the IQN-ILS method such as [57] which improves the parallel efficiency of the method. As opposed to the latter, Haeltermann et al. [33] approximate for the Jacobians of the interface operators of the respective field solvers $\boldsymbol{F}$ and $\boldsymbol{S}$ and solve the resulting system, which refers to the concept of Broyden's first method. Here, this method is referred to as IQN-LSHJ, see Sect. 4.1.3. Bogaers et al. presented a method that implicitly re-uses previous information in the approximation process of the system Jacobian in combination with a block-iterative system. The rank-m IQN-IMVJ method in Section 4.3 is a variant of the latter.

Fang et al. [23] considered quasi-Newton methods as standalone solvers for non-linear equations that have the characteristics listed above. They present a class of generalized Broyden-like quasi-Newton methods that update a previous approximation of the inverse Jacobian. Depending on the form of the update they differentiate between Type I and Type II methods. While Type II methods update the approximation of the inverse Jacobian by minimizing the change in the approximation of the inverse Jacobian, Type I methods do so by minimizing the change in the approximation of the Jacobian and not its inverse. This results in a alternative update formula. In this notation, the generalized Broyden Type II method corresponds to the IQN-IMVJ method in Section 4.3. Walker et al. [60] used the same classification but fixed the previous approximation of the inverse Jacobian to zero which results in the IQN-ILS method for the Type II update and in the IQN-LSHJ method for the Type I update after applying the Woodbury formula. Furthermore it turns out that the Walker Type II method is equivalent with the Anderson Acceleration scheme [2] for fixed-point iterations [22, 60].

For the methods described by Fang et al., it is essential to have a (non-trivial) initial approximation of the inverse Jacobian. Dealing with partitioned FSI coupling this is often not possible which renders a lot of their variants inapplicable here, including the hybrid variants that mix the Type I and Type II update. Choosing trivial initial guesses such as the zero-matrix translates the IQN-IMVJ method to the IQN-ILS method for the Type II update in the very first iteration, but leads to linear dependencies in the secant equation for the Type I update.

In the following, we present a subset of the above mentioned quasi-Newton variants together with some further extensions. Thereby, structural differences, similarities as well as possible

drawbacks and advantages are outlined. The subsequent methods can merely be categorized by three criteria:

(1) To obtain a unique solution for the Jacobian approximation, the secant equation is enhanced by a additional minimization of the Jacobian difference to an initial guess for the current time step. The initial guess can either be taken over from the approximation for the last time step or can be fixed to the zero-matrix (trivial guess).

(2) The variants can be discriminated by the operator that is to be approximated. As presented above, it is advantageous to approximate for the inverse Jacobian of the residual operator $\tilde{\boldsymbol{R}}$, however, it is likewise possible to solve for an approximation of the Jacobian of the residual operator or the Jacobian of the fixed-point operator $\boldsymbol{H}$.

(3) Some methods need to set up the approximated Jacobian matrix explicitly, while for others it suffices to provide a procedure that gives the result applying part of the approximate Jacobian to $-\tilde{\boldsymbol{R}}(\tilde{\boldsymbol{x}}^k) = -\boldsymbol{R}(\boldsymbol{x}^k)$, respectively, based on this secant equation.

The following sections consider a variety of quasi-Newton methods and refer to the discriminating characteristics above.

4.1 Interface Quasi-Newton with Minimal Jacobian Norm (IQN-LS)

4.1.1 Residual Inverse Jacobian Approximation (IQN-ILS)

The interface quasi-Newton least squares method introduced by Degroote et. al [17] estimates the inverse of the Jacobian of the residual operator $\boldsymbol{R}(\boldsymbol{x}) = \boldsymbol{H}(\boldsymbol{x}) - \boldsymbol{x}$, i. e., $\widehat{\boldsymbol{J}_{\tilde{R}}}^{-1}(\tilde{\boldsymbol{x}}^k)$. It is shown to be equivalent [22] to a variant of the Anderson Acceleration scheme [2] and refers to the Type II update in Walker et al. [60].

In order to find an approximation of the inverse Jacobian, we choose[1] $\widehat{\boldsymbol{J}_{\tilde{R}}}^{-1}(\tilde{\boldsymbol{x}}^k)$, such that it fulfills a linearization of the residual equation (4.3). To be specific, we collect input-output data throughout the iterations within a time step and generate the following matrices[2]

$$\begin{aligned} \boldsymbol{W}_k &= (\boldsymbol{w}_i^k)_{i=0}^{k-1} = \left[\Delta\tilde{\boldsymbol{x}}_0^k, \Delta\tilde{\boldsymbol{x}}_1^k, \cdots, \Delta\tilde{\boldsymbol{x}}_{k-1}^k\right], \qquad \text{with } \Delta\boldsymbol{H}_i^k = \Delta\tilde{\boldsymbol{x}}_i^k = \tilde{\boldsymbol{x}}^k - \tilde{\boldsymbol{x}}^i \\ \boldsymbol{V}_k &= (\boldsymbol{v}_i^k)_{i=0}^{k-1} = \left[\Delta\boldsymbol{R}_0^k, \Delta\boldsymbol{R}_1^k, \cdots, \Delta\boldsymbol{R}_{k-1}^k\right], \qquad \text{with } \Delta\boldsymbol{R}_i^k = \boldsymbol{R}(\boldsymbol{x}^k) - \boldsymbol{R}(\boldsymbol{x}^i). \end{aligned} \tag{4.4}$$

The inverse Jacobian then approximately fulfills the secant equation, i.e.,

$$\boldsymbol{J}_{\tilde{R}}^{-1}(\tilde{\boldsymbol{x}}^k)\, \boldsymbol{V}_k \approx \boldsymbol{W}_k.$$

This system of equations serves as a set of constraints for the entries of the approximate inverse Jacobian $\widehat{\boldsymbol{J}_{\tilde{R}}}^{-1}(\tilde{\boldsymbol{x}}^k)$.As the number of iterations per time step k is in general much smaller than the number of degrees of freedom at the coupling interface, $\boldsymbol{W}_k \in \mathbb{R}^{n\times k}$ and $\boldsymbol{V}_k \in \mathbb{R}^{n\times k}$ are tall and thin matrices. The system of equations for the entries of $\widehat{\boldsymbol{J}_{\tilde{R}}}^{-1}(\tilde{\boldsymbol{x}}^k)$ is therefore underdetermined:

$$\widehat{\boldsymbol{J}_{\tilde{R}}}^{-1}(\tilde{\boldsymbol{x}}^k)\boldsymbol{V}_k = \boldsymbol{W}_k. \tag{4.5}$$

To enhance equation (4.5) to a system with a unique solution for $\widehat{\boldsymbol{J}_{\tilde{R}}}^{-1}(\tilde{\boldsymbol{x}}^k)$, the IQN-ILS method uses the norm minimization

$$\left\|\widehat{\boldsymbol{J}_{\tilde{R}}}^{-1}(\tilde{\boldsymbol{x}}^k)\right\|_F \to \min. \tag{4.6}$$

Here, $\|\cdot\|_F$ is the Frobenius norm. This can be seen as the minimum change with respect to the zero-matrix. From this point of view the IQN-ILS corresponds to a member in the generalized Broyden family described by Fang et al. [23] with the previous inverse Jacobian approximation fixed to zero.

[1] Using Taylor expansion, we find a linearization of the residual equation (4.3) as $\boldsymbol{T}(\tilde{\boldsymbol{R}} \mid \boldsymbol{x}, \tilde{\boldsymbol{x}}^k) = \tilde{\boldsymbol{R}}(\tilde{\boldsymbol{x}}^k) + \boldsymbol{J}_{\tilde{\boldsymbol{R}}}(\tilde{\boldsymbol{x}}^k)(\boldsymbol{x} - \tilde{\boldsymbol{x}}^k) + \mathcal{R}(\tilde{\boldsymbol{R}} \mid \boldsymbol{x}, \tilde{\boldsymbol{x}}^k)$. Hence, we find an approximation of the Jacobian in terms of input-output modes as $\boldsymbol{J}_{\tilde{R}}^{-1}(\tilde{\boldsymbol{x}}^k)(\tilde{\boldsymbol{x}}^{k+1} - \tilde{\boldsymbol{x}}^k) \approx \tilde{\boldsymbol{R}}(\tilde{\boldsymbol{x}}^{k+1}) - \tilde{\boldsymbol{R}}(\tilde{\boldsymbol{x}}^k) = \boldsymbol{R}(\boldsymbol{x}^{k+1}) - \boldsymbol{R}(\boldsymbol{x}^k)$. Consequently, each pair of input-output modes generates a constraint for the approximation of the Jacobian or its inverse.

[2] In order to facilitate the presentation of the quasi-Newton methods, we assume that we have already completed $k+1$ FSI coupling iterations. As the approximation of the system Jacobian needs at least two input-output pairs, a constant underrelaxation is performed in the first iteration, i. e., $\boldsymbol{x}^1 = \boldsymbol{x}^0 + \omega\boldsymbol{R}(\boldsymbol{x}^0)$

This finally leads to the approximate inverse Jacobian

$$\widehat{\boldsymbol{J}_{\tilde{R}}}^{-1}(\tilde{\boldsymbol{x}}^k) = \boldsymbol{W}_k \left(\boldsymbol{V}_k^T \boldsymbol{V}_k\right)^{-1} \boldsymbol{V}_k^T \tag{4.7}$$

and the resulting quasi-Newton update formula

$$\boldsymbol{x}^{k+1} = \tilde{\boldsymbol{x}}^k + \boldsymbol{W}_k \underbrace{(\boldsymbol{V}_k^T \boldsymbol{V}_k)^{-1} \boldsymbol{V}_k \left(-\boldsymbol{R}(\boldsymbol{x}^k)\right)}_{=: \boldsymbol{\alpha}}.$$

The above can be seen as follows. Solving for the minimum of the Jacobian in the Frobenius norm under the side constraints of the secant equation (4.5) can be done using Lagrangian multipliers. The Lagrange functional reads as follows

$$\boldsymbol{L}((\mathcal{J}_{i,j})_{i,j=1}^n, \boldsymbol{\lambda}) = \frac{1}{2} \left\| \widehat{\boldsymbol{J}_{\tilde{R}}}^{-1}(\tilde{\boldsymbol{x}}^k) \right\|_F^2 + \boldsymbol{\lambda} \left(\widehat{\boldsymbol{J}_{\tilde{R}}}^{-1}(\tilde{\boldsymbol{x}}^k) \boldsymbol{V}_k - \boldsymbol{W}_k \right)$$

with the unknowns $\boldsymbol{\lambda}$ and $\widehat{\boldsymbol{J}_{\tilde{R}}}^{-1} = (\mathcal{J}_{i,j})_{i,j=1}^n$.
A solution $\left(\mathcal{J}_{i,j}^*\right)_{i,j=1}^n, \boldsymbol{\lambda}^*$ has to fulfill the optimality conditions

$$\begin{aligned} \nabla_{(\mathcal{J}_{i,j})} \boldsymbol{L} &= \widehat{\boldsymbol{J}_{\tilde{R}}^*}^{-1} + \boldsymbol{\lambda}^* \boldsymbol{V}_k^T \stackrel{!}{=} 0 \\ \nabla_{\lambda} \boldsymbol{L} &= \widehat{\boldsymbol{J}_{\tilde{R}}^*}^{-1} \boldsymbol{V}_k - \boldsymbol{W}_k \stackrel{!}{=} 0, \end{aligned} \tag{4.8}$$

that can be solved for $\lambda^* = -\boldsymbol{W}_k \left(\boldsymbol{V}_k^T \boldsymbol{V}_k\right)^{-1}$, inserting the first equation into the second. This finally leads to the expression for the approximate inverse Jacobian, depicted above in equation (4.7).

IQN-ILS as Matrix-Free Approach

For this quasi-Newton variant we do not have to explicitly compute the inverse Jacobian matrix, but it suffices to provide a procedure that computes the result of parts of the Jacobian with the vector $-\boldsymbol{R}(\boldsymbol{x}^k)$. Hence, we can restrict ourselves to compute only the vector $\boldsymbol{\alpha}$ (compare [57]). This can be realized very efficiently by solving the least squares problem

$$\min_{\alpha \in \mathbb{R}^k} \|\boldsymbol{V}_k \boldsymbol{\alpha} + \boldsymbol{R}(\boldsymbol{x}^k)\|_2 \,, \tag{4.9}$$

where $\|\cdot\|_2$ denotes the Euclidian norm.[3]

[3] From the update formula for $\boldsymbol{x}^{k+1}$ one sees, that performing a quasi-Newton step without performing a fixed-point iteration (computing $\tilde{\boldsymbol{x}}^k = \boldsymbol{H}(\boldsymbol{x}^k)$) in advance would have led to linearly independent columns in $\boldsymbol{W}_k$. In this case we would have to use the input modes $\boldsymbol{W}_k = (\Delta\boldsymbol{x}_0^k, \Delta\boldsymbol{x}_1^k, \ldots, \Delta\boldsymbol{x}_{k-1}^k)$ with $\Delta\boldsymbol{x}_i^k = \boldsymbol{x}^k - \boldsymbol{x}^i$ and hence, $\boldsymbol{x}^k$ would be corrected to $\boldsymbol{x}^{k+1}$ by adding multiples of differences from $\boldsymbol{x}^k - \boldsymbol{x}^i$ from previous iterations. As a result, all columns of $\boldsymbol{W}_k$ would be in the space spanned by $\boldsymbol{x}^0$ and $\boldsymbol{x}^1 - \boldsymbol{x}^0$, which can be seen by induction over the iterations.

To see this, we consider the estimation of the inverse Jacobian from an other perspective which turns out to be equivalent to the above derivation. To find $\boldsymbol{x}^{k+1}$, we construct $\Delta\tilde{\boldsymbol{x}}^k = \boldsymbol{x}^{k+1} - \tilde{\boldsymbol{x}}^k$ in the column space of $\boldsymbol{W}_k$: $\Delta\tilde{\boldsymbol{x}}^k = \boldsymbol{W}_k\boldsymbol{\alpha}$, such that a linearization of the residual equation (4.3) is fulfilled as accurately as possible. To this end we solve the least squares problem depicted in eq. (4.9), i. e., we do a l_2-projection of the residuum $-\boldsymbol{R}(\boldsymbol{x}^k)$ onto the column space of $\boldsymbol{V}_k$, such that $\boldsymbol{V}_k\boldsymbol{\alpha} \doteq -\boldsymbol{R}(\boldsymbol{x}^k)$.

The quasi-Newton update can then be approximated as $\Delta\tilde{\boldsymbol{x}}^k := \boldsymbol{W}_k\boldsymbol{\alpha}$, since

$$\widehat{\boldsymbol{J}_{\tilde{R}}}^{-1}(\boldsymbol{x}^k)(\boldsymbol{x}^{k+1} - \tilde{\boldsymbol{x}}^k) = \widehat{\boldsymbol{J}_{\tilde{R}}}^{-1}(\boldsymbol{x}^k)\boldsymbol{W}_k\boldsymbol{\alpha} = \widehat{\boldsymbol{J}_{\tilde{R}}}^{-1}(\boldsymbol{x}^k)\sum_{i=0}^{k-1}\alpha_i\Delta\tilde{\boldsymbol{x}}_i^k \doteq \sum_{i=0}^{k-1}\alpha_i\Delta\boldsymbol{R}_i^k = \boldsymbol{V}_k\boldsymbol{\alpha} \approx -\boldsymbol{R}(\boldsymbol{x}^k)$$

and we find the next iterate as $\boldsymbol{x}^{k+1} = \tilde{\boldsymbol{x}}^k + \boldsymbol{W}_k\boldsymbol{\alpha}$.

Remark 4.1.1. The above matrix-free version of the IQN-ILS is clearly superior to the explicit matrix approach with respect to computational effort and storage requirement. Still, the explicit matrix approach is at least of theoretical interest. It should yield exactly the same result as the matrix free approach which is however not the case due to different numerics and finite precision. The explicit matrix approach – denoted by IQN-ILSJ – is expected to be slightly more robust.

Benefit from Information from Previous Time Steps

The convergence properties of the IQN-ILS method can be greatly improved if the input-output information from previous time steps is incorporated into the secant equation, i.e., into $\boldsymbol{W}_k$ and $\boldsymbol{V}_k$. To achieve this, the difference matrices $\boldsymbol{V}^{\{n+1-R\}}, \cdots, \boldsymbol{V}^{\{n\}}, \boldsymbol{V}^{\{n+1\}}$ and $\boldsymbol{W}^{\{n+1-R\}}, \cdots, \boldsymbol{W}^{\{n\}}, \boldsymbol{W}^{\{n+1\}}$ from the previous $R \in \mathbb{N}$ time steps are stored and included in the secant equation of the current time step, i. e., we replace $\boldsymbol{W}_k$ and $\boldsymbol{V}_k$ by the enhanced versions

$$\boldsymbol{W}_k^{(R)} = \left[\boldsymbol{W}^{\{n+1-R\}}, \boldsymbol{W}^{\{n-R\}}, \cdots, \boldsymbol{W}_k^{\{n+1\}}\right],$$
$$\boldsymbol{V}_k^{(R)} = \left[\boldsymbol{V}^{\{n+1-R\}}, \boldsymbol{V}^{\{n-R\}}, \cdots, \boldsymbol{V}_k^{\{n+1\}}\right].$$

This additional information significantly improves the convergence as shown in [17] and Chapter 5 and 7. However, the optimal parameter R of reused time steps is highly problem dependent and there is no analytical method available to determine the optimal R. Thus, in practice, R has to be determined based on experiences which is a costly try and error process. Also linear dependencies and contradicting information within the accumulated difference matrices need to be handled properly. In Section 4.3 we present an alternative quasi-Newton approach that provides an automatic implicit incorporation of information from passed time steps and, thus, avoids these drawbacks of the IQN-ILS method. However, this requires to explicitly compute $\widehat{\boldsymbol{J}_{\tilde{R}}}^{-1}$ instead of only the short vector $\boldsymbol{\alpha} \in \mathbb{R}^k$.

A pseudo code of the IQN-ILS method is depicted in Algorithm 2.

Input: initial value $\boldsymbol{x}^0 = \text{extr}(\boldsymbol{x}^*_n, \boldsymbol{x}^*_{n-1}, \boldsymbol{x}^*_{n-2})$ time extrapolation from previous time steps
Output: $\boldsymbol{x}^*_{n+1}$, converged configuration of current time step.

IQN-ILS$(\boldsymbol{x}^0)$
$\tilde{\boldsymbol{x}}^0 = \boldsymbol{H}(\boldsymbol{x}^0)$ and $\boldsymbol{R}^0 = \tilde{\boldsymbol{x}}^0 - \boldsymbol{x}^0$
$\boldsymbol{x}^1 = \boldsymbol{x}^0 + \omega_0 \cdot \boldsymbol{R}^0$
for $k = 1 \ldots$ **do**
  $\tilde{\boldsymbol{x}}^k = \boldsymbol{H}(\boldsymbol{x}^k)$ and $\boldsymbol{R}^k = \tilde{\boldsymbol{x}}^k - \boldsymbol{x}^k$
  $\boldsymbol{V}_k = [\Delta\boldsymbol{R}^k_0, \ldots, \Delta\boldsymbol{R}^k_{k-1}]$ with $\Delta\boldsymbol{R}^k_i = \boldsymbol{R}^i - \boldsymbol{R}^k$
  $\boldsymbol{W}_k = [\Delta\tilde{\boldsymbol{x}}^k_0, \ldots, \Delta\tilde{\boldsymbol{x}}^k_{k-1}]$ with $\Delta\tilde{\boldsymbol{x}}^k_i = \tilde{\boldsymbol{x}}^i - \tilde{\boldsymbol{x}}^k$
  decompose $\boldsymbol{V}^k = \boldsymbol{Q}^k\boldsymbol{U}^k$
  solve the first k lines of $\boldsymbol{U}^k\boldsymbol{\alpha} = -\boldsymbol{Q}^{k^T}\boldsymbol{R}^k$
  $\Delta\tilde{\boldsymbol{x}}^k = \boldsymbol{W}_k\boldsymbol{\alpha}$
  $\boldsymbol{x}^{k+1} = \tilde{\boldsymbol{x}}^k + \Delta\tilde{\boldsymbol{x}}^k$
end

Algorithm 2: IQN-ILS quasi-Newton method. Pseudo code of the IQN-ILS algorithm, a matrix-free approach that approximates for the inverse Jacobian of the residual operator.

Relation to Aitken's Relaxation

There is a relation of the IQN-ILS method to the dynamic Aitken's relaxation, presented in Chapter 3. Indeed, the latter can be seen as the simplest quasi-Newton method available. Assume, that solely the most recent input-output modes $\boldsymbol{m}^k_{k-1} = \boldsymbol{x}^k - \boldsymbol{x}^{k-1}$ and $\boldsymbol{v}^k_{k-1} = \boldsymbol{r}_k - \boldsymbol{r}_{k-1}$, rather than all $k+1$ available modes are used to approximate the inverse system Jacobian. Then, following the above derivation of the IQN-ILS method leads to a formula for the next iterate that is very similar to the formula for Aitken's dynamic relaxation (3.10). Note that we use the input mode $\boldsymbol{m}^k_{k-1}$ instead of $\boldsymbol{w}^k_{k-1} = \tilde{\boldsymbol{x}}^k - \tilde{\boldsymbol{x}}^i$ for this excursus which is equivalent to skipping the fixed-point iteration step. For the IQN-ILS method, using $\boldsymbol{m}^k_i$ would lead to linear dependent columns[3] in $\boldsymbol{W}_k$, see Remark 4.3.2.

We shortly recapitulate the formula for Aitken's dynamic relaxation

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \omega_k\boldsymbol{r}^k = \boldsymbol{x}^k - \omega_{k-1}\frac{(\boldsymbol{r}_{k-1})^T(\boldsymbol{r}_k - \boldsymbol{r}_{k-1})}{\|\boldsymbol{r}_k - \boldsymbol{r}_{k-1}\|^2_{L^2}}\boldsymbol{r}^k$$

From this we see that for Aitken's dynamic relaxation the most recent input mode $\boldsymbol{m}^k_{k-1}$ can be written as $\boldsymbol{x}^k - \boldsymbol{x}^{k-1} = -\omega_{k-1}\boldsymbol{r}^{k-1}$. Following the derivation of the IQN-ILS method, we now replace the matrices $\boldsymbol{V}$ and $\boldsymbol{W}$ by $\underline{\boldsymbol{V}} = [\boldsymbol{v}^k_{k-1}]$ and $\underline{\boldsymbol{M}} = [\boldsymbol{m}^k_{k-1}]$ and try to approximate the system Jacobian with the result of only two iterations by solving for the minimum norm of

the Jacobian, i. e., $\|\widehat{\boldsymbol{J}_{\tilde{R}}}^{-1}\|_F \rightarrow$ min, with respect to the secant equation $\widehat{\boldsymbol{J}_{\tilde{R}}}^{-1}\underline{\boldsymbol{V}} = \underline{\boldsymbol{M}}$.
A solution to the above minimization problem is given by

$$\widehat{\boldsymbol{J}_{\tilde{R}}}^{-1} = \boldsymbol{m}_{k-1}^k \left((\boldsymbol{v}_{k-1}^k)^T \boldsymbol{v}_{k-1}^k\right)^{-1} (\boldsymbol{v}_{k-1})^T,$$

compare eq. (4.7). Thus, the resulting iteration reads

$$\begin{aligned}
\boldsymbol{x}^{k+1} &= \boldsymbol{x}^k + \boldsymbol{m}_{k-1}^k \left((\boldsymbol{v}_{k-1}^k)^T \boldsymbol{v}_{k-1}^k\right)^{-1} (\boldsymbol{v}_{k-1}^k)^T \left(-\boldsymbol{R}(\boldsymbol{x}^k)\right) \\
&= \boldsymbol{x}^k + \frac{(\boldsymbol{x}^k - \boldsymbol{x}^{k-1})(\boldsymbol{r}^k - \boldsymbol{r}^{k-1})^T}{\|\boldsymbol{r}^k - \boldsymbol{r}^{k-1}\|_{L^2}^2}(-\boldsymbol{r}^k) \\
&= \boldsymbol{x}^k - \omega_{k-1}\frac{\boldsymbol{r}^{k-1}(\boldsymbol{r}^k - \boldsymbol{r}^{k-1})^T}{\|\boldsymbol{r}^k - \boldsymbol{r}^{k-1}\|_{L^2}^2}\boldsymbol{r}^k
\end{aligned}$$

For the last equality, we used the fact that for Aitken's relaxation the difference between iterates can be written as $\boldsymbol{x}^k - \boldsymbol{x}^{k-1} = -\omega_{k-1}\boldsymbol{r}^k$. The above formula looks very similar to the update formula for Aitken's relaxation (3.11), yet, there is a not irrelevant difference. Aitken's method enforces the Jacobian approximation to be a scalar $\omega_k = {}^{(\cdot)^T(\cdot)}/_{\|\cdot\|_2^2}$ while the above derivation results in a matrix approximation ${}^{(\cdot)(\cdot)^T}/_{\|\cdot\|_2^2}$. For the one-dimensional case, both representations are equal.

4.1.2 Residual Jacobian Approximation (IQN-LSJ)

This variant of the interface quasi-Newton least squares method estimates the Jacobian of the residual operator $\boldsymbol{R}(\boldsymbol{x}) = \boldsymbol{H}(\boldsymbol{x}) - \boldsymbol{x}$ and constructs the Jacobian matrix explicitly. Clearly, this alternative does not have any practical utility as it does not offer any advantages over the IQN-ILS method described earlier while entailing two remarkably drawbacks: Firstly, the Jacobian matrix needs to be set up explicitly and secondly, it involves to solve a linear system of equation in order to obtain the Newton update, in equation (4.2).
Still, the method is of theoretical interest, especially in comparison to the IQN-ILS method. Both methods are mathematically identical and are therefore expected to perform similar. Due to different numerics and different Newton update steps the two methods are, however, likely to perform different.

As for the IQN-ILS, we minimize the Frobenius norm of the Jacobian to enhance the secant equation to have a unique solution, i. e., we solve

$$\left\|\widehat{\boldsymbol{J}_{\tilde{R}}}(\tilde{\boldsymbol{x}}^k)\right\|_F \rightarrow \text{min} \quad \textbf{subject to} \quad \widehat{\boldsymbol{J}_{\tilde{R}}}(\tilde{\boldsymbol{x}}^k)\boldsymbol{W}_k = \boldsymbol{V}_k$$

using Lagrangian multipliers, obtaining the approximation of the Jacobian

$$\widehat{\boldsymbol{J}_{\tilde{R}}}(\tilde{\boldsymbol{x}}^k) = \boldsymbol{V}_k \left(\boldsymbol{W}_k^T \boldsymbol{W}_k\right)^{-1} \boldsymbol{W}_k^T \tag{4.10}$$

and the resulting quasi-Newton update formula reads

$$\begin{aligned}
\text{solve} \quad & \widehat{\boldsymbol{J}_{\tilde{R}}}(\tilde{\boldsymbol{x}}^k)\Delta\tilde{\boldsymbol{x}}^k = -\boldsymbol{R}(\boldsymbol{x}^k), \\
\text{set} \quad & \boldsymbol{x}^{k+1} = \tilde{\boldsymbol{x}}^k + \Delta\tilde{\boldsymbol{x}}^k.
\end{aligned}$$

Alternative Derivation from a Least Squares Point of View

We briefly consider the derivation of eq. (4.10) from the least squares point of view (compare [33]) in order to clarify the understanding that both variants are equivalent. This is valuable as the derivations are pretty much mixed up in literature. Estimating a Jacobian that fulfills the linearized residual equation as good as possible, we write the Newton update as linear combination of the input modes, see eq. (4.4),

$$\Delta\tilde{\boldsymbol{x}}^k = \sum_{i=0}^{k-1} \alpha_i w_i^k + \boldsymbol{\epsilon}$$

where α_i denotes the coordinates of $\Delta\tilde{\boldsymbol{x}}^k$ with respect to the input modes and $\boldsymbol{\epsilon}$ the part of $\Delta\tilde{\boldsymbol{x}}^k$ that lies outside the range of $\boldsymbol{W}_k$, i. e., $\boldsymbol{\epsilon} \notin \mathcal{R}(\boldsymbol{W}_k)$.
We want $\boldsymbol{\epsilon}$ to be minimal with respect to the l_2-norm which is basically a least squares approximation:

$$\boldsymbol{\alpha} = \text{argmin}_\beta \|\boldsymbol{\epsilon}\|_2 = \text{argmin}_\beta \|\Delta\tilde{\boldsymbol{x}}^k - \boldsymbol{W}_k\boldsymbol{\beta}\|_2$$

To this end, impose $\boldsymbol{\epsilon} \perp \mathcal{R}(\boldsymbol{W}_k)$ w. r. t. the standard scalar product. We can now rewrite $\Delta\tilde{\boldsymbol{x}}^k$ as $\Delta\tilde{\boldsymbol{x}}^k = \boldsymbol{W}_k\boldsymbol{\alpha} + \boldsymbol{\epsilon}$ which leads to

$$\begin{aligned} 0 = \boldsymbol{W}_k^T\boldsymbol{\epsilon} &= \boldsymbol{W}_k^T(\Delta\tilde{\boldsymbol{x}}^k - \boldsymbol{W}_k\boldsymbol{\alpha}) \\ \boldsymbol{\alpha} &= (\boldsymbol{W}_k^T\boldsymbol{W}_k)^{-1}\boldsymbol{W}_k^T\Delta\tilde{\boldsymbol{x}}^k \end{aligned}$$

The vector $\boldsymbol{\alpha}$ is now used to make a prediction to the output, i. e., we write the same linear combination with respect to the output modes

$$\begin{aligned} \boldsymbol{R}(\boldsymbol{x}^{k+1}) - \boldsymbol{R}(\boldsymbol{x}^k) &\approx \boldsymbol{V}_k\alpha = \boldsymbol{V}_k(\boldsymbol{W}_k^T\boldsymbol{W}_k)^{-1}\boldsymbol{W}_k^T\Delta\tilde{\boldsymbol{x}}^k \\ \boldsymbol{R}(\boldsymbol{x}^{k+1}) &\approx \boldsymbol{R}(\boldsymbol{x}^k) + \underbrace{\boldsymbol{V}_k(\boldsymbol{W}_k^T\boldsymbol{W}_k)^{-1}\boldsymbol{W}_k^T}_{\approx\widehat{J_{\tilde{R}}}=\frac{\partial R(x)}{\partial x}}\Delta\tilde{\boldsymbol{x}}^k \end{aligned}$$

From this, we see, that the expression $\boldsymbol{V}_k(\boldsymbol{W}_k^T\boldsymbol{W}_k)^{-1}\boldsymbol{W}_k^T$ approximately fulfills the role of the derivative of $\boldsymbol{R}$ which is exactly identical to eq. (4.10).

4.1.3 Fixed-Point Operator Jacobian Approximation (IQN-LSHJ)

This section briefly revisits a quasi-Newton variant which was introduced in [33] for the serial system and in [59] for a block-iterative system, respectively. It approximates the Jacobian of the fixed-point operator $\widehat{\boldsymbol{J}_H}(\boldsymbol{x})$ as the solution to the optimality problem

$$\left\|\widehat{\boldsymbol{J}_H}(\tilde{\boldsymbol{x}}^k)\right\|_F \;\rightarrow\; \min \quad \textbf{subject to} \quad \widehat{\boldsymbol{J}_H}(\tilde{\boldsymbol{x}}^k)\boldsymbol{M}_k = \boldsymbol{W}_k$$

with the adjusted input modes

$$\boldsymbol{M}_k = (\boldsymbol{m}_i^k)_{i=0}^{k-1} = \left[\Delta\boldsymbol{x}_0^k, \Delta\boldsymbol{x}_1^k, \cdots, \Delta\boldsymbol{x}_{k-1}^k\right], \quad \text{with} \;\; \Delta\boldsymbol{x}_i^k = \boldsymbol{x}^k - \boldsymbol{x}^i \tag{4.11}$$

The solution is given by

$$\widehat{\boldsymbol{J}_H}(\boldsymbol{x}^k) = \boldsymbol{W}_k \left(\boldsymbol{M}_k^T \boldsymbol{M}_k\right)^{-1} \boldsymbol{M}_k^T \tag{4.12}$$

which leads to the resulting Newton iteration

$$\begin{aligned} \text{solve} \quad & \left[\widehat{\boldsymbol{J}_H}(\boldsymbol{x}^k) - I\right] \Delta\boldsymbol{x}^k = -\boldsymbol{R}(\boldsymbol{x}^k), \\ \text{set} \quad & \boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \Delta\boldsymbol{x}^k. \end{aligned}$$

Note, that $\boldsymbol{W}_k = (\boldsymbol{w}_i^k)_{i=0}^{k-1}$ with $\boldsymbol{w}_i^k = \Delta\tilde{\boldsymbol{x}}_i^k = \Delta\boldsymbol{H}_i^k = \boldsymbol{H}(\boldsymbol{x}^k) - \boldsymbol{H}(\boldsymbol{x}^i)$, now contains the output modes.

Remark 4.1.2. Here, we derived the approximation formula (4.12) using input-output modes for the fixed-point operator H. However, eq. (4.12) can be written in terms of $\widehat{\boldsymbol{J}_H}^{-1}(\boldsymbol{x}^k)$, applying the Shermann-Morrison-Woodbury formula which shows the (theoretical) equivalence of the IQN-LSHJ to the Type I update presented in [60]. The methods clearly differ in application due to different numerics.

4.2 Interface Quasi-Newton Methods Based on Broyden's Method

Broyden's method is a very basic rank-1 update quasi-Newton method. It does not restrict the update to the system Jacobian to be either symmetric or positive definite, like other popular rank-1 modifications as for example DFP, BFGS and SR1. It is therefore a versatile method and it is a bit surprising, that Broyden's method has received little attention in the field of partitioned FSI coupling.

We briefly revisit Broyden's second method [10, 39] that approximates for a rank-1 update of the inverse Jacobian approximation. We mention the Broyden method in this exposure of quasi-Newton methods, as it depicts the base case of generalized Broyden methods and motivates the IQN-IMVJ method from the next section. For the previous methods we minimized the difference of the approximation to the zero-matrix, whereas here, we store a previous approximation and minimize its difference to the newly approximation in the Frobenius norm. For the Broyden method, the difference is minimized to the approximation from the last iteration.

Unlike the quasi-Newton methods seen so far, the Broyden method is a rank-1 update method that solely uses the information from the two most recent iterations, i. e., $\boldsymbol{w}_{k-1}^k = \tilde{\boldsymbol{x}}^k - \tilde{\boldsymbol{x}}^{k-1}$ and $\boldsymbol{v}_{k-1}^k = \boldsymbol{r}_k - \boldsymbol{r}_{k-1}$. Hence, instead of the matrices $\boldsymbol{V}_k$ and $\boldsymbol{W}_k$ we use the input-output modes $\boldsymbol{v}_{k-1}^k$ and $\boldsymbol{w}_{k-1}^k$ and set up the corresponding secant equation. This underdetermined system of equations with regards to $\widehat{\boldsymbol{J}_R}^{-1}(\tilde{\boldsymbol{x}}^k)$ is solved by minimizing the difference of two consecutive approximations of the system Jacobian in the Frobenius norm, i. e., we solve for the solution of the optimization problem

$$\left\| \widehat{\boldsymbol{J}_{\tilde{R}}}^{-1}(\tilde{\boldsymbol{x}}^k) - \widehat{\boldsymbol{J}_{\tilde{R}}}^{-1}(\tilde{\boldsymbol{x}}^{k-1}) \right\|_F \rightarrow \min \quad \textbf{subject to} \quad \widehat{\boldsymbol{J}_{\tilde{R}}}^{-1}(\tilde{\boldsymbol{x}}^k)\boldsymbol{v}_{k-1}^k = \boldsymbol{w}_{k-1}^k$$

which results in a Jacobian update formula

$$\begin{aligned} \widehat{\boldsymbol{J}_{\tilde{R}}}^{-1}(\tilde{\boldsymbol{x}}^k) &= \widehat{\boldsymbol{J}_{\tilde{R}}}^{-1}(\tilde{\boldsymbol{x}}^{k-1}) + \left(\boldsymbol{w}_{k-1}^k - \widehat{\boldsymbol{J}_{\tilde{R}}}^{-1}(\tilde{\boldsymbol{x}}^{k-1})\boldsymbol{v}_{k-1}^k \right) \left([\boldsymbol{v}_{k-1}^k]^T \boldsymbol{v}_{k-1}^k \right)^{-1} [\boldsymbol{v}_{k-1}^k]^T \\ &= \widehat{\boldsymbol{J}_{\tilde{R}}}^{-1}(\tilde{\boldsymbol{x}}^{k-1}) + \frac{\boldsymbol{w}_{k-1}^k - \widehat{\boldsymbol{J}_{\tilde{R}}}^{-1}(\tilde{\boldsymbol{x}}^{k-1})\boldsymbol{v}_{k-1}^k}{\|\boldsymbol{v}_{k-1}^k\|_2^2} [\boldsymbol{v}_{k-1}^k]^T \end{aligned} \tag{4.13}$$

The resulting Newton iteration reads as follows

$$\begin{aligned} \boldsymbol{x}^{k+1} &= \tilde{\boldsymbol{x}}^k + \widehat{\boldsymbol{J}_{\tilde{R}}}^{-1}(\tilde{\boldsymbol{x}}^k) \left(-\boldsymbol{R}(\boldsymbol{x}^k) \right) \\ &= \tilde{\boldsymbol{x}}^k - \left(\widehat{\boldsymbol{J}_{\tilde{R}}}^{-1}(\tilde{\boldsymbol{x}}^{k-1}) + \frac{\boldsymbol{w}_{k-1}^k - \widehat{\boldsymbol{J}_{\tilde{R}}}^{-1}(\tilde{\boldsymbol{x}}^{k-1})\boldsymbol{v}_{k-1}^k}{\|\boldsymbol{v}_{k-1}^k\|_2^2} [\boldsymbol{v}_{k-1}^k]^T \right) \boldsymbol{R}(\boldsymbol{x}^k). \end{aligned}$$

Minimizing for the difference of two consecutive system Jacobian approximations allows implicit reutilization of information from the past. Theoretically, this is equally possible when approximating the residual Jacobian $\widehat{\boldsymbol{J}_{\tilde{R}}}$ or the fixed-point operator Jacobian $\widehat{\boldsymbol{J}_{\boldsymbol{H}}}$. The IQN-MVJ method, presented in the next section, differs from the Broyden method as it is not a rank-1 update because it utilizes all previous input-output modes within one time step.

4.3 Interface Quasi-Newton Multiple Vector Method (IQN-MVJ)

4.3.1 Residual Inverse Jacobian Approximation (IQN-IMVJ)

The IQN-IMVJ method presented here is a newly developed quasi-Newton fluid-structure coupling approach. In conjunction with FSI coupling, the concept was first presented in [8] by Bogaers et al. for Jacobian approximations in the context of a block-iterative Newton method (B-system). Here, we combine the idea of approximating the Newton iteration defined by (4.1) and (4.2) based on the secant equation (4.5) with the concepts of Bogaers et al. This approach is equivalent to the generalized Broyden method with Type II update in the classification of Fang et al. [23], imposing the natural choice of minimizing over time steps, i. e., performing a rank-k update.

This practice of minimizing the change of the inverse Jacobian approximation in the Frobenius norm with respect to the approximation from the last time step enhances the method to implicitly use information from previous time steps in directions where it is needed. The resulting minimization problem reads as follows

$$\begin{aligned} &\left\| \widehat{\boldsymbol{J}_{\tilde{R}}}^{-1}(\tilde{\boldsymbol{x}}^k) - \widehat{\boldsymbol{J}_{\tilde{R}prev}}^{-1} \right\|_F \rightarrow \min \\ \textbf{subject to} \quad &\left(\widehat{\boldsymbol{J}_{\tilde{R}}}^{-1}(\tilde{\boldsymbol{x}}^k) - \widehat{\boldsymbol{J}_{\tilde{R}prev}}^{-1} \right) \boldsymbol{V}_k = \left(\boldsymbol{W}_k - \widehat{\boldsymbol{J}_{\tilde{R}prev}}^{-1} \boldsymbol{V}_k \right) \end{aligned} \tag{4.14}$$

Here, $\widehat{\boldsymbol{J}_{\tilde{R}prev}}^{-1}$ denotes the last inverse Jacobian approximation of the previous time step. We get the approximate inverse Jacobian

$$\widehat{\boldsymbol{J}_{\tilde{R}}}^{-1}(\tilde{\boldsymbol{x}}^k) = \widehat{\boldsymbol{J}_{\tilde{R}prev}}^{-1} + \left(\boldsymbol{W}_k - \widehat{\boldsymbol{J}_{\tilde{R}prev}}^{-1} \boldsymbol{V}_k \right) \left(\boldsymbol{V}_k^T \boldsymbol{V}_k \right)^{-1} \boldsymbol{V}_k^T \tag{4.15}$$

and the corresponding quasi-Newton update formula

$$\begin{aligned} \boldsymbol{x}^{k+1} &= \tilde{\boldsymbol{x}}^k + \widehat{\boldsymbol{J}_{\tilde{R}}}^{-1}(\tilde{\boldsymbol{x}}^k) \left(-\boldsymbol{R}(\boldsymbol{x}^k) \right) \\ &= \tilde{\boldsymbol{x}}^k - \left(\widehat{\boldsymbol{J}_{\tilde{R}prev}}^{-1} + \left(\boldsymbol{W}_k - \widehat{\boldsymbol{J}_{\tilde{R}prev}}^{-1} \boldsymbol{V}_k \right) \left(\boldsymbol{V}_k^T \boldsymbol{V}_k \right)^{-1} \boldsymbol{V}_k^T \right) \boldsymbol{R}(\boldsymbol{x}^k). \end{aligned}$$

Equation (4.14) illustrates that the current approximations always stay as close as possible to the approximation from the last time step. This automatically guarantees that we profit from past information without having to explicitly re-use previous $\boldsymbol{W}$ and $\boldsymbol{V}$ matrices, as deployed for the IQN-ILS method. This lessens the possibility of having linear depending columns within $\boldsymbol{W}$ and $\boldsymbol{V}$, and first and foremost renders the tuning of the optimal parameter R of reused previous time steps unnecessary. Furthermore, as information from previous time steps is matched in a minimum norm sense only, it is naturally less emphasized in the approximation. Hence, if newer information falls along the same direction as existing old information, the latter is replaced which entirely banns the risk of having contradicting information.

This advantageous behaviour, however, comes at the cost of the necessity for an explicit inverse Jacobian matrix. This is due to the fact, that we reuse the estimation of the inverse Jacobian from the previous time step and solve for an update of the approximation

$$\widehat{\boldsymbol{J}_{\tilde{R}}}^{-1}(\tilde{\boldsymbol{x}}^k) = \widehat{\boldsymbol{J}_{\tilde{R}prev}}^{-1}(\tilde{\boldsymbol{x}}^k) + \widehat{\boldsymbol{J}_{\tilde{R}prev}}^{-1}$$

Remark 4.3.1. We obtain the inverse Jacobian update rule in eq. (4.15) if we solve the minimization problem (4.8) for the Jacobian update $\widetilde{\boldsymbol{J}_{\tilde{R}}}^{-1}(\tilde{\boldsymbol{x}}^k)$. This shows the relation of the IQN-IMVJ method to the IQN-ILS method. For the latter, it holds that $\widehat{\boldsymbol{J}_{\tilde{R}}}^{-1}(\tilde{\boldsymbol{x}}^k) = \widetilde{\boldsymbol{J}_{\tilde{R}}}^{-1}(\tilde{\boldsymbol{x}}^k)$ as the difference to the zero-matrix is minimized in the Frobenius norm. Setting $\widehat{\boldsymbol{J}_{\tilde{R}prev}}^{-1} := \boldsymbol{0}$ The minimization problem together with the secant equation in eq. (4.14) turns out to be identical to the minimization problem for the IQN-ILS method (4.6).

A clearly listing of the IQN-IMVJ method is depicted in Algorithm 3.

Input: initial value $\boldsymbol{x}^0 = \text{extr}(\boldsymbol{x}^*_n, \boldsymbol{x}^*_{n-1}, \boldsymbol{x}^*_{n-2})$ time extrapolation, initial value for $\widehat{\boldsymbol{J}_{\tilde{R}prev}}^{-1}$

Output: $\boldsymbol{x}^*$ and $\widehat{\boldsymbol{J}_{\tilde{R}}}^{-1}(\tilde{\boldsymbol{x}}^*)$ for converged configuration of current time step.

IQN-IMVJ($\boldsymbol{x}^0$)
$\tilde{\boldsymbol{x}}^0 = \boldsymbol{H}(\boldsymbol{x}^0)$ and $\boldsymbol{R}^0 = \tilde{\boldsymbol{x}}^0 - \boldsymbol{x}^0$
$\boldsymbol{x}^1 = \boldsymbol{x}^0 + \omega_0 \cdot \boldsymbol{R}^0$
for $k = 1 \ldots$ **do**
  $\tilde{\boldsymbol{x}}^k = \boldsymbol{H}(\boldsymbol{x}^k)$ and $\boldsymbol{R}^k = \tilde{\boldsymbol{x}}^k - \boldsymbol{x}^k$
  $\boldsymbol{V}_k = [\Delta\boldsymbol{R}^k_0, \ldots, \Delta\boldsymbol{R}^k_{k-1}]$ with $\Delta\boldsymbol{R}^k_i = \boldsymbol{R}^i - \boldsymbol{R}^k$
  $\boldsymbol{W}_k = [\Delta\tilde{\boldsymbol{x}}^k_0, \ldots, \Delta\tilde{\boldsymbol{x}}^k_{k-1}]$ with $\Delta\tilde{\boldsymbol{x}}^k_i = \tilde{\boldsymbol{x}}^i - \tilde{\boldsymbol{x}}^k$
  $\widehat{\boldsymbol{J}_{\tilde{R}}}^{-1}(\tilde{\boldsymbol{x}}^k) = \widehat{\boldsymbol{J}_{\tilde{R}prev}}^{-1} + \left(\boldsymbol{W}_k - \widehat{\boldsymbol{J}_{\tilde{R}prev}}^{-1}\boldsymbol{V}_k\right)\left(\boldsymbol{V}_k^T\boldsymbol{V}_k\right)^{-1}\boldsymbol{V}_k^T$
  $\Delta\tilde{\boldsymbol{x}}^k = -\widehat{\boldsymbol{J}_{\tilde{R}}}^{-1}(\tilde{\boldsymbol{x}}^k)\boldsymbol{R}^k$
  $\boldsymbol{x}^{k+1} = \tilde{\boldsymbol{x}}^k + \Delta\tilde{\boldsymbol{x}}^k$
end

Algorithm 3: IQN-IMVJ method. Pseudo code of the IQN-IMVJ algorithm which explicitly stores the matrix of the Jacobian estimation. The inverse of the Jacobian of the residual operator is approximated using information from previous time steps implicitly. In practice, the Jacobian update is performed using a QR-decomposition, see Chapter 5

Relation to the Broyden Method

As already mentioned, the IQN-IMVJ method can be seen as a generalized Broyden method, performing a rank-k update rather than a rank-one update. Both methods minimize distances between successively computed Jacobian approximations. However, Broyden minimizes between two successive quasi-Newton iterations $\|\widehat{\boldsymbol{J}_{\tilde{R}}}^{-1}(\tilde{\boldsymbol{x}}^k) - \widehat{\boldsymbol{J}_{\tilde{R}}}^{-1}(\tilde{\boldsymbol{x}}^{k-1})\|_F$, whereas the IQN-IMVJ method minimizes the distance of approximations between two successive time steps $\|\widehat{\boldsymbol{J}_{\tilde{R}}}^{-1}(\tilde{\boldsymbol{x}}^k) - \widehat{\boldsymbol{J}_{\tilde{R}prev}}^{-1}(\tilde{\boldsymbol{x}}^k)\|_F$. This is due to the fact, that all input-output modes available within the current time step are used for the approximation.

4.3.2 Residual Jacobian Approximation (IQN-MVJ)

Generalizing Broyden's first method, we can also approximate for the Jacobian of the residual operator $\boldsymbol{R}$, instead of its inverse. Likewise, it is possible to approximate for the Jacobian of the fixed-point operator $\boldsymbol{H}$, following the IQN-MVJ scheme. For both variants, the derivation is straightforward an can be obtained in analogy to the above version. Yet, the secant equations need to be adapted properly, i. e., in terms of the update matrix they read $\widetilde{\boldsymbol{J}_{\tilde{R}}}(\tilde{\boldsymbol{x}}^k)\boldsymbol{W}_k = \boldsymbol{V}_k$ for the residual operator and $\widetilde{\boldsymbol{J}_{\tilde{H}}}(\tilde{\boldsymbol{x}}^k)\boldsymbol{M}_k = \boldsymbol{W}_k$ for the fixed-point operator, respectively.

The Newton iteration then reads as in eq. (4.1), but involves the solution of a linear system of equations.

Remark 4.3.2. Towards the issue with linear dependent iterates for the IQN-ILS method. In equation (4.4) we collected differences of the intermediate results from the fixed point-equation $\tilde{\boldsymbol{x}}^k - \tilde{\boldsymbol{x}}^i$ in the matrix $\boldsymbol{W}_k$ in order to ensure proper working of the IQN-ILS method[3]. Although it is not unfavourable to use the matrix $\boldsymbol{W}_k$ for the Broyden and IQN-IMVJ methods, this is theoretical not essential, if a initial guess for the approximation of the Jacobian or its inverse is available. Compare for example [23] where all generalized Broyden methods are introduced using the plain input modes $\boldsymbol{M}_k$.

To see that, we recapitulate the definition of the matrices, gathering the input-output modes within one time step.

$$
\begin{aligned}
\boldsymbol{M}_k &= (\boldsymbol{m}_i^k)_{i=0}^{k-1} = \left[\Delta\boldsymbol{x}_0^k, \Delta\boldsymbol{x}_1^k, \cdots, \Delta\boldsymbol{x}_{k-1}^k\right], && \text{with } \Delta\boldsymbol{x}_i^k = \boldsymbol{x}^k - \boldsymbol{x}^i \\
\boldsymbol{W}_k &= (\boldsymbol{w}_i^k)_{i=0}^{k-1} = \left[\Delta\tilde{\boldsymbol{x}}_0^k, \Delta\tilde{\boldsymbol{x}}_1^k, \cdots, \Delta\tilde{\boldsymbol{x}}_{k-1}^k\right], && \text{with } \Delta\boldsymbol{H}_i^k = \Delta\tilde{\boldsymbol{x}}_i^k = \tilde{\boldsymbol{x}}^k - \tilde{\boldsymbol{x}}^i \\
\boldsymbol{V}_k &= (\boldsymbol{v}_i^k)_{i=0}^{k-1} = \left[\Delta\boldsymbol{R}_0^k, \Delta\boldsymbol{R}_1^k, \cdots, \Delta\boldsymbol{R}_{k-1}^k\right], && \text{with } \Delta\boldsymbol{R}_i^k = \boldsymbol{R}(\boldsymbol{x}^k) - \boldsymbol{R}(\boldsymbol{x}^i).
\end{aligned}
$$

The quasi-Newton update for the IQN-ILS method reads

$$
\boldsymbol{x}^{k+1} = \tilde{\boldsymbol{x}}^k + \boldsymbol{W}_k \underbrace{(\boldsymbol{V}_k^T\boldsymbol{V}_k)^{-1}\boldsymbol{V}_k\left(-\boldsymbol{R}(\boldsymbol{x}^k)\right)}_{=:\,\boldsymbol{\alpha}}.
$$

Thus, $\boldsymbol{x}^{k+1}$ is $\tilde{\boldsymbol{x}}^k$ plus a linear combination of the columns of $\boldsymbol{W}_k$. Replacing $\tilde{\boldsymbol{x}}^k$ by $\boldsymbol{x}^k$ and $\boldsymbol{W}_k$ by $\boldsymbol{M}_k$ would lead to iterates that are all linear combinations of $\boldsymbol{x}^0$ and $\boldsymbol{x}^1 - \boldsymbol{x}^0 = \tilde{\boldsymbol{x}}^0 - \boldsymbol{x}^0$. For the IQN-ILS method, it is therefore essential to use the matrix $\boldsymbol{W}_k$.

On the contrary, the quasi-Newton update rule for the IQN-IMVJ reads

$$\boldsymbol{x}^{k+1} = \tilde{\boldsymbol{x}}^k + \left(\widehat{\boldsymbol{J}_{\tilde{R}prev}}^{-1} + \left(\boldsymbol{W}_k - \widehat{\boldsymbol{J}_{\tilde{R}prev}}^{-1}\boldsymbol{V}_k\right)\left(\boldsymbol{V}_k^T\boldsymbol{V}_k\right)^{-1}\boldsymbol{V}_k^T\right)\left(-\boldsymbol{R}(\boldsymbol{x}^k)\right).$$

and $\boldsymbol{x}^{k+1}$ is a linear combination of the columns of $\boldsymbol{W}_k$, $\widehat{\boldsymbol{J}_{\tilde{R}prev}}^{-1}$ and $\widehat{\boldsymbol{J}_{\tilde{R}prev}}^{-1}\boldsymbol{V}_k$. Hence, replacing $\boldsymbol{W}_k$ by $\boldsymbol{M}_k$ would not lead to linear dependent iterates for the IQN-IMVJ method. Though, this holds true only if a suitable initial guess for the approximation of the inverse Jacobian is available. As this is not the case in the face of partitioned FSI coupling, we fix the initial guess to zero, which in particular, is equivalent to performing an IQN-ILS iteration for the first time step (compare Remark 4.3.1) and finally leads to linear dependent iterates.

The above turns out to be the main reason, why the different Type I and Type II variations in Fang et al. [23], but especially the therein introduced hybrid methods that combine both update types are not applicable to the present setting.

To stay on top of things, the methods presented above are summarized and categorized in Table 4.1 with respect to the discriminating criteria, mentioned in the introduction to this detailed consideration of quasi-Newton methods. It views all the considered variants, pointing out the form of applied norm minimization, the approximated operator and whether a matrix-free computation is possible or an explicit computation of the matrix is required.

	Norm Minimization			
	$\|\mathcal{J}\|_F \to \min$		$\|\mathcal{J} - \mathcal{J}_{prev}\|_F \to \min$	
	INVERSE JACOBIAN	JACOBIAN	INVERSE JACOBIAN	JACOBIAN
explicit Jacobian		**IQN-LSJ** $\mathcal{J} := \widehat{\boldsymbol{J}_{\tilde{R}}}$ $\widehat{\boldsymbol{J}_{\tilde{R}}}\Delta\tilde{\boldsymbol{x}}^k = -\boldsymbol{R}^k$	**IQN-IMVJ** $\mathcal{J} := \widehat{\boldsymbol{J}_{\tilde{R}}}^{-1}$ $\Delta\tilde{\boldsymbol{x}}^k := -\widehat{\boldsymbol{J}_{\tilde{R}}}^{-1}R^k$	**IQN-MVJ** $\mathcal{J} := \widehat{\boldsymbol{J}_{\tilde{R}}}$ $\widehat{\boldsymbol{J}_{\tilde{R}}}\Delta\tilde{\boldsymbol{x}}^k = -\boldsymbol{R}^k$
		IQN-LSJH $\mathcal{J} := \widehat{\boldsymbol{J}_H}$ $(\widehat{\boldsymbol{J}_H} - I)\Delta\tilde{\boldsymbol{x}}^k = -\boldsymbol{R}^k$	**Broyden-IJ** $\mathcal{J} := \widehat{\boldsymbol{J}_{\tilde{R}}}^{-1}$ $\Delta\tilde{\boldsymbol{x}}^k := -\widehat{\boldsymbol{J}_{\tilde{R}}}^{-1}\boldsymbol{R}^k$	
provides $-\widehat{\boldsymbol{J}_{\tilde{\mathbf{R}}}}^{-1}\mathbf{R^k}$	**IQN-ILS** $\mathcal{J} := \widehat{\boldsymbol{J}_{\tilde{R}}}^{-1}$ $\Delta\tilde{\boldsymbol{x}}^k := -\widehat{\boldsymbol{J}_{\tilde{R}}}^{-1}\boldsymbol{R}^k$			

Table 4.1: Categorization of quasi-Newton coupling schemes, examined in this thesis with respect to the following three criteria: The minimization expression for the Jacobian approximation with respect to the Frobenius norm, the estimated operator, i. e., $\boldsymbol{R}$, $\boldsymbol{H}$ or $\boldsymbol{F}$ and $\boldsymbol{S}$ as well as approximating for the inverse or forward Jacobian.

4.4 Combining the Block-Iterative System with a Quasi-Newton Method

In this section, we focus on the solution of the fixed-point equation (3.5) that stems from the serial-implicit block-iterative coupling system, using a generic quasi-Newton method for post-processing. As mentioned earlier, this coupling system offers some kind of speciality compared to the two remaining systems, i. e., the serial-implicit and parallel-implicit system. The rather complicated formulation of the fixed-point equation stems from the intention to solve the system using a block-iterative system with interlaced quasi-Newton post-processing, as depicted in the schematic representation in Algorithm 1.

Obviously, it is likewise possible to solve the B-system following the methods presented above, that means to approximate the Jacobian or its inverse for the residual operator or the fixed-point operator itself. This would however not allow for a block-iterative solution to the fixed-point equation including the interlacing of post-processing. It would rather yield the same result that would be obtained if the squared V-system was solved. The following elaboration is based on versions of the method in [19] and [31].

We start from the serial block-iterative fixed-point equation (3.5), which can be written in residual form

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix} = \begin{pmatrix} \boldsymbol{R}_F(x) \\ \boldsymbol{R}_S(x) \end{pmatrix} = \begin{pmatrix} \boldsymbol{F} \circ \boldsymbol{S}(\boldsymbol{x}_f) - \boldsymbol{x}_f \\ \boldsymbol{S} \circ \boldsymbol{F}(\boldsymbol{x}_d) - \boldsymbol{x}_d \end{pmatrix} = \begin{bmatrix} \boldsymbol{F} \circ \boldsymbol{S} - I & 0 \\ 0 & \boldsymbol{S} \circ \boldsymbol{F} - I \end{bmatrix} \begin{pmatrix} \boldsymbol{x}_f \\ \boldsymbol{x}_d \end{pmatrix}$$

This coupled system can be solved by computing the system Jacobian and solving for an Newton update in direction

$$\begin{bmatrix} \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{S}} \frac{\partial \boldsymbol{S}}{\partial \boldsymbol{x}_f} - I & \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{S}} \\ \frac{\partial \boldsymbol{S}}{\partial \boldsymbol{F}} & \frac{\partial \boldsymbol{S}}{\partial \boldsymbol{F}} \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{x}_d} - I \end{bmatrix} \begin{pmatrix} \Delta \boldsymbol{x}_f \\ \Delta \boldsymbol{x}_d \end{pmatrix} = \begin{bmatrix} \boldsymbol{R}_F(\boldsymbol{x}) \\ \boldsymbol{R}_S(\boldsymbol{x}) \end{bmatrix} \tag{4.16}$$

It can be easily seen, that $\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{S}} = \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{x}_d}$ and $\frac{\partial \boldsymbol{S}}{\partial \boldsymbol{F}} = \frac{\partial \boldsymbol{S}}{\partial \boldsymbol{x}_f}$. As we aim to solve this system in a block-iterative manner, we define $\boldsymbol{J}_F := \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{x}_d}$ and $\boldsymbol{J}_S := \frac{\partial \boldsymbol{S}}{\partial \boldsymbol{x}_f}$ and try to find approximations $\widehat{\boldsymbol{J}_F}$ and $\widehat{\boldsymbol{J}_S}$ for the Jacobians of the fluid and structural interface operators, that are updated in each coupling iteration.

As suggested in Alg. 1 the block Gauß-Seidel solution of (4.16) implies a special ordering of the solver inputs and outputs together with the interlaced post-processing, i. e.,

$$^{k-1} \cdots \curvearrowright \underbrace{\tilde{\boldsymbol{x}}_f^k = \boldsymbol{F}(\boldsymbol{x}_d^k)}_{\text{invoke } \boldsymbol{F}} \overset{*}{\underset{\widehat{\boldsymbol{J}_F}'}{\curvearrowright}} \underbrace{\tilde{\boldsymbol{x}}_f^k \overset{(PP)}{\rightsquigarrow} \boldsymbol{x}_f^{k+1}}_{\text{QN update}} \curvearrowright \underbrace{\tilde{\boldsymbol{x}}_d^k = \boldsymbol{S}(\boldsymbol{x}_f^{k+1})}_{\text{invoke } \boldsymbol{S}} \overset{*}{\underset{\widehat{\boldsymbol{J}_S}'}{\curvearrowright}} \underbrace{\tilde{\boldsymbol{x}}_d^k \overset{(PP)}{\rightsquigarrow} \boldsymbol{x}_d^{k+1}}_{\text{QN update}} \curvearrowright \cdots \, ^{k+1}$$

The fluid and solid solvers do not directly compute the coupling values used as input by the respective counterpart solver, unlike in the S-system. Here, $\widehat{\boldsymbol{J}_F}'$ and $\widehat{\boldsymbol{J}_S}'$ at the marked positions $\overset{*}{\curvearrowright}$ indicate, that the approximations for the Jacobian of the fluid or structural

interface operator are updated before the respective post-processing step – namely performing a quasi-Newton iteration.

In order to solve the block-iterative coupling scheme, we assume that $k-1$ coupling iterations have been performed. As a first step of coupling iteration k, the fluid solver is called, which returns an interface traction $\boldsymbol{x}_f$ as an intermediate result. Consequently, the residual $\boldsymbol{R}_F(\boldsymbol{x}_f, \boldsymbol{x}_d)$ changes and the approximation for the Jacobian of the fluid interface operator is updated $\widehat{\boldsymbol{J}_F}^{k-1} \rightsquigarrow \widehat{\boldsymbol{J}_F}^{k}$.

Then, $\Delta\boldsymbol{x}_f$ is computed by solving the system

$$\left(\widehat{\boldsymbol{J}_F}^{k}\widehat{\boldsymbol{J}_S}^{k-1} - I\right)\Delta\boldsymbol{x}_f^k = -\underbrace{\boldsymbol{R}_F^k(\boldsymbol{x}_f, \boldsymbol{x}_d)}_{\tilde{\boldsymbol{x}}_f^k - \boldsymbol{x}_f^k} - \widehat{\boldsymbol{J}_F}^{k}\underbrace{\boldsymbol{R}_S^{k-1}(\boldsymbol{x}_f, \boldsymbol{x}_d)}_{\tilde{\boldsymbol{x}}_d^{k-1} - \boldsymbol{x}_d^{k-1}} \tag{4.17}$$

and the solid solver input is computed by the quasi-Newton update $\boldsymbol{x}_f^{k+1} = \boldsymbol{x}_f^k + \Delta\boldsymbol{x}_f^k$ (post-processing).

Similarly, following a call to the solid solver with the generation of the displacements $\tilde{\boldsymbol{x}}_d^k$ as an intermediate result, the residual $\boldsymbol{R}_S(\boldsymbol{x}_f, \boldsymbol{x}_d)$ changes and the estimation of the Jacobian of the solid interface operator is updated $\widehat{\boldsymbol{J}_S}^{k-1} \rightsquigarrow \widehat{\boldsymbol{J}_S}^{k}$. Thereafter, $\Delta\boldsymbol{x}_d^k$ is computed from the updated system (4.16) by solving the system

$$\left(\widehat{\boldsymbol{J}_S}^{k}\widehat{\boldsymbol{J}_F}^{k} - I\right)\Delta\boldsymbol{x}_d^k = -\underbrace{\boldsymbol{R}_S^k(\boldsymbol{x}_f, \boldsymbol{x}_d)}_{\tilde{\boldsymbol{x}}_d^k - \boldsymbol{x}_d^k} - \widehat{\boldsymbol{J}_S}^{k}\underbrace{\boldsymbol{R}_F^k(\boldsymbol{x}_f, \boldsymbol{x}_d)}_{\tilde{\boldsymbol{x}}_f^k - \boldsymbol{x}_f^k} \tag{4.18}$$

Finally, the input for the invocation of the fluid solver in the next coupling iteration $k+1$ is computed as $\boldsymbol{x}_d^{k+1} = \boldsymbol{x}_d^k + \Delta\boldsymbol{x}_d^k$.

Having said this, the only thing that needs to be discussed is how the approximations to the solver Jacobians $\widehat{\boldsymbol{J}_F}$ and $\widehat{\boldsymbol{J}_S}$ are obtained. For this purpose, we can use the methods presented above, especially the IQN-LSJ and IQN-MVJ quasi-Newton method, which ultimately act as different post-processing methods.
In order to approximate for the Jacobians $\widehat{\boldsymbol{J}_F}$ and $\widehat{\boldsymbol{J}_S}$ of the fluid and structural interface operators, rather than for the residual operator $\boldsymbol{R}$ or fixed-point operator $\boldsymbol{H}$, one needs to adjust the matrices $\boldsymbol{V}_k$ and $\boldsymbol{W}_k$ appropriately. As opposed to the previous Jacobian estimations, we now need a pair of input-output matrices for each of the interface operators, i. e., $\boldsymbol{V}_k^F, \boldsymbol{W}_k^F$ and $\boldsymbol{V}_k^S, \boldsymbol{W}_k^S$.

For the Jacobian of the fluid solver, input coupling value modes $\Delta\boldsymbol{x}_d|_{k-1}^k = \boldsymbol{x}_d^k - \boldsymbol{x}_d^{k-1}$ and (intermediate) output coupling value modes $\Delta\tilde{\boldsymbol{x}}_f|_{k-1}^k = \tilde{\boldsymbol{x}}_f^k - \tilde{\boldsymbol{x}}_f^{k-1}$ are computed and collected during the coupling iterations

$$\begin{aligned}
\boldsymbol{W}_k^F &= (\boldsymbol{w}^F|_i^k)_{i=0}^{k-1} = \left[\Delta\boldsymbol{x}_d|_0^k, \Delta\boldsymbol{x}_d|_1^k, \cdots, \Delta\boldsymbol{x}_d|_{k-1}^k\right], \qquad \text{with} \quad \Delta\boldsymbol{x}_d|_i^k = \boldsymbol{x}_d^k - \boldsymbol{x}_d{}^i \\
\boldsymbol{V}_k^F &= (\boldsymbol{v}^F|_i^k)_{i=0}^{k-1} = \left[\Delta\tilde{\boldsymbol{x}}_f|_0^k, \Delta\tilde{\boldsymbol{x}}_f|_1^k, \cdots, \Delta\tilde{\boldsymbol{x}}_f|_{k-1}^k\right], \qquad \text{with} \quad \Delta\tilde{\boldsymbol{x}}_f|_i^k = \tilde{\boldsymbol{x}}_f^k - \tilde{\boldsymbol{x}}_f^i
\end{aligned} \tag{4.19}$$

Having defined these matrices, an approximation of the fluid interface operator $\widehat{\boldsymbol{J}_F}$ can be obtained by the methods presented above that use a forward approximation of the Jacobian (do not approximate the inverse Jacobian), i. e., the IQN-LSJ method, the IQN-MVJ method as well as the Broyden method. These can easily be modified to approximate the forward Jacobian. To this end, we need to replace the matrices $\boldsymbol{V}_k$ and $\boldsymbol{W}_k$ from above: $\boldsymbol{V}_k := \boldsymbol{V}_k^F$ and $\boldsymbol{W}_k := \boldsymbol{W}_k^F$.

The reduced order model for the solid interface operator $\widehat{\boldsymbol{J}_S}$ is established in an analogous way. Input modes $\Delta\boldsymbol{x}_f|_k^{k+1} = \boldsymbol{x}_f^{k+1} - \boldsymbol{x}_f^k$ and output modes $\Delta\tilde{\boldsymbol{x}}_d|_{k-1}^k = \tilde{\boldsymbol{x}}_d^k - \tilde{\boldsymbol{x}}_d^{k-1}$ are collected in matrices

$$\begin{aligned} \boldsymbol{W}_k^S &= (\boldsymbol{w}^S|_i^k)_{i=0}^{k-1} = \left[\Delta\boldsymbol{x}_f|_1^{k+1}, \Delta\boldsymbol{x}_f|_2^{k+1}, \cdots, \Delta\boldsymbol{x}_f|_k^{k+1}\right], \quad \text{with } \Delta\boldsymbol{x}_f|_i^{k+1} = \boldsymbol{x}_d^{k+1} - \boldsymbol{x}_d{}^i \\ \boldsymbol{V}_k^S &= (\boldsymbol{v}^S|_i^k)_{i=0}^{k-1} = \left[\Delta\tilde{\boldsymbol{x}}_d|_0^k, \Delta\tilde{\boldsymbol{x}}_d|_1^k, \cdots, \Delta\tilde{\boldsymbol{x}}_d|_{k-1}^k\right], \quad \text{with } \Delta\tilde{\boldsymbol{x}}_d|_i^k = \tilde{\boldsymbol{x}}_d^k - \tilde{\boldsymbol{x}}_d^i \end{aligned} \tag{4.20}$$

that can be used by the respective quasi-Newton method presented earlier.

Remark 4.4.1. It is also possible to build the differences of subsequent input and output pairs and collect them in the matrices $\boldsymbol{W}_k^*$ and $\boldsymbol{V}_k^*$, which is clearly more efficient, i. e.,

$$\boldsymbol{W}_k^S = \left[\Delta\boldsymbol{x}_f|_1^2, \Delta\boldsymbol{x}_f|_2^3, \cdots, \Delta\boldsymbol{x}_f|_k^{k+1}\right] \text{ and } \boldsymbol{V}_k^S = \left[\Delta\tilde{\boldsymbol{x}}_d|_0^1, \Delta\tilde{\boldsymbol{x}}_d|_1^2, \cdots, \Delta\tilde{\boldsymbol{x}}_d|_{k-1}^k\right]$$

Remark 4.4.2. For the IQN-LS method it suffices to provide a method that gives the result of the estimated Jacobian approximation multiplied with a vector, i. e., $\widehat{\boldsymbol{J}_F}^k\boldsymbol{v}$ and $\widehat{\boldsymbol{J}_S}^k\boldsymbol{v}$. That means it is not indispensable to set up an explicit representation for the Jacobians of the interface solvers.

A nicely structured overview of the block-iterative quasi-Newton coupling scheme is given by the pseudo code in Algorithm 4.

Input: initial value ${\boldsymbol{x}_d}^0 = \text{extr}(\boldsymbol{x}_n^*, \boldsymbol{x}_{n-1}^*, \boldsymbol{x}_{n-2}^*)$ time extrapolation,
initial Jacobains $\widehat{\boldsymbol{J}_F}^0$ and $\widehat{\boldsymbol{J}_S}^0$
Output: $\boldsymbol{x}^*$ for converged configuration of current time step.

Block-Iterative-QN($\boldsymbol{x}^0$)

$\boldsymbol{x}_d^0$ prediction from previous time steps
$\boldsymbol{x}_f^1 = \tilde{\boldsymbol{x}}_f^0 = \boldsymbol{F}(\boldsymbol{x}_d^0)$
$\tilde{\boldsymbol{x}}_d^0 = \boldsymbol{S}(\boldsymbol{x}_f^1)$
$\boldsymbol{x}_d^1 = \boldsymbol{x}_d^0 + \omega_0(\tilde{\boldsymbol{x}}_d^0 - \boldsymbol{x}_d^0)$ ▷ underrelaxation as PP
for $k = 1 \ldots$ **do**
  $\tilde{\boldsymbol{x}}_f^k = \boldsymbol{F}(\boldsymbol{x}_d^k)$ and $\boldsymbol{R}_F^k = \tilde{\boldsymbol{x}}_f^k - \boldsymbol{x}_f^k$
  update matrices $\boldsymbol{V}_k^F$ and $\boldsymbol{W}_k^F$ (4.19)
  update $\widehat{\boldsymbol{J}_F}^{k-1} \rightsquigarrow \widehat{\boldsymbol{J}_F}^k$ ▷ according to QN post-processing
  solve for $\Delta\boldsymbol{x}_f^k$: equation (4.17), i. e.,
    $(\widehat{\boldsymbol{J}_F}^k \widehat{\boldsymbol{J}_S}^{k-1} - I)\Delta\boldsymbol{x}_f^k = -\boldsymbol{R}_F^k - \widehat{\boldsymbol{J}_F}^k \boldsymbol{R}_S^{k-1}$
  $\boldsymbol{x}_f^{k+1} = \boldsymbol{x}_f^k + \Delta\boldsymbol{x}_f^k$ ▷ post-processing correction
  $\tilde{\boldsymbol{x}}_d^k = \boldsymbol{S}(\boldsymbol{x}_f^{k+1})$ and $\boldsymbol{R}_S^k = \tilde{\boldsymbol{x}}_d^k - \boldsymbol{x}_d^k$
  update matrices $\boldsymbol{V}_k^S$ and $\boldsymbol{W}_k^S$ (4.20)
  update $\widehat{\boldsymbol{J}_S}^{k-1} \rightsquigarrow \widehat{\boldsymbol{J}_S}^k$ ▷ according to QN post-processing
  solve for $\Delta\boldsymbol{x}_d^k$: equation (4.18), i. e.,
    $(\widehat{\boldsymbol{J}_S}^k \widehat{\boldsymbol{J}_F}^k - I)\Delta\boldsymbol{x}_d^k = -\boldsymbol{R}_S^k - \widehat{\boldsymbol{J}_S}^k \boldsymbol{R}_F^k$
  $\boldsymbol{x}_d^{k+1} = \boldsymbol{x}_d^k + \Delta\boldsymbol{x}_d^k$ ▷ post-processing correction
end

Algorithm 4: Pseudo code for the serial-implicit block-iterative coupling scheme (3.5) using a quasi-Newton method as post-processing. The algorithm shows the coupling scheme for one time step, where the for-loop iterates until convergence is achieved. Input values for the initial displacement and, depending on the quasi-Newton variant, initial values for the estimated solver Jacobians are expected. The initial displacement is usually extrapolated from previous time steps. In the first time step the initial estimations for the Jacobians are set to zero. The solution is computed in a block-iterative manner, i. e., intermediate results for forces and displacements are corrected by a post-processing scheme before they are taken as input for the complementary solver.

4.5 Summary of Coupling Schemes – All valid Combinations

Summarizing the findings from this chapter together with Chapter 3, we are given a construction kit for implicit, partitioned FSI coupling, which consists of three different coupling schemes, induced by the respective fixed-point equations (3.3), (3.4) and (3.5) as well as a variety of quasi-Newton solvers that act as a post-processing scheme to improve convergence. Although it is not possible or practical to use all of the combinations, i. e.,

$$\left\{ \begin{array}{c} \text{S-system,} \\ \text{V-system,} \\ \text{B-system} \end{array} \right\} \times \left\{ \begin{array}{ccc} \text{IQN-ILS,} & \text{IQN-LSJ,} & \text{IQN-LSHJ,} \\ \text{IQN-IMVJ,} & \text{IQN-MVJ,} & \text{Broyden} \end{array} \right\},$$

the majority of the combinations are valid and they are either of practical relevance by being competitive to other (e. g., monolithic) FSI coupling solvers, or of theoretical interest. All the valid combinations of a coupling scheme with one of the post-processing schemes are depicted in Table 4.2. While the others are merely of theoretical interest, the V-IQN-ILS and V-IQN-IMVJ are of practical interest, as they provide good convergence rates and are robust with respect to instabilities due to the added mass effect. The S-IQN-ILS, S-IQN-IMVJ and B-IQN-MVJ methods are also very robust and show small coupling iteration numbers, however, they provide a bad load-balancing for massive parallel applications which results in a poor parallel efficiency. A comprehensive evaluation and comparison of all the listed coupling schemes from Table 4.2 is conducted in Chapter 6.

	Serial System	**Vectorial System**	**Block System**
IQN-ILS	S-IQN-ILS$^{\natural}$ [17]	V-IQN-ILS$^{\maltese}$ [57]	–
	S-IQN-LS	V-IQN-LS	–
IQN-LSHJ	S-IQN-LSHJ† [33]	V-IQN-LSHJ	B-IQN-LS* [59]
IQN-IMVJ	S-IQN-IMVJ$^{\pounds}$	V-IQN-IMVJ	–
	S-IQN-MVJ	V-IQN-MVJ	B-IQN-MVJ‡ [8]
S-Broyden	S-Broyden	V-Broyden	B-Broyden

Table 4.2: Numerical coupling methods examined in this thesis: each method is defined by choosing either the serial fixed-point equation (3.3), the vectorial fixed-point equation (3.4) or the block-iterative scheme (3.5) and either the least-squares quasi-Newton solver with inverse or forward Jacobian approximation, the Broyden method or the multiple-vector quasi-Newton approach with both flavours: forward and inverse Jacobian approximation.

$^{\natural}$ The S-IQN-ILS method is equivalent to the known IQN-ILS method by Degroote et al. [17] and to the Type II update presented by Walker et al. [60], which is proven to be equivalent to the Anderson Acceleration [2].

$^{\maltese}$ The V-IQN-ILS method is the parallel version of the IQN-ILS, introduced by Uekermann et al. [57].

* The B-IQN-LS method was first introduced by Vierendeels et al. [59].

† The S-IQN-LSHJ method was first described by Haeltermann et al. [33].

$^{\pounds}$ The IQN-MVJ is equivalent to a member of the generalized Broyden family, presented by Fang et al. [23] using the Type II update.

‡ The B-IQN-MVJ method is referred to the MVQN method, presented by Bogaers et al. [8]

Chapter 5

Software and Implementation

All numerical experiments in Chapter 7 are carried out using the coupling library preCICE, together with the numerical toolbox openFOAM. In Section 5.1 we give a brief introduction to both simulation tools. A very detailed elaboration on preCICE, including all the internals of used concepts and algorithms together with the description of the application programming interface (API), is given in [31]. The reader can reference this for further information.

The remainder of this chapter discusses implementation details and used concepts for the realization of the most practical quasi-Newton coupling schemes from Chapter 3 and 4 into preCICE. This includes the definition of proper initial values as well as practical considerations for realizing the Jacobian update with the focus on a efficient but robust and stable implementation.

5.1 Coupling and Solver Software

5.1.1 preCICE: A Generic Library for Black Box Coupling

preCICE[1] (precise Code Interaction Coupling Environment) is a flexible and powerful library for the numerical coupling of single physics solvers. preCICE is developed at the Technische Universität München and the Universität Stuttgart with the main purpose to facilitate the partitioned coupling approach for surface problems. It is designed for the coupling of black box simulation software with the focus on FSI simulation, and provides ready-to-go solutions for the main problems occurring in partitioned coupling. To that end, it aims to meet two major design goals, i. e., to minimize the effort of preparing an existing solver for partitioned coupling, and secondly, to maximize the coupling flexibility that comes with the partitioned approach. Its software architecture as a library, together with a high level API that hides all coupling functionality, allows for a minimal-invasive integration into existing solver software. With this, solvers can be exchanged in a plug-and-play manner and a pool of ready-to-use

[1] http://www5.in.tum.de/wiki/index.php/PreCICE Webpage

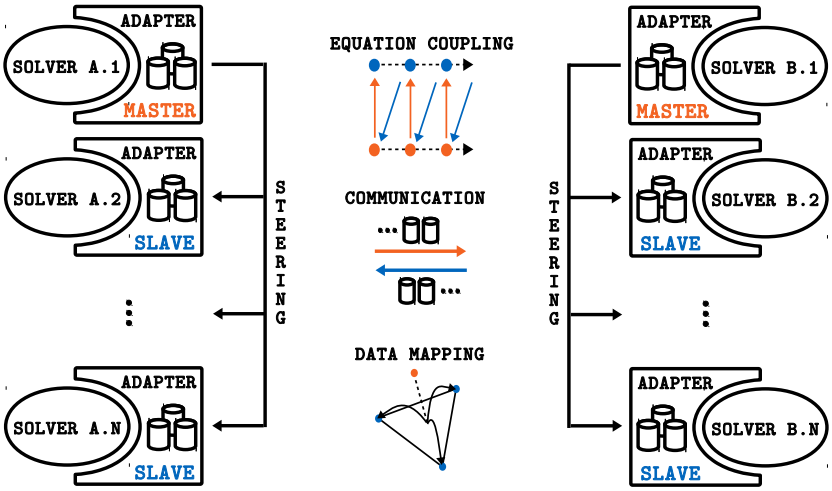


Figure 5.1: The complete picture of preCICE. The main functionalities of preCICE are depicted in the middle. Two single physics solvers are coupled. preCICE supplies solver adapters and allows for parallel or serial execution of the solvers.

solvers can be employed in any physical meaningful combination, i. e., it is possible to choose the best combination of single physics solvers for a particular problem.

A schematic representation of the main components is given in Figure 5.1. preCICE supplies solutions for three main functionalities that are essential for partitioned coupling:

1. Equation Coupling
2. Data Mapping
3. Data Communication

A geometry interface representation for fixed cartesian grid solvers with a efficient treatment of spatial queries is provided. For the exchange of coupling data between the solvers, a peer-to-peer communication concept without a central control instance is used. Various methods for data mapping between non-matching grids are available, ranging from simple projection to approaches based on radial-basis function interpolation. The equation coupling module provides a wide variety of runtime configurable aspects of numerical coupling, like serial or parallel as well as explicit and implicit coupling schemes along with several acceleration schemes, so called post-processing methods. The latter category of implicit schemes with suitable post-processing methods is of particular interest with regards to this work. preCICE comes with a XML configuration module that allows for runtime specification and steering of coupling algorithms and auxiliary tools.

preCICE is written in C++ and features a clean and modern software design, including unit and integration tests. It provides efficient coupling concepts while maintaining minimal external library dependencies. All application programming interfaces are available in C++, C and Fortran.

5.1.2 OpenFOAM

OpenFOAM[2] (Open Source Field Operation and Manipulation) is a open source numerical simulation toolbox for problems in continuum mechanics with emphasis on flow simulation. It provides a variety of single physics solvers using different discretizations such as FVM and FEM. The fluid-structure simulations within this work were conducted using single physics solvers from a software based on openFOAM, called the *foam-extend-3.1 project*[3]. The preCICE adapter as well as the actual used solver were developed by David Blom et al. from the TU Delft.

The fluid solver uses a second order finite volume discretization of the incompressible *Navier-Stokes* equations. A coupled solution algorithm as proposed in [15] is used, instead of a standard PISO (Pressure Implicit with Splitting of Operators) approach. The governing fluid equations are formulated in the *arbitrary-lagrangian-eulerian* point of view and mesh movement is done via radial-basis function interpolation as in [16]. Time integration is done using a second order backward differencing scheme.

The structural solver uses a full *Lagrangian formulation* and a Saint-Venant-Kirchhoff material model.

5.2 Implementation Details

A considerable amount in this work comprised the realization and implementation of the coupling scheme variants as presented in Chapter 3 and 4. These turned out to be valuable for partitioned FSI coupling, namely the serial IQN-IMVJ method and its parallel counterpart. The integration of a stable, robust and efficient implementation of those schemes into the partitioned black box library preCICE was intended. While several explicit schemes as well as some important implicit schemes, i. e., the dynamic Aitken's relaxation and the serial and parallel IQN-ILS scheme are already available in the preCICE library, the implementation of the multi-vector schemes has been done within this work.

In the following, we consider a few aspects of the implementation that have substantial influence on the stability, robustness and efficiency of the coupling schemes. Section 5.2.1 and 5.2.2 show some aspects that are readily realized in the preCICE implementation.

In order to facilitate the findings in this section, we simplify the notation. In the remainder, we refer $\boldsymbol{W}$ to be the matrix composing the input modes and $\boldsymbol{V}$ to be the matrix that keeps all the output modes, regardless of which method is used. Furthermore, we drop the distinction between the Jacobian and its inverse and do not differentiate between the Jacobian of the residual operator or any other. The approximation of the Jacobian is referred to as $\widehat{\mathcal{J}}$.

[2] http://www.openfoam.org/
[3] http://www.extend-project.de

5.2.1 Advance and Initialize the Coupling Schemes

In Chapter 3 and 4 we particularly focused on the solution of the fixed-point equation within one time step but suppressed how to initialize the coupling algorithm and how the coupling is advanced in between time steps. These aspects are covered in what follows.

Initialization. The initialization of a coupling scheme that uses one of the quasi-Newton methods for acceleration and stabilization comprises to provide initial values for the input-output matrices $\boldsymbol{V}$ and $\boldsymbol{W}$ as well as initial guesses for the Jacobian approximation $\widehat{\mathcal{J}}$.

As we have seen in Chapter 4 and in particular in Remark 4.3.2, the initialization of the Jacobian approximation is of great importance, especially for the Broyden and multi-vector methods. In absence of a suitable initial guess[4] for the Jacobian, the initial approximation is fixed to the zero-matrix for all methods within this work. Indeed, this choice showed reasonable results for all the inverse methods as well as for the forward methods in conjunction with the block-iterative coupling system. For the latter, the initial matrix that is involved in the solution process for the quasi-Newton update, thus is the identity matrix. For the IQN-ILS method, the initial inverse Jacobian is irrelevant and for the multi-vector method, the zero initial guess renders the method to be exactly identical to the IQN-ILS method for the first time step.

All of the quasi-Newton methods work with differences of subsequent input and output values of the fixed-point equation. Hence, data from at least two iterations is needed to provide initial data for the matrices $\boldsymbol{V}$ and $\boldsymbol{W}$. Therefore, the first iteration performs a constant underrelaxation step to provide the data, i. e.,

$$\boldsymbol{x}^1 = \boldsymbol{x}^0 + \omega_0 \boldsymbol{R}(\boldsymbol{x}^0)$$

Clearly, the underrelaxation step does not result in a drop of the residual comparable to one quasi-Newton iteration. As the problem advances in time, it is possible to provide a more sophisticated first iteration for all time steps, except the first. For now, assume that the quasi-Newton method only utilizes information from the current time step for the estimation of the Jacobian, i. e., the input-output matrices do not contain any column from previous time steps. For all time steps $t = 2, \cdots$, the first iteration performs a quasi-Newton update using the input-output matrices from the previous converged configuration. Thereafter, enough input-output data from the current time step is available and the old information is discarded.

This technique decreases the required average iterations per time step by one iteration. As the IQN-ILS method usually is applied in a mode that reuses information from several time steps in $\boldsymbol{V}$ and $\boldsymbol{W}$, the effect of the alternative first iteration is vanishingly low[5]. However, for the IQN-IMVJ method it is of great benefit, as this methods usually works best if no information from old time steps is incorporated in $\boldsymbol{V}$ and $\boldsymbol{W}$.

[4] For partitioned FSI coupling it is in general not possible to provide a initial guess for the Jacobian, as we aim to o approximate the Jacobian of entire single-physics solvers that additionally do highly depend on time.

[5] If columns from previous time steps are kept in the input-output matrices, this improved first iteration is performed implicitly.

Advance Scheme in Between Time Steps. While for all post-processing schemes it is beneficial to have a start value that is not too far from the sought fixed-point, for the quasi-Newton methods this is essential due to their inherent local convergence property. While for the first time step a good initial guess $\boldsymbol{x}_0^0$ is required, an extrapolation of coupling values from previous time steps $\boldsymbol{x}_n^*, \boldsymbol{x}_{n-1}^*, \cdots$ can be employed for all succeeding time steps, to achieve a better initial guess. Only the coupling data that is directly used as input by one of the black-box solvers is extrapolated in time, i. e., displacements for the serial and block-iterative system and both, displacements and pressure values for the parallel system. In Chapter 4 this extrapolation step was referred to as $\boldsymbol{x}_{n+1}^0 = \text{extr}(\boldsymbol{x}_n^*, \boldsymbol{x}_{n-1}^*, \boldsymbol{x}_{n-2}^*)$ and is implemented in preCICE to perform a second-order extrapolation in time, reading

$$\boldsymbol{x}_{n+1}^0 = \frac{5}{2}\boldsymbol{x}_n^* - 2\boldsymbol{x}_{n-1}^* + \frac{1}{2}\boldsymbol{x}_{n-2}^*$$

which falls back to a first-order extrapolation

$$\boldsymbol{x}_{n+1}^0 = 2\boldsymbol{x}_n^* - \boldsymbol{x}_{n-1}^*$$

for the second time step, since no two time step solutions are available by then.

5.2.2 Realizing the Newton Update and Solving the Linear System of Equations

All the rank-m quasi-Newton update schemes presented in Chapter 4 use either of the two update formulas

$$\mathcal{J} = \boldsymbol{W}\left(\boldsymbol{V}^T\boldsymbol{V}\right)^{-1}\boldsymbol{V}^T \tag{5.1}$$

$$\mathcal{J} = \mathcal{J}_{prev} + (\boldsymbol{W} - \mathcal{J}_{prev}\boldsymbol{V})\left(\boldsymbol{V}^T\boldsymbol{V}\right)^{-1}\boldsymbol{V}^T, \tag{5.2}$$

Both include the solution of a linear system of equations or the solution to a least squares problem, respectively. Here, we have $\boldsymbol{V} \in \mathbb{R}^{n\times m}$, $\boldsymbol{W} \in \mathbb{R}^{n\times m}$ and $\mathcal{J} \in \mathbb{R}^{n\times n}$ where $m \ll n$ is the number of columns in the difference matrices. If no information from previous time steps is re-used, it holds $m = k$. For those quasi-Newton methods with explicit representation of the Jacobian, that use the second update formula, we need to compute $\boldsymbol{Z} = (\boldsymbol{V}^T\boldsymbol{V})^{-1}\boldsymbol{V}^T$ with $\boldsymbol{Z} \in \mathbb{R}^{k\times n}$. Whereas for the matrix-free methods it suffices to compute the vector $\boldsymbol{\alpha} = -\boldsymbol{Z}\boldsymbol{r}^k$. The latter has been shown to be equivalent to solving the least squares problem $\boldsymbol{\alpha} = \text{argmin}_\beta\|\boldsymbol{V}\boldsymbol{\beta} + \boldsymbol{r}^k\|_2$ in Section 4.1. Either way, both methods need to solve a system of equations, whose realization is discussed in the following with regards to stability and conditioning as well as complexity of the solution.

Solving the Normal Equations. A straightforward and simple approach is to solve the least-square linear system using the normal equations, that is, solving the system directly as stated in the update formulas. It is naturally to replace $(\boldsymbol{V}^T\boldsymbol{V})^{-1}\boldsymbol{V}^T$ by the system

$$(\boldsymbol{V}^T\boldsymbol{V})\boldsymbol{z} = \boldsymbol{V}^T\boldsymbol{y} \tag{5.3}$$

for arbitrary $\boldsymbol{y}$ and corresponding $\boldsymbol{z}$. This system can be solved using LU-decomposition for the matrix $(\boldsymbol{V}^T\boldsymbol{V})$, followed by forward and backward substitution. The matrix $\boldsymbol{Z}$ is obtained by solving this system for all unit vectors $\boldsymbol{y} = \boldsymbol{e}_i$. A pseudo code is given in Alg. 5, left. Note, that for the matrix-free methods it suffices to solve the normal equations once for $\boldsymbol{y} = -\boldsymbol{r}^k$.

Usually $\boldsymbol{V}$ is close to rank-deficient and has a high condition number. Computing the product $(\boldsymbol{V}^T\boldsymbol{V})$ even doubles the condition number, which results in a bad conditioned problem when solving the normal equations and is thus impractical for a lot of applications. Using this approach, the overall complexity can be easily seen from the pseudo code to be in $\mathcal{O}(m^2n + m^3)$.

Solving the Least Squares Problem Using QR-Decomposition. An alternative solution is to solve an equivalent least-squares problem that entails a much better condition number. The resulting algorithm turns out to be numerically more stable than solving the normal equations. Solving the normal equations (5.3) is equivalent to solving the unconstraint least-squares problem

$$\mathbf{min}_{\boldsymbol{z}\in\mathbb{R}^n} \ \|\boldsymbol{V}\boldsymbol{z} - \boldsymbol{y}\|_2 \tag{5.4}$$

This formulation is convenient for storing and updating information from previous time steps (no re-computation of $\boldsymbol{V}^T\boldsymbol{V}$ required) and, in particular, it is efficient when solving successive least-squares problems over a number of iterations. Looking at eq. (5.4), it suffices to solve $\boldsymbol{V}\boldsymbol{z} = \boldsymbol{y}$, which is done using a economy-size QR-decomposition, i. e., $\boldsymbol{V} = \boldsymbol{Q}\boldsymbol{U}$ with the orthogonal matrix $\boldsymbol{Q} \in \mathbb{R}^{n\times n}$ and the upper triangular matrix $\boldsymbol{U} \in \mathbb{R}^{n\times m}$.

Then, $\boldsymbol{z}$ is computed from solving the quadratic $m \times m$ system

$$\tilde{\boldsymbol{U}}\boldsymbol{z} = \tilde{\boldsymbol{Q}}^T\boldsymbol{y}$$

via backward substitution, where $\tilde{\boldsymbol{U}} \in \mathbb{R}^{m\times m}$ denotes the first m rows of $\boldsymbol{U}$ and $\tilde{\boldsymbol{Q}}$ contains the first m rows of $\boldsymbol{Q}$. As before, we obtain the matrix $\boldsymbol{Z}$ by solving this system n-times for the unit vectors $\boldsymbol{y} = \boldsymbol{e}_i$, while for the matrix-free versions it suffices to solve this system once for $\boldsymbol{y} = -\boldsymbol{r}^k$. A pseudo code for the computation of $\boldsymbol{Z}$ is depicted in Alg. 5, right.

With this straightforward implementation, the QR-decomposition is re-computed in every iteration, requiring approximately $2m^2(n - {}^m/_3) \in \mathcal{O}(m^2n)$ flops using Householder QR-decomposition. However, the complexity can be reduced if the decomposition is updated rather than recomputed throughout the iterations. Since each $\boldsymbol{V}_k$ is obtained by its predecessor $\boldsymbol{V}_{k-1}$ by adding one column to the right and probably dropping one or more columns to the left, it suffices to compute one additional Householder transformation for the rightmost column to update the previous QR-decomposition. This can be done in $\mathcal{O}(mn)$ arithmetic operations. It is likewise possible to add and delete columns at arbitrary position for the sake

solve $\boldsymbol{V}^T\boldsymbol{V}\boldsymbol{z} = \boldsymbol{V}^T\boldsymbol{y}$ via LU-dec.

compute product $\boldsymbol{V}^T\boldsymbol{V} \in \mathbb{R}^{m\times m}$ $\in \mathcal{O}(m^2 n)$
compute $\boldsymbol{L}\boldsymbol{U} = \boldsymbol{V}^T\boldsymbol{V}$ $\in \frac{2}{3}\mathcal{O}(m^3)$
for $i = 1$ *to* n **do**
  forw. subst. $\boldsymbol{L}\boldsymbol{w} = \boldsymbol{V}^T(\cdot, i)$ $\in \frac{1}{2}\mathcal{O}(m^2)$
  backw. subst. $\boldsymbol{U}\boldsymbol{z} = \boldsymbol{w}$ $\in \frac{1}{2}\mathcal{O}(m^2)$
  $\boldsymbol{Z}(\cdot, i) = \boldsymbol{z}$
end

solve $\boldsymbol{V}\boldsymbol{z} = \boldsymbol{y}$ via QR-dec.

compute $\boldsymbol{Q}\boldsymbol{U} = \boldsymbol{V}$ $\in \frac{4}{3}\mathcal{O}(m^2 n)$
for $i = 1$ *to* n **do**
  backw. subst. $\tilde{\boldsymbol{U}}\boldsymbol{z} = \boldsymbol{Q}^T(\cdot, i)$ $\in \frac{1}{2}\mathcal{O}(m^2)$
  $\boldsymbol{Z}(\cdot, i) = \boldsymbol{z}$
end

Algorithm 5: Solving the least squares linear system $\boldsymbol{Z} = (\boldsymbol{V}^T\boldsymbol{V})^{-1}\boldsymbol{V}^T$ **left:** via solving the normal equations using LU-decomposition of $\boldsymbol{V}^T\boldsymbol{V}$, and **right:** via solving a equivalent least squares problem and QR-decomposition of $\boldsymbol{V}$. The latter approach is favourable because of better conditioning and good stability properties.

of stability and well-conditionedness using suitable Givens rotations within $\mathcal{O}(mn)$ flops. A very detailed elaboration on updated QR-decomposition, including an exact comparison of complexity with respect to required arithmetic operations and implementation details, is given in [36]. In evaluating a large set of test cases, Hammerling et al. showed that the updated QR-decomposition appears to be way more efficient than recomputing the entire factorization if changed.

In preCICE, the update for all quasi-Newton methods is done using a updated QR-decomposition of the matrix $\boldsymbol{V}_k$ via a modified Gram-Schmidt algorithm and a set of suitable Givens rotations [14]. The respective system of equations is solved in every iteration. In literature it is also proposed to use singular-value decomposition or rank-revealing QR-decomposition with full pivoting [23]. Although those methods perform better in case of rank-deficiency, they are more costly and, in particular, perhaps less necessary if additional effort is made to maintain a acceptable conditioning of the system, as described in the following.

Maintaining Acceptable Conditioning. As we store differences of subsequent input-output data, $\boldsymbol{V}$ is often close to rank-deficient and the condition of the problem is critical. The condition number highly depends on the number of columns m in $\boldsymbol{V}_k$ as the risk of linear dependence increases with growing m. Furthermore, a large m entails the risk of having outdated or contradicting information, especially if a large number of time steps are reused. On the contrary, restricting m to be small bears the possibility that the secant information is too limited to provide sufficient convergence of the method. To solve this dilemma, the preCICE implementation monitors the condition number of the least-squares coefficient matrix, which is just the condition number of the matrix $\tilde{\boldsymbol{U}}$, and keeps it at tolerable size by dropping critical columns. To be specific, if the diagonal element $\boldsymbol{U}_{ii}$ falls below a prescribed tolerance, the corresponding column is removed from $\boldsymbol{V}_k$ and the QR-decomposition is updated.

Scaling of Coupling Data. In practice, the input-output data of forces and displacements often differs by orders of magnitude. Usually the changes of the displacements are much smaller than the changes in the forces exerted by the fluid. Once again, this causes bad conditioning for the least squares system that needs to be solved. To overcome this stability problem, the coupling data can be scaled previous to the post-processing process, so that both, forces and displacements are in the same order of magnitude. So far, the optimal ratio $r_f^d = \|\boldsymbol{x}_d\|_2 / \|\boldsymbol{x}_f\|_2$ for the different methods is obscure and an automatic choice of the respective scaling factor is not possible. Thus, to obtain optimal performance of the coupling scheme, a parameter study, searching for the best scaling factor, has to be performed in a post-processing stage. Although, it might not be optimal a ratio of $r_f^d \approx 1$ always guarantees proper working of the considered quasi-Newton methods. Note, that this is only necessary for the parallel-implicit coupling scheme, as the serial-implicit coupling scheme solely uses displacement input-output data to approximate the Jacobian of the System.

Chapter 6

Comparison and Evaluation of Implicit Coupling Schemes

In this chapter, the previously introduced implicit quasi-Newton coupling schemes are evaluated and compared for a one-dimensional model for the internal flow through a flexible tube, based on [18]. To allow for a comprehensive and feasible comparison of different coupling schemes and solver variants, the fluid and structure solver as well as the coupling algorithm has been implemented in MatLab, supporting fast prototyping. The description of the scenario, including the physical model and discretization, is given in Section 6.1.

In Chapter 4 a whole range of coupling schemes and quasi-Newton post-processing methods was presented. In order to figure out the most promising variants for real world scenarios and benchmark examples, a qualitative comparison of all the coupling schemes from Sect. 4.5 is elaborated in Section 6.2. The preCICE implementation for newly developed coupling schemes is validated in Section 6.3 for the one-dimensional test scenario.

6.1 One-Dimensional Flexible Tube: Scenario Description

This scenario considers a flexible tube with internal inviscid and incompressible flow, where gravity impacts are neglected. It is taken from [18] and the description of the physical model is based thereon and on [31].

Physical Model. Due to the axis-symmetry we obtain a one-dimensional model by averaging over the tube in radial direction as shown in Figure 6.1. The simplified equations for conservation of mass and momentum read

$$\frac{\partial(av)}{\partial t} + \frac{\partial(au^2)}{\partial x} + a\frac{\partial p}{\partial x} = 0 \tag{6.1}$$

$$\frac{\partial a}{\partial t} + \frac{\partial(au)}{\partial x} = 0, \tag{6.2}$$

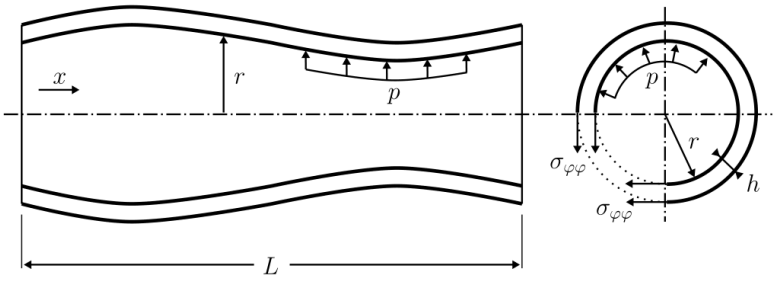


Figure 6.1: Schematic representation of the deformed flexible tube with internal flow in x-direction, radius r, wall thickness h and total length L. The fluid caused pressure acting on the inner tube walls in radial direction causes circumferential stresses $\sigma_{\varphi\varphi}(x)$ that result in a deformation in radial direction. This illustration is taken from [31, p. 134].

where $v(x)$ is the flow velocity in x-direction, $a(x) = \pi r(x)^2$ denotes the cross sectional area of the tube and $p(x)$ is the fluid pressure in radial direction. A sinus-shaped time-varying inlet velocity and non-reflecting outlet boundary conditions are imposed.

$$v_{in} = v_0 - \frac{v_0}{100} \sin^2(\pi \frac{t}{T})$$

The elastic tube wall is described by a linear elastic constitutive law with the scalar circumferential stress

$$\sigma_{\varphi\varphi} = E \frac{r - r_0}{r_0} \sigma_0, \tag{6.3}$$

where E is the Young's modulus and r_0 and σ_0 denote the reference position of the wall and its initial circumferential stress, respectively. The tube wall is assumed to contain no mass, since the inertia of the latter is neglected. The motion of the wall is restricted in radial direction. The dynamic FSI interface conditions are given by

$$\rho p r = \sigma_{\varphi\varphi} h \tag{6.4}$$

with the thickness h of the tube wall. Substituting the constitutive law (6.3) in (6.4) and introducing the Moens-Korteweg wave speed $c_{mk} = \sqrt{Eh/2\rho r_0}$, the cross sectional area finally can be rewritten as an explicit function of the pressure

$$a(p) = a_0 \left(\frac{p_0 - 2c_{mk}^2}{p - 2c_{mk}^2} \right)^2 \tag{6.5}$$

with the reference values r_0 and a_0. For this scenario, the general coupling variables $\boldsymbol{x}_f$ for the forces and $\boldsymbol{x}_d$ for the displacements can be substituted by the pressure p and the cross sectional area (csa) a. For further details of the model, refer to [18].

Discretization. To discretize the physical model, the spatial domain $[0, L]$ is subdivided into N_S equidistant cells of size $\Delta x = L/N_S$. The pressure and velocity unknowns v_i and p_i as

well as the cross sectional area values a_i are located at the cell centers, i. e., the same mesh is used for fluid and structural solver. Finite volumes using central discretization schemes are applied to discretize all terms in the fluid and structural equations, except for the convective flow terms, where a stabilizing first-order upwind scheme is used. To account for the absence of a diffusive term, an additional pressure stabilization term is added to the continuity equation (6.2). At the inflow and outflow boundary $x = 0$ and $x = L$, respectively, the values from the cell centers are linearly extrapolated to the boundary.

The time interval $[0, T]$ is analogously subdivided into N_T time steps of the same length Δt. The time derivatives for both, the structure and fluid equations are discretized by means of a first-order backward Euler scheme.

In order to control the hardness of the problem and to study the behaviour of different coupling schemes, two characteristic quantities are defined, referring to [18], namely the dimensionless stiffness parameter κ and the dimensionless time step τ, reading

$$\kappa := \frac{1}{U_0} = \frac{\sqrt{\frac{Eh}{2\rho r_0} - \frac{p_0}{2\rho}}}{v_0}, \qquad \tau = \frac{U_0}{D_0 N_T} = \frac{v_0 \Delta t}{L}$$

Here, the initial velocity v_0, the initial dimensionless velocity U_0, the initial pressure p_0 and the dimensionless spatial discretization width D_0 are used. Stability analysis in [18] and [12] showed, that obtaining stability for the FSI coupling is more challenging for decreasing structural stiffness κ, decreasing time step size τ and increasing spatial resolution. The spatial resolution is fixed to $N_S = 100$ for all the conducted experiments. A full period of the time dependent sine-shaped inlet velocity is simulated.

6.2 Numerical Results and Comparison of Coupling Schemes

Because of the simplicity of the scenario, it is possible to analyze a whole range of methods and parameters with relatively low implementation effort. Supporting this, both single-physics solver as well as the coupling algorithm were implemented in simple MatLab code. This section aims for a qualitative comparison of the coupling schemes introduced in Section 4.5, i. e., all valid combinations of coupling systems $\{S, V, B\}$ and post-processing schemes $\{\text{IQN-ILS}, \text{IQN-LSJ}, \text{IQN-LSHJ}, \text{IQN-IMVJ}, \text{IQN-MVJ}, \text{Broyden}\}$.

Although the here considered one-dimensional example is rather easy to implement and analyze, it shows some inherent bad-conditioning which is particularly intensified by the assumption of having a tube wall with no mass. Further, the pressure and cross sectional area values differ by orders of magnitude. Consequently, all the numerical experiments carried out in this chapter aim to investigate and compare the rough behaviour of the coupling schemes in order to spot the most promising candidates for further experiments. Here, we are interested in the qualitative properties of the coupling algorithms to be able to formulate tendencies and do not focus on quantitative results.

We obtain the same physical values up to an absolute l_2-precision of 10^{-9} for all investigated coupling variants, provided that convergence is achieved. In the very first iteration, a constant underrelaxation step is performed, while using the approximation from the last time step for the first iteration of all the subsequent ones. A second-order time extrapolation in between time steps is computed for the coupling data. The fluid solver solves up to a l_2-precision of 10^{-15} and the coupling algorithm proceeds until the l_2-norm of the residual drops below a relative convergence criterion of 10^{-7} for the cross sectional area $a = \boldsymbol{x}_d$ and the pressure $p = \boldsymbol{x}_f$, respectively, i. e.,

$$\|\hat{a}_k - \hat{a}_{k-1}\|_2 \leq 10^{-7}\|\hat{a}_k\|_2 \quad \text{and} \quad \|\hat{p}_k - \hat{p}_{k-1}\|_2 \leq 10^{-7}\|\hat{p}_k\|_2.$$

The offsets a_0 and p_0 are subtracted before the measurement of the residual, i. e., $\hat{a} = a - a_0$ and $\hat{p} = p - p_0$.

The update of the Jacobian matrix or its inverse is done naively, solving the normal equations using built-in MatLab functions. No effort is made to stabilize the system or maintain a good conditioning of the matrices beyond the internal MatLab implementation.

In Section 6.2.1, we investigate the behaviour of the different coupling systems, i. e., $\{S, V, B\}$, followed by a comparison of the different quasi-Newton methods in Section 6.2.2. In this connection a few prototypes of additional variants are considered.

6.2.1 Comparison of Coupling Systems

The serial-implicit coupling system as described in Section 3.2.1 represents the standard approach to partitioned FSI coupling. However, it offers some drawbacks with regards to parallel efficiency due to bad load balancing. The parallel-implicit coupling system is designed to overcome these drawbacks. A priori the latter introduces the double amount of work, though, allowing for a fully parallel execution of the field solvers. This can be seen from solving both systems with a plain fixed-point iteration, as examined in [57], which requires twice as many minus one iterations for the V-system than for the S-system to converge. The authors show, that only quasi-Newton methods (namely the IQN-ILS method) turn out to be powerful enough to make one iteration of the V-system comparable to one iteration of the S-system. This is mainly due to the fact, that additional input-output information from the present and previous time steps for the force or pressure data can be incorporated into the approximation process.

Likewise, the B-system ought to be an advancement to the serial-implicit system, however coming from a different perspective. The block-iterative execution allows for a better modularity as the Jacobians for the respective field solvers need to be approximated directly. This can be beneficial if any information of solver internals or initial Jacobian guesses is available, in contrast to our assumptions. However, the execution order of the field solvers is restricted to be serial.

Table 6.1 shows the mean iteration numbers over the first hundred time steps for the different coupling schemes, applied to the one-dimensional flexible tube scenario. The dimensionless

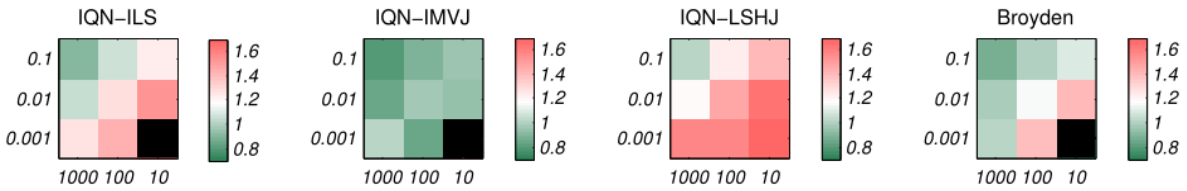


Figure 6.2: Comparison of mean iteration numbers for the S-system and the V-system and different values for τ (y-axis) and κ (x-axis). The factor of how much more or less iterations the V-system required with respect to the mean iterations for the S-system is visualized for each parameter setting, i. e., the factor $f_S^V = {}^{\text{mean iterations V-system}}/_{\text{mean iterations S-system}}$. The lower right corner is undefined for all cases, except for the IQN-LSHJ method.

stiffness parameter κ and the time step size τ control the amount of present oscillations and instabilities and are varied, i. e., $\kappa \in \{1000, 100, 10\}$ and $\tau \in \{0.1, 0.01, 0.001\}$. The hardness of the coupling scenario increases from the top left corner to lower right corner of the iteration tables. Firstly, the findings from [57] can be confirmed when comparing the iteration numbers for the serial- and parallel-implicit coupling system. For all the considered quasi-Newton post-processing schemes, the average number of iterations per time step for the V-system is comparable to the S-system. To be specific, we have a deviation of plus/minus one iteration for the easier cases, and we require a multiple of 1.3 of the mean iterations of the S-system in order for the V-system to converge for the harder cases, averaged over all methods. This factor

$$f_S^V = \frac{\text{mean iterations for V-system}}{\text{mean iterations for S-system}}$$

is visualized in Figure 6.2 for the different quasi-Newton methods. The V-system yields especially good results for the IQN-IMVJ method where it is always superior to the S-system, but also shows good performance for the IQN-ILS and Broyden method. As opposed to the V-system, the S-system appears to have difficulties with the most challenging parameter setting $(\tau, \kappa) = (0.001, 10)$, where the coupling fails to converge for most methods. This however, is not in the least due to the bad conditioning of the scenario.

Comparing the block-iterative coupling system with the S-system, we see from Table 6.1 that there is no remarkable difference between the two coupling variants. To be more accurate, the B-system shows slightly worse average iteration numbers than the S-system. A possible explanation for this is that the B-system requires the approximation of the forward Jacobians, which turned out to be less stable than the approximation of the inverse Jacobian of one of the operators (see Sect. 6.2.2). Similarly, the B-system offers no identifiable advantage over the V-system, while having several drawbacks. While providing no better convergence properties, the computational effort of the B-system is comparable to that of the V-system, but is not suitable for a massive parallel computation. Furthermore, it involves the solution step of a linear system of equations in each iteration, as it approximates for the solvers' Jacobians.

Summarizing the above, we can conclude, that the B-system offers several drawbacks without having a better convergence behavior. The performance of the S-system and the V-system is quite comparable. Together with its better parallel efficiency, this yields an advantage of the V-system over the S-system.

	S-System			V-System			B-System		
IQN-ILS	**S-IQN-ILS**			**V-IQN-ILS**					
	$\tau \setminus \kappa$	1000	100	10	1000	100	10		
	0.1	2.98	3.06	4.15	0.1	2.70	3.25	5.14	
	0.01	3.03	3.41	7.26	0.01	3.19	4.39	11.14	
	0.001	3.45	6.96	–	0.001	4.39	10.09	35.75	✠
IQN-LSJ	**S-IQN-LSJ**†			**V-IQN-LSJ**†					
	$\tau \setminus \kappa$	1000	100	10	$\tau \setminus \kappa$	1000	100	10	
	0.1	2.98	3.07	4.24	0.1	2.69	3.65	–	
	0.01	3.02	3.16	–	0.01	3.15	–	–	
	0.001	3.39	–	–	0.001	11.65	–	–	✠
IQN-LSHJ	**S-IQN-LSHJ**			**V-IQN-LSHJ**				**B-IQN-LSJ**		
	$\tau \setminus \kappa$	1000	100	10	$\tau \setminus \kappa$	1000	100	10	$\tau \setminus \kappa$	1000	100	10
	0.1	2.98	3.08	4.07	0.1	3.04	3.84	5.77	0.1	3.02	3.31	5.29
	0.01	3.03	3.49	7.64	0.01	3.61	5.15	12.57	0.01	3.07	4.80	8.39
	0.001	3.25	7.21	20.11	0.001	5.16	11.41	35.25	0.001	4.36	8.19	–
IQN-IMVJ	**S-IQN-IMVJ**			**V-IQN-IMVJ**						
	$\tau \setminus \kappa$	1000	100	10	$\tau \setminus \kappa$	1000	100	10		
	0.1	2.98	3.10	3.40	0.1	2.39	2.78	3.27		
	0.01	3.01	3.19	4.34	0.01	2.57	3.13	4.08		
	0.001	3.07	4.45	–	0.001	3.14	3.78	8.04	✠	
IQN-MVJ	**S-IQN-MVJ**†			**V-IQN-MVJ**†				**B-IQN-MVJ**		
	$\tau \setminus \kappa$	1000	100	10	$\tau \setminus \kappa$	1000	100	10	$\tau \setminus \kappa$	1000	100	10
	0.1	2.98	3.06	4.23	0.1	3.11	3.29	5.25	0.1	3.01	3.04	3.32
	0.01	3.01	3.33	–	0.01	3.45	4.46	–	0.01	2.06	3.09	5.01
	0.001	3.38	–	–	0.001	4.47	8.65	–	0.001	3.05	4.24	–
Broyden	**S-Broyden**			**V-Broyden**				**B-Broyden**		
	$\tau \setminus \kappa$	1000	100	10	$\tau \setminus \kappa$	1000	100	10	$\tau \setminus \kappa$	1000	100	10
	0.1	2.98	3.12	4.63	0.1	2.61	3.13	5.06	0.1	3.02	3.28	6.07
	0.01	3.08	3.19	6.18	0.01	3.05	3.70	8.74	0.01	3.18	4.50	–
	0.001	3.44	5.80	–	0.001	3.49	8.07	–	0.001	5.05	–	–

Table 6.1: One-dimensional elastic tube. All valid coupling schemes, as summarized in Sect. 4.5, are evaluated for the one-dimensional flexile tube scenario. The average number of iterations over the first 100 time steps is depicted for different parameter combinations of the dimensionless stiffness parameter κ and the dimensionless time step size τ. The occurring instabilities become more intense for decreasing structural stiffness κ and decreasing time step size τ. ✠ Left blank, because the block-iterative scheme is only valid for a approximation of the Jacobian of $\boldsymbol{F}$ and $\boldsymbol{S}$. † IQN-MVJ and IQN-LSJ methods that approx. the Jacobian matrix do only work properly with an initial guess $1 \cdot 10^{10} I$, which is kind of complementary to the zero-matrix for the inverse approximation. Though, they are highly sensitive to the choice of the scaling factor for the identity.

6.2.2 Comparison of Quasi-Newton Post-Processing Methods.

IQN-IMVJ Method. This generalized Broyden method, approximating the inverse Jacobian of the residual operator, is newly introduced within this thesis and up to now only tested for the B-system in Bogaers et al. [8]. It yields the overall best results for the considered one-dimensional flexible tube scenario, as can be seen from Table 6.1. The mean iteration numbers for the S-system as well as for the V-system dominate the results for the remaining methods. Here, the V-system even appears to outperform the S-system and the method provides a good overall robustness. Even the case $(\tau, \kappa) = (0.001, 10)$, where severe instabilities are present, converges fast and reliable. The good convergence behaviour for this method can be explained considering its properties and features as explained in Section 4.3. Information from previous time steps is implicitly incorporated into the approximation process in a minimum norm sense which lessens the risk of linear dependence and contradicting information in the secant equation. Additionally reusing columns from old time steps in the matrices $\boldsymbol{V}$ and $\boldsymbol{W}$ showed further improvements for this scenario, however, this behaviour could not be confirmed for more realistic applications, see Chapter 7.

Consequently, this method yields excellent results without explicitly re-using past information in the secant equation, and thus renders the tuning of the highly problem dependent parameter R of re-used time steps unnecessary. This is beneficial for two reasons: Firstly, it saves us a very costly pre-processing step. Secondly, the number of columns in the secant equation is usually significantly smaller. Yet, this comes at the cost of having to store an explicit representation of the Jacobian approximation matrix.

IQN-ILS Method. The well-known IQN-ILS method shows comparable results to previous investigations [57] for the serial and the parallel coupling system. The iteration numbers in Table 6.1 were obtained without re-using any information from previous time steps in the difference matrices $\boldsymbol{V}$ and $\boldsymbol{W}$. In doing so, the convergence behaviour of this method can be greatly improved, which allows for a fair comparison to the IQN-IMVJ method. In Table 6.2 the mean iteration numbers are given for the S-system and the V-system, re-using the information from 4, 8, 16 and 32 previous time steps in the approximation process. For both coupling systems we observe a significantly improved convergence behaviour and robustness, especially for the harder cases in the lower right corner. Unlike the serial coupling system, the V-system further improves for the re-use of 8 or 16 previous time steps, which suggests that it is beneficial to re-use more previous information for the V-system than for the S-system. Nonetheless, the performance cannot be arbitrarily improved, as the risk of linear dependencies and contradicting information within the secant equation increases if more and more outdated information is retained. Although this scenario is exceptional well suited for retaining passed information, the effect is visible from Table 6.2 in that the S-system presents worsened mean iteration numbers for a large number R and the V-system is at least not further improved reusing 16 or 32 passed time steps. Thus, in general, the optimal number of reused time steps R is highly problem dependent and needs to be tuned for every application in a costly pre-processing stage.

S-IQN-ILS(4)				**S-IQN-ILS(8)**				**S-IQN-ILS(16)**				**S-IQN-ILS(32)**			
$\tau \setminus \kappa$	1000	100	10	$\tau \setminus \kappa$	1000	100	10	$\tau \setminus \kappa$	1000	100	10	$\tau \setminus \kappa$	1000	100	10
0.1	2.98	3.04	3.18	0.1	2.98	3.04	3.54	0.1	2.98	3.04	9.08	0.1	2.98	3.17	7.47
0.01	3.02	3.08	3.61	0.01	3.02	3.11	3.62	0.01	3.02	3.23	4.27	0.01	3.02	4.23	4.64
0.001	3.08	3.27	–	0.001	3.08	3.32	–	0.001	3.08	3.54	–	0.001	3.08	3.83	–

V-IQN-ILS(4)				**V-IQN-ILS(8)**				**V-IQN-ILS(16)**				**V-IQN-ILS(32)**			
$\tau \setminus \kappa$	1000	100	10	$\tau \setminus \kappa$	1000	100	10	$\tau \setminus \kappa$	1000	100	10	$\tau \setminus \kappa$	1000	100	10
0.1	2.07	2.13	2.34	0.1	2.08	2.14	2.51	0.1	2.09	2.34	2.50	0.1	2.28	2.89	2.93
0.01	2.10	2.21	3.73	0.01	2.11	2.19	3.14	0.01	2.13	2.25	2.76	0.01	2.86	2.38	2.92
0.001	2.20	3.28	10.72	0.001	2.16	2.74	8.77	0.001	2.17	2.44	8.01	0.001	2.25	2.46	7.98

Table 6.2: One-dimensional elastic tube. Average iteration numbers over the first 100 time steps for the IQN-ILS method, re-using information from previous time steps in the approximation process. For the serial-implicit and the parallel-implicit coupling system, the mean iteration numbers are given for 4, 8, 16 and 32 retained passed time steps (indicated in brackets).

It has to be mentioned, that for this scenario the IQN-ILS(R) variant yields better results than the IQN-IMVJ method. This is partly due to the fact that the benefit from re-using previous time steps is enhanced for this example, which also explains an improved performance for the IQN-IMVJ if information is re-used explicitly in $\boldsymbol{V}$ and $\boldsymbol{W}$. Although the IQN-ILS(R) shows quite good results, one has to keep in mind, that this optimum with respect to the parameter R is found in a costly try and error process.

IQN-LSHJ Method. The IQN-LSHJ method shows no better convergence behaviour than the IQN-ILS and the IQN-IMVJ method, while carrying some severe drawbacks. Firstly, it approximates the Jacobian matrix of the fixed-point operator, which means that a linear system of equations needs to be solved in each quasi-Newton iteration and the approximate Jacobian matrix has to be computed explicitly. As a consequence, this method is considered inferior and is not investigated further.

IQN-MVJ and IQN-LSJ Method. Both methods correspond to the forward approximation variants to their inverse counterparts IQN-ILS and IQN-IMVJ. As opposed to the latter, the forward approximations appear to be very unstable and offer bad conditioning. The main problem here is to find a suitable initial guess for the Jacobian approximation. The zero-matrix initial guess is impractical, as a rank-one update for the very first iteration does not lead to a regular matrix for the Jacobian approximation. Thus, it is not possible to solve the respective system.
A natural solution is to choose a scaled identity matrix $f_{init} \cdot I$ as initial guess, which is kind of complementary to the initial zero-guess for the inverse versions. This choice appears to work for the multi-vector variant, but fails to stabilize the harder cases properly as can be seen from Table 6.1. For both methods, convergence was achieved only for the easy cases for which a pure fixed-point iteration yields comparable results. Furthermore, the stability and conditioning is highly sensitive to the scaling parameter f_{init}.

The above, in conjunction with the other drawbacks that come with the approximation of the Jacobian instead of the inverse Jacobian, render these methods useless and impractical for realistic applications in partitioned FSI coupling.

Broyden Method and Generalized Broyden Methods. The results for the Broyden method are of particular interest in comparison with those from the IQN-IMVJ method, as it simply minimizes the change in the approximate inverse Jacobian within one iteration, finally doing a rank-one update. It will now be interesting to see if the additional computational effort and storage requirement for the IQN-IMVJ rank-m update in terms of convergence speed and stabilizing behaviour pays off. At first, we see from Table 6.1 that the Broyden method works quite good for the easier cases and, in particular, performs far better than dynamic Aitken's underrelaxation (compare [57]). The latter can be seen as the rank-one update version of the IQN-ILS method and has comparable complexity to the Broyden method. Furthermore, it yields even better results than the IQN-ILS method, if the latter is applied without retaining past information. This shows that the implicit incorporation of old information works good, even if the minimization ranges over no more than one iteration. However, it does not measure up with the generalized rank-m variant which is clearly superior, especially for the harder cases.

Considering the complexity of both methods, we see from eq. (6.6) that the Broyden rank-one update has an overall complexity of $\mathcal{O}(2n^2 + n)$ in terms of required multiplications.

$$\mathcal{J}_k = \mathcal{J}_{k-1} + \overbrace{\left(\boldsymbol{w}_{k-1}^k - \underbrace{\mathcal{J}_{k-1}\boldsymbol{v}_{k-1}^k}_{\mathcal{O}(n^2)}\right)\left[\boldsymbol{v}_{k-1}^k\right]^T}^{\mathcal{O}(n^2)} / \underbrace{\|\boldsymbol{v}_{k-1}^k\|_{L^2}^2}_{\mathcal{O}(n)} \tag{6.6}$$

As opposed to this, the IQN-IMVJ rank-m update involves matrices of size $m \times n$ and the solution to a linear system of equations.

$$\mathcal{J}_k = \mathcal{J}_{prev} + \overbrace{\left(\boldsymbol{W}_k - \underbrace{\mathcal{J}_{prev}\boldsymbol{V}_k}_{\mathcal{O}(n^2m)}\right)\underbrace{\left(\boldsymbol{V}_k^T\boldsymbol{V}_k\right)^{-1}\boldsymbol{V}_k^T}_{\mathcal{O}(m^2n+mn)}}^{\mathcal{O}(n^2m)} \tag{6.7}$$

The latter can be efficiently computed using an economy-size QR-decomposition which is updated in each iteration, requiring $\mathcal{O}(mn)$ multiplications (compare Section 5.2). This results in an overall complexity of $\mathcal{O}(2n^2m + m^2n + mn)$ in terms of required multiplications, which can be seen from eq. (6.7). For a less efficient implementation of the QR-decomposition the complexity increases to $\mathcal{O}(3n^2m)$. Consequently, as m is usually rather small, the updated QR-decomposition allows for efficient implementation of the generalized rank-m update, which clearly pays off due to better convergence. In case of having a more costly QR-decomposition implementation, it could be worth considering the Broyden method, if moderate instabilities are present.

The generalized Broyden methods a priori allow for an arbitrary choice of m, i. e., partitioning of input-output data into groups to perform the rank-m update. The very natural choice for partitioned FSI coupling of $m = k$, the number of iterations performed in the current time

V-Broyden				**V-IQN-IMVJ**				**V-IQN-IMVJ$_{-2}$**				**V-IQN-IMVJ$_{-4}$**			
$\tau \setminus \kappa$	1000	100	10	$\tau \setminus \kappa$	1000	100	10	$\tau \setminus \kappa$	1000	100	10	$\tau \setminus \kappa$	1000	100	10
0.1	2.61	3.13	5.06	0.1	2.39	2.78	3.27	0.1	2.53	2.91	3.39	0.1	2.59	3.07	3.59
0.01	3.05	3.70	8.74	0.01	2.57	3.13	4.08	0.01	2.91	3.30	5.25	0.01	3.07	3.47	6.00
0.001	3.49	8.07	–	0.001	3.14	3.78	8.04	0.001	3.25	4.37	12.06	0.001	3.38	5.13	15.04
$\min \left\| \widehat{\boldsymbol{J}}_{\tilde{R}k}^{-1} - \widehat{\boldsymbol{J}}_{\tilde{R}k-1}^{-1} \right\|_F$				$\min \left\| \widehat{\boldsymbol{J}}_{\tilde{R}n}^{-1} - \widehat{\boldsymbol{J}}_{\tilde{R}n-1}^{-1} \right\|_F$				$\min \left\| \widehat{\boldsymbol{J}}_{\tilde{R}n}^{-1} - \widehat{\boldsymbol{J}}_{\tilde{R}n-2}^{-1} \right\|_F$				$\min \left\| \widehat{\boldsymbol{J}}_{\tilde{R}n}^{-1} - \widehat{\boldsymbol{J}}_{\tilde{R}n-4}^{-1} \right\|_F$			

Table 6.3: One-dimensional elastic tube. Comparison of mean iteration numbers for different minimization distances in the context of generalized Broyden methods. Left: The already introduced Broyden and IQN-IMVJ method, that minimize the difference in the Jacobian approximation within one iteration and within one complete time step, respectively. Right: Two modifications for the IQN-IMVJ method, that minimize the change in the Jacobian approximation within two, resp. four entire time steps.

step, yields the IQN-IMVJ method. However, it is equally conceivable to minimize over the change in the Jacobian approximation over more than one time step and setting m accordingly. Table 6.3 shows a test series that considers minimization over more than one time step for the one-dimensional flexible tube scenario. Mean iteration numbers are given for two modified multi-vector methods that minimize the difference of the Jacobian approximation over two and four time steps, respectively. Comparing the resulting mean Iteration numbers with the reference case, namely the IQN-IMVJ method, we see that for this scenario no benefit is gained from minimizing over more than one time step. The convergence behaviour gets even worse when minimized over a larger distance. Thus, these alternatives are not promising, while having higher computational costs and memory requirements.

On the other side, it is equally possible to choose m small but constant, that means the used information is limited to a small number $L \in \mathbb{N}$, e. g., $L \in \{1, 2, 3, 4\}$, of most recent input-output modes. Here, two different forms are conceivable. It is either possible to limit the number of utilized input-output modes to a small constant number L, but persist with the minimizing within one entire time step. This method is referred to as IQN-IMVJ-truncated(L) in the following. Or, an update can be performed each time the number of columns in $\boldsymbol{V}$ and $\boldsymbol{W}$ exceeds the limit. This alternative, denoted as IQN-IMVJ-limited(L), constitutes the counterpart to the earlier introduced IQN-IMVJ$_{-R}$ methods, performing a true rank-L update in the sense of generalized Broyden methods. Note that the IQN-IMVJ-limited(1) method coincides with the rank-one Broyden method. Both alternatives provide a Jacobian update that is computationally cheap and independent on the number of required coupling iterations. In particular, the complexity of the update appears to be a constant multiple of the Broyden's update complexity. Table 6.4 shows mean iteration numbers for both limited-information alternatives, applied with the vectorial coupling scheme for different parameter settings for the one-dimensional flexible tube scenario. While the results for the IQN-IMVJ-truncated(L) methods deteriorate sharply compared to the reference case IQN-IMVJ, the IQN-IMVJ-limited(L) methods show worsened but satisfying results. Thus, due to their rather low computational complexity these limited-information alternatives seem to be promising in cases, where moderate to little instabilities are present.

IQN-IMVJ-truncated(2)				**IQN-IMVJ-truncated(3)**				**IQN-IMVJ-limited(2)**				**IQN-IMVJ-limited(3)**			
$\tau \setminus \kappa$	1000	100	10	$\tau \setminus \kappa$	1000	100	10	$\tau \setminus \kappa$	1000	100	10	$\tau \setminus \kappa$	1000	100	10
0.1	4.06	4.25	7.05	0.1	2.59	3.13	4.00	0.1	2.59	3.07	3.58	0.1	2.52	2.82	3.40
0.01	4.10	4.98	–	0.01	3.07	3.42	–	0.01	2.93	3.28	7.45	0.01	2.88	3.24	5.04
0.001	4.82	–	–	0.001	3.35	9.60	–	0.001	3.25	4.74	–	0.001	3.24	4.24	–

Table 6.4: One-dimensional elastic tube. Mean iteration numbers for different limited-information V-IQN-IMVJ variants that restrict the utilized input-output modes to a small, constant number. Left: IQN-IMVJ-truncated(L) methods. Here the number of columns in difference matrices is restricted to L, but the minimization spans over the entire time step. Right: IQN-IMVJ-limited(L) methods. True rank-L updates where the minimization distance is aligned to the constant number L of utilized input-output modes.

Nonetheless, the IQN-IMVJ method appears to be the optimal choice, which is a very nice result for two reasons: Firstly, m usually is limited to be rather small and secondly, (for the IQN-IMVJ method) we do not have a problem dependent control parameter, neither for the number of re-used time steps in $\boldsymbol{V}$ and $\boldsymbol{W}$, nor for the distance that is optimal to minimize about, that needs to be tuned beforehand.

Summarizing this section, it is worth considering the serial-implicit and the parallel-implicit coupling system for further more sophisticated applications and benchmarks. The V-systems offers a better parallel efficiency while the S-system often results in slightly better convergence behaviour. The block-iterative coupling system showed no advantages over the other coupling variants, but suffers from the same drawbacks with regards to parallel efficiency as the S-system. Investigating the quasi-Newton post-processing methods from Chapter 4, all methods that approximate for the (forward) Jacobian appeared to be very unstable and yield unsatisfactory overall convergence behaviour while having a larger computational effort. Thus, we focus on the inverse Jacobian approximations in the following, especially on the comparison of the IQN-IMVJ method with the IQN-ILS method, re-using information from previous time steps.

A Remark Towards the Methods' Independence w. r. t. Spatial Resolution. For the sake of completeness a few results are added, that indicate the robustness of the considered quasi-Newton method, especially the IQN-ILS(R) and the IQN-IMVJ, with respect to different spatial resolutions, i. e., different numbers of unknowns at the coupling interface. For the pure fixed-point iteration, i. e., Gauß-Seidel or Jacobi like iteration, the mean iteration numbers for the considered parameter settings deteriorate if the spatial resolution is increased, cf. [31, p. 135]. Similar results have been found in a stability analysis by Degroote et al. [18]. However, several experiments using quasi-Newton methods as stabilizing post-processing methods show that the convergence behaviour of the coupling appears to be independent of the spatial resolution. Table 6.5 illustrates exemplary results for the IQN-ILS(8) and the IQN-IMVJ method for spatial resolutions of $N = 100$ and $N = 1000$ unknowns at the fluid-structure interface. Both methods show unchanged convergence behaviour apart from some very small fluctuations. Similar results are obtained for different numbers of N and other quasi-Newton methods.

V-IQN-IMVJ								V-IQN-ILS(8)							
$N = 100$				$N = 1000$				$N = 100$				$N = 1000$			
$\tau \setminus \kappa$	1000	100	10	$\tau \setminus \kappa$	1000	100	10	$\tau \setminus \kappa$	1000	100	10	$\tau \setminus \kappa$	1000	100	10
0.1	2.39	2.78	3.27	0.1	2.39	2.78	2.92	0.1	2.08	2.14	2.51	0.1	2.08	2.14	2.50
0.01	2.57	3.13	4.08	0.01	2.77	3.22	4.53	0.01	2.11	2.19	3.14	0.01	2.13	2.21	3.34
0.001	3.14	3.78	8.04	0.001	2.93	3.50	9.21	0.001	2.16	2.74	8.77	0.001	2.16	2.82	8.86

Table 6.5: One-dimensional elastic tube. Mean iteration numbers for the parallel coupling with IQN-ILS(8) and IQN-IMVJ, using different resolutions in spatial discretization. The number of unknowns at the interface is set to $N = 100$ and $N = 1000$.

6.3 Validation of the preCICE Implementation

Within this work, the IQN-IMVJ post-processing method has been implemented into the preCICE library for partitioned FSI coupling. The implementation follows the considerations from Chapter 5, particularly Section 5.2. In order to validate the implementation, the here considered one-dimensional flexible tube scenario is carried out using the preCICE library for the coupling of the fluid and solid single physics solver, written in C++. The results for the serial and parallel coupling system, together with the IQN-IMVJ post-processing method as well as for the parallel system accelerated by the IQN-ILS post-processing, are given in Table 6.6.

While for the easy cases the preCICE implementation confirms or even improves the results from the MatLab test bench, the preCICE implementation shows slightly worse convergence behaviour for the harder cases. This can be explained by means of several reasons and is mainly caused by the bad conditioning and sensitivity of the considered scenario. Firstly, MatLab provides a whole range of stabilizing internal techniques that are chosen adaptively, such as rank-revealing decompositions or singular-value decompositions in our case. Thus, it is likely that the MatLab implementation maintains stability in cases where the preCICE implementation becomes ill-conditioned. As this scenario is very sensitive, small differences in the conditioning

	S-IQN-IMVJ				V-IQN-IMVJ				V-IQN-ILS			
MatLab	$\tau \setminus \kappa$	1000	100	10	$\tau \setminus \kappa$	1000	100	10	$\tau \setminus \kappa$	1000	100	10
	0.1	2.98	3.10	3.40	0.1	2.39	2.78	3.27	0.1	2.70	3.25	5.14
	0.01	3.01	3.19	4.34	0.01	2.57	3.13	4.08	0.01	3.19	4.39	11.14
	0.001	3.07	4.45	–	0.001	3.14	3.78	8.04	0.001	4.39	10.09	35.75
preCICE	$\tau \setminus \kappa$	1000	100	10	$\tau \setminus \kappa$	1000	100	10	$\tau \setminus \kappa$	1000	100	10
	0.1	2.99	3.11	3.73	0.1	2.04	2.09	3.10	0.1	2.04	2.17	4.87
	0.01	3.04	3.17	6.20	0.01	3.07	2.22	7.76	0.01	2.14	4.05	15.98
	0.001	4.07	5.08	–	0.001	2.14	5.67	–	0.001	3.79	13.30	–

Table 6.6: One-dimensional elastic tube. Validation of the preCICE implementation of the newly introduced IQN-IMVJ method for the one-dimensional flexible tube scenario. Mean iteration numbers for different parameter settings are displayed.

of the least-squares problem easily increase the average number of coupling iterations required. For the most challenging case $(\tau, \kappa) = (0.001, 10)$ the preCICE implementation does not fail completely, but diverges in one of the last time steps due to bad-conditioning of the least-squares problem. The average number of iterations by then is comparable to the MatLab implementation. Lastly, the preCICE implementation does not subtract the offset for the cross sectional area and the pressure values before measuring the convergence, as this is not a typical task for FSI coupling. Though, the latter further fosters higher iteration numbers on average. Similar behaviour can be observed for the IQN-ILS method and an already existing implementation within the preCICE library.

In conclusion, the preCICE implementation of newly post-processing methods can be validated using the MatLab results. Especially the findings from the previous chapter, i e., the improved convergence and robustness properties of the IQN-IMVJ method in comparison to the IQN-ILS method, are clearly visible.

Chapter 7

Benchmarks and Applications

In Chapter 6 a whole range of coupling schemes, using quasi-Newton methods for stabilization and acceleration have been tested and investigated for a rather simple, one-dimensional test problem. The herein considered FSI applications are more sophisticated and are motivated from real world problems in the field of hemodynamics and aeronautics. All the considered scenarios are investigated for the most promising coupling variants as suggested in Chapter 6, i. e., the IQN-ILS and the IQN-IMVJ method for the serial as well as for the parallel coupling system. In doing so, we aim to confirm the findings from earlier experiments but also want to gain a deeper understanding of the behaviour of the two quasi-Newton methods to carry out a detailed comparison.

All applications were conducted using the in-house coupling library preCICE together with the open source toolbox openFOAM for the single physics solvers. Thereby, the newly integrated parts and improvements for the preCICE library, as described in Chapter 5, are deployed. In the following, three different scenarios are considered. Section 7.1 discusses a three-dimensional flow over an elastic structure based on [49]. The wave propagation through a three-dimensional flexible tube [17, 5, 8], induced by an initial pressure pulse, is investigated in Section 7.2. As a closing example, the FSI3 benchmark scenario [55] is carried out for the considered coupling schemes in Section 7.3.

7.1 Three-Dimensional Flow over an Elastic Structure

7.1.1 Scenario Description

This fluid-structure interaction application consists of a three-dimensional laminar and incompressible flow over an elastic structure based on [49]. The problem is considered to be symmetric in the x/y-plane, hence, the simulation is only performed in one half of the domain. The computational domain has a width and height of 0.4 m and spans a length of 1.5 m. The elastic structure consists of a rectangular obstacle with dimensions 0.2 m $\times$ 0.2 m $\times$ 0.2 m, which is mounted on the wall. The geometry of the scenario is given in Figure 7.1.

The fluid flow is driven by a parabolic velocity profile v, which is imposed as a Dirichlet boundary condition at the inflow boundary with a peak velocity of $v_{max} = 0.2\,\frac{\mathrm{m}}{\mathrm{s}}$. The profile is faded in smoothly, i. e.,

$$v(t) = \begin{cases} v\frac{1-cos(\frac{\pi}{2}t)}{2} & \text{if } t < 2.0, \\ v & \text{otherwise} \end{cases} \tag{7.1}$$

At the outflow boundary, standard outflow conditions are applied for velocity and pressure and no-slip conditions on all remaining boundaries. On the symmetry surface free-slip conditions need to be applied. The considered incompressible fluid has a density of $1 \times 10^3\,\frac{\mathrm{kg}}{\mathrm{m}^3}$ and a dynamic viscosity of $1 \times 10^{-3}\,\mathrm{Pa\,s}$. The density of the solid obstacle is $1 \times 10^3\,\frac{\mathrm{kg}}{\mathrm{m}^3}$, the Poisson ratio is 0.4 and the Young's modulus is set to $1 \times 10^4\,\frac{\mathrm{N}}{\mathrm{m}^2}$.

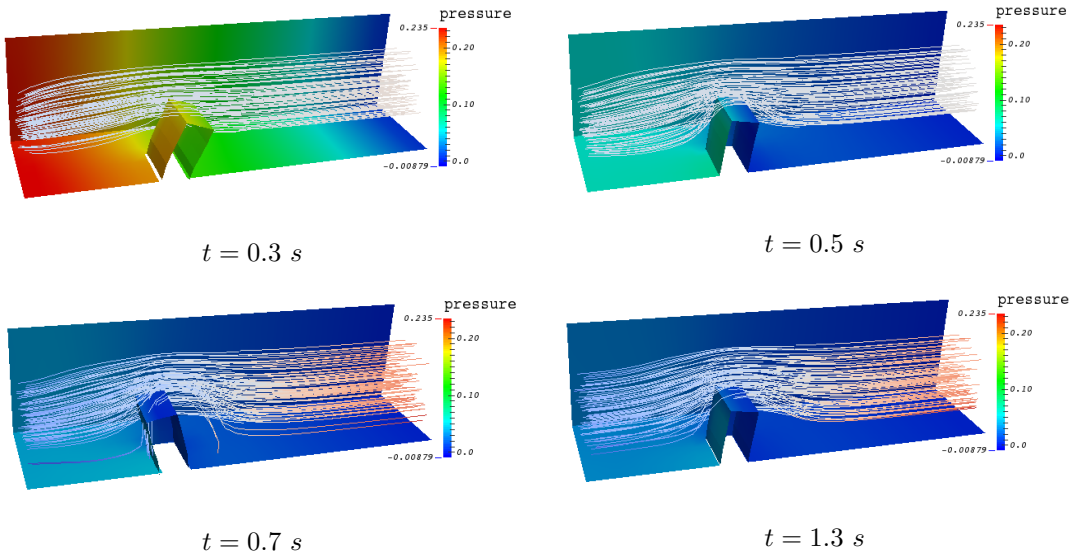


Figure 7.1: Three-dimensional flow over an elastic structure. The geometry and deformation of the elastic structure in the cross flow are depicted at different points in time. Pressure contours and streamlines for the parabolic flow are given.

7.1.2 Numerical Results

The numerical simulation is conducted using the single-physics solvers from openFOAM, augmented with the preCICE library functions to realise the partitioned coupling, as described in Section 5.1.2. The domain is decomposed into 1632 cells for the fluid mesh and 32 cells for the solid mesh. The scenario is simulated for a physical timeframe of 4 s, divided into 40 time steps of size 0.1 s. For both coupling variables, a relative convergence measure of 10^{-5} is used, i. e.,

$$\|\boldsymbol{x}_d^k - \boldsymbol{x}_d^{k-1}\|_2 \leq 10^{-5}\|\boldsymbol{x}_d^k\|_2, \qquad \|\boldsymbol{x}_f^k - \boldsymbol{x}_f^{k-1}\|_2 \leq 10^{-5}\|\boldsymbol{x}_f^k\|_2.$$

Figure 7.1 shows pressure contours and the physical deformation of the elastic structure at different time instances.

scaling factor	10^{0}	10^{-1}	10^{-2}	10^{-3}	10^{-4}	$4 \cdot 10^{-4}$	$6 \cdot 10^{-4}$	$8 \cdot 10^{-4}$	10^{-5}
ratio $r_f^d = \frac{\|\boldsymbol{x}_d\|_2}{\|\boldsymbol{x}_f\|_2}$	1.09^{-05}	1.09^{-04}	1.09^{-03}	1.09^{-02}	1.09^{-01}	4.36^{-01}	6.54^{-01}	8.72^{-01}	1.09
V-IQN-IMVJ(0)	13.92	11.30	8.97	7.40	<u>6.65</u>	6.82	6.82	7.05	7.17
V-IQN-IMVJ(5)	–	–	–	–	7.72	<u>7.65</u>	7.97	8.22	8.27
V-IQN-ILS(0)	16.22	14.97	12.37	10.75	9.40	<u>9.22</u>	9.37	9.32	9.37
V-IQN-ILS(5)	–	11.50	8.97	7.45	<u>6.67</u>	6.80	7.05	7.22	7.27
V-IQN-ILS(8)	–	–	–	9.02	7.57	<u>7.45</u>	7.67	7.85	8.10

Table 7.1: Three-dimensional flow over an elastic structure. Average number of iterations over the first 40 time steps with $\tau = 10^{-1}$ for the parallel-implicit coupling system, combined with different quasi-Newton post-processing methods is depicted for various scalings of the force data. For a well-conditioned V-system, a scaling of data has to be done previous to the post-processing, so that forces and displacements have the same order of magnitude. The V-IQN-IMVJ and the V-IQN-ILS were conducted with different numbers of reused time steps, indicated in brackets. The optimal value for each coupling scheme is underlined.

For the vectorial coupling, a scaling of the coupling data is necessary in order to pre-condition the system. To that end, the forces are scaled prior to each coupling iteration but with a constant factor for the entire simulation, such that forces and displacements have the same order of magnitude. Table 7.1 shows the results of a parameter study for the optimal scaling parameter for the respective quasi-Newton method. The average iteration numbers over the first 40 time steps are depicted for the IQN-ILS and the IQN-IMVJ post-processing methods for different number of reused time steps in the columns of $\boldsymbol{V}$ and $\boldsymbol{W}$. From this, several observations can be made. Firstly, it is obvious that the pre-conditioning is essential, as it improves the convergence speed by factor two for this scenario and, more importantly, most post-processing variants fail to converge at all if no scaling is performed in advance. Further, we see that for this scenario, the average iteration numbers for the optimal scaling factor do not differ remarkably from those that stem from a l_2-norm ratio of approximately one between the force and displacement data, as depicted in the last column. For optimal results in the following experiments, the V-system is applied with a pre-conditioning using a factor of $1 \cdot 10^{-4}$ for the IQN-IMVJ method and a factor of $4 \cdot 10^{-4}$ for the IQN-ILS method to scale the forces.

The re-use of information from previous time steps in the approximation process is of major importance for quasi-Newton methods. This is exactly the characteristic in which the IQN-IMVJ and the IQN-ILS methods differ the most. Table 7.2 shows the mean iteration numbers over the first 40 time steps for both variants, applied with the serial and vectorial system, for different numbers of retained previous time steps in the secant equation, i. e., in the matrices $\boldsymbol{V}$ and $\boldsymbol{W}$. The multi-vector method IQN-IMVJ that implicitly re-uses old information in a minimum norm sense, does not benefit from additionally incorporating previous information in an explicit sense via the secant equation. Instead, its convergence behaviour even deteriorates if done so. As opposed to this, the latter greatly improves the convergence speed for the IQN-ILS quasi-Newton method, but the optimal value of re-used time steps is apparently unknown and highly problem dependent. Thus, using the newly introduced IQN-IMVJ method renders the very costly tuning of the parameter R unnecessary, as suggested in Chapter 4 and 6.

reused steps (R)	0	1	2	3	4	5	6	7	8
S-IQN-ILS(R)	5.15	4.75	4.50	<u>4.45</u>	4.65	5.05	5.65	5.70	6.15
S-IQN-IMVJ(R)	<u>5.15</u>	5.22	5.57	5.87	6.32	6.70	7.00	7.45	7.62
V-IQN-ILS(R)	9.30	7.85	7.07	6.90	<u>6.70</u>	6.80	7.05	7.15	7.45
V-IQN-IMVJ(R)	<u>6.65</u>	6.75	6.82	7.27	7.47	7.72	8.02	8.10	8.52

Table 7.2: Three-dimensional flow over an elastic structure. Average numbers of coupling iterations over first 40 time steps for the IQN-ILS and IQN-IMVJ method, and for serial and parallel coupling. Different numbers of reused time steps are evaluated. † For a well-conditioned V-system, the forces are scaled by a factor of $4 \cdot 10^4$ for the IQN-ILS method and by a factor of $1 \cdot 10^4$ for the IQN-IMVJ method.

Furthermore, this experiments confirm the observations from Section 6.2.1 in that the V-system yields comparable results to the S-system, while achieving slightly worse convergence behaviour. This can also be seen from Figure 7.2, where the iterations per time step, required by the optimal (underlined) quasi-Newton variants from Table 7.2, are depicted for the entire simulation. Here, all variants show relatively fast and robust convergence behaviour after a attack time of ten time steps. The V-system, on average, requires two iterations more per time step than the S-system. However, the difference between the coupling systems is higher for the IQN-ILS method.

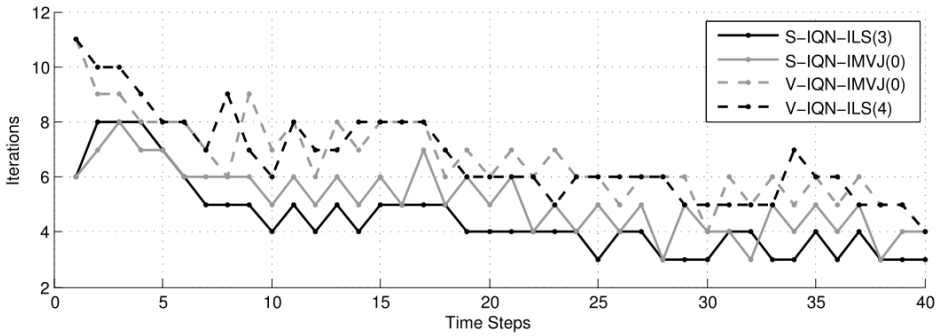


Figure 7.2: Three-dimensional flow over an elastic structure. Iteration numbers per time step for the serial (solid line) and vectorial (dashed line) coupling system are plotted for the first 40 time steps for both, the IQN-IMVJ and IQN-ILS post-processing method. The V-system is pre-conditioned with a scaling factor for the forces of $4 \cdot 10^{-4}$ for the IQN-ILS method and a factor of $1 \cdot 10^{-4}$ for the IQN-IMVJ method.

7.2 Wave Propagation in a Three-Dimensional Elastic Tube

7.2.1 Scenario Description

Let us now consider a three-dimensional internal flow problem through a flexible tube, referring to [17, 5, 8]. The scenario is inspired by the type of flow problems encountered in hemodynamics and constitutes a very strongly coupled FSI problem, as the density ratio of fluid and solid are near unity and incompressible internal flow is considered. The scenario simulates the wave propagation in a straight, three-dimensional flexible tube, induced by the pressure that is exerted on the tube walls by the internal flow. The tube has a length of 0.05 m and a wall thickness of 0.001 m. The internal fluid domain has a diameter of 0.003 m and both ends of the tube are fixed.

A pressure driven flow is generated by setting the boundary condition for the pressure inlet to a peak value of 1333.2 Pa for a initial duration of 0.003 s. To avoid spurious pressure waves, that may occur if a high and sharp pressure profile is applied from the start of the simulation, the pressure is faded in smoothly for $t < 0.003$ s. Thereafter the pressure is set to zero at the inlet boundary. Zero Dirichlet boundary conditions are imposed for the pressure at the outlet at every time instant.

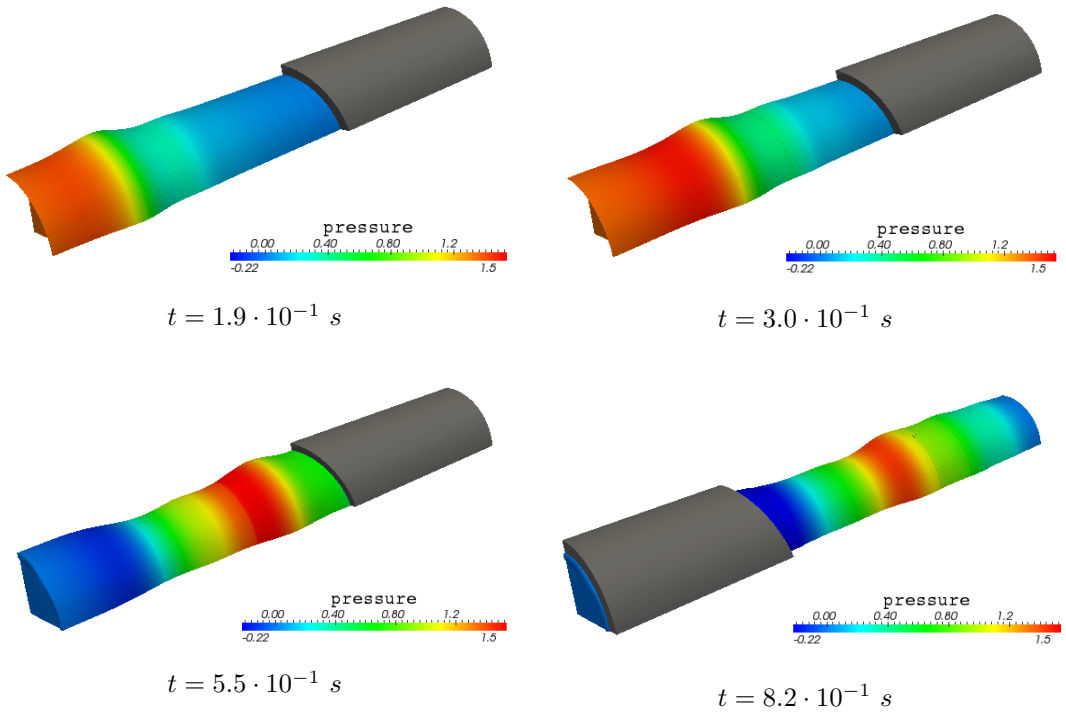


Figure 7.3: Wave propagation in a three-dimensional elastic tube. Geometry and pressure contours on the fluid-structure interface at different time instances. A pressure pulse propagates through the tube. Fluid-structure interface displacements are amplified.

For the tube, a hyper-elastic material model is assumed with Young's modulus of $E = 3 \times 10^5 \frac{\mathrm{N}}{\mathrm{m}^2}$, a Poisson's ratio of 0.3 and density $\rho_s = 1.2 \times 10^3 \frac{\mathrm{kg}}{\mathrm{m}^3}$. The fluid has a density of $\rho_f = 1 \times 10^3 \frac{\mathrm{kg}}{\mathrm{m}^3}$ and a dynamic viscosity of $\nu = 3 \times 10^{-3}\,\mathrm{Pa\,s}$. The pressure pulse that propagates through the tube and the arising wave in the tube wall are shown in Figure 7.3 for different time instances.

7.2.2 Numerical Results

For the simulation of this test case, the fluid and solid solvers from openFOAM are coupled using the preCICE library. A fluid mesh consisting of 17600 cells and a solid mesh with 800 cells are used. Hundred time steps of $1 \times 10^{-4}\,\mathrm{s}$ are performed. A relative convergence measure of 10^{-5} for both, the traction and the displacements on the fluid-structure interface is used. Mesh deformations are computed using a mesh motion solver based on radial basis function interpolation.

As a first experiment, the pre-conditioning for the V-system is considered. A parameter study for the scaling parameter of the pressure data is given in Table 7.3. It can be seen that the IQN-IMVJ and the IQN-ILS method have their optimum at displacement-force ratios that differ by factor 25. In particular, a tendency for the coherence of the displacement-force ratio and the optimal convergence behaviour of the quasi-Newton methods can hardly be detected, referring to the results from Scenario 7.1. However, the iteration numbers are not considerably worse for an automatic scaling to the same order of magnitude, i. e., $r_f^d \approx 1$. Moreover, it turns out, that the IQN-IMVJ method is slightly more robust with respect to the pre-conditioning.

A second experiment investigates the performance of the multi-vector and IQN-ILS quasi-Newton method for serial and parallel coupling with different amount of retained information form previous time steps. The results are given in Table 7.4. When comparing the serial and vectorial coupling system, the choice of the convergence criterion is of major importance. Table 7.4 shows two test series for the S-system. The first measures the convergence only for the displacement data, while the second demands the residual of both, displacements

scaling factor	$1 \cdot 10^{-6}$	$1 \cdot 10^{-7}$	$1 \cdot 10^{-8}$	$4 \cdot 10^{-8}$	$8 \cdot 10^{-8}$	$1 \cdot 10^{-9}$	$1 \cdot 10^{-10}$	$1 \cdot 10^{-11}$
ratio $r_f^d = \frac{\|\boldsymbol{x}_d\|_2}{\|\boldsymbol{x}_f\|_2}$	3.59^{-03}	3.59^{-02}	3.59^{-01}	1.43	2.87	3.59	3.59^{01}	3.59^{02}
V-IQN-IMVJ(0)	17.26	12.62	9.70	<u>8.67</u>	9.02	9.24	10.70	15.76
V-IQN-ILS(0)	37.85	29.77	24.45	22.12	21.65	21.50	<u>21.28</u>	25.62
V-IQN-ILS(5)	16.64	14.35	13.40	13.61	13.35	13.12	<u>12.90</u>	15.31
V-IQN-ILS(8)	15.07	12.90	12.08	12.20	12.01	<u>11.93</u>	11.99	14.63

Table 7.3: Wave propagation in a three-dimensional elastic tube. Average number of iterations over the first 100 time steps with $\tau = 10^{-4}$ for the parallel-implicit coupling system, combined with different quasi-Newton post-processing methods is depicted for various scalings of the force data. For a well-conditioned V-system a scaling of data has to be done previous to the post-processing, so that forces and displacements have the same order of magnitude. The V-IQN-IMVJ and the V-IQN-ILS were conducted with different numbers of reused time steps, indicated in brackets. The optimal value for each coupling scheme is underlined.

reused steps (R)	0	1	3	5	7	9	11	13	15	17
S-IQN-ILS(R)	9.97	9.45	6.80	6.35	6.17	<u>6.05</u>	6.05	6.06	6.15	6.21
S-IQN-MVJ(R)	<u>5.36</u>	8.97	9.47	9.32	9.25	9.21				
S-IQN-ILS(R)*	13.27	12.89	10.05	8.25	7.65	<u>7.47</u>	7.51	7.56	7.66	7.81
S-IQN-IMVJ(R)*	<u>7.19</u>	10.49	10.40	10.24	11.07	11.02				
V-IQN-ILS(R)†	21.41	18.63	13.73	12.90	12.28	11.95	11.71	13.36	13.36	<u>11.47</u>
V-IQN-IMVJ(R)†	<u>8.67</u>	15.18	17.66	17.60	16.99	17.13				

Table 7.4: Wave propagation in a three-dimensional elastic tube. Average numbers of coupling iterations over the first 100 time steps for the IQN-ILS and IQN-MVJ method combined with the serial and parallel coupling system. Different numbers of reused time steps are evaluated. † For a well-conditioned V-system, the forces are scaled by a factor of 10^{-10} for the IQN-ILS and $4.0 \cdot 10^{-8}$ for the IQN-IMVJ method. * To ensure proper conditions for the comparison with the V-system, the S-system also needs to fulfill both relative convergence criteria, i. e., for the displacements and the pressure. The test series for the S-system differ in that the first only uses one criterion for the displacements while the second (*) applies both.

and pressure, to fall below the relative convergence measure. As a consequence, the required numbers of coupling iterations increase by 1.4 iterations for the IQN-ILS(9) method and by 1.8 iterations for the IQN-IMVJ(0) method, on average. This can be explained by the observation, that the residual for the displacements usually comes down faster than the residual for the pressure values, as can be seen in Figure 7.6. Therefore, to ensure a fair comparison between the coupling systems, the serial system additionally needs to fulfill the relative convergence criterion for the pressure.

Regarding Table 7.4 we see once again that the IQN-IMVJ method yields best results if no information is re-used explicitly within the secant equation, that means it appears to be optimal without any tuning of problem dependent parameters. Furthermore, it outperforms the optimal configurations for the IQN-ILS method for both coupling systems, but especially for the vectorial coupling, where the difference almost amounts to three iterations on average. On the contrary, for the IQN-ILS method, retaining information from 17 time steps in the past results in a halving of the required coupling iterations for the vectorial system. All these observations confirm the suppositions and findings from Chapter 6.

Similar to previous findings for the IQN-ILS method [57], the IQN-IMVJ method almost retains the convergence order when switching from the serial to the parallel coupling scheme. This can be seen nicely in Figure 7.4-(left), where the required coupling iterations per time step are given for the optimal coupling scheme configurations from Table 7.4, after a attack time of 15 time steps. The curves for the iteration numbers required by the V-system are pretty similar to those for the S-system, but they show higher peaks at somehow critical positions in time, like for example at time step 31.

Considering the attack time of the coupled system within the first 15 time steps, depicted in Figure 7.4-(right), we see that the IQN-ILS method entails very high numbers of required coupling iterations at the very beginning of the simulation, while the IQN-IMVJ method shows very robust behaviour and fast convergence. This improved convergence during the

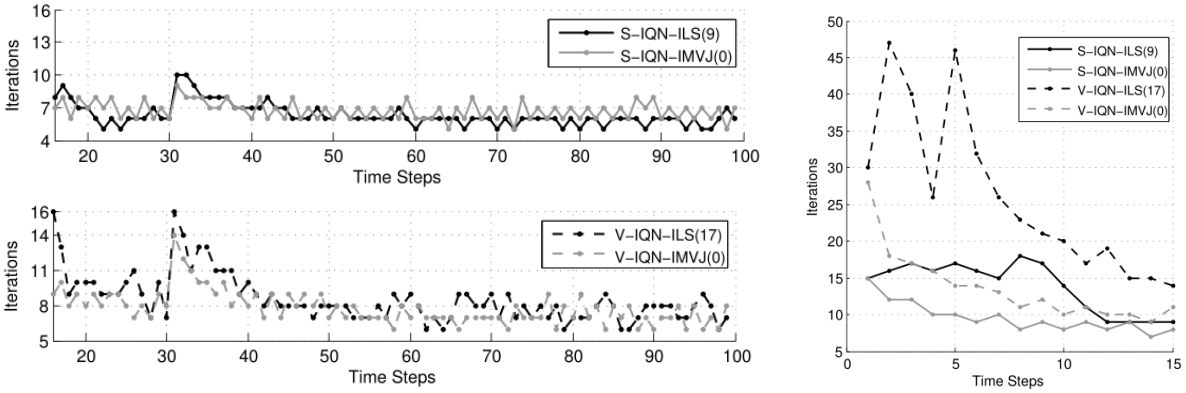


Figure 7.4: Wave propagation in a three-dimensional elastic tube. Iteration numbers for IQN-ILS and IQN-IMVJ, and for serial (solid line) and parallel (dashed line) coupling. **left:** time steps 16 to 100, **right:** time steps 1 to 15.

attack time first and foremost accounts for the dominance of the IQN-IMVJ(0) method over the IQN-ILS(17) method, cf. Table 7.4. A similar behaviour can be observed for the serial coupling system, but is not as pronounced as for the parallel.

For a more detailed comparison of the two quasi-Newton methods, we investigate the convergence rates for pressure and displacement values for different time steps. Figure 7.5 shows the relative l_2-norm of the pressure residual along with the relative convergence criterion, plotted over the required numbers of coupling iterations for serial and parallel coupling. Here, the earlier discussed different start performance is reflected and clearly visible. Considering the S-system in Figure 7.5, we see, that the convergence behaviour of the IQN-ILS method improves considerably as the Jacobian is trained through the simulation process. Indeed, this holds true from the very beginning of the simulation. On the contrary, the IQN-ILS method shows a stagnating convergence behaviour in the early stage, in that the convergence rates for the first ten time steps do not improve. Thereafter, the convergence behaviour is fairly comparable. Note, that the convergence curve for the first time step is the same for both variants, as the IQN-IMVJ is identical to the IQN-ILS at the very beginning, due to the initial zero-guess for the Jacobian. Thus, we can conclude, that retaining old information in an implicit, norm-minimizing manner endorses the quasi-Newton iteration to be faster and more robust.

These observations can equally be drawn from the vectorial coupling scheme, where the effect of the different convergence behaviour for the IQN-ILS(17) and IQN-IMVJ(0) method in the early stage of the simulation is far more pronounced (compare Figure 7.5). While the IQN-IMVJ(0) method supplies fast and steep Newton-like convergence after the first time step, the IQN-ILS(17) method shows stagnation or even deterioration for the convergence rates of the early time steps. As a result, the IQN-ILS(17) method requires up to 20 coupling iterations more then the multi-vector method for certain time steps within the attack time. Note, that the reason for the different convergence curves of the V-IQN-ILS(17) and V-IQN-IMVJ(0) for the first time step is found in the different scaling of pressure values which results in a different conditioning for least-squares problem.

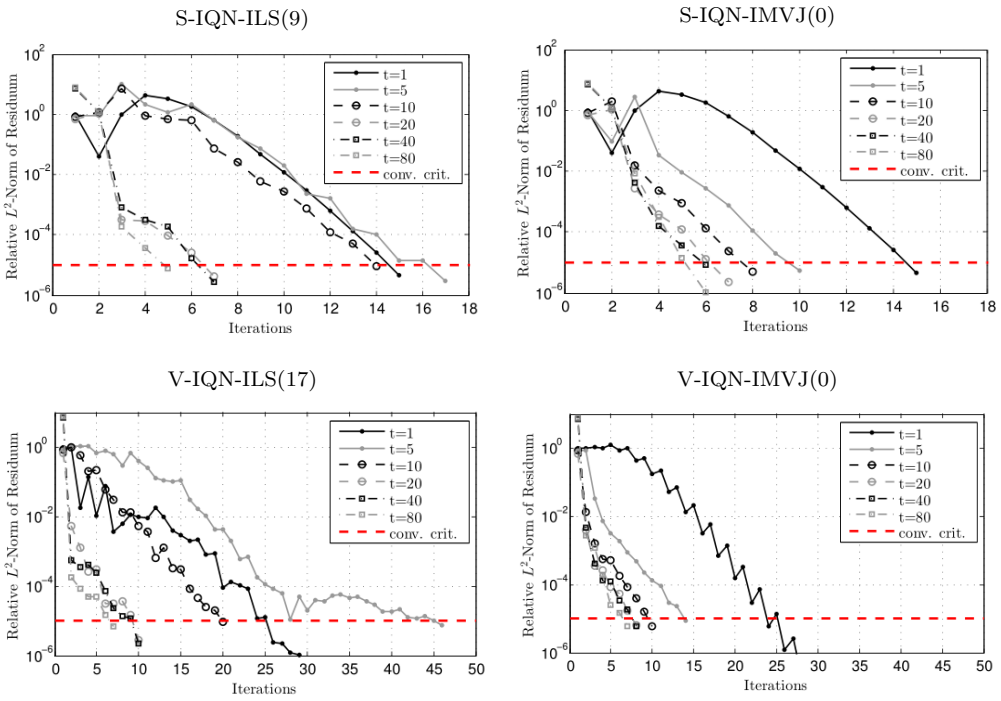


Figure 7.5: Wave propagation in a three-dimensional elastic tube. The l_2-norm of the pressure residual is plotted over the number of required coupling iterations for different time steps spread over the entire simulation. Typical convergence rates for both quasi-Newton variants are shown for the serial (top row) and vectorial (bottom row) coupling scheme. The relative convergence criterion is depicted as a dashed red line.

Up to this point, we solely considered the drop with respect to the l_2-norm of the relative residual for the pressure. In general, looking at the convergence rates for the relative residual of the displacement values, a faster and slightly better convergence behaviour can be observed. Figure 7.6 shows the convergence rates for the pressure and displacement residuals at different time instances within the early stage of the simulation. For the serial system, we observe, that the residual of the displacement drops below the relative convergence criterion at a maximum of two iterations earlier as for the pressure. This in turn fairly explains the different numbers of required coupling iterations for the two test series for the S-system in Table 7.4 with and without convergence criterion for the pressure residual.

Considering the vectorial coupling, the number of additional iterations, required by the IQN-ILS(17) method to drop the relative residual of the pressure below the convergence criterion, seems to be arbitrary large. This, in particular applies to time steps within the attack time of the simulation, i. e., we observe a difference of 25 coupling iterations between the satisfaction of the relative convergence criterion for the displacements and for the pressure values for the second time step, as can be seen from Figure 7.6. This is not an exceptional phenomenon, but can be equally observed for the fifth, sixth, seventh and more time steps in comparable proportions. On the contrary, the IQN-IMVJ(0) method appears to reduce both residuals with comparable speed and reliability, which once more accounts for a better robustness of this method.

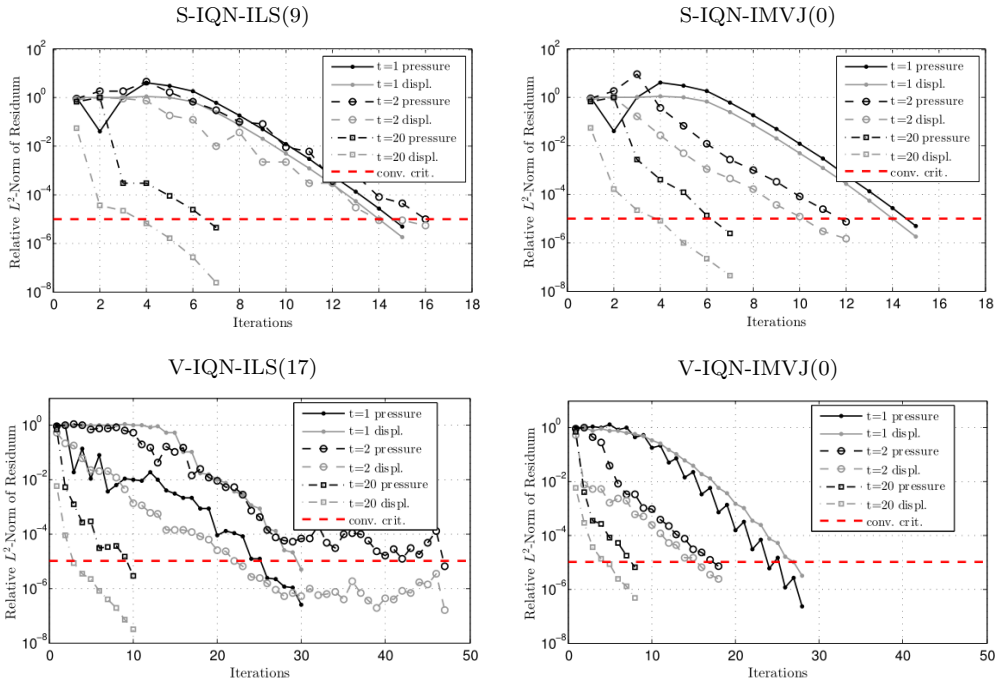


Figure 7.6: Wave propagation in a three-dimensional elastic tube: Convergence rates for the relative pressure and displacement residual, plotted over the required number of coupling iterations for various coupling schemes at different time steps. The relative displacement residual is shown in light gray and the relative pressure displacement is shown in dark gray.

As a last experiment, the discussed coupling schemes along with their respective optimal amount of retained information from the previous time steps are conducted for different relative convergence criterions. While all the experiments are carried out using a relative convergence criterion of $\epsilon_r = 10^{-5}$, two additional series for $\epsilon_r = 10^{-7}$ and $\epsilon_r = 10^{-10}$ are realised. For the rather small convergence criterion of $\epsilon_r = 10^{-10}$ all methods fail to converge. Here, the least squares system becomes ill-conditioned, as a lot of differences in the secant equation are near machine precision. The mechanism to preserve good conditioning as described in Chapter 5 would discard almost all recent information, thus no convergence can be obtained. The results for $\epsilon_r = 10^{-7}$ are given in Figure 7.8-(left) for the different coupling schemes. Both quasi-Newton variants show a deterioration of the convergence behaviour for the very first time steps of the simulation, where little or no information can be retained. However, while the IQN-IMVJ(0) method reveals fast and monotonic advancing convergence rates after few time steps, the IQN-ILS(17) method once more presents the earlier discussed problematic convergence behaviour within an attack time of 20 time steps, but far more pronounced due to the smaller convergence criterion. This can be seen clearly in Figure 7.7 for time step six where we obtain good reduction of the pressure residual for the IQN-IMVJ(0) method, but deterioration of convergence speed for the IQN-ILS(17) method. Note that this is not an

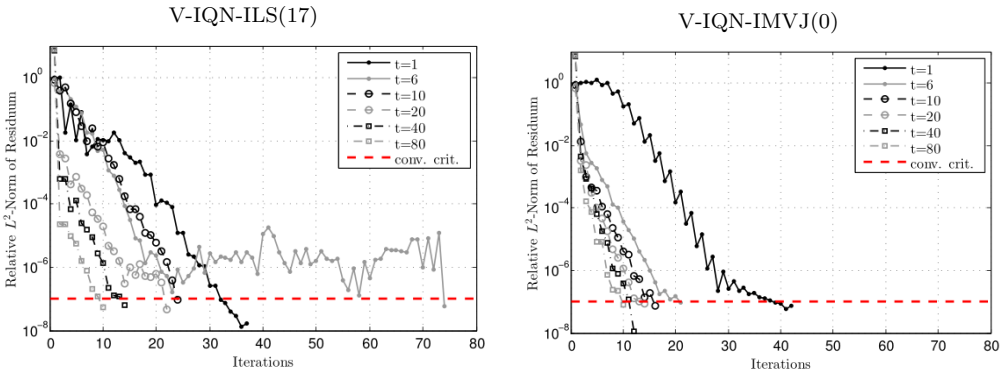


Figure 7.7: Wave propagation in a three-dimensional elastic tube. Convergence rates for the pressure residual with a relative convergence criterion of 10^{-7} are plotted for different time steps. Results are shown for the V-IQN-ILS(17) and V-IQN-IMVJ(0) method.

individual case, but can be observed frequently within the first 20 time steps. This can be confirmed by Figure 7.8-(right). Here, the average convergence rate per time step

$$\bar{\rho}_{l_2} = \left(\frac{\|\boldsymbol{r}^k\|_2 / \|\boldsymbol{x}^k\|_2}{\|\boldsymbol{r}^0\|_2 / \|\boldsymbol{x}^0\|_2} \right)^{\frac{1}{k}} \tag{7.2}$$

is plotted for the pressure over the entire simulation, applying the vectorial coupling scheme. The V-IQN-ILS(17) method presents poor average convergence rates or even stagnation in the early stage of the simulation. Though, after this attack time, we obtain steep and reliable linear convergence rates with a good constant for both methods as well as a minor plus for the V-IQN-IMVJ(0). This is endorsed by the iteration numbers from Figure 7.8-(left). Here, the mean iteration numbers are given, cutting off the attack time. From this, two observations can be made that confirm the previous findings. Firstly, the number of required iterations increases linearly if a smaller convergence criterion is applied, and secondly, the dominance of the multi-vector method over the IQN-ILS method is mainly caused by the unfavourable behaviour of the latter in the early stage of the simulation.

conv. measure	10^{-5}	10^{-7}	$10^{-5}_{\dagger}$	$10^{-7}_{\dagger}$
S-IQN-ILS(9)	7.47	10.54	$4.93^{\dagger}$	$8.78^{\dagger}$
S-IQN-IMVJ(0)	7.19	9.80	$4.92^{\dagger}$	$8.89^{\dagger}$
V-IQN-ILS(17)	11.47	18.81	$8.65^{\dagger}$	$12.39^{\dagger}$
V-IQN-IMVJ(0)	8.67	15.17	$7.90^{\dagger}$	$11.48^{\dagger}$

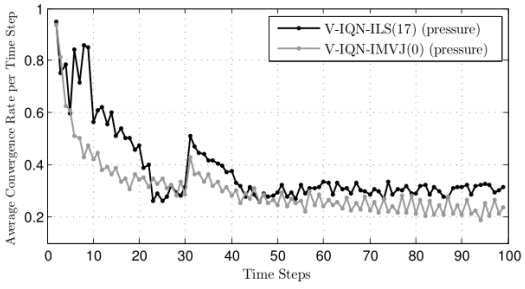


Figure 7.8: Wave propagation in a three-dimensional elastic tube. **left:** Mean iteration numbers for a relative convergence measure of 10^{-5} and 10^{-7}, respectively. † Mean iteration numbers, averaged over time step 16-100, cutting off the attack time. **right:** Average convergence rates for the pressure residual of both methods using the V-system.

limited columns (L)	4	3	2	1 \ Broyden	V-IQN-IMVJ(0)†	V-IQN-ILS(8)†
IQN-IMVJ-limited(L)†	11.97	12.73	14.50	16.50	8.67	12.20

Table 7.5: Wave propagation in a three-dimensional elastic tube. Average numbers of coupling iterations over the first 100 time steps for limited-information variants of the multi-vector method combined with the vectorial coupling scheme. The number of utilized input-output information is limited to be small and constant. The norm-minimization ranges over the distance L. † For a well-conditioned V-system, the forces are scaled by a factor of $4.0 \cdot 10^{-8}$ such that a displacement-force ratio of approximately one $r_f^d \approx 1$ is ensured.

In attempting to reduce the overall computational complexity of the multi-vector method, some limited-information variants had been introduced in Chapter 6, where the number of allowed columns in the secant equation system was limited to a small but constant number L. For the case of restricting the method to only use the single most recent input-output information, this exactly coincides with the Broyden rank-one update method. The results for the one-dimensional flexible tube lead us to hope that such limited versions are interesting and beneficial for less demanding scenarios due to their low computationally complexity. For further evaluation, a test series with $L \in \{1, 2, 3, 4\}$ is carried out for the three-dimensional flexible tube scenario in conjunction with the vectorial coupling scheme. The results are depicted in Table 7.5, together with some reference values for the original IQN-IMVJ(0) method and the IQN-ILS(8) method.

While the full-information V-IQN-IMVJ(0) method clearly shows highly superior results, the mean iteration numbers for the limited-information versions with $L \in \{3, 4\}$ appear to be quite comparable to the V-IQN-ILS(8). It is to be signalized that the number of utilized columns for the V-IQN-ILS(8) is up to 96 on average, while for the limited-information methods it never exceeds $L \leq 4$. Additionally, the system that needs to be solved for the Jacobian update is small with a quadratic $L \times L$ matrix and thus even allows for direct inversion. On the downside, the Jacobian approximation has to be stored explicitly. Concluding, the limited-information variants indeed offer an interesting and computationally efficient alternative with satisfying coupling properties. However, for a more informed utterance, a detailed comparison including computational complexity, overall memory consumption and robustness issues has to be done.

7.3 FSI3 Benchmark–Cylinder Flap

Turek et al. proposed a series of FSI benchmarking scenarios as described in [55]. Here, a two-dimensional scenario for laminar incompressible flow around a fixed cylinder is considered, which is referred to FSI3 benchmark scenario. The test case appears to be quite challenging despite being only two-dimensional, since it involves large deformations of the elastic structure and thus of the fluid-structure interface. The scenario involves the initiation of oscillations, therefore a large number of oscillation periods has to be simulated until a quasi-stationary state is reached. This makes the simulation very time consuming. The original formulation of the benchmark scenarios comprises three different parameter settings with a different level of difficulty with respect to FSI coupling. Here, only the third benchmark scenario is considered, as it offers the highest Reynolds number and the lowest density of the structure and thus constitutes the most challenging setting for FSI coupling algorithms.

7.3.1 Scenario Description

The FSI3 benchmark scenario describes a two-dimensional incompressible flow around a fixed cylinder. A flexible Saint-Venant-Kirchhoff cantilever is attached to the cylinder and placed near the middle of the flow channel, but with a small vertical offset such that oscillations are induced. The geometry of the scenario and all essential dimensions of the channel, the cylinder and the cantilever are given in Figure 7.9. A parabolic inflow profile is prescribed at the left boundary, while free outflow conditions are imposed on the right boundary. All remaining boundaries, i. e., the top and the lower wall of the flow channel as well as the surface of the cylinder and cantilever are no-slip boundaries.

An incompressible Newtonian fluid with density $\rho_f = 1 \times 10^3 \, \frac{\mathrm{kg}}{\mathrm{m}^3}$, a dynamic viscosity of $\nu_f = 1 \times 10^{-3} \, \frac{\mathrm{m}^2}{\mathrm{s}}$ and a Reynolds number $Re = 200$ is considered with a mean inflow velocity of $\bar{U}$ =2 $\frac{\mathrm{m}}{\mathrm{s}}$. The elastic cantilever is modeled using the Saint-Venant-Kirchhoff material model, allowing for large displacements. Here, a Young's modulus of E =$1.4 \times 10^6 \, \frac{\mathrm{N}}{\mathrm{m}^2}$ a Poisson's ratio of $\nu_s = 0.4$ and a density of $\rho_s = 1 \times 10^3 \, \frac{\mathrm{kg}}{\mathrm{m}^3}$ is assumed.

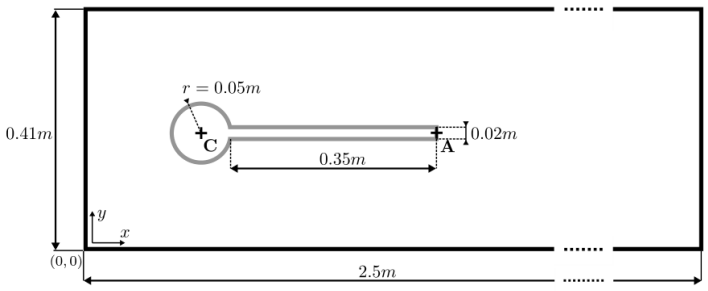


Figure 7.9: FSI3 Benchmark. Sketch of the geometry for the FSI3 scenario, as proposed in [55]. A fixed cylinder with attached elastic cantilever is placed in the middle of a channel, but with a small vertical offset at point C = (0.2m,0.2m). This picture is taken from [31].

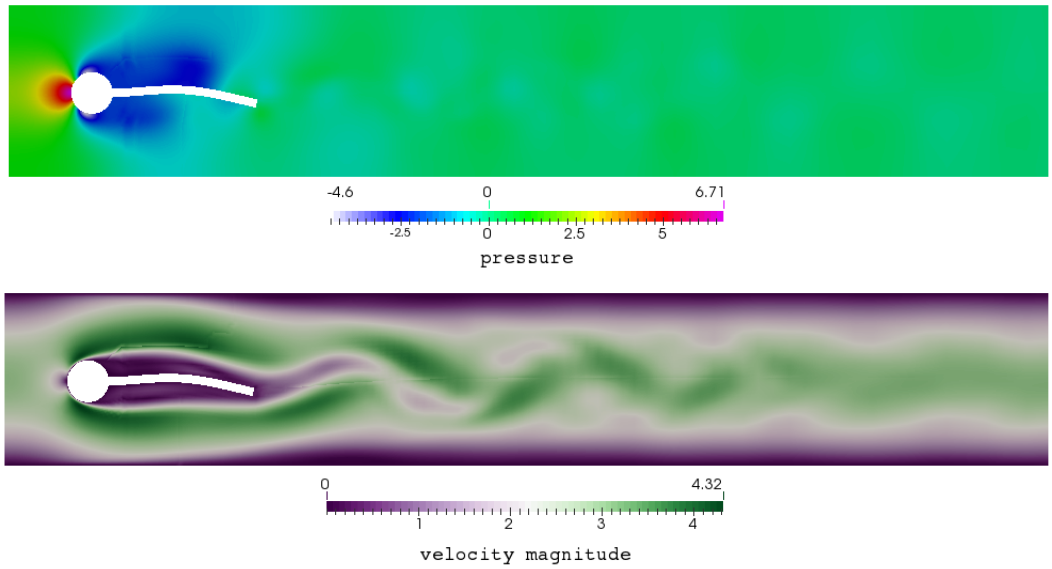


Figure 7.10: FSI3 Benchmark. Geometry of the scenario with pressure (upper) and velocity magnitude (lower) contours at t =0.697 s. The pressure values are given in Pa and the velocity magnitude is given in $\frac{\text{m}}{\text{s}}$.

7.3.2 Numerical Results

As for the previous scenarios, fluid and solid single-physics solvers from openFOAM are coupled using the preCICE library. For the fluid solver, the mesh is decomposed into 5981 cells, while the solid mesh consists of 82 cells. A timeframe of T =20 s is simulated and subdivided into 20000 time steps of length 1×10^{-3} s. A relative convergence measure for the traction and the displacements at the fluid-structure interface is used with a stopping criterion of $\epsilon_r = 10^{-3}$. The dynamic movement of the cantilever converges to a periodic oscillation with an average period of 128 time steps, i. e., 0.128 s, after an attack time of the system. Figure 7.10 shows pressure contours and the velocity magnitude of the flow at t =0.697 s.

A parameter study to supply the optimal pre-conditioning for the vectorial coupling system results in a force-scaling factor of 10^{-8} for the V-IQN-ILS method and of 10^{-9} for the V-IQN-IMVJ method. This refers to a displacement-force ratio of $r_f^d \approx 1$ for the parallel multi-vector method and to a ratio of $r_f^d \approx 0.1$ for the V-IQN-ILS method. Experiments for the multi-vector method and the IQN-ILS method have been carried out for serial and vectorial coupling, retaining information from zero, four and eight previous time steps. The results are shown in Table 7.6. For this scenario, the vectorial coupling in combination with either of the two quasi-Newton post-processing methods appears to be superior to the serial coupling system. Regarding the multi-vector method this accounts to an improvement of 1.5 iterations on average for the best configuration. Considering the re-use of information from the past, we see that the vectorial coupling benefits from retaining more time steps than the serial coupling. This confirms the observations from the previous scenarios.

reused steps (R)	0	4	8
S-IQN-ILS(R)*	7.60	4.27	4.88
S-IQN-IMVJ(R)*	6.08	5.44	5.54
V-IQN-ILS(R)†	11.86	5.26	4.23
V-IQN-IMVJ(R)†	5.81	3.98	4.82

Table 7.6: FSI3 Benchmark. Average numbers of coupling iterations over first 20000 time steps for IQN-ILS and IQN-MVJ method combined with the serial and parallel coupling system. Different numbers of reused time steps are evaluated. † For a well-conditioned V-system, we scale the forces by a factor of 10^{-8} for IQN-ILS and 10^{-9} for IQN-IMVJ. * For a fair comparison between the V-system and the S-system, the latter needs to fulfill both relative convergence criteria, i. e., for the displacements and the pressure.

For the multi-vector method we observe a new phenomenon, which is partially contradictory to the findings from the previous scenarios. Here, the IQN-IMVJ method shows an improved convergence behaviour that results in nearly two iterations less on average, if a few number of previous time steps are retained explicitly within the secant equation. With this configuration, the V-IQN-IMVJ(4) method outperforms the V-IQN-ILS(8) method. Although, without tuning the parameter R of re-used time steps, the multi-vector method on average requires approximately one iteration more to converge than the V-IQN-ILS(8) method with optimal configuration, but is twice as fast as the V-IQN-ILS(0) method.

In compliance with previous findings, we observe the typical poor performance of the IQN-ILS method in the early stage of the simulation. This can be seen from Figure 7.11-(left), where the average convergence rates $\bar{\rho}_{l_2}$ of the pressure residual are plotted every 250 time steps over the entire simulation. The V-IQN-IMVJ(4) method shows robust and reliable convergence behaviour from the very beginning of the simulation. In Figure 7.11-(right), the average convergence rates are depicted for one oscillation period of the cantilever, approximately ranging from time step 3066 to time step 3248. Here, the multi-vector method shows superior

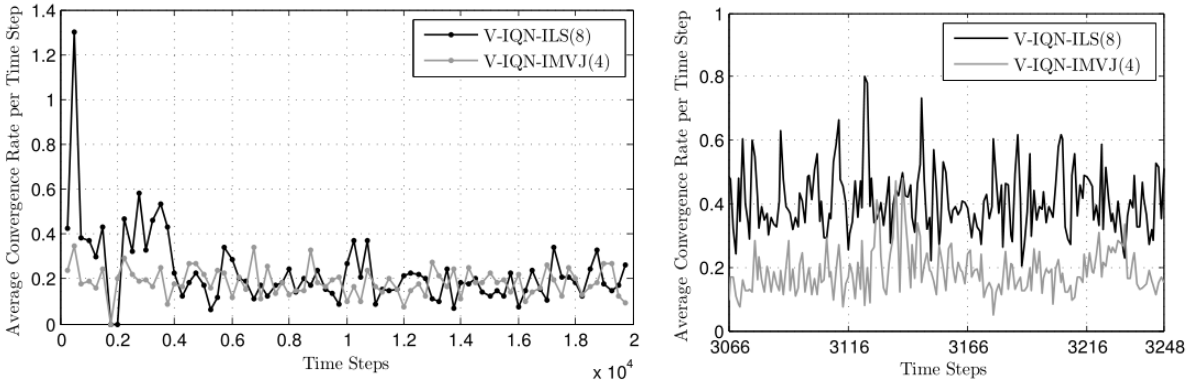


Figure 7.11: FSI3 Benchmark. Average convergence rates $\bar{\rho}_{l_2}$, cf. eq. (7.2), of the pressure residual for the IQN-ILS(8) and the IQN-IMVJ(4) post-processing method and a parallel coupling scheme. **left:** Average convergence rates over the entire simulation, plotted every 250 time steps. **right:** Average convergence rates for the duration of one oscillation period, approximately from time step 3066 to time step 3248.

convergence rates, which accounts to the fact that the oscillation has not fully converged and is still within its attack time.

Summary

Summarizing the findings from this chapter, the multi-vector method appears to have an edge over the IQN-ILS method for all the conducted experiments. Especially the IQN-IMVJ method in general renders the tuning of the problem dependent parameter R of re-used previous time steps unnecessary, which accounts to a major advantage of this method. The dominance of the multi-vector method above the IQN-ILS(R) method has been found to be mainly due to a poor convergence behaviour of the latter during the early stage of the simulation, when little information from the past is available. As opposed to this, the IQN-IMVJ method provides robust and reliable convergence from the very beginning, which suggests that it is beneficial to incorporate past information implicitly in a minimum norm sense, rather than explicitly in the secant equation. After a certain time, both quasi-Newton variants show comparable convergence rates right up to very high precision.
Comparing the serial and the vectorial coupling system, the latter appears to achieve quite similar performance than the former, if coupled with the multi-vector method. Similar results have been found earlier for the IQN-ILS method in [57] and can be confirmed for the herein considered scenarios. For the vectorial coupling system a pre-conditioning in terms of scaling the force values has to be done in advance. While the coherence between the l_2-displacement-force ratio r_f^d and the optimal performance of the respective quasi-Newton method varies a lot throughout the conducted experiments, at least for the V-IQN-IMVJ method an automatic scaling mechanism to $r_f^d \approx 1$ appears to be practical.

Concluding, the multi-vector method clearly shows some noticeable advantages compared to the IQN-ILS method, i. e., providing a more robust convergence behaviour and rendering the tuning of additional problem dependent parameters unnecessary. However, the IQN-IMVJ method requires to store an explicit representation of the Jacobian approximation, which constitutes to a higher memory requirement and higher computational costs for the Jacobian update.

Chapter 8

Summary and Outlook

8.1 Summary of Findings and Achievements

This thesis' main goal was to supply a comprehensive categorization and comparison of various existing implicit coupling schemes for the coupling of partitioned FSI simulation as well as the introduction of some new combinations of quasi-Newton methods and fixed-point equation coupling systems. This comprises a clear and uniform exposition of different coupling algorithms, mainly the quasi-Newton post-processing methods, along with the linkage to recent literature as well as the implementation and evaluation for a qualitative and quantitative comparison.

To provide a lucid overview of the different coupling algorithms, we separated the formulation of the coupling system, i. e., the calling-order of the single-physics solvers which results in a certain fixed-point equation, from the description of the so called post-processing methods that accelerate and stabilize the pure fixed-point iteration. In doing so, three different coupling systems have been formulated in Chapter 3, that differ with respect to parallel efficiency and modularity properties, namely the serial implicit coupling scheme, the vectorial- or parallel-implicit coupling scheme as well as the block-iterative serial coupling scheme. This thesis considers the case of strongly coupled FSI problems, thus the pure fixed-point iteration fails to converge and needs to be stabilized and accelerated by additional advanced methods, i. e., quasi-Newton methods. A comprehensive exposition and classification of quasi-Newton methods has been provided in Chapter 4. Various generalized Broyden methods are described, that differ in the way in which past information is incorporated into the approximation process and which operator is to be approximated. Emphasis is put on the newly introduced inverse multi-vector method that re-uses past information in an implicit way. Different coupling schemes are obtained, combining one of the coupling systems with an appropriate post-processing quasi-Newton method.

A substantial share of this work was furthermore the implementation and realization of robust and fast converging quasi-Newton methods into the preCICE framework. This is a flexible and powerful library for the numerical coupling of partitioned FSI problems, optimized for black box single physics solvers. In order to spot the most promising coupling algorithms for the integration, a whole range of coupling schemes has been analyzed, evaluated and compared

in Chapter 6 for a simple one-dimensional test problem. The investigation of the different coupling systems with the resulting fixed-point equations showed, that the vectorial-implicit coupling appears to have quite comparable performance in terms of required coupling iterations than the serial-implicit scheme, while bearing the chance of improving the parallel efficiency up to a factor of two compared to the latter. Thus, the better parallel efficiency of the V-system clearly outperforms a potential minor loss in convergence speed. The block-iterative serial-implicit coupling scheme showed no advantages over the remainder coupling schemes, while having the same drawbacks with regards to parallel efficiency as the S-system.
In comparing the different quasi-Newton post-processing methods, all the forward approximations of the Jacobian for the residual- or fixed-point operator have been shown to provide poor robustness and convergence behaviour. We therefore focused on the inverse Jacobian approximations that also spare us from solving a additional system of equations in the quasi-Newton updating process. Here, the multi-vector method (IQN-IMVJ) showed the best results and appeared to outperform even standard coupling algorithms for FSI coupling, such as the IQN-ILS method. To the best of the authors' knowledge the multi-vector method with approximation of the inverse Jacobian for the residual operator has not been applied to FSI coupling problems in conjunction with the vectorial coupling scheme before. Due to its implicit minimum norm like incorporation of past information into the approximation process, the IQN-IMVJ method occurs to be mostly optimal without the need of tuning problem dependent parameters such as the number of re-used time steps for the IQN-ILS method. The Broyden method and other variations of the multi-vector method, such as limited secant information methods IQN-IMVJ-limited(L), have been found to provide worse, but satisfying results and are, due to their lower computational costs, a striking alternative for weakly coupled problems that bear fewer instabilities.

Based on these insights the multi-vector method has been integrated into the preCICE library for the serial and vectorial coupling scheme. Here, special emphasis has been put on the robust and efficient realization of the Jacobian update step. To that end, an economy-size QR-decomposition that is updated and computed successively, together with a mechanism to maintain a good conditioning of the least squares system, has been implemented.

In Chapter 7 a multitude of experiments for three different FSI applications have been carried out using the preCICE implementation in conjunction with the openFOAM toolbox. Thereby, mainly two objectives were pursued. Firstly, the preCICE implementation had to be tested and validated for more realistic and real world scenarios. Secondly, we aimed to confirm the suppositions and findings concerning the performance of the quasi-Newton methods proposed earlier. IN doing so we gained a deeper insight to the behaviour of the different coupling schemes. It has been manifested that the multi-vector method appears to outperform the IQN-ILS(R) method with an optimal number of re-used time steps R. On top of that, the IQN-IMVJ method shows optimal behaviour without any tuning of problem dependent parameters in almost all considered cases. This means that the parameter R of explicitly retained previous time steps can be set to zero and the scaling parameter to pre-condition the V-system can be chosen automatically such that both displacements and forces have the same order of magnitude. The dominance of the multi-vector method above the IQN-ILS method has been found in a very robust convergence behaviour of the former from the very beginning of the simulation, while the latter suffers from poor convergence properties or even

stagnation in the early stage of the simulation. After a certain attack time both quasi-Newton methods provide reliable and fast convergence with comparable rates. However, the improved performance of the IQN-IMVJ method and its better robustness properties come at the cost of a more costly computation and higher memory-requirements, as an explicit representation of the Jacobian matrix needs to be available.

Obviously, if the solver complexity highly dominates the overall simulation and the coupling complexity is diminishable, the IQN-IMVJ post-processing method is to be preferred either way, i. e., there always exists a break-even-point with respect to the complexity ratio of field solvers versus coupling algorithm at which the IQN-IMVJ clearly pays off. On the other hand, the IQN-IMVJ method is impractical if the number of coupling variables is very high. This situation mainly occurs for volume-coupled problems such as flow transport problems.

8.2 Outlook and Ideas for Further Research

The findings and contributions within this work enable a robust and efficient simulation of a whole range of strongly coupled multi-physics simulations and provide a deep insight into the characteristics and properties of various coupling schemes. Nonetheless, there is space for further improvements as well as a lot of possibilities and ideas for more in-depth research. This section serves as a pool of ideas for future extensions and research directions.

As preCICE supplies the possibility of massive parallel execution of the single-physics solvers, the coupling algorithm itself appears to become a major bottleneck in the simulation. Thus, it would be desirable to parallelize the considered coupling schemes. In particular, this concerns the efficient parallel computation of the Jacobian updating step, i. e., implementing an efficient parallel QR-decomposition (or updating, respectively) as well as parallel matrix operations. A massive parallel execution of the coupling algorithm would in fact be especially beneficial for the multi-vector quasi-Newton variant such that its higher computational effort would pass unnoticed.

Furthermore, an all-encompassing quantitative and qualitative comparison of the IQN-ILS and the IQN-IMVJ method as well as reliable statements for the usage of the respective method given certain assumptions and scenario characteristics, would be desirable. To that end, an accurate time and complexity analysis for all coupling algorithms is indispensable. This includes cost-effectiveness considerations for the IQN-ILS(R) method, the IQN-IMVJ multi-vector method and its limited-information alternatives IQN-IMVJ-limited(L). As a matter of course this requires highly optimized implementations for all alternatives. For weakly-coupled problems with less instabilities, a more in-depth cost-benefit analysis for the limited-information methods seems to be of special interest.

Aside from the herein considered surface-coupled FSI applications, there are plenty of coupled problems that still await to be coupled using the presented coupling schemes. To mention just a few examples, acoustic fluid coupled problems considering the near field or pure acoustics problems that include the wave equation and far field simulation, are interesting examples for surface coupled problems. Here, a bi-directional coupling is required in case of

reflections of acoustic waves going back in the acoustic fluid domain. Considering volume coupled problems, flow with transport, chemical reactions an the simulation of electric fields are worthwhile applications.

Bibliography

[1] H. Altenbach, *Kontinuumsmechanik: Einführung in die materialunabhängigen und materialabhängigen Gleichungen*, SpringerLink : Bücher, Springer Berlin Heidelberg, 2012. (Cited on page 8)

[2] D. G. Anderson, *Iterative procedures for nonlinear integral equations*, Journal of the ACM (JACM), 12 (1965), pp. 547–560. (Cited on pages 29, 31 and 47)

[3] I. Babuška and T. Strouboulis, *The finite element method and its reliability*, Oxford university press, 2001. (Cited on page 11)

[4] S. Badia, A. Quaini, and A. Quarteroni, *Splitting methods based on algebraic factorization for fluid-structure interaction*, SIAM Journal on Scientific Computing, 30 (2008), pp. 1778–1805. (Cited on page 14)

[5] K.-J. Bathe and G. A. Ledezma, *Benchmark problems for incompressible fluid flows with structural interactions*, Computers & structures, 85 (2007), pp. 628–644. (Cited on pages 71 and 75)

[6] Y. Bazilevs, K. Takizawa, and T. E. Tezduyar, *Computational fluid-structure interaction: methods and applications*, John Wiley & Sons, 2013. (Cited on page 1)

[7] D. S. Blom, A. Van Zuylen, and H. Bijl, *Acceleration of strongly coupled fluid-structure interaction with manifold mapping*, in Proceedings-WCCM XI: 11th World Congress on Computational Mechanics; ECCM V: 5th European Conference on Computational Mechanics; ECFD VI: 6th European Conference on Computational Fluid Dynamics, Barcelona, Spain, 20-25 July 2014, CIMNE, 2014. (Cited on pages 18, 19 and 28)

[8] A. Bogaers, S. Kok, B. Reddy, and T. Franz, *Quasi-newton methods for implicit black-box fsi coupling*, Computer Methods in Applied Mechanics and Engineering, 279 (2014), pp. 113–132. (Cited on pages 25, 39, 47, 63, 71 and 75)

[9] L. Brekhovskikh and V. Goncharov, *Mechanics of continua and wave dynamics*, Springer Verlag Springer Series on Wave Phenomena, 1 (1985). (Cited on page 8)

[10] C. G. Broyden, *A class of methods for solving nonlinear simultaneous equations*, Mathematics of computation, (1965), pp. 577–593. (Cited on pages 27, 28 and 38)

[11] H.-J. Bungartz, F. Lindner, M. Mehl, and B. Uekermann, *A plug-and-play coupling approach for parallel multi-field simulations*, Computational Mechanics, (2014), pp. 1–11. (Cited on page 20)

[12] P. Causin, J. Gerbeau, and F. Nobile, *Added-mass effect in the design of partitioned algorithms for fluid–structure problems*, Computer Methods in Applied Mechanics and Engineering, 194 (2005), pp. 4506 – 4527. (Cited on pages 1, 17 and 59)

[13] L. Cavagna, G. Quaranta, and P. Mantegazza, *Application of navier–stokes simulations for aeroelastic stability assessment in transonic regime*, Computers & Structures, 85 (2007), pp. 818–832. (Cited on page 1)

[14] J. W. Daniel, W. B. Gragg, L. Kaufman, and G. Stewart, *Reorthogonalization and stable algorithms for updating the gram-schmidt qr factorization*, Mathematics of Computation, 30 (1976), pp. 772–795. (Cited on page 55)

[15] M. Darwish, I. Sraj, and F. Moukalled, *A coupled finite volume solver for the solution of incompressible flows on unstructured grids*, Journal of Computational Physics, 228 (2009), pp. 180–201. (Cited on page 51)

[16] A. de Boer, A. van Zuijlen, and H. Bijl, *Radial basis functions for interface interpolation and mesh deformation*, in Advanced Computational Methods in Science and Engineering, Springer, 2010, pp. 143–178. (Cited on page 51)

[17] J. Degroote, K.-J. Bathe, and J. Vierendeels, *Performance of a new partitioned procedure versus a monolithic procedure in fluid–structure interaction*, Computers & Structures, 87 (2009), pp. 793–801. (Cited on pages 2, 18, 28, 29, 31, 33, 47, 71 and 75)

[18] J. Degroote, P. Bruggeman, R. Haelterman, and J. Vierendeels, *Stability of a coupling technique for partitioned solvers in fsi applications*, Computers & Structures, 86 (2008), pp. 2224–2234. (Cited on pages 17, 57, 58, 59 and 67)

[19] J. Degroote, R. Haelterman, S. Annerel, P. Bruggeman, and J. Vierendeels, *Performance of partitioned procedures in fluid–structure interaction*, Computers & Structures, 88 (2010), pp. 446 – 457. (Cited on pages 19 and 43)

[20] N. Diniz dos Santos, J. Gerbeau, and J. Bourgat, *Partitioned fsi strategy for simulations of a thin elastic valve*, in ECCOMAS CFD 2006: Proceedings of the European Conference on Computational Fluid Dynamics, Egmond aan Zee, The Netherlands, September 5-8, 2006, Delft University of Technology; European Community on Computational Methods in Applied Sciences (ECCOMAS), 2006. (Cited on page 1)

[21] K. Dumont, J. Vierendeels, R. Kaminsky, G. Van Nooten, P. Verdonck, and D. Bluestein, *Comparison of the hemodynamic and thrombogenic performance of two bileaflet mechanical heart valves using a cfd/fsi model*, Journal of Biomechanical Engineering, 129 (2007), pp. 558–565. (Cited on page 1)

[22] V. Eyert, *A comparative study on methods for convergence acceleration of iterative vector sequences*, Journal of Computational Physics, 124 (1996), pp. 271–285. (Cited on pages 27, 29 and 31)

[23] H.-r. Fang and Y. Saad, *Two classes of multisecant methods for nonlinear acceleration*, Numerical Linear Algebra with Applications, 16 (2009), pp. 197–221. (Cited on pages 27, 29, 31, 39, 41, 42, 47 and 55)

[24] C. Farhat, *Cfd-based nonlinear computational aeroelasticity*, Encyclopedia of computational mechanics, (2004). (Cited on page 1)

[25] C. Farhat, P. Geuzaine, and G. Brown, *Application of a three-field nonlinear fluid–structure formulation to the prediction of the aeroelastic parameters of an f-16 fighter*, Computers & Fluids, 32 (2003), pp. 3 – 29. (Cited on page 1)

[26] C. Farhat and M. Lesoinne, *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 (2000), pp. 499–515. (Cited on pages 2 and 19)

[27] C. Farhat, K. G. Van der Zee, and P. Geuzaine, *Provably second-order time-accurate loosely-coupled solution algorithms for transient nonlinear computational aeroelasticity*, Computer methods in applied mechanics and engineering, 195 (2006), pp. 1973–2001. (Cited on page 1)

[28] C. A. Felippa, K. Park, and C. Farhat, *Partitioned analysis of coupled mechanical systems*, Computer methods in applied mechanics and engineering, 190 (2001), pp. 3247–3270. (Cited on pages 16 and 19)

[29] C. Fletcher, *Computational techniques for fluid dynamics 1*, vol. 1, Springer Science & Business Media, 1991. (Cited on page 6)

[30] C. Förster, W. A. Wall, and E. Ramm, *Artificial added mass instabilities in sequential staggered coupling of nonlinear structures and incompressible viscous flows*, Computer methods in applied mechanics and engineering, 196 (2007), pp. 1278–1293. (Cited on page 17)

[31] B. Gatzhammer, *Efficient and flexible partitioned simulation of fluid-structure interactions*, PhD thesis, Technische Universität München, Inst. für Informatik, 2015. (Cited on pages 5, 6, 8, 11, 15, 16, 17, 18, 43, 49, 57, 58, 67 and 83)

[32] M. Griebel, T. Dornseifer, and T. Neunhoeffer, *Numerical Simulation in Fluid Dynamics, a Practical Introduction*, SIAM, Philadelphia, 1998. (Cited on pages 7 and 11)

[33] R. Haelterman, J. Degroote, D. V. Heule, and J. A. Vierendeels, *The quasi-newton least squares method: A new and fast secant method analyzed for linear systems*, SIAM J. Numerical Analysis, (2009), pp. 2347–2368. (Cited on pages 29, 36, 37 and 47)

[34] E. Hairer, S. Nørsett, and G. Wanner, *Solving Ordinary Differential Equations I: Nonstiff Problems*, Solving Ordinary Differential Equations, Springer, 1993. (Cited on page 12)

[35] ——, *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*, Lecture Notes in Economic and Mathematical Systems, Springer, 1993. (Cited on page 12)

[36] S. Hammarling and C. Lucas, *Updating the qr factorization and the least squares problem*. (Cited on page 55)

[37] J.-P. P. R.-F. J. Donea, A. Hurta, *Arbitrary lagrangian-eulerian methods*, Encyclopedia of Computational Mechanics, (2004), pp. 1:413–437. (Cited on page 5)

[38] D. Johnson, *Modified broyden's method for accelerating convergence in self-consistent calculations*, Physical Review B, 38 (1988), p. 12807. (Cited on page 27)

[39] E. Kvaalen, *A faster broyden method*, BIT Numerical Mathematics, 31 (1991), pp. 369–372. (Cited on pages 28 and 38)

[40] P. Le Tallec, J.-F. Gerbeau, P. Hauret, and M. Vidrascu, *Fluid structure interaction problems in large deformation*, Comptes Rendus Mecanique, 333 (2005), pp. 910–922. (Cited on page 1)

[41] M. Lesoinne and C. Farhat, *Higher-order subiteration-free staggered algorithm for nonlinear transient aeroelastic problems*, AIAA journal, 36 (1998), pp. 1754–1757. (Cited on page 16)

[42] R. J. LeVeque, *Finite volume methods for hyperbolic problems*, vol. 31, Cambridge university press, 2002. (Cited on page 11)

[43] G. Link, M. Kaltenbacher, M. Breuer, and M. Döllinger, *A 2d finite-element scheme for fluid–solid–acoustic interactions and its application to human phonation*, Computer Methods in Applied Mechanics and Engineering, 198 (2009), pp. 3321–3334. (Cited on page 1)

[44] J. M. Martínez, *A family of quasi-newton methods for nonlinear equations with direct secant updates of matrix factorizations*, SIAM Journal on Numerical Analysis, 27 (1990), pp. 1034–1049. (Cited on page 27)

[45] J. M. Martinez, *Practical quasi-newton methods for solving nonlinear systems*, Journal of Computational and Applied Mathematics, 124 (2000), pp. 97–121. (Cited on page 27)

[46] E. Oñate, S. R. Idelsohn, F. Del Pin, and R. Aubry, *The particle finite element method—an overview*, International Journal of Computational Methods, 1 (2004), pp. 267–307. (Cited on page 11)

[47] S. Piperno and C. Farhat, *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 (2001), pp. 3147–3170. (Cited on page 16)

[48] S. Piperno, C. Farhat, and B. Larrouturou, *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 (1995), pp. 79–112. (Cited on page 16)

[49] T. Richter, *Goal-oriented error estimation for fluid–structure interaction problems*, Computer Methods in Applied Mechanics and Engineering, 223 (2012), pp. 28–42. (Cited on page 71)

[50] K. Riemslagh, J. Vierendeels, and E. Dick, *Coupling of a navier-stokes solver and an elastic boundary solver for unsteady problems*, COMPUTATIONAL FLUID DYNAMICS '98, VOL 1, PARTS 1 AND 2, (1998), pp. 1040–1045. (Cited on page 1)

[51] H. Schwarz, *Gesammelte Mathematische Abhandlungen*, 1890. (Cited on page 23)

[52] M. A. Schweitzer, *A Parallel Multilevel Partition of Unity Method for Elliptic Partial Differential Equations*, vol. 29 of Lecture Notes in Computational Science and Engineering, Springer, 2003. (Cited on page 11)

[53] B. Smith, P. Bjorstad, and W. Gropp, *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*, Cambridge University Press, 2004. (Cited on page 23)

[54] J. Thomas, *Numerical Partial Differential Equations: Finite Difference Methods*, no. Bd. 1 in Graduate Texts in Mathematics, Springer, 1995. (Cited on page 11)

[55] S. Turek and J. Hron, *Proposal for numerical benchmarking of fluid-structure interaction between an elastic object and laminar incompressible flow*, Springer, 2006. (Cited on pages 71 and 83)

[56] S. Turek, J. Hron, M. Madlik, M. Razzaq, H. Wobker, and J. Acker, *Numerical simulation and benchmarking of a monolithic multigrid solver for fluid-structure interaction problems with application to hemodynamics*, Springer, 2010. (Cited on page 13)

[57] B. Uekermann, H.-J. Bungartz, B. Gatzhammer, and M. Mehl, *A parallel, black-box coupling algorithm for fluid-structure interaction*, in Proceedings of 5th International Conference on Computational Methods for Coupled Problems in Science and Engineering, 2013. (Cited on pages 18, 20, 25, 27, 28, 29, 32, 47, 60, 61, 63, 65, 77 and 86)

[58] R. van Loon, P. D. Anderson, and F. N. van de Vosse, *A fluid–structure interaction method with solid-rigid contact for heart valve dynamics*, Journal of computational physics, 217 (2006), pp. 806–823. (Cited on page 1)

[59] J. Vierendeels, L. Lanoye, J. Degroote, and P. Verdonck, *Implicit coupling of partitioned fluid–structure interaction problems with reduced order models*, Computers & Structures, 85 (2007), pp. 970–976. (Cited on pages 1, 29, 37 and 47)

[60] H. F. Walker and P. Ni, *Anderson acceleration for fixed-point iterations*, SIAM Journal on Numerical Analysis, 49 (2011), pp. 1715–1735. (Cited on pages 29, 31, 37 and 47)

Declaration

I declare that this thesis is the solely effort of the author. I did not use any other sources and references than the listed ones. I have marked all contained direct or indirect statements from other sources as such. Neither this work nor significant parts of it were part of another review process. I did not publish this work partially or completely yet. The electronic copy is consistent with all submitted copies.

Ort, Datum, Unterschift