In this Ph.D. Thesis, multigrid-reduction-in-time (MGRIT) is considered as means to reduce the *time-to-solution* for numerical algorithms concerned with the solution of time-dependent partial differential equations (PDEs) arising in the field of fluid-structure interaction (FSI) modeling. As a parallel-in-time integration method, the MGRIT algorithm significantly increases the potential for parallel speedup by employing modern computer architectures, ranging from small-scale clusters to massively parallel high-performance computing platforms.
In this work, the MGRIT algorithm is considered as a true multilevel method that can exhibit optimal scaling. Convergence of MGRIT is studied for the solution of linear and nonlinear (systems of) PDEs: from single- to multiphysics applications relevant to FSI problems in two and three dimensions.
A multilevel convergence framework for MGRIT is derived that establishes *a priori* upper bounds and approximate convergence factors for a variety of cycling strategies (e.g., V- and F-cycles), relaxation schemes and parameter settings. The convergence framework is applied to a number of test problems relevant to FSI modeling, both linear and nonlinear as well as parabolic and hyperbolic in nature.
An MGRIT variant is further proposed that exploits the time-periodicity that is present in many biomedical engineering applications, e.g., cyclic blood flow in the human heart. The time-periodic MGRIT algorithm proves capable of consistently reducing the *time-to-solution* of an existing simulation model with significant observed speedups.

Andreas Hessenthaler

ISBN 978-3-946412-03-8

A. Hessenthaler

Parallel-in-Time Integration for FSI Problems

CBM-04 (2020)

Multilevel Convergence Analysis: Parallel-in-Time Integration for Fluid-Structure Interaction Problems with Applications in Cardiac Flow Modeling

Andreas Hessenthaler

vorgelegt an der
Universität Stuttgart

Multilevel Convergence Analysis: Parallel-in-Time Integration for Fluid-Structure Interaction Problems with Applications in Cardiac Flow Modeling

Von der Fakultät Bau- und Umweltingenieurwissenschaften
und dem Stuttgart Research Centre for Simulation Technology
der Universität Stuttgart
zur Erlangung der Würde eines Doktor-Ingenieurs (Dr.-Ing.)
genehmigte Abhandlung

Vorgelegt von

Andreas Manuel Hessenthaler, M.Sc.

aus Ostfildern

Hauptberichter:	Prof. Oliver Röhrle, Ph.D.
Mitberichter:	Prof. Dr. Matthias Bolten
	Assoc. Prof. David Nordsletten, DPhil

Tag der mündlichen Prüfung: 7. Februar 2020

Institut für Modellierung und Simulation Biomechanischer Systeme
der Universität Stuttgart
2020

Report No.: CBM-04 (2020)
Institute for Modelling and Simulation of Biomechanical Systems
Continuum Biomechanics and Mechanobiology Research Group
University of Stuttgart, Germany, 2020

Editor:
Prof. O. Röhrle, PhD

ISBN 978-3-946412-03-8
(D 93 – Dissertation, Universität Stuttgart)

Declaration of Originality

I hereby declare that this thesis and the work reported herein was composed by and originated entirely from me. Information derived from the published and unpublished work of others has been acknowledged in the text and references are given in the list of sources.

Stuttgart, September 2019

_____

Andreas Manuel Hessenthaler

Acknowledgements

The work presented in this Ph.D. Thesis was carried out during my time at the *Institute for Modelling and Simulation of Biomechanical Systems* and the *Institute of Applied Mechanics (CE)* at the *University of Stuttgart*, and during research stays and internships at *King's College London*, UK, at the *Lawrence Livermore National Laboratory*, CA, USA, and at the *University of Michigan*, MI, USA, from September 2015 to September 2019.

First and foremost, I would like to gratefully acknowledge the supervision, guidance and advice from my Ph.D. Advisor Oliver Röhrle, and my Ph.D. Co-Advisor David Nordsletten. I would not have been able to achieve the goals set for my Ph.D. project without their constant support, without the numerous discussions, without them challenging and encouraging my creative potential and without them giving me the freedom to do research visits and internships during my Ph.D., as well as to explore new, sometimes controversial, and fun ideas.

I would further like to thank my collaborators and supervisors Robert D. Falgout, Jacob B. Schroder and Ben S. Southworth at the *Lawrence Livermore National Laboratory*, at the *University of New Mexico* and at *CU Boulder*, who have inspired and helped me to always think one step further and who have contributed to the success of my thesis.

A shout-out goes out to everyone that I have met during my time as a Ph.D. student and to all those who have sticked with me during exciting, frustrating, challenging, fun and rewarding times in Stuttgart, London, Livermore, Auckland, Uppsala, Hildesheim, Taiwan, Maastricht, and many other places: my workmates, fellow Ph.D. students, flatmates and dear friends. Thanks for all the fun hours spent at work, hiking, camping, climbing, skiing, surfing, photographing, dining, watching sports and theatre plays, listening to orchestras and poetry, nerding out about technology, science and the arts, watching our nephews, nieces, (god)sons and (god)daughters grow up, laughing, loving and caring, and everyone and everything else that makes life worth living!

I could not finish without expressing my sincere gratitude and appreciation to my parents and family for their unlimited support, love and understanding. You have always encouraged me to follow my own path that has led me to finding my happy places with good friends all around me.

Stuttgart, September 2019 Andreas Manuel Hessenthaler

Contents

Abstract

In this work, parallel-in-time integration methods are considered as means to reduce the *time-to-solution* for numerical algorithms concerned with the solution of time-dependent partial differential equations (PDEs) arising in the field of fluid-structure interaction (FSI) modeling.

Parallel-in-time integration methods introduce parallelism in the temporal domain and complement parallelization techniques that are applied to the spatial domain, such as domain decomposition methods. This significantly increases the potential for parallel speedup by employing modern computer architectures, ranging from small-scale clusters to massively parallel high-performance computing platforms.

In this work, the multigrid-reduction-in-time (MGRIT) algorithm is considered as a true multilevel method that can exhibit optimal scaling. MGRIT is based on multigrid reduction and takes inspiration from well-established spatial multigrid methods. Convergence of MGRIT is studied for the solution of linear and nonlinear (systems of) PDEs: from single- to multiphysics applications relevant to FSI problems in two and three dimensions.

A multilevel convergence framework is derived that generalizes and extends previous two-level theory for linear PDEs. The convergence framework establishes *a priori* upper bounds and approximate convergence factors for a variety of cycling strategies (e.g., V- and F-cycles), relaxation schemes and parameter settings. The derived upper bounds range from sharp but computationally expensive numerical upper bounds to cheap analytic formulae bounding the worst-case convergence of MGRIT. Furthermore, approximate convergence factors are proposed that capture MGRIT convergence with reasonable quality for cases, when analytic formulae are not available and the computational cost of numerical bounds is prohibitive.

The multilevel convergence framework is applied to a number of test problems relevant to FSI modeling: solving time-dependent linear PDEs, such as the anisotropic diffusion equation, the wave equation, the linear elasticity equation and the Stokes equation. Extensions of the theory to the nonlinear PDE case are explored as well.

This work further proposes an MGRIT variant that exploits the time-periodicity that is present in many biomedical engineering applications, e.g., cyclic blood flow in the human heart. The time-periodic MGRIT algorithm is assessed for simple flow problems, for a novel class of analytic solutions for linear and nonlinear FSI, as well as nonlinear flow in a patient-specific model of the left atrium and left ventricle. For the range of considered test problems, the time-periodic MGRIT algorithm proves capable of consistently reducing the *time-to-solution* of an existing simulation model with significant observed speedups.

Deutsche Zusammenfassung

In dieser Arbeit werden parallele Zeitintegrationsmethoden zur Reduktion von Lösungszeiten numerischer Algorithmen betrachtet. Dies gilt insbesondere für numerische Algorithmen zur Lösung von zeitabhängigen partiellen Differentialgleichungen im Bereich der Modellierung von Fluid-Struktur-Interaktionsproblemen.

Parallele Zeitintegrationsmethoden ermöglichen Parallelität in der Zeitvariablen und ergänzen Parallelisierungstechniken, die traditionellerweise auf die räumliche Komponente von partiellen Differentialgleichungen angewandt werden, wie beispielsweise Gebietszerlegungsmethoden. Durch die Ausnutzung modernster Computerarchitekturen, von kleinen Rechenclustern bis massiv parallelen Höchstleistungsrechnern, erhöht sich das Potential zur Beschleunigung von Lösungsalgorithmen substantiell.

In dieser Arbeit wird der Zeit-Mehrgitterreduktionsalgorithmus MGRIT betrachtet, für den für bestimmte Anwendungen bereits eine optimale parallele Skalierfähigkeit nachgewiesen wurde. MGRIT ist eine Mehrgitterreduktionsmethode und lehnt sich an die etablierten und bereits vielseitig eingesetzten, räumlichen Mehrgitteralgorithmen an. Im Rahmen dieser Dissertation wird die Konvergenz des Zeit-Mehrgitteralgorithmus für die Lösung von linearen und nichtlinearen (Systemen von) partiellen Differentialgleichungen untersucht: von einfachen Problemstellungen hin zu gekoppelten Multiphysiksystemen im Bereich der Fluid-Struktur-Interaktionsmodellierung, sowohl im (vereinfachten) zweidimensionalen Raum, als auch für realistische dreidimensionale Geometrien.

Eine Mehrgitterkonvergenztheorie wird hergeleitet und vorgestellt, die bereits veröffentlichte Zweigittertheorie für lineare Differentialgleichungen verallgemeinert und erweitert. Die postulierte Konvergenztheorie umfasst obere Schranken und Abschätzungen von Konvergenzraten des MGRIT-Algorithmus, die bereits *a priori* ausgewertet werden können und daher zur informierten Parameterwahl vor der eigentlichen Lösung der partiellen Differentialgleichung dienen. Die theoretischen Ergebnisse schließen verschiedene MGRIT-Varianten mit ein: es können sowohl V-, als auch F-Zyklen und unterschiedliche Relaxierungsstrategien beziehungsweise MGRIT-Parameter betrachtet werden. In diesem Sinne ist die Konvergenztheorie sehr allgemein gehalten und erlaubt eine umfassende Voruntersuchung und Einschätzung der zu erwartenden MGRIT-Performance. Die theoretischen Ergebnisse umfassen sowohl scharfe obere Schranken an die Konvergenzrate, die numerisch und mit einigem Berechnungsaufwand ausgewertet werden, als auch analytische obere Schranken, die einer einfachen Funktionsauswertung bedürfen. Desweiteren werden analytische Ausdrücke postuliert, die Konvergenzraten approximieren. Diese approximierenden Ausdrücke können in solchen Fällen eingesetzt werden, in denen der Berechnungsaufwand numerischer oberer Schranken zu hoch wäre, aber keine alternative analytische obere Schranke verfügbar ist.

Die Mehrgitterkonvergenztheorie wird für eine große Anzahl von Testproblemen angewandt, die für Fluid-Struktur-Interaktionsmodelle relevant sind: beispielsweise für die Lösung zeitabhängiger partieller Differentialgleichungen, wie die anisotrope Diffusions-

gleichung, die Wellengleichung, die lineare Elastizitätsgleichung und die Stokes-Gleichung für lineare Strömungen. Ideen zur Erweiterungen der Theorie für den Fall nichtlinearer partieller Differentialgleichungen werden ebenfalls ergründet und evaluiert.

Desweiteren wird in der vorliegenden Dissertationsschrift eine neue MGRIT-Variante vorgeschlagen, die die Zeitperiodizität von Systemen im Bereich der biomedizinischen Ingenieursanwendungen gezielt ausnutzen kann, wie zum Beispiel bei der Untersuchung zyklischer Blutströmungen im menschlischen Herzen. Der zeitperiodische MGRIT-Algorithmus wird dann für eine Reihe von Problemstellungen evaluiert: für einfache Strömungsprobleme, für eine neue Klasse analytischer Lösungen im Bereich Fluid-Struktur-Interaktion und für nichtlineare Strömungen innerhalb einer patientenspezifischen Geometrie des linken Vorhofs und der linken Herzkammer. Für alle betrachteten Anwendungsfälle liefert der neue zeitperiodische MGRIT-Algorithmus konstant gute Laufzeitreduktionen existierender numerischer Modelle mit übergreifend guten Speedup-Faktoren.

Nomenclature

Subscript and superscript conventions

d	dimensional index
i	iteration index
k	spatial mode index / eigenvalue index
ℓ	level index
n	time index

Sets	**Description**
$\mathbb{Z}$	set of integers
$\mathbb{Z}_0^+$	set of non-negative integers
$\mathbb{Z}^+$	set of positive integers
$\mathbb{Q}$	set of rational numbers
$\mathbb{R}$	set of real numbers
$\mathbb{C}$	set of complex numbers

Symbols	**Description**
m_ℓ	temporal coarsening factor from level ℓ to level $\ell + 1$
n_ℓ	number of time grid levels
N_ℓ	number of points on time grid level ℓ
N_x	number of spatial degrees-of-freedom
$\mathcal{E}$	error propagation operator (V-cycle)
$\mathcal{F}$	error propagation operator (F-cycle)
$\mathcal{R}$	residual propagation operator
$\kappa(\boldsymbol{A})$	condition number of $\boldsymbol{A}$
$\imath$	denotes imaginary part of a complex number, i.e. $\imath = \sqrt{-1}$

Acronyms	**Description**
1D	one-dimensional
2D	two-dimensional
3D	three-dimensional
ALE	arbitrary Lagrangian-Eulerian

Acronyms	Description
CT	computed tomography
DOF	degree-of-freedom
DOFs	degrees-of-freedom
EDV	end-diastolic volume
ESV	end-systolic volume
FSI	fluid-structure interaction
FMG	full multigrid
LVEF	left ventricular ejection fraction
MGRIT	multigrid-reduction-in-time
MRI	magnetic resonance imaging
MV	mitral valve
ODE	ordinary differential equation
ODEs	ordinary differential equations
PDE	partial differential equation
PDEs	partial differential equations
PinT	parallel-in-time
PITA	parallel implicit time-integrator
SPD	symmetric positive-definite
SV	stroke volume

1 Introduction

1.1 Fluid-structure interaction and cardiac flow problems

"Mathematical modeling and numerical simulation have become important tools in the investigation of complex multiphysics phenomena [86] including the interaction between fluids and solids [4, 9, 17, 103]. In the field of biomedical engineering, fluid-structure interaction (FSI) modeling is playing an increasingly important role due to the coupling of fluid flow and tissue mechanics vital to many physical phenomena. Use of FSI models for the assessment of medical devices as well as clinical evaluation [102, 104, 128] is becoming increasingly common. *In silico* testing of devices using FSI models can help to expedite and augment preproduction development as well as assist in understanding the implications of an implant and its interaction in the human body. Moreover, in the domain of diagnostics and therapy planning, patient-specific models of the cardiovascular system [1, 132, 141, 143] are actively pursued, with the vision of eventually providing clinicians with guidance on treatment."[1]

A diseased heart may be affected from an asynchronous contraction of the tissue. This can be caused, for example, by scarred tissue in infarct patients that alters the propagation of electrical waves, and thus, affects the activation of the contraction of the heart, leading to impaired cardiac output. A typical model of such a diseased heart may involve the interaction of blood flow in the atria and ventricles of the human heart with heart valves, the active and passive behavior of the surrounding tissue, the electophysiology (possibly including chemical processes at the cellular level), as well as other relevant phenomena. While many macroscopic quantities can be easily assessed through medical imaging techniques, e.g., computed tomography (CT) or magnetic resonance imaging (MRI), *in silico* models can fill the gaps when such data cannot be recorded, the effect of therapy needs to be predicted (e.g., for cardiac resynchronization therapy planning) or if the imaging data lack spatiotemporal resolution simply because the patient cannot stay in an MRI scanner long enough.

Depending on the level of complexity and detail of the *in silico* model, the computational cost may be enormous. Processes and phenomena need to be resolved accurately in space and time. Oftentimes, models of phenomena in the human body show a naturally occuring time-periodic behavior. That is, processes or states occur repeatedly with a given period length T. In general, to reach such a *periodic steady-state* for an *in silico* model, multiple such periods or cycles need to be simulated, which increases the computational cost. Thus, the *time-to-solution* scales with the number of simulated cycles. In this work, parallel-in-time (PinT) integration is assessed as a means to reduce the time-to-solution, and new ways to compute a periodic steady-state more efficiently are introduced.

In the following sections, the concepts of numerical solvers and parallelization techniques are explained in more detail to develop an intuition for the multigrid-reduction-in-

[1] Paragraph from [64].

time (MGRIT) algorithm. MGRIT, which is introduced in Chapter 3, will be the main focus of this work to develop a framework that enables the acceleration of numerical simulations of FSI models and cardiac flow problems.

1.2 Solvers

In many cases, numerical models (such as those described in Section 1.1) result in the task of solving a system of linear equations,

$$\boldsymbol{A}\boldsymbol{u} = \boldsymbol{b}, \tag{1.1}$$

with solution $\boldsymbol{u}$ and right-hand-side $\boldsymbol{b}$. Depending on the properties of the system matrix $\boldsymbol{A}$, the choice of an optimal solution strategy may vary. A broad classification of solution methods is given as *direct* and *iterative* methods. Direct methods solve (1.1) exactly (up to rounding errors), whereas iterative methods compute successively improved approximations $\boldsymbol{u}_i$ (iteration index i) up to a given tolerance. For example, as a *convergence criterion* one may require that the ℓ^2-norm of the residual at iteration i,

$$\boldsymbol{r}_i = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{u}_i, \tag{1.2}$$

is less than a given tolerance,

$$\|\boldsymbol{r}_i\|_2 < \text{tol} \qquad \text{for some } i, \tag{1.3}$$

where the ℓ^2-norm is defined as,

$$\|\boldsymbol{r}_i\|_2 = \sqrt{\sum_k [\boldsymbol{r}_i]_k^2}, \tag{1.4}$$

and $[\boldsymbol{r}_i]_k$ refers to the k^{th} component of the residual at iteration i.

In the following, examples for direct and iterative solvers are discussed[2] to develop an intuition for the key differences and to set the stage for the discussion of parallelization techniques in Section 1.3, and in particular for the multigrid-reduction-in-time algorithm as the main focus of this work. To simplify the discussion, consider:

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & & a_{2N} \\ \vdots & & \ddots & \vdots \\ a_{N1} & a_{N2} & \cdots & a_{NN} \end{bmatrix} \in \mathbb{R}^{N\times N}, \qquad \boldsymbol{u} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_N \end{bmatrix} \in \mathbb{R}^N, \qquad \boldsymbol{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_N \end{bmatrix} \in \mathbb{R}^N, \tag{1.5}$$

with $N \in \mathbb{Z}^+$. Further, assume that,

$$a_{ij} \neq 0 \quad \text{for } i,j \in \{1,2,\ldots,N\}, \tag{1.6}$$

and that $\boldsymbol{A}$ has full rank, is symmetric, i.e. $\boldsymbol{A} = \boldsymbol{A}^T$, and positive-definite.

[2] All solvers discussed in Section 1.2 were implemented in Python [151] for demonstration purposes [58].

Definition 1 (Positive-definiteness)**.** A real matrix $\boldsymbol{A} \in \mathbb{R}^{N\times N}$ with $N \in \mathbb{Z}^+$ is called *positive-definite* [81], if

$$\boldsymbol{z}^T\boldsymbol{A}\boldsymbol{z} > 0, \tag{1.7}$$

for all nonzero $\boldsymbol{z} \in \mathbb{R}^N$.

We further use the notion of *time complexity* and *computational complexity* [11, 88] interchangeably, as defined in the following.

Definition 2 (Time complexity)**.** Consider an input of size N, fixed constants $c, d > 0$ and let $f(N)$ be the time required by a given algorithm. Then, the *time complexity* of the algorithm is defined as $f(N) = O(g(N))$, that is,

$$0 \leq f(N) \leq cg(N) \qquad \forall\ N \geq d. \tag{1.8}$$

In Definition 2, $g(N)$ describes the growth of the number of elementary operations in the algorithm. For example, $O(1)$ refers to *constant* complexity, $O(N)$ refers to *linear* complexity, $O(N^2)$ refers to *quadratic* complexity, etc. A similar definition can be made for the *memory complexity* of an algorithm.

1.2.1 Direct solvers

Direct methods solve Equation (1.1) exactly. For example, they find the solution $\boldsymbol{u}$ as,

$$\boldsymbol{u} = \boldsymbol{A}^{-1}\boldsymbol{b}, \tag{1.9}$$

where $\boldsymbol{A}^{-1}$ is the inverse of $\boldsymbol{A}$. Forming the inverse explicitly and computing the matrix-vector product $\boldsymbol{A}^{-1}\boldsymbol{b}$, however, is computationally expensive.

1.2.1.1 Gaussian elimination

An alternative to forming $\boldsymbol{A}^{-1}$ explicitly is Gaussian elimination [3, pp. 508–515], which is *the* textbook algorithm for computing the solution $\boldsymbol{u}$ of (1.1). Through linear combinations of the rows of $\boldsymbol{A}$ and manipulation of the right-hand-side $\boldsymbol{b}$, the following steps are performed:

1. Compute upper-diagonal form $\tilde{\boldsymbol{A}}$ with new right-hand-side $\tilde{\boldsymbol{b}}$
2. Compute diagonal form of $\bar{\boldsymbol{A}}$ with new right-hand-side $\bar{\boldsymbol{b}}$
3. Compute solution $\boldsymbol{u} = \bar{\boldsymbol{b}}/\operatorname{diag}(\bar{\boldsymbol{A}})$ (element-wise)

Gaussian elimination has a time complexity of $O(N^3)$, which is not optimal [140].

1.2.2 Iterative solvers

In general, direct solvers are computationally expensive. One typical remedy is the use of iterative solvers, that can often be much faster at the expense of solution accuracy.

1.2.2.1 Jacobi method

The Jacobi method [3, 79, 119] is a typical fixed-point method in the class of iterative methods and is based on the decomposition,

$$\boldsymbol{A} = \boldsymbol{D} + \boldsymbol{R}, \tag{1.10}$$

where $\boldsymbol{D} = \mathrm{diag}(\boldsymbol{A})$ is the diagonal of $\boldsymbol{A}$, and $\boldsymbol{R}$ is the remainder. Thus, rewriting Equation (1.1) yields,

$$\boldsymbol{A}\boldsymbol{u} = (\boldsymbol{D} + \boldsymbol{R})\,\boldsymbol{u} = \boldsymbol{b}. \tag{1.11}$$

The Jacobi method now selects an initial guess $\boldsymbol{u}_0$ of the solution and defines a fixed-point iteration by the following recurrence,

$$\boldsymbol{u}_i = -\boldsymbol{D}^{-1}\boldsymbol{R}\boldsymbol{u}_{i-1} + \boldsymbol{D}^{-1}\boldsymbol{b}, \qquad \text{for } i \in \mathbb{Z}^+, \tag{1.12}$$

with iteration matrix,

$$\boldsymbol{G} = -\boldsymbol{D}^{-1}\boldsymbol{R}. \tag{1.13}$$

A more general fixed-point iteration is given by,

$$\boldsymbol{u}_i = \left[(1-\omega)\boldsymbol{I} - \omega\boldsymbol{D}^{-1}\boldsymbol{R}\right]\boldsymbol{u}_{i-1} + \omega\boldsymbol{D}^{-1}\boldsymbol{b}, \qquad \text{for } i \in \mathbb{Z}^+, \tag{1.14}$$

and referred to as *weighted* Jacobi method, with the identity operator $\boldsymbol{I}$ and the iteration matrix,

$$\boldsymbol{G}(\omega) = \left[(1-\omega)\boldsymbol{I} - \omega\boldsymbol{D}^{-1}\boldsymbol{R}\right], \tag{1.15}$$

with a weigthing factor ω that can improve convergence if selected carefully.

An important question is *if* the iteration (1.14) converges (that is, $\boldsymbol{u}_i \to \boldsymbol{u}$ for $i \to \infty$) and *how quickly* it yields an approximation with sufficient accuracy. To this end, consider the error at iteration i,

$$\boldsymbol{e}_i = \boldsymbol{u} - \boldsymbol{u}_i. \tag{1.16}$$

If the iteration converges, then the solution $\boldsymbol{u}$ satisfies the fixed-point iteration [119],

$$\boldsymbol{u} = \boldsymbol{G}(\omega)\boldsymbol{u} + \omega\boldsymbol{D}^{-1}\boldsymbol{b}, \tag{1.17}$$

and we can write,

$$\boldsymbol{e}_i = \boldsymbol{u} - \boldsymbol{u}_i = \boldsymbol{G}(\omega)\,(\boldsymbol{u} - \boldsymbol{u}_{i-1}) = \ldots = \boldsymbol{G}^i\,(\boldsymbol{u} - \boldsymbol{u}_0) = \boldsymbol{G}^i\boldsymbol{e}_0 \tag{1.18}$$

As shown in [119], the iteration (1.14) converges for any initial guess $\boldsymbol{u}_0$ and right-hand-side $\boldsymbol{b}$ (and thus, $\boldsymbol{e}_i \to 0$ for $i \to \infty$) if the spectral radius of $\boldsymbol{G}(\omega)$ is less than unity, i.e. $\rho\left(\boldsymbol{G}(\omega)\right) < 1$. In fact, this basic observation provides key ingredients for developing convergence theory in later parts of this work (see Chapter 4).

Since it is typically difficult or impossible to precicely measure the error in practical applications, other measures of convergence have to be defined. It is, for example, possible

to measure the residual, and stopping criteria for iterative methods are often based on a residual tolerance, see Equation (1.3). For the approximation of the solution of the linear system of equations in (1.1) at iteration i, the relationship between error $\boldsymbol{e}_i$ and residual $\boldsymbol{r}_i$ is surprisingly simple,

$$\boldsymbol{A}\boldsymbol{u} = \boldsymbol{A}\left(\boldsymbol{u}_i + \boldsymbol{e}_i\right) = \boldsymbol{b} \qquad \Leftrightarrow \qquad \boldsymbol{A}\boldsymbol{e}_i = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{u}_i = \boldsymbol{r}_i. \tag{1.19}$$

To illustrate the notion of convergence further, consider the symmetric positive-definite (SPD) matrix $\boldsymbol{A}$ and right-hand-side $\boldsymbol{b}$ with dimension $N = 500$ and coefficients $a_{ij}, b_i \in (0, 1)$ for all i, j. Further, consider the weighted Jacobi method with weights $\omega = 1$ and $\omega = 0.807$, and corresponding spectral radii $\rho(\boldsymbol{G}(\omega = 1)) = 0.498$ and $\rho(\boldsymbol{G}(\omega = 0.807)) = 0.209$. Using (1.18) and (1.19), the residual reduction of the weighted Jacobi method can be predicted,

$$\begin{aligned} \|\boldsymbol{r}_i\|_2 = \|\boldsymbol{A}\boldsymbol{G}^i\boldsymbol{A}^{-1}\boldsymbol{r}_0\|_2 \leq \|\boldsymbol{G}\|_2^i \|\boldsymbol{r}_0\|_2 = \rho\left(\boldsymbol{G}\right)^i \|\boldsymbol{r}_0\|_2 \\ \Leftrightarrow \qquad \|\boldsymbol{r}_i\|_2 / \|\boldsymbol{r}_0\|_2 \leq \rho\left(\boldsymbol{G}\right)^i, \end{aligned} \tag{1.20}$$

and similarly for the error reduction,

$$\begin{aligned} \|\boldsymbol{e}_i\|_2 = \|\boldsymbol{G}^i\boldsymbol{e}_0\|_2 \leq \|\boldsymbol{G}\|_2^i \|\boldsymbol{e}_0\|_2 = \rho\left(\boldsymbol{G}\right)^i \|\boldsymbol{e}_0\|_2 \\ \Leftrightarrow \qquad \|\boldsymbol{e}_i\|_2 / \|\boldsymbol{e}_0\|_2 \leq \rho\left(\boldsymbol{G}\right)^i. \end{aligned} \tag{1.21}$$

Equation (1.20) and Equation (1.21) give *upper bounds* on the residual and error reduction of the weighted Jacobi method. Figure 1.1 highlights that the bound (1.20) is indeed sharp, i.e. observed residual reduction does not noticeably differ from the predicted upper bound. As such, the development of convergence theory for numerical algorithms is desirable, and has the potential to help understand an algorithm's strengths and weaknesses.

Lastly, it is noted that the complexity of the (weighted) Jacobi method is $O(N^2)$.

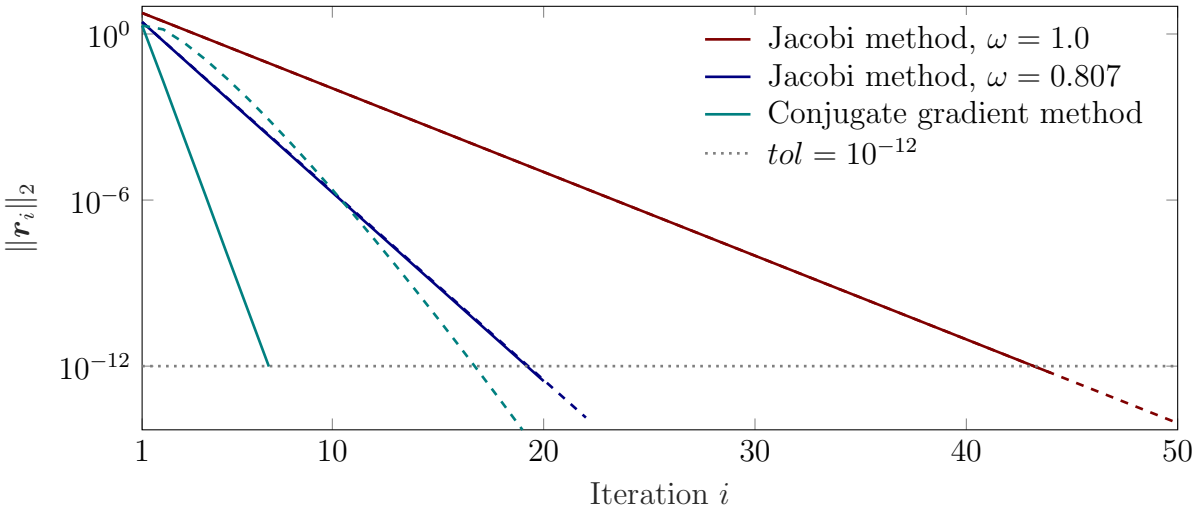


Figure 1.1: *Observed convergence for the Jacobi (solid, red), weighted Jacobi (solid, blue) and conjugate gradient (solid, green) method for an SPD matrix $\boldsymbol{A}$. The predicted residual reduction for each method is shown in dashed lines.*

1.2.2.2 Conjugate gradient method

In the previous section, it was observed that the convergence speed of the weighted Jacobi method depends on the weight ω. The conjugate gradient (CG) method is a particularly fast method for solving (1.1) when $\boldsymbol{A}$ is real and SPD. It is motivated from minimizing,

$$\frac{1}{2}\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}^T\boldsymbol{x} \qquad \text{for } \boldsymbol{x} \in \mathbb{R}^N, \tag{1.22}$$

which is equivalent to solving (1.1), see [3, Section 8.9]. The CG method follows a similar idea as the *method of steepest descent*, however, instead of finding the minimum along the direction of steepest descent (i.e. in the direction of the gradient of (1.22)) it selects search directions that are $\boldsymbol{A}$-orthogonal to each other. This yields a method that is guaranteed to converge in N iterations. Pseudo-code is provided in Algorithm 1.

Algorithm 1 Conjugate gradient method [3]

procedure CG
  Set initial guess: $\boldsymbol{u}_0 = \boldsymbol{0}$
  Set initial residual: $\boldsymbol{r}_0 = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{u}_0$
  Set initial search direction: $\boldsymbol{p}_0 = \boldsymbol{r}_0$
  Set $i = 0$
  while $i < N$ **do**
    Compute coefficient $\alpha_i = \boldsymbol{r}_i^T\boldsymbol{r}/(\boldsymbol{p}_i^T\boldsymbol{A}\boldsymbol{p}_i)$
    Compute approximation $\boldsymbol{u}_{i+1} = \boldsymbol{u}_i + \alpha_i\boldsymbol{p}_i$
    Compute residual $\boldsymbol{r}_{i+1} = \boldsymbol{r}_i - \alpha_i\boldsymbol{A}\boldsymbol{p}_i$
    if $\|\boldsymbol{r}_{i+1}\| < tol$ **then break**
    Compute coefficient $\beta_i = (\boldsymbol{r}_{i+1}^T\boldsymbol{r}_{i+1})/(\boldsymbol{r}_i^T\boldsymbol{r}_i)$
    Compute search direction $\boldsymbol{p}_{i+1} = \boldsymbol{r}_{i+1} + \beta_i\boldsymbol{p}_i$
    Set $i \leftarrow i + 1$

Residual reduction of the CG method can be bounded (see [82–84]) by,

$$\|\boldsymbol{r}_i\|_2/\|\boldsymbol{r}_0\|_2 \leq \left(\kappa(\boldsymbol{A})^{1/i} - 1\right)^{i/2}, \tag{1.23}$$

where $\kappa(\boldsymbol{A}) = \|\boldsymbol{A}^{-1}\|_2\|\boldsymbol{A}\|_2$ is the condition number of $\boldsymbol{A}$.

Applying the CG method to the system from Section 1.2.2.1 yields a much faster converging method, see Figure 1.1. For example, the CG method converges in 7 iterations compared to 44 and 20 for the weighted Jacobi method with weights $\omega = 1$ and $\omega = 0.807$, respectively. On the other hand, the upper bound on residual reduction for the CG method is less sharp than for the weighted Jacobi method. This emphasizes that convergence theory can help understand the worst-case performance of an algorithm, however, performance for practical applications may be (significantly) better. While one generally wants to design a method that converges in less iterations, it is important to consider that one also wants to minimize the *per-iteration cost*, and thus the *wall clock time* of the algorithm.

1.2.3 Best solver for a particular application

The *best* solver for a particular application may not be a good or even applicable solver for other applications. For example, the CG method (see Section 1.2.2.2) is a fast solver

for systems of linear equations, when the matrix is real and SPD; however, if the matrix is not symmetric or not positive-definite, the CG method does not converge. Important considerations in the selection of a suitable solver are matrix properties, such as symmetry, definiteness, condition number, sparsity, etc. These aspects may depend on the underlying PDE, the discretization strategy, the parallelization technique, and others.

1.3 Parallelization techniques

For many applications, the solution of a given problem is found by solving a PDE that is formulated over a given domain Ω. This domain is discretized, e.g., using finite elements [160], to create a discrete representation Ω^h of the domain (see Figure 1.2) and to solve the PDE at a finite number of (mesh) points. To find the solution of a given problem often requires the solution of a system of linear equations similar to Equation (1.1), where the dimension of $\boldsymbol{A}$ corresponds to the number of degrees-of-freedom (DOFs), i.e. the number of mesh points.

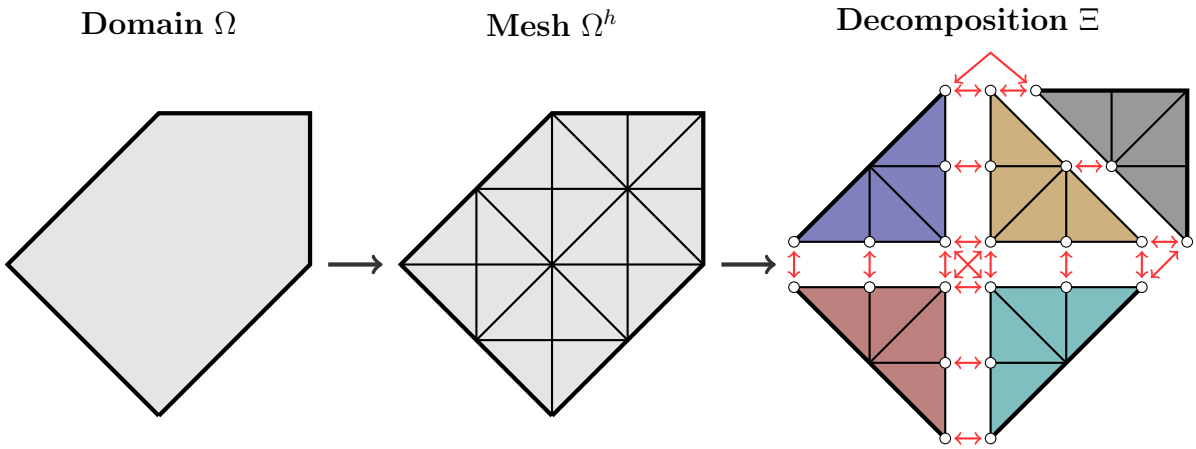


Figure 1.2: *Parallelization using mesh partitioning: A given domain Ω is triangulated to create a discrete representation of the domain (or mesh) Ω^h, which is further decomposed into discrete subdomains. Each of the subdomains of the decomposition Ξ can be handled on a (separate) processor and communication (red arrows) between points at the interface (circles) is employed to enforce contraints, e.g., continuity of the solution. Schematic inspired from [15, Figure 2.14].*

The wall clock time of an algorithm to solve (1.1) scales with the number of DOFs, see Section 1.2. There are a number of ways to reduce the wall clock time. For example:

1. **Hardware**: Reducing the wall clock time can be achieved by employing faster hardware to execute more operations per second. While this was a viable path until the mid-2000's, clock rates have become stagnant (see Figure 1.3) in recent years.

2. **Algorithm**: As discussed in Section 1.2, selecting a different solution strategy can yield better performance and shorter time-to-solution.

3. **Parallelism**: Around the time when clock rates stagnated, processing units with multiple logical cores have become available (see Figure 1.3). Two major categories

of solution strategies to exploit parallelism for time-dependent PDEs were established:

a) Spatial component: One of the most common parallelization techniques is the spatial domain decomposition (DD) method. The key idea is to divide the spatial domain into subdomains, distribute tasks for each subdomain over parallel processors and communicate information between neighboring subdomains. See Figure 1.2 and Section 1.3.1.

b) Temporal component: The classical approach for solving time-dependent PDEs is to solve a spatial problem at a number of discrete time steps sequentially, based on an initial state. This approach is referred to as *time-stepping*. In general, the solution at each time step is dependent on the solution at the previous step. Methods that explore parallelism in the temporal component are, e.g., Parareal [97], multigrid-reduction-in-time [35], parallel implicit time-integrator [22, 33], and others [39]. In the context of Section 1.2, sequential time-stepping can be interpreted as a direct solver, whereas time-parallel methods usually fall into the category of iterative solvers. See Section 1.3.2.

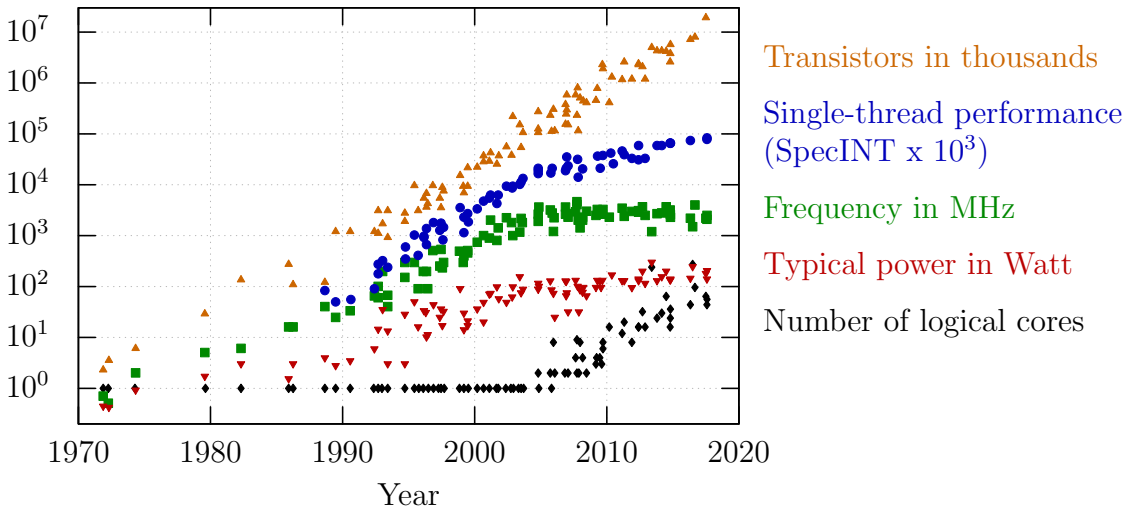


Figure 1.3: 42 *years of microprocessor trend data: The single-thread performance and processor frequency have plateaued, while the growth of the number of transistors still follows Moore's law. In contrast, the availability of multiple logical cores per processing unit has caused a paradigm shift in computing towards more parallelism in recent years. Data for years* $1970 - 2010$ *collected by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten. Data for years* $2010 - 2017$ *collected by K. Rupp. See [118].*

1.3.1 Spatial domain decomposition method

A natural way to reduce the wall clock time is to find a decomposition Ξ of the discrete domain Ω^h (see Figure 1.2), and thus, to distribute the computational cost of solving (1.1)

over a large number of parallel processors. In general, the solution over each subdomain depends on the solution of its neighbors and communication at the interface (see Figure 1.2) is required to enforce constraints, such as continuity of the solution at the interface.

In the context of solving (1.1), the decomposition in Figure 1.2 can be illustrated as follows:

$$\begin{bmatrix} \boldsymbol{A}_{rr} & \boldsymbol{A}_{rb} & \boldsymbol{A}_{rg} & \boldsymbol{A}_{ro} & \boldsymbol{A}_{rk} \\ \boldsymbol{A}_{br} & \boldsymbol{A}_{bb} & \boldsymbol{A}_{bg} & \boldsymbol{A}_{bo} & \boldsymbol{A}_{bk} \\ \boldsymbol{A}_{gr} & \boldsymbol{A}_{gb} & \boldsymbol{A}_{gg} & \boldsymbol{A}_{go} & \boldsymbol{A}_{gk} \\ \boldsymbol{A}_{or} & \boldsymbol{A}_{ob} & \boldsymbol{A}_{og} & \boldsymbol{A}_{oo} & \boldsymbol{A}_{rk} \\ \boldsymbol{A}_{kr} & \boldsymbol{A}_{kb} & \boldsymbol{A}_{kg} & \boldsymbol{A}_{ko} & \boldsymbol{A}_{kk} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_r \\ \boldsymbol{u}_b \\ \boldsymbol{u}_g \\ \boldsymbol{u}_o \\ \boldsymbol{u}_k \end{bmatrix} = \begin{bmatrix} \boldsymbol{b}_r \\ \boldsymbol{b}_b \\ \boldsymbol{b}_g \\ \boldsymbol{b}_o \\ \boldsymbol{b}_k \end{bmatrix}. \tag{1.24}$$

The diagonal blocks in Equation (1.24) correspond to DOFs in each of the subdomains of the decomposition in Figure 1.2 and the offdiagonal blocks are coupling blocks that correspond to shared DOFs at the interface. Note, for example, that the gray subdomain in Figure 1.2 is decoupled from the red subdomain and vice-versa, i.e. $\boldsymbol{A}_{kr} = \boldsymbol{0}$ and $\boldsymbol{A}_{rk} = \boldsymbol{0}$. This is an important observation because this means that there will be no communication between processors that compute the solution on the gray and red subdomains. In practical cases, the surface-to-volume ratio for each subdomain is an intuitive measure for the relative dominance of tasks that correspond to communication (surface) and computation (volume), and the amount of parallelism that can be exploited. For example, less communication means that the offdiagonal blocks in (1.24) are sparse.

In practical applications, finding a *good* decomposition of the spatial domain is often formulated as finding a partitioning of a nodal and / or dual graph, see Figure 1.4. The goal is to distribute the computational cost of each subproblem equally over the number of available processors (intuitively: same dimension of diagonal blocks in Equation (1.24)) while reducing the amount of communication required (intuitively: increasing the sparsity of the offdiagonal blocks in Equation (1.24)). The partitioning of the nodal or dual graph can be finetuned by introducing nodal, edge or element weights. A popular library to solve graph partitioning problems is, e.g., ParMETIS [85].

Depending on the dimension of $\boldsymbol{A}$, the amount of parallelism that can be exploited through mesh partitioning may be limited. Typically, there is an optimum number of processors at which the wall clock time is minimized and the achieved speedup is maximized. The speedup is defined as,

$$\text{Speedup} = \frac{\text{Wall clock time with 1 processor}}{\text{Wall clock time with } p \text{ processors}}. \tag{1.25}$$

If more than the optimum number of processors is used, however, the time-to-solution may be constant (or even grow) due to aspects like an increase in communication.

For example, consider solving a transient isotropic diffusion equation for 10 time steps (using backward Euler) on the unit square with a uniform finite element discretization (linear quadrilateral elements). Figure 1.5 illustrates how the use of more parallel processors initially yields a larger speedup. But, eventually, the observed speedup reaches a plateau. While more parallelism can be used for larger spatial problems (e.g., in Figure 1.5, the observed speedup is larger for larger numbers of DOFs), it may still be significantly less parallelism than the available hardware would enable. For example, as of June 14, 2019

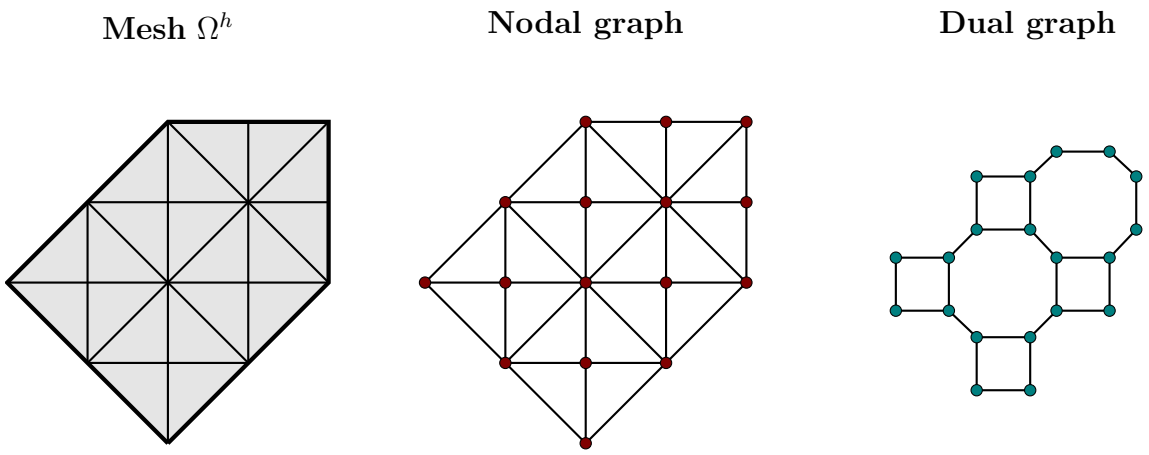


Figure 1.4: *To partition a mesh, nodal or dual graphs are constructed. The vertices of a nodal graph are comprised of the nodes of the mesh and the edges correspond to edges of the mesh. The vertices of a dual graph represent mesh elements and its edges represent the connection between neighboring elements.*

the Cray XC40 (Hazel Hen) system at the University of Stuttgart had a total number of 185088 processors. In most practical applications, time-dependent PDEs need to be solved at hundreds to thousands (or more) time steps, and thus, parallelization techniques to exploit parallelism in the temporal domain need to be employed to enable a further reduction of the time-to-solution.

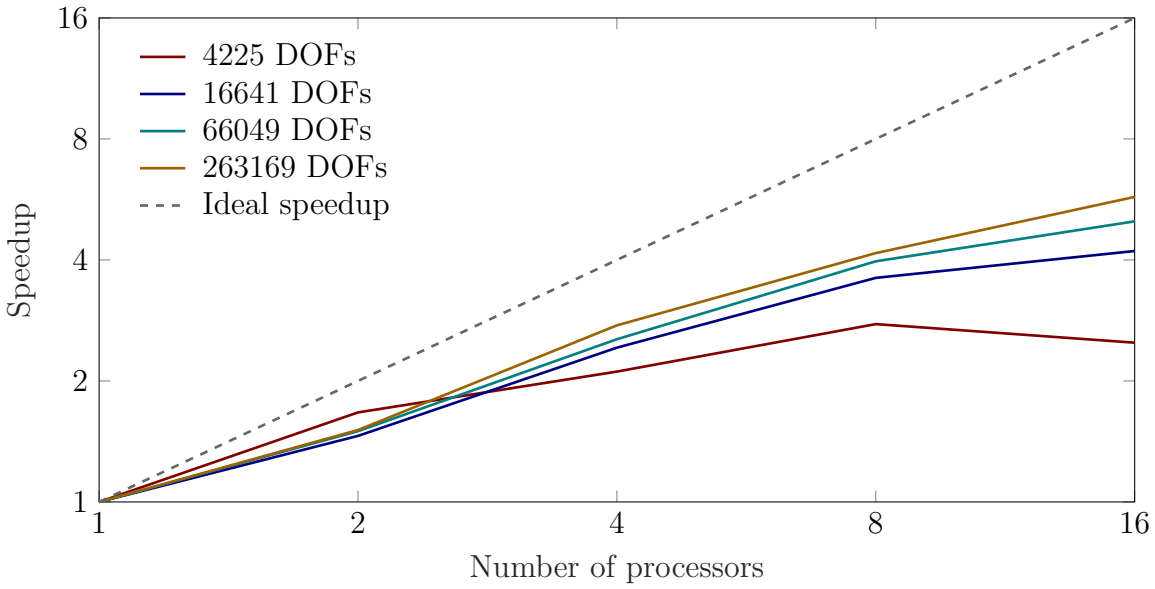


Figure 1.5: *Observed speedup for solving a 2D isotropic diffusion equation using spatial parallelism, depending on the number of spatial DOFs.*

1.3.2 Time parallelism

To solve a time-dependent ODE or PDE over N_0 time-steps, sequential time-stepping takes exactly N_0 steps and is, thus, considered optimal. Time-stepping, however, is an inherently sequential process since the solution at each time step depends on the solution at the previous step. Intuitively, with this dependency between time steps, time integration may seem impossible to parallelize.

On the other hand, spatial DOFs that arise from the discretization of time-dependent PDEs depend on their (spatial) neighbors as well, e.g., through spatial gradient operators. While this dependency is generally present with respect to all spatial directions, time-stepping has a one-sided dependency.[3] With this analogy, however, one may appreciate that such dependencies do not necessarily prohibit parallel computations.

In fact, parallelization techniques that are applied to the temporal component of time-dependent ODEs were first introduced more than a half century ago [108]. The advent of massively parallel computing hardware has sparked a large number of new developments and applications over the last decade, see Figure 1.6. Nowadays, many different approaches exist and can be categorized according to the review article by Gander [39]:

- Shooting-type time-parallel methods
- Domain decomposition methods in space-time
- Multigrid methods in space-time
- Direct solvers in space-time

Each of these categories is composed of many different methods and variants of these methods; for example, "waveform relaxation [99, 152], space-time multigrid [74], parallel implicit time-integrator [22, 33, 34], revisionist integral deferred correction [21], spectral deferred correction [28, 56, 136], Parareal [97] and multigrid-reduction-in-time [30, 35]" [68], to name a few.[4] These methods have been developed for various application areas and with varying degree of intrusiveness, ease of implementation, level of parallelism, and potential for speedup.

In this work, the multigrid-reduction-in-time algorithm is considered for the following reasons:

- Reuse of application codes:

  MGRIT is a nonintrusive PinT method that wraps existing simulation codes. Application code developers can maintain a single code base for legacy behavior and MGRIT-enabled time-parallelism.

- Introduction of parallel-in-time integration:

  Additional parallelism in time is introduced through MGRIT wrappers to further reduce the time-to-solution. In particular, the user may switch between space-parallelism, time-parallelism and space-time parallelism at runtime. This offers great flexibility depending on the given application and the available / employed hardware.

[3] Note, that forward and backward dependencies may also exist, e.g., when solving adjoint problems [51].
[4] Note, that this list is not exhaustive and more references can be found in [39].

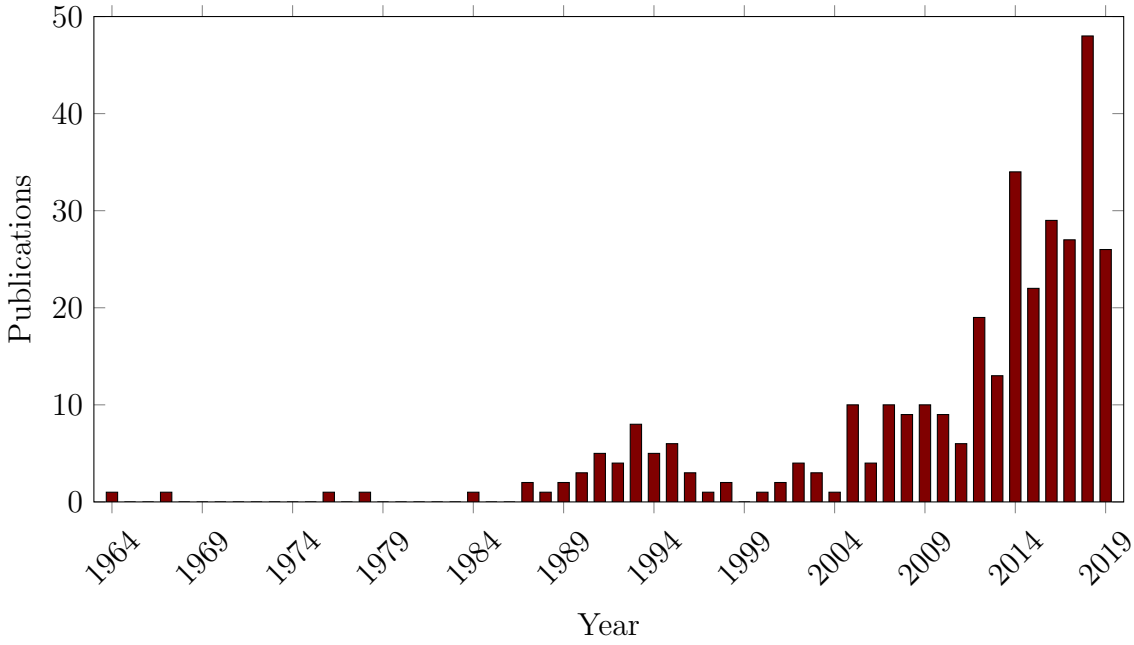


Figure 1.6: *Number of publications in the field of parallel-in-time integration methods and algorithms since* 1964. *Data from the community website* `parallel-in-time.org` *(accessed: September 4, 2019).*

- Optimal parallel scaling property:

  Multigrid has the desirable property of "optimal algorithmic scaling for both parallel communication and number of operations" [111].

- Availability of open-source implementation:

  The availability of XBraid [158] as an open-source implementation of the MGRIT algorithm simplifies the task of introducing time-parallelism in application codes, such as OpenCMISS [13, 112] and CHeart [18, 94]. It further provides a platform for developing extensions or variants of MGRIT and making these available to the community.

1.4 Thesis outline

This Ph.D. Thesis is organized as follows: In Chapter 2, a novel class of analytic solutions is introduced that aims to address the core need for method verification and spatiotemporal convergence analysis in the field of linear and nonlinear FSI research. A subset of these analytic solutions (transient linear FSI in two dimensions, transient nonlinear FSI in three dimensions) is later employed to assess the convergence and performance of MGRIT and a novel time-periodic MGRIT variant.

In Chapter 3, the MGRIT algorithm is introduced and an extension for time-periodic problems is presented. Chapter 4 reviews an existing two-level convergence analysis framework and discusses its strenghts and limitations. This motivates an extension to the true multilevel case, which is subsequently derived.

Chapter 5 assesses MGRIT performance for PDEs that are typically part of coupled FSI models in the cardiac field: first as single-physics problems (diffusion equation, wave equation, solid mechanics, fluid dynamics) and then as multiphysics problems (fluid-structure interaction with diffusion model to capture fluid domain motion). In Chapter 6, the novel time-periodic MGRIT algorithm is applied to a coupled left ventricle / left atrium flow model based on a patient-specific geometry from computed tomography (CT) data.

A summary of this Ph.D. Thesis is given in Chapter 7 along with a discussion of possible future extensions.

1.5 List of publications

Accepted manuscripts:

- *Experiment for validation of fluid-structure interaction models and algorithms* [64]
- *Validation of a non-conforming monolithic fluid-structure interaction method using phase-contrast MRI* [67]
- *3D Fluid-Structure Interaction Experiment and Benchmark Results* [63]
- *Enabling Detailed, Biophysics-Based Skeletal Muscle Models on HPC Systems* [14]
- *Convergence of the multigrid reduction in time algorithm for the linear elasticity equations* [66]
- *Multilevel convergence analysis of multigrid-reduction-in-time* [68]
- *A Class of Analytic Solutions for Verification and Convergence Analysis of Linear and Nonlinear Fluid-Structure Interaction Algorithms* [59]

In preparation:

- *Non-invasive estimation of relative pressure for intracardiac flows using virtual work-energy* [101]
- *Tight two-level convergence of Linear Parareal and MGRIT: Extensions and Implications in Practice* [134]
- *Multigrid-reduction-in-time: Estimating F-cycle convergence analytically* [60]
- *Time-periodic multigrid-reduction-in-time for a stenosed valve problem* [61]
- *Time-periodic steady-state solution of fluid-structure interaction and cardiac flow problems through multigrid-reduction-in-time* [62]
- *Solving the multidomain equations with multigrid-reduction-in-time* [65]

2 A Class of Analytic Solutions for Verification and Convergence Analysis of Linear and Nonlinear Fluid-Structure Interaction Algorithms[1]

2.1 Introduction

Within computational engineering, examples of fluid-structure interaction (FSI) are pervasive and represent an increasingly important set of problems. Addressing the disparate requirements of different FSI applications, the scientific community has responded by generating a broad range of numerical methods. From now classic arbitrary Lagrangian-Eulerian (ALE) boundary fitted approaches [26, 70, 77], to space-time ALE variational multiscale [142, 144, 145], unified continuum methods [72, 80], immersed boundary methods [105, 113, 114], fictitious domain methods [46, 47], immersed structural potential methods [44, 45], and overlapping domain methods [5, 20, 75, 137, 138, 154] Trailing this methodological development enabling complex simulations was a boom in applications, further pushing the numerical envelope to accommodate bigger problems with more physical models that emulate the complex fluid-structure dynamics of real-world systems.

Part and parcel to the development of FSI techniques, which often involve bespoke or in-house codes, comes the consistent need for verification and validation. Extending the tradition of documented numerical results established in fluid mechanics (e.g., lid-driven cavity [87, 123, 124]), FSI verification problems have been developed in both two and three dimensions [6, 43, 107, 147, 153] (a more exhaustive list can be found in [64], Table 1), often involving elastic structures immersed in a steady or periodic flow. These methods have been extensively used to compare results [57, 148] across codes and provide a measure of numerical consistency. Similarly, experiments have been proposed [10, 23, 48, 49, 67, 78] that provide experimental data for validation.

While these approaches provide important mechanisms for observing numerical performance and fidelity, the above benchmark problems and experiments typically present challenges for assessing the accuracy and convergence of FSI methods and implementations. Given the complexity of FSI solutions, spatiotemporal convergence analysis presents a critical, yet difficult, aspect of FSI method verification. The method of manufactured solutions [117, 120, 139], whereby a compatible solution is selected and appropriate forcing terms are prescribed, provides a potential avenue for addressing these needs. However, im-

[1] This chapter was submitted as [59]

plementation of forcing terms can be complex, code specific, and prone to spatiotemporal errors introduced through their numerical inclusion. An alternative is to provide analytic solutions, as are used in other fields, for spatiotemporal convergence analysis [29]. In this case, the fluid and solid state variables are known to satisfy the FSI problem, enabling comprehensive numerical assessment extending to the limits of machine precision. While promising, in theory, the availability of analytic solutions for FSI remain severely limited due, in part, to the complexity of the system.

In this work, we aim to address this core need for method verification and spatiotemporal convergence analysis by introducing a novel class of analytic solutions for FSI. Building from Womersley's solution for pulsatile fluid flow [155], we introduce analytic solutions that characterize fluid flow and solid motion in shear. To enable incremental testing, solutions are derived for two (channel) and three dimensions (tube) for transient and steady fluids and solids. Additionally, we introduce analytic solutions for both linear elastic and nonlinear hyperelastic (neo-Hookean) solid models. In total, 16 analytic solutions are presented, exhibiting a range of solution complexity in both space and time. Code implementing these solutions is provided (see Section 2.6.5), enabling comparisons with numerical results and evaluation of FSI solutions for different parameter combinations. Derived solutions are subsequently used to examine spatiotemporal convergence and accuracy of our previously published FSI method and implementation [94, 110], demonstrating the efficacy of these analytic solutions for providing meaningful analysis.

In what follows, we begin by outlining the general FSI boundary value problem in both two and three dimensions and deriving their analytic solutions (Section 2.2). Details of the previously published FSI method [110] are briefly reviewed (Section 2.3) along with details of the spatiotemporal convergence analysis protocol (Section 2.4). In Section 2.5, results illustrating the behavior of the analytic solutions for two specific cases (transient two-dimensional fluid / linear solid, transient three-dimensional fluid / nonlinear solid) are demonstrated, accompanied by convergence results showing expected numerical behavior with spatiotemporal refinement. These results are discussed in-depth in Section 2.6, followed by concluding remarks in Section 2.7.

2.2 Methodology

2.2.1 Derivations

In the interest of condensing the material presented in this chapter, we first derive the linear case in two dimensions before extending it to the nonlinear case. In three dimensions, we take the opposite approach and start from the nonlinear form before we simplify it to derive the analytic solution for the linear case. Each category is divided to include all possible permutations of quasi-static and transient behaviors. In the following, we derive a general solution for the fluid and solid problems separately. Subsequently, depending on the combination of temporal behaviors, we derive a unique FSI solution which satisfies the kinematic and traction coupling conditions.

Physical and numerical parameters	
Material properties	
ρ_f	Fluid density
μ_f	Fluid viscosity
ρ_s	Solid density
μ_s	Solid stiffness
Discretization	
δ_t	Time step size
$\delta_x, \delta_y, \delta_z$	Spatial step size
Domain dimensions and forces	
H_i	Fluid domain height
H_o	Fluid-solid domain height
L	Domain length
T	Temporal cycle length
P	Pressure over domain length
Derived constants	
k_f	$\sqrt{\rho_f i\omega/\mu_f}$
k_s	$\omega\sqrt{\rho_s/\mu_s}$
α	$e^{k_f H_i} + e^{-k_f H_i}$
β	$\mu_f k_f \left(e^{k_f H_i} - e^{-k_f H_i}\right)$
γ	$J^r_{0,s}/Y^r_{0,s}$
$J^r_{0,s}$	$J_0(-k_s H_o)$
$Y^r_{0,s}$	$Y_0(-k_s H_o)$
$J^*_{0,f}$	$J_0(ik_f H_i)$
$J^*_{1,f}$	$k_f J_1(ik_f H_i)$
$J^*_{0,s}$	$J_0(-k_s H_i)$
$J^*_{1,s}$	$ik_s J_1(-k_s H_i)$
$Y^*_{0,s}$	$Y_0(-k_s H_i)$
$Y^*_{1,s}$	$ik_s Y_1(-k_s H_i)$
Δ_0	$J^*_{0,s} - \gamma Y^*_{0,s}$
Δ_1	$J^*_{1,s} - \gamma Y^*_{1,s}$
ν_0	$Y^*_{0,s}/Y^r_{0,s}$
ν_1	$Y^*_{1,s}/Y^r_{0,s}$
ξ_1	$\sin(k_s H_i) + \cot(k_s H_o)\cos(k_s H_i)$
ξ_2	$\cot(k_s H_o)\sin(k_s H_i) - \cos(k_s H_i)$
ζ_1	$\csc(k_s H_o)\cos(k_s H_i)$
ζ_2	$1 - \sin(k_s H_i)\csc(k_s H_o)$

Table 2.1: *Material parameters and constants.*

2.2.1.1 Two space dimensions - linear case

We begin the series of derivations with the simplest case presented in this work: the interaction between a pulsatile-flowing liquid in a two dimensional channel and a linear elastic solid, on the top and bottom, which undergoes shear deformation, see Figure 2.1. One of the aims of this section is to help the reader get familiarized with the notation, main assumptions and steps taken throughout the derivation, all of which are reused and adapted in the more complex cases. For simplicity, we assume that the physics are symmetrical with respect to the axis running along the middle of the channel, allowing us to limit the problem domain to one of the halves. In the reference space, the solid and fluid domains are represented by two quadrilaterals, Ω_f^0 and Ω_s^0, which are separated by an interface Γ^λ. Their length, the width of the channel and the width of the domain are denoted by L, H_i and H_o. The top boundary represents the outer wall of the solid and is denoted by Γ_s^W, while the bottom one corresponds to the axis of symmetry of the channel and we refer to it as Γ_f^W. The left and right boundaries are the inlet (Γ_f^I and Γ_s^I) and the outlet (Γ_f^O and Γ_s^O).

The FSI problem takes the following general form:

$$\rho_f \partial_t \boldsymbol{v}_f - \nabla \cdot \boldsymbol{\sigma}_f = \boldsymbol{0} \qquad \text{in } \Omega_f, \tag{2.1}$$
$$\nabla \cdot \boldsymbol{v}_f = 0 \qquad \text{in } \Omega_f, \tag{2.2}$$
$$[\boldsymbol{v}_f]_y = 0 \qquad \text{on } \Gamma_f^I \cup \Gamma_f^O \cup \Gamma_f^W, \tag{2.3}$$
$$(\boldsymbol{\sigma}_f \cdot \boldsymbol{n}_f) \cdot \boldsymbol{e}_x = [\boldsymbol{t}_f]_x \qquad \text{on } \Gamma_f^I \cup \Gamma_f^O, \tag{2.4}$$
$$\partial \boldsymbol{v}_f / \partial y = \boldsymbol{0} \qquad \text{on } \Gamma_f^W, \tag{2.5}$$
$$\boldsymbol{v}_f(\cdot, 0) = \boldsymbol{v}_f^0 \qquad \text{in } \Omega_f(0), \tag{2.6}$$

$$\rho_s \partial_{tt} \boldsymbol{u}_s - \nabla \cdot \boldsymbol{\sigma}_s = \boldsymbol{0} \qquad \text{in } \Omega_s, \tag{2.7}$$
$$\nabla \cdot \partial_t \boldsymbol{u}_s = 0 \qquad \text{in } \Omega_s, \tag{2.8}$$
$$\boldsymbol{u}_s(\cdot, t) = \boldsymbol{0} \qquad \text{on } \Gamma_s^W, \tag{2.9}$$
$$[\boldsymbol{u}_s]_y = 0 \qquad \text{on } \Gamma_s^I \cup \Gamma_s^O, \tag{2.10}$$
$$(\boldsymbol{\sigma}_s \cdot \boldsymbol{n}_s) \cdot \boldsymbol{e}_x = [\boldsymbol{t}_s]_x \qquad \text{on } \Gamma_s^I \cup \Gamma_s^O, \tag{2.11}$$
$$\boldsymbol{u}_s(\cdot, 0) = \boldsymbol{u}_s^0 \qquad \text{in } \Omega_s(0), \tag{2.12}$$
$$\boldsymbol{v}_s(\cdot, 0) = \boldsymbol{v}_s^0 \qquad \text{in } \Omega_s(0), \tag{2.13}$$

$$\boldsymbol{\sigma}_f \cdot \boldsymbol{n}_f + \boldsymbol{\sigma}_s \cdot \boldsymbol{n}_s = \boldsymbol{0} \qquad \text{on } \Gamma^\lambda, \tag{2.14}$$
$$\boldsymbol{v}_f - \boldsymbol{v}_s = \boldsymbol{0} \qquad \text{on } \Gamma^\lambda, \tag{2.15}$$

with the Cauchy stress tensor for the fluid and solid defined as:

$$\boldsymbol{\sigma}_f = \mu_f [\nabla \boldsymbol{v}_f + \nabla^T \boldsymbol{v}_f] - p_f \boldsymbol{I} \quad \text{and} \quad \boldsymbol{\sigma}_s = \mu_s [\nabla \boldsymbol{u}_s + \nabla^T \boldsymbol{u}_s] - p_s \boldsymbol{I},$$

respectively. Naturally, individual transient and quasi-static cases may be obtained by nullifying or assigning strictly positive values to the two density parameters, ρ_f and ρ_s. The solid stiffness and fluid viscosity are denoted by μ_s and μ_f. For now we will assume that the initial value fields (i.e. $\boldsymbol{v}_f^0$, $\boldsymbol{u}_s^0$ and $\boldsymbol{v}_s^0$), and the inlet and outlet surface tractions,

(i.e. $\boldsymbol{t}_f$ and $\boldsymbol{t}_s$) are given. As we shall see, knowing what these fields are is not prerequisite to the derivation of the analytical solutions. Here, they are presented in anticipation of the weak form of the problem used in the numerical results section. Also, note that the domains at time zero do not necessarily coincide with reference ones (e.g., when $\boldsymbol{u}_s^0 \neq \boldsymbol{0}$, $\Omega_s(0) \neq \Omega_s^0$).

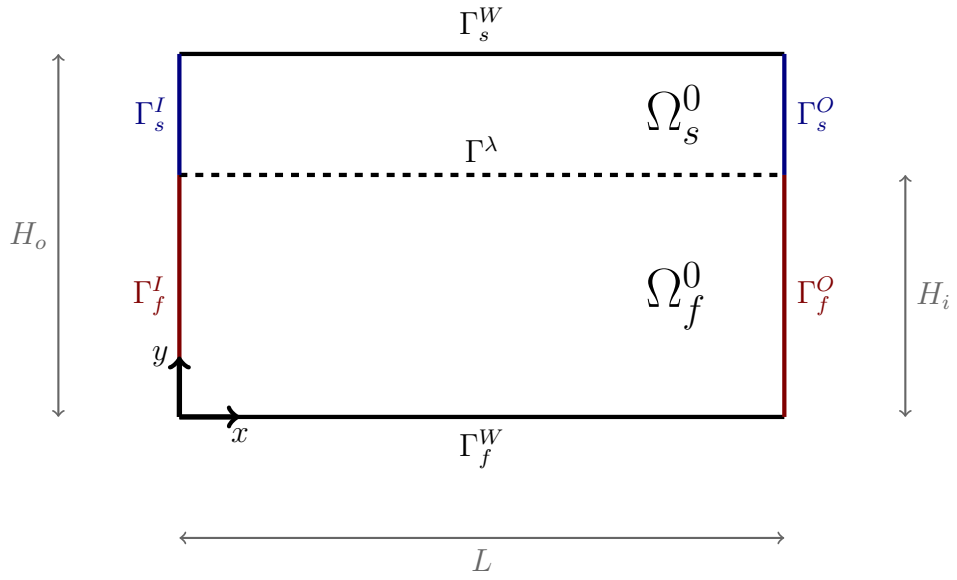


Figure 2.1: *Fluid and solid reference domains, Ω_f^0 and Ω_s^0, in two dimensions with respective boundaries at the inlet (Γ_f^I and Γ_s^I), the outlet (Γ_f^O and Γ_s^O) and the wall (Γ_f^W and Γ_s^W). The common interface boundary is denoted as Γ^λ. Further, the domain length is given as L, the fluid domain height as H_i and the fluid / solid domain height as H_o.*

In a similar fashion to the classical pulsatile channel flow problem, our strategy is based on the assumptions that the fluid velocity and solid displacement are constant in the x-direction. Hence, the solid displacement and fluid velocity take the form:

$$\boldsymbol{v}_f = v_f(y,t)\boldsymbol{e}_x \quad \text{and} \quad \boldsymbol{u}_s = u_s(y,t)\boldsymbol{e}_x.$$

To obtain a general form of the fluid solution, we first rewrite the momentum balance equation in (2.1) in component form and apply our knowledge of the flow behavior:

$$\rho_f \partial_t v_f = -\frac{\partial p_f}{\partial x} + \mu_f \frac{\partial^2 v_f}{\partial y^2}, \tag{2.16}$$

$$0 = -\frac{\partial p_f}{\partial y}. \tag{2.17}$$

From (2.17), we see that p_f is constant in y. Furthermore, applying $\partial(\cdot)/\partial x$ to (2.16) yields:

$$-\frac{\partial^2}{\partial x^2} p_f(x,t) = 0.$$

Thus, p_f is a constant or linear function in x. We then suppose the velocity and pressure are periodic and separable, where the pulsatile behavior is defined by a single harmonic frequency, ω, i.e.:

$$v_f(y,t) = \Re\{\mathrm{v}_f(y)e^{i\omega t}\} \quad \text{and} \quad p_f(x,t) = \Re\{P_f(L-x)e^{i\omega t}\}, \tag{2.18}$$

where $P_f \in \mathbb{C}$ and $\mathrm{v}_f : [0, H_i] \to \mathbb{C}$ denote the fluid pressure over the domain length L and velocity amplitudes, respectively. The first momentum balance equation reduces to:

$$i\omega\rho_f \mathrm{v}_f = P_f + \mu_f \mathrm{v}_f''. \tag{2.19}$$

In this case, the general solution takes the form:

$$\begin{aligned} \mathrm{v}_f(y) &= -\frac{P_f}{2\mu_f}y^2 + c_2 y + c_1, && (\rho_f = 0), \\ \mathrm{v}_f(y) &= -\frac{iP_f}{\rho_f\omega} + c_1 e^{k_f y} + c_2 e^{-k_f y}, && (\rho_f > 0). \end{aligned} \tag{2.20}$$

where $k_f = \sqrt{i\rho_f\omega/\mu_f}$ (see Table 2.1 for a list of short-form constants). Note that the integration constants are reused here to simplify notation and they are not related to each other. Furthermore, based on the smoothness condition in Equation (2.6), which is equivalent to $\mathrm{v}_f' = 0$, we can also conclude that $c_2 = 0$ (quasi-static case) or $c_2 = c_1$ (transient case).

Moving on to the solid component of the problem, the corresponding momentum balance equation can be rewritten to incorporate our assumptions:

$$\rho_s \partial_{tt} u_s = -\frac{\partial p_s}{\partial x} + \mu_s \frac{\partial^2 u_s}{\partial y^2}, \tag{2.21}$$

$$0 = -\frac{\partial p_s}{\partial y}. \tag{2.22}$$

Assuming periodicity, separability and the same period length for both fluid and solid, we write:

$$u_s(y,t) = \Re\{\mathrm{u}_s(y)e^{i\omega t}\} \quad \text{and} \quad p_s(x,t) = \Re\{P_s(L-x)e^{i\omega t}\}, \tag{2.23}$$

where $P_s \in \mathbb{C}$ and $\mathrm{u}_s : [R_i, R_0] \to \mathbb{C}$ are the amplitudes of the solid solution. Here, we skipped the derivation of p_s as it is analogous to that of p_f. Consequently, Equation (2.21) reduces to:

$$-\rho\omega^2 \mathrm{u}_s = P_s + \mu_s \mathrm{u}_s''.$$

resulting in the general solution for solid displacement:

$$\begin{aligned} \mathrm{u}_s(y) &= -\frac{P_s}{2\mu_s}y^2 + c_3 y + c_4, && (\rho_s = 0), \\ \mathrm{u}_s(y) &= -\frac{P_s}{\rho_s\omega^2} + c_3 \sin(k_s y) + c_4 \cos(k_s y), && (\rho_s > 0), \end{aligned} \tag{2.24}$$

where $k_s = \sqrt{\rho_s \omega^2 / \mu_s}$. The last unknown integration constants (i.e. c_1, c_3 and c_4) can only be identified by ensuring that the coupling conditions are satisfied. In the following we identify their closed form depending on the different temporal behavior combinations.

Quasi-static fluid and quasi-static solid ($\rho_f = \rho_s = 0$)
We begin the coupling process by expanding the traction balance boundary condition in (2.14):

$$\mu_f \left.\frac{\partial \mathrm{v}_f}{\partial y}\right|_{y=H_i} - \mu_s \left.\frac{\partial \mathrm{u}_s}{\partial y}\right|_{y=H_i} = 0, \tag{2.25}$$

$$- p_f|_{y=H_i} + p_s|_{y=H_i} = 0. \tag{2.26}$$

Based on Equation ((2.26)) it can easily be shown that $P_f = P_s = P$. Note that, as we will see, this result is independent of the quasi-static/transient property of the FSI problem. Furthermore, substituting Equation (2.20) and Equation (2.24) into Equation (2.25), we can see that:

$$-PH_i - \mu_s \left(-\frac{PH_i}{\mu_s} + c_3\right) = 0 \;\; \Rightarrow \;\; c_3 = 0. \tag{2.27}$$

Considering the fixed wall constraint on the solid (i.e. $\mathrm{u}_s(H_o) = 0$), we can use Equation (2.24) to show that:

$$c_4 = \frac{PH_o^2}{2\mu_s}. \tag{2.28}$$

Finally, by expanding the kinematic constraint in (2.15), we obtain the final unknown constant:

$$c_1 = \frac{PH_i^2}{2\mu_f} + i\omega \frac{P}{2\mu_s}(H_o^2 - H_i^2). \tag{2.29}$$

Quasi-static fluid and transient solid ($\rho_f = 0$, $\rho_s > 0$)
Based on our general formulations for the fluid velocity and solid displacement in (2.18), (2.20), (2.23) and (2.24), the kinematic interface condition (2.15) reduces to:

$$c_1 - c_3 i\omega \sin(k_s H_i) - c_4 i\omega \cos(k_s H_i) = \frac{PH_i^2}{2\mu_f} - \frac{iP}{\rho_s \omega}. \tag{2.30}$$

A second equation relating the integration constants can be obtained from the condition (2.9) for fixed outer walls:

$$c_3 \sin(k_s H_o) + c_4 \cos(k_s H_o) = \frac{P}{\rho_s \omega^2}. \tag{2.31}$$

The third and final equation can be obtained through the expansion of the shear component of the traction balance equation (2.14), e.g.,

$$c_3 \mu_s k_s \cos(k_s H_i) - c_4 \mu_s k_s \sin(k_s H_i) = -PH_i. \tag{2.32}$$

Finally, c_1, c_3 and c_4 can be determined by solving the system of equations formed by (2.30), (2.31) and (2.32). Provided there is a unique solution, the three constants can be written as the following closed-form expressions:

$$c_1 = \frac{PH_i^2}{2\mu_f} - \frac{iP}{\rho_s\omega} + PH_i\frac{i\omega}{\mu_s k_s}\tan[k_s(H_o - H_i)] + \frac{iP}{\rho_s\omega}\sec[k_s(H_o - H_i)], \tag{2.33}$$

$$c_3 = \left[\frac{1}{\rho_s\omega^2}\sin(k_s H_i) - \frac{H_i}{\mu_s k_s}\cos(k_s H_o)\right] P\sec[k_s(H_o - H_i)], \tag{2.34}$$

$$c_4 = \left[\frac{H_i}{\mu_s k_s}\sin(k_s H_o) + \frac{1}{\rho_s\omega^2}\cos(k_s H_i)\right] P\sec[k_s(H_o - H_i)]. \tag{2.35}$$

This approach of finding the closed form of the constants by solving a system of equations is repeated throughout the paper. Consequently, it should be noted that the existence of unique solutions is conditioned by the non-singularity of the system matrix. As we will see, identifying the appropriate set of parameters which lead to a singular matrix is not a trivial problem and, as a consequence, we will only perform this analysis for this specific case, which is more tractable. Here, it can be shown that for a set of problem parameters with finite values, the determinant of the system is null when:

$$\cos\left[k_s(H_o - H_i)\right] = 0.$$

Rearranging this, we find a series of resonance frequencies for which this is true:

$$\omega_n = \frac{(2n+1)\pi}{2(R_o - R_i)}\sqrt{\frac{\mu_s}{\rho_s}}, \quad \text{for } n \in \mathbb{Z}. \tag{2.36}$$

Transient fluid and quasi-static solid ($\rho_f > 0,\ \rho_s = 0$)
As in the previous problem permutation, the constants (i.e. c_1, c_3 and c_4) can be identified by solving the system of equations derived from (2.14), (2.9) and (2.15):

$$\left(e^{k_f H_i} + e^{-k_f H_i}\right) c_1 - i\omega H_i c_3 - i\omega c_4 = \frac{i\omega P}{\rho_f\omega} - \frac{i\omega PH_i^2}{2\mu_s}, \tag{2.37}$$

$$H_o c_3 + c_4 = \frac{PH_o^2}{2\mu_s}, \tag{2.38}$$

$$\mu_f k_f\left(e^{k_f H_i} - e^{-k_f H_i}\right) c_1 - \mu_s c_3 = -PH_i. \tag{2.39}$$

The resulting closed-form expressions for the three constants are:

$$c_1 = \frac{\frac{i\mu_s P}{\rho_f\omega} + \frac{i\omega P}{2}(H_o^2 - H_i^2) - i\omega P(H_o - H_i)H_i}{\alpha\mu_s + i\omega(H_o - H_i)\beta} \tag{2.40}$$

$$c_3 = \frac{\left[\frac{iP}{\rho_f\omega} + \frac{i\omega P}{2\mu_s}(H_o^2 - H_i^2)\right]\beta + PH_i\alpha}{\alpha\mu_s + i\omega(H_o - H_i)\beta} \tag{2.41}$$

$$c_4 = \frac{PH_o^2}{2\mu_s} - H_o\frac{\left[\frac{iP}{\rho_f\omega} + \frac{i\omega P}{2\mu_s}(H_o^2 - H_i^2)\right]\beta + PH_i\alpha}{\alpha\mu_s + i\omega(H_o - H_i)\beta} \tag{2.42}$$

where $\alpha = e^{k_f H_i} + e^{-k_f H_i}$ and $\beta = \mu_f k_f\left(e^{k_f H_i} - e^{-k_f H_i}\right)$.

Transient fluid and transient solid ($\rho_f,\ \rho_s > 0$)
In the transient fluid / transient solid case, the three equations obtained from the expansion of (2.14), (2.9) and (2.15) are:

$$\left(e^{k_f H_i} + e^{-k_f H_i}\right) c_1 - i\omega \sin(k_s H_i) c_3 - i\omega \cos(k_s H_i) c_4 \left(\frac{P}{i\rho_f \omega} + \frac{iP}{\rho_s \omega}\right) = 0, \tag{2.43}$$

$$\mu_f k_f \left(e^{k_f H_i} - e^{-k_f H_i}\right) c_1 - \mu_s k_s \cos(k_s H_i) c_3 + \mu_s k_s \sin(k_s H_i) c_4 = 0, \tag{2.44}$$

$$\sin(k_s H_o) c_3 + \cos(k_s H_o) c_4 - \frac{P}{\rho_s \omega^2} = 0. \tag{2.45}$$

The resulting closed-form expressions are:

$$c_1 = \frac{\mu_s k_s \cos\left[k_s(H_i - H_o)\right] \left(\frac{P}{i\rho_f \omega} + \frac{iP}{\rho_s \omega}\right) - i\omega \mu_s k_s \frac{P}{\rho_s \omega^2}}{-\mu_s k_s \alpha \cos\left[k_s(H_i - H_o)\right] + i\omega\beta \sin\left[k_s(H_i - H_o)\right]}, \tag{2.46}$$

$$c_3 = \frac{\beta \cos(k_s H_o) \left(\frac{P}{i\rho_f \omega} + \frac{iP}{\rho_s \omega}\right) - \left[i\omega\beta \cos(k_s H_i) + \alpha \mu_s k_s \sin(k_s H_i)\right] \frac{P}{\rho_s \omega^2}}{-\mu_s k_s \alpha \cos\left[k_s(H_i - H_o)\right] + i\omega\beta \sin\left[k_s(H_i - H_o)\right]}, \tag{2.47}$$

$$c_4 = \frac{P}{\rho_s \omega^2} \sec(k_s H_o) - \tan(k_s H_o) c_3. \tag{2.48}$$

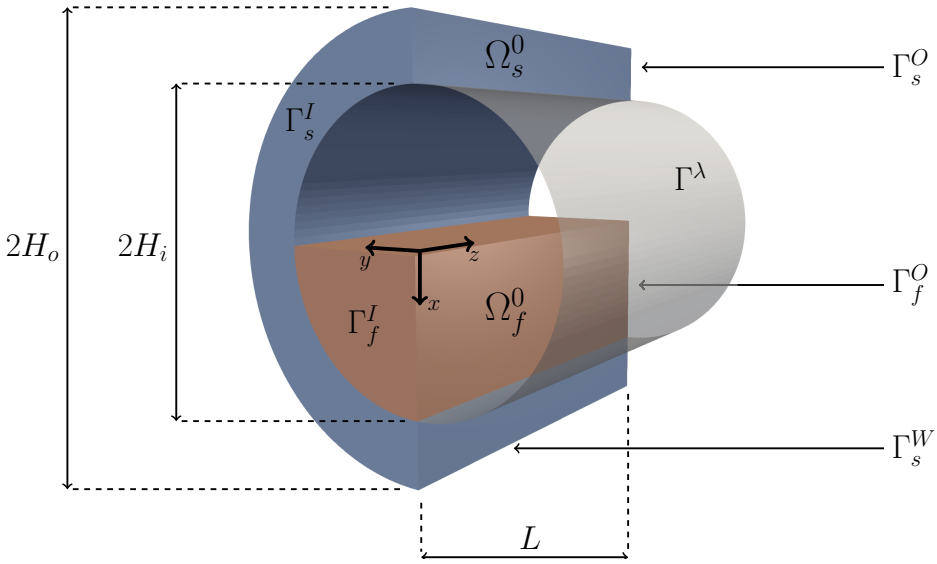


Figure 2.2: *Fluid and solid reference domains in three dimensions: The fluid reference domain Ω_f^0 is shown in red for $x, y > 0$, the solid reference domain Ω_s^0 is shown in blue for $y > 0$, and the the common interface boundary is indicated in opaque gray. The respective boundaries are denoted as Γ_f^I and Γ_s^I at the inlet, Γ_f^O and Γ_s^O at the outlet, and Γ_f^W and Γ_s^W at the wall. Furthermore, the domain length is given as L, the fluid domain radius as H_i and the fluid / solid domain radius as H_o.*

2.2.1.2 Three space dimensions - nonlinear case

In the three-dimensional setting, the channel is replaced with a tube. Similar to the previous case, we now consider the interaction between the pulsatile flow and the hyperelastic wall which undergoes a shearing deformation along the flow direction. The domains

of the two media in the reference configuration are shown in Figure 2.2, where we used an analogous notation for domains, boundaries and interfaces as in the case of the two-dimensional FSI problem. Similarly, H_i and H_o now denote the inner and outer radii of the tube. For ease, we use a cylindrical coordinate system defined by radial (r), angular (θ) and axial (z) positions.

The FSI problem takes the following general form:

$$\rho_f \partial_t \boldsymbol{v}_f + \rho_f \boldsymbol{v}_f \cdot \nabla_{\boldsymbol{x}} \boldsymbol{v}_f + \nabla_{\boldsymbol{x}} \cdot \boldsymbol{\sigma}_f = \boldsymbol{0} \qquad \text{in } \Omega_f, \tag{2.49}$$

$$\nabla_{\boldsymbol{x}} \cdot \boldsymbol{v}_f = 0 \qquad \text{in } \Omega_f, \tag{2.50}$$

$$[\boldsymbol{v}_f]_k = 0 \qquad \text{on } \Gamma_f^I \cup \Gamma_f^O, \text{ for } k \in \{r, \theta\}, \tag{2.51}$$

$$(\boldsymbol{\sigma}_f \cdot \boldsymbol{n}_f) \cdot \boldsymbol{e}_z = [\boldsymbol{t}_f]_z \qquad \text{on } \Gamma_f^I \cup \Gamma_f^O, \tag{2.52}$$

$$\boldsymbol{v}_f(\cdot, 0) = \boldsymbol{v}_f^0 \qquad \text{in } \Omega_f(0), \tag{2.53}$$

$$\rho_s \partial_{tt} \boldsymbol{u}_s - \nabla_{\boldsymbol{X}} \cdot \boldsymbol{P}_s = \boldsymbol{0} \qquad \text{in } \Omega_s(0) \times [0, T], \tag{2.54}$$

$$J - 1 = 0 \qquad \text{in } \Omega_s(0) \times [0, T], \tag{2.55}$$

$$\boldsymbol{u}_s(\cdot, t) = \boldsymbol{0} \qquad \text{on } \Gamma_s^W, \tag{2.56}$$

$$[\boldsymbol{u}_s]_k = 0 \qquad \text{on } \Gamma_s^I \cup \Gamma_s^O, \text{ for } k \in \{r, \theta\} \tag{2.57}$$

$$(\boldsymbol{P}_s \cdot \boldsymbol{N}_s) \cdot \boldsymbol{e}_z = \boldsymbol{t}_s \qquad \text{on } \Gamma_s^I \cup \Gamma_s^O, \tag{2.58}$$

$$\boldsymbol{u}_s(\cdot, 0) = \boldsymbol{u}_s^0 \qquad \text{in } \Omega_s(0), \tag{2.59}$$

$$\boldsymbol{v}_s(\cdot, 0) = \boldsymbol{v}_s^0 \qquad \text{in } \Omega_s(0), \tag{2.60}$$

$$\boldsymbol{\sigma}_f \cdot \boldsymbol{n}_f + \boldsymbol{P}_s \cdot \boldsymbol{N}_s = \boldsymbol{0} \qquad \text{on } \Gamma^\lambda, \tag{2.61}$$

$$\boldsymbol{v}_f - \boldsymbol{v}_s = \boldsymbol{0} \qquad \text{on } \Gamma^\lambda, \tag{2.62}$$

where the fluid Cauchy stress tensor is the same as in the linear case. For the solid, the first Piola-Kirchhoff stress tensor is defined as:

$$\boldsymbol{P}_s = \frac{\mu_s}{J^{2/3}} \left[\boldsymbol{F} - \frac{\boldsymbol{F} : \boldsymbol{F}}{3} \boldsymbol{F}^{-T} \right] - p_s \boldsymbol{F}^{-T}.$$

The domains at time zero, $\Omega_s(0)$ and $\Omega_f(0)$, do not necessarily coincide with the reference domains, Ω_s^0 and Ω_f^0, e.g., when $\boldsymbol{u}_s^0 \neq \boldsymbol{0}$. For now, we will consider $\boldsymbol{u}_s^0$, $\boldsymbol{v}_s^0$, $\boldsymbol{t}_f$ and $\boldsymbol{t}_s$ to be known, since their definitions are not used to arrive to the analytical solutions.

Similar to the 2D problem in Section 2.2.1.1, our derivation is based on the assumption that the flow and deformation field are axisymmetric and axially invariant, i.e. $\boldsymbol{v}_f = v_f(r, t)\boldsymbol{e}_z$ and $\boldsymbol{u}_s = u_s(r, t)\boldsymbol{e}_z$. A first consequence of these is that the advective term in Equation (2.49) is always null. Secondly, we can keep using ρ_f as a switch between the quasi-static and transient modes. Thirdly, we can use the same coordinate vectors (i.e. $\boldsymbol{e}_r$, $\boldsymbol{e}_\theta$ and $\boldsymbol{e}_z$), to refer to both the reference and current configurations. The same notation simplification is also valid for r and θ. Conversely, we use Z and z to distinguish the axial coordinates in the reference and deformed configuration.

In order to derive the general analytical solution for the fluid, we first observe that:

$$\nabla_{\boldsymbol{x}} \boldsymbol{v}_f = \frac{\partial v_f(r, t)}{\partial r} \boldsymbol{e}_r \otimes \boldsymbol{e}_z.$$

Based on this, we can expand the fluid momentum equation (2.49) into its three vector components:

$$\frac{\partial p_f}{\partial r}\boldsymbol{e}_r + \frac{1}{r}\frac{\partial p_f}{\partial \theta}\boldsymbol{e}_\theta + \left[\rho_f \partial_t v_f - \frac{\mu_f}{r}\frac{\partial}{\partial r}\left(r\frac{\partial v_f}{\partial r}\right) + \frac{\partial p_f}{\partial z}\right]\boldsymbol{e}_z = \boldsymbol{0}. \tag{2.63}$$

From the radial and circumferential directions of the equation, we can conclude that p_f is constant in the plane perpendicular to the flow direction. Taking the partial derivative in the axial direction, we obtain that $\partial^2 p_f/\partial z^2 = 0$. Thus, p_f is either a constant or a linear function in z, as observed in the 2D linear case. If we assume the velocity and pressure fields are periodic (with a single harmonic frequency) and that they are separable, we arrive at the following forms:

$$\begin{aligned} v_f &= \Re\{\mathrm{v}_f(r)e^{i\omega t}\}, \\ p_f &= \Re\{P(L-z)e^{i\omega t}\}, \end{aligned} \tag{2.64}$$

where $\mathrm{v}_f : [0, H_i] \to \mathbb{C}$ and $P_f \in \mathbb{C}$ are the velocity and pressure over domain length amplitudes.

Returning to the axial component of (2.63) and simplifying the complex exponential term, we obtain the following ODE:

$$\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial \mathrm{v}_f}{\partial r}\right) - k_f^2 \mathrm{v}_f + \frac{P}{\mu_f} = 0,$$

which, depending on the value of ρ_s, is satisfied by the following general solutions:

$$\begin{aligned} \mathrm{v}_f &= -\frac{Pr^2}{4\mu_f} + c_1 + c_2\ln(r), && (\rho_s = 0), \\ \mathrm{v}_f &= \frac{P}{\mu_f k_f^2} + c_1 J_0(ik_f r) + c_2 Y_0(ik_f r), && (\rho_s > 0). \end{aligned} \tag{2.65}$$

Here, J_0 and Y_0 denote the first and second kind Bessel functions of order 0. Finally, we can remark that $c_2 = 0$ in order to prevent a singularity at $r = 0$.

Moving on to the solid, we begin by defining the deformation tensor as $\boldsymbol{F} = \boldsymbol{I} + \frac{\partial u_s}{\partial r}\boldsymbol{e}_r \otimes \boldsymbol{e}_z$. Thus, we can write the first Piola-Kirchhoff into its vector components:

$$\boldsymbol{P} = \mu_s\left[2\frac{\partial u_s}{\partial r}\mathrm{sym}(\boldsymbol{e}_r \otimes \boldsymbol{e}_z) - \left(\frac{1}{3}\left[\frac{\partial u_s}{\partial r}\right]^2 + \frac{p_s}{\mu_s}\right)\left(\boldsymbol{I} - \frac{\partial u_s}{\partial r}\boldsymbol{e}_z \otimes \boldsymbol{e}_r\right)\right]. \tag{2.66}$$

Substituting this into (2.54), we can expand the momentum balance equation as:

$$\begin{aligned} &\left[-\frac{\partial}{\partial r}\left(\frac{\mu_s}{3}\left[\frac{\partial u_s}{\partial r}\right]^2 + p_s\right) + \frac{\partial u_s}{\partial r}\frac{\partial p_s}{\partial z}\right]\boldsymbol{e}_r - \frac{1}{r}\frac{\partial p_s}{\partial \theta}\boldsymbol{e}_\theta \\ &\qquad + \left[\frac{\mu_s}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u_s}{\partial r}\right) - \frac{\partial p_s}{\partial z} - \rho_s \partial_{tt} u_s\right]\boldsymbol{e}_z = \boldsymbol{0}. \end{aligned} \tag{2.67}$$

Applying the axial partial derivative to the $\boldsymbol{e}_z$-component, we obtain that $\partial^2 p_s/\partial z^2 = 0$. From this we conclude that p_s is either a constant or linear with respect to z. Furthermore,

by applying the axial derivative to the $\boldsymbol{e}_r$-component, we find that $\partial^2 p_s/\partial r \partial z = 0$, i.e. $\partial p_s/\partial z$ is radially constant. Based on these results, we can integrate the radial component to obtain the following formula for the pressure:

$$p_s = c_I(t) + (Z + u_s)c_{II}(t) - \frac{\mu_s}{3}\left[\frac{\partial u_s}{\partial r}\right]^2. \tag{2.68}$$

Similar to the 2D linear case, we assume that the solid displacement and the c_{II} constant are separable periodic (with a single harmonic frequency):

$$u_s = \Re\{\mathrm{u}_s e^{i\omega t}\} \quad \text{and} \quad c_{II} = \Re\{c_3 e^{i\omega t}\}$$

with $\mathrm{u}_s : [H_i, H_o] \to \mathbb{C}$ and $c_3 \in \mathbb{C}$. Consequently, we can simplify the temporal complex exponentials in Equation (2.67), which yields:

$$\frac{1}{r}\frac{d}{dr}\left(r\frac{d\mathrm{u}_s}{dr}\right) + k_s^2 \mathrm{u}_s - \frac{c_3}{\mu_s} = 0. \tag{2.69}$$

For the transient case, i.e. $\rho_s > 0$ and $k_s \neq 0$, we can recognize this as a nonhomogeneous Bessel's ODE, with the order of the Bessel function equal to 0. Hence, the general solid displacement solutions are:

$$\begin{aligned} \mathrm{u}_s &= \frac{c_3 r^2}{4\mu_s} + c_4 \ln(r) + c_5, && (\rho_s = 0) \\ \mathrm{u}_s &= \frac{c_3}{\mu_s k_s^2} + c_4 J_0(-k_s r) + c_5 Y_0(-k_s r), && (\rho_s > 0). \end{aligned}$$

Furthermore, by including the fixed wall boundary condition, i.e. $\mathrm{u}_s(H_o) = 0$, we arrive at the following simplified forms:

$$\begin{aligned} \mathrm{u}_s &= \frac{c_3}{4\mu_s}(r^2 - H_o^2) + c_4 \ln\left(\frac{r}{H_o}\right), && (\rho_s = 0) \\ \mathrm{u}_s &= \frac{c_3}{\mu_s k_s^2}\left[1 - \frac{Y_0(-k_s r)}{Y_{0,s}^r}\right] + c_4\left[J_0(-k_s r) - \gamma Y_0(-k_s r)\right], && (\rho_s > 0). \end{aligned} \tag{2.70}$$

The remaining unknown constants (i.e. c_1, c_3, c_4 and c_I) can only be identified by verifying that the kinematic and traction balance boundary conditions are satisfied. In the following, we look at the different transient / quasi-static permutations and derive for each of them the appropriate closed formulations for the set of constants.

Quasi-static fluid and quasi-static solid ($\rho_f = \rho_s = 0$)
For the fluid and solid problems to be coupled, the traction balance and kinematic conditions in Equation (2.61) and Equation (2.62), respectively, need to hold. The former can now be expanded as follows:

$$\begin{aligned} &\left[\frac{\mu_s}{3}\left(\frac{\partial u_s}{\partial r}\right)^2\bigg|_{r=H_i} + p_s(H_i, z, t) - p_f(z, t)\right]\boldsymbol{e}_r \\ &\qquad + \left[\mu_f \left.\frac{\partial v_f}{\partial r}\right|_{r=H_i} - \mu_s \left.\frac{\partial u_s}{\partial r}\right|_{r=H_i}\right]\boldsymbol{e}_z = \boldsymbol{0}, \end{aligned} \tag{2.71}$$

for all $z \in [0, L]$ and $t \in [0, T]$. Based on the solid and pressure formulations in (2.64) and (2.68), the radial component can be further reduced to:

$$c_I(t) + [Z + u_s(H_i, t)]c_3 e^{i\omega t} - P(L - Z)e^{i\omega t} = 0. \tag{2.72}$$

From this, we can deduce that $c_3 = -P$ and $c_I(t) = [L + u_s(H_i, t)]\Re\{Pe^{i\omega t}\}$, two results which are valid for all transient / quasi-static permutation. Consequently, (2.68) can be expanded using (2.70) in quasi-static form to yield the following pressure formula:

$$\begin{aligned} p_s(r, Z, t) = & \left[L - Z + u_s(H_i, t) - u_s(r, t)\right] \Re\{Pe^{i\omega t}\} - \frac{\mu_s}{3}\left[\frac{\partial u_s(r,t)}{\partial r}\right]^2 \\ = & \left[L - Z + \Re\left\{c_4 e^{i\omega t} \ln\left(\frac{H_i}{r}\right)\right\} + \Re\left\{\frac{P}{4\mu_s}(r^2 - H_i^2)e^{i\omega t}\right\}\right] \Re\{Pe^{i\omega t}\} \\ & - \frac{\mu_s}{3}\left[\Re\left\{\frac{c_4 e^{i\omega t}}{r} - \frac{Pre^{i\omega t}}{2\mu_s}\right\}\right]^2. \end{aligned} \tag{2.73}$$

In the case of the shear component of the traction balance coupling condition, we obtain:

$$-\mu_s \frac{PH_i}{2\mu_s} + \mu_s \frac{PH_i}{2\mu_s} + \frac{c_4}{H_i} = 0,$$

which is equivalent to $c_4 = 0$. Using the kinematic coupling condition, we identify the last integration constant:

$$c_1 = \frac{PH_i^2}{4\mu_f} + i\omega \frac{P(H_o^2 - H_i^2)}{4\mu_s}. \tag{2.74}$$

Transient fluid and quasi-static solid ($\rho_f > 0,\ \rho_s = 0$)
Similar to the previous example, the two integration constants can be found by solving the system of two equations formed by the shear components of the traction balance and the kinematic coupling conditions:

$$\begin{bmatrix} i\mu_f J_{1,f}^* & \mu_s/H_i \\ J_{0,f}^* & i\omega \ln(H_o/H_i) \end{bmatrix} \begin{bmatrix} c_1 \\ c_4 \end{bmatrix} = \begin{bmatrix} PH_i/2 \\ i\omega P(H_o^2 - H_i^2)/(4\mu_s) - P/(\mu_f k_f^2) \end{bmatrix}, \tag{2.75}$$

where $J_{0,f}^* = J_0(ik_f H_i)$ and $J_{1,f}^* = k_f J_1(ik_f H_i)$. The resulting closed-formulation of the two constants are:

$$c_1 = -iP \frac{\omega H_i^2 \ln(H_i/H_o)/2 + \omega(H_o^2 - H_i^2)/4 + i\mu_s/(\mu_f k_f^2)}{\mu_f \omega H_i \ln(H_i/H_o) J_{1,f}^* - \mu_s J_{0,f}^*}, \tag{2.76}$$

$$c_4 = -P \frac{\mu_s J_{0,f}^* H_i/2 + \mu_f \omega J_{1,f}^* (R_o^2 - R_i^2)/4 + i\mu_s J_{1,f}^*/k_f^2}{\mu_f \mu_s \omega \ln(H_i/H_o) J_{1,f}^* - \mu_s^2 J_{0,f}^*/H_i}. \tag{2.77}$$

The general pressure solution in Equation (2.73) continues to apply in this case as well (with $c_3 = -P$), provided one uses the definition of c_4 found in (2.77) rather than $c_4 = 0$.

Quasi-static fluid and transient solid ($\rho_f = 0,\ \rho_s > 0$)
The closed-formulation for the c_4 constant is obtained by rearranging the kinematic condition in (2.62):

$$c_4 = -P\frac{2Y^*_{1,s} + iH_i k_s^2 Y^r_{0,s}}{2\mu_s k_s^2 Y^r_{0,s}\Delta_1}, \tag{2.78}$$

where $J^*_{0,s} = J_0(-k_sH_i)$, $J^*_{1,s} = ik_sJ_1(-k_sH_i)$, $Y^*_{0,s} = Y_0(-k_sH_i)$, $Y^*_{1,s} = ik_sY_1(-k_sH_i)$, $\Delta_0 = J^*_{0,s} - \gamma Y^*_{0,s}$ and $\Delta_1 = J^*_{1,s} - \gamma Y^*_{1,s}$. Using this result and expanding the axial component of the traction balance coupling condition in (2.54), we can also derive the closed-formulation of the last constant:

$$c_1 = \frac{PH_i^2}{4\mu_f} - \frac{i\omega P}{\mu_s k_s^2}(1-\nu_0) - P\omega\frac{2iY^*_{1,s} - H_i k_s^2 Y^r_{0,s}}{2\mu_s k_s^2 Y^r_{0,s}}\frac{\Delta_0}{\Delta_1}, \tag{2.79}$$

where $\nu_k = Y^*_{k,s}/Y^r_{0,s}$ and $k \in \{0,1\}$.

To find the solution for the solid pressure, we expand the more general result in Equation (2.68) using the results of Equation (2.72) and the corollaries, $c_3 = -P$ and $c'(t) = [L+u_s(H_i,t)]\Re\{Pe^{i\omega t}\}$, plus the transient version of the solid displacement formula in Equation 2.70:

$$\begin{aligned}
p_s(r,Z,t) =& \left[L - Z + u_s(H_i,t) - u_s(r,t)\right]\Re\{Pe^{i\omega t}\} - \frac{\mu_s}{3}\left[\frac{\partial u_s(r,t)}{\partial r}\right]^2 \\
=& \left[L - Z - \Re\left\{\frac{P}{\mu_s k_s^2}\left(\frac{Y_0(-k_sr)}{Y^r_{0,s}} - \nu_0\right)e^{i\omega t}\right\}\right]\Re\{Pe^{i\omega t}\} \\
&+ \Re\{c_4\left[\Delta_0 - J_0(-k_sr) + \gamma Y_0(-k_sr)\right]e^{i\omega t}\}\Re\{Pe^{i\omega t}\} \\
&- \frac{\mu_s}{3}\left[\Re\left\{\frac{P}{\mu_s k_s}\frac{Y_1(-k_sr)}{Y^r_{0,s}}e^{i\omega t} + c_4k_s(J_1(-k_sr) - \gamma Y_1(-k_sr))e^{i\omega t}\right\}\right]^2.
\end{aligned} \tag{2.80}$$

Transient fluid and transient solid ($\rho_f, \rho_s > 0$)
In this case, the system of equations resulting from the axial components of the traction and kinematic coupling conditions can be written as:

$$\begin{bmatrix} -\mu_f J^*_{1,f} & \mu_s\Delta_1 \\ -J^*_{0,f} & -i\omega\Delta_0 \end{bmatrix}\begin{bmatrix} c_1 \\ c_4 \end{bmatrix} = \begin{bmatrix} -P\nu_1/k_s^2 \\ -i\omega P(1-\nu_0)/(\mu_s k_s^2) + P/(\mu_s k_f^2) \end{bmatrix}. \tag{2.81}$$

Solving the system gives the following closed forms:

$$c_1 = -\frac{i\omega\Delta_0\nu_1\frac{P}{k_s^2} + \mu_s\Delta_1\left[(1-\nu_0)\frac{i\omega P}{\mu_s k_s^2} + \frac{P}{\mu_f k_f^2}\right]}{\mu_s J^*_{0,f}\Delta_1 - i\omega\mu_f J^*_{1,f}\Delta_0}, \tag{2.82}$$

$$c_4 = -\frac{J^*_{0,f}\nu_1\frac{P}{k_s^2} + \mu_f J^*_{1,f}\left[(1-\nu_0)\frac{i\omega P}{\mu_s k_s^2} + \frac{P}{\mu_f k_f^2}\right]}{\mu_s J^*_{0,f}\Delta_1 - i\omega\mu_f J^*_{1,f}\Delta_0}. \tag{2.83}$$

The analytical solution for the solid pressure retains the form presented in (2.80) with $c_3 = -P$, but using the version of c_4 in (2.83).

2.2.1.3 Two space dimensions - nonlinear case

Again, we consider the nonlinear FSI problem discussed in Section 2.2.1.2 and defined by Equations (2.49) -(2.62)), with the mention that the definition of the first Piola-Kirchhoff stress tensor is now adjusted for the 2D problem:

$$\boldsymbol{P}_s = \frac{\mu_s}{J_s}\left[\boldsymbol{F} - \frac{\boldsymbol{F} : \boldsymbol{F}}{2}\boldsymbol{F}^{-T}\right] - J_s p_s \boldsymbol{F}^{-T}. \tag{2.84}$$

In line to Section 2.2.1.1, we adapt the boundary conditions to the 2D setting and use the same general assumption with regards to the behaviour of the solution. Based on these considerations, it can be shown that the advective term, $\boldsymbol{v}_f \cdot \nabla_{\boldsymbol{x}} \boldsymbol{v}_f$ is null and that, in effect, the fluid problem is identical to the one in the linear case. Thus, the general solutions for the fluid pressure is given by Equations (2.18). Similarly, the general quasi-static and transient fluid velocity solutions are given by Equations (2.18) and (2.20).

To find a general solution for the solid problem, we first observe that the deformation gradient takes the form $\boldsymbol{F} = \boldsymbol{I} + \partial_Y u_s (\boldsymbol{e}_Y \otimes \boldsymbol{e}_X)$, where u_s is the x-component of the deformation field. Thus, we can express the first Piola-Kirchhoff tensor in terms of u_s and p_s as follows:

$$\boldsymbol{P}_s = \begin{bmatrix} -\frac{\mu_s}{2}\left(\frac{\partial u_s}{\partial Y}\right)^2 - p_s & \mu_s \frac{\partial u_s}{\partial Y} \\ \mu_s \frac{\partial u_s}{\partial Y} + \frac{\mu_s}{2}\left(\frac{\partial u_s}{\partial Y}\right)^3 + p_s \frac{\partial u_s}{\partial Y} & -\frac{\mu_s}{2}\left(\frac{\partial u_s}{\partial Y}\right)^2 - p_s \end{bmatrix}. \tag{2.85}$$

Using this result, we expand the momentum balance equation in (2.54) to obtain:

$$\begin{aligned} &\left(-\frac{\partial p_s}{\partial X} + \mu_s \frac{\partial^2 u_s}{\partial Y^2} - \rho_s \partial_{tt} u_s\right) \boldsymbol{e}_X \\ &\quad + \left[\frac{\partial p_s}{\partial X}\frac{\partial u_s}{\partial Y} - \frac{\partial p_s}{\partial Y} - \frac{\mu_s}{2}\frac{\partial}{\partial Y}\left(\frac{\partial u_s}{\partial Y}\right)^2\right] \boldsymbol{e}_Y = \boldsymbol{0}. \end{aligned} \tag{2.86}$$

By applying ∂_X to this vector, we find that $\partial^2 p_s/\partial X^2 = 0$ and $\partial^2 p_s/(\partial X \partial Y) = 0$. Based on these results, we can integrate the $\boldsymbol{e}_y$ component of (2.86) with respect to Y to obtain the general solution for the pressure field:

$$p_s = c_I(t) + c_{II}(t)(X + u_s) - \frac{\mu_s}{2}\left(\frac{\partial u_s}{\partial Y}\right)^2. \tag{2.87}$$

Furthermore, assuming that $u_s = \Re\{\mathrm{u}_s(Y) e^{i\omega t}\}$, $\mathrm{u}_s : [H_i, H_o] \to \mathbb{C}$ and $c_{II}(t) = \Re\{c_3 e^{i\omega t}\}$, the X-component of (2.86) can be simplified:

$$\mu_s \frac{\partial^2 \mathrm{u}_s}{\partial Y^2} + \rho_s \omega^2 \mathrm{u}_s - c_3 = 0. \tag{2.88}$$

Given the fixed wall boundary condition $\mathrm{u}_s(H_o) = 0$, the general solutions of this ODE take the form:

$$\begin{aligned} \mathrm{u}_s &= \frac{c_3}{2\mu_s}(Y^2 - H_o^2) + c_4(Y - H_o), && (\rho_s = 0), \\ \mathrm{u}_s &= \frac{c_3}{\rho_s \omega^2}\left[1 - \sin(k_s Y)\csc(k_s H_o)\right] \\ &\quad + c_4\left[\cos(k_s Y) - \cot(k_s H_o)\sin(k_s Y)\right], && (\rho_s > 0). \end{aligned} \tag{2.89}$$

The last four integration constants, namely c_1, c_3 c_4 and c_I, can only be identified by verifying that the coupling conditions are satisfied. In the following, we consider the different permutations of quasi-static / transient behaviors and the resulting sets of closed formulations for the constants.

Quasi-static fluid and quasi-static solid ($\rho_f = 0,\ \rho_s = 0$)
Let us first consider the expansion of the traction balance coupling condition in Equation (2.61):

$$\left[\mu_f \left.\frac{\partial v_f}{\partial y}\right|_{y=H_i} - \mu_s \left.\frac{\partial u_s}{\partial Y}\right|_{Y=H_i}\right] \boldsymbol{e}_x + \left[\left.\frac{\mu_s}{2}\left(\frac{\partial u_s}{\partial Y}\right)^2\right|_{Y=H_i} + p_s|_{Y=H_i} - p_f|_{y=H_i}\right] \boldsymbol{e}_y = 0.$$

Taking the axial component and replacing the pressure fields with the results in Equation (2.18) and Equation (2.87), we obtain:

$$c_I(t) + \Re\{c_3 e^{i\omega t}\}\,[x + u_s(H_i, t)] - \Re\{P(L-x)e^{i\omega t}\} = 0. \tag{2.90}$$

From this, it can be shown that $c_3 = -P$ and $c_I(t) = \Re\{P[L + u_s(H_i, t)]e^{i\omega t}\}$. Similar to the 3D nonlinear case, these results are valid for all quasi-static and transient permutations due to the fact that the general structure of the pressure solutions is independent from these factors. Thus, we can expand the general nonlinear solid pressure solution in Equation (2.68) to obtain a quasi-static specific version:

$$\begin{aligned} p_s(X,Y,t) &= [L - X + u_s(H_i,t) - u_s]\,\Re\{Pe^{i\omega t}\} - \frac{\mu_s}{2}\left[\frac{\partial \mathrm{u}_s}{\partial Y}\right]^2 \\ &= \left[L - X + \Re\left\{\frac{P}{2\mu_s}(Y^2 - H_i^2)e^{i\omega t} + c_4(H_i - Y)e^{i\omega t}\right\}\right]\Re\{Pe^{i\omega t}\} \\ &\quad - \frac{\mu_s}{2}\left[\Re\left\{c_4 e^{i\omega t} - \frac{PY}{\mu_s}e^{i\omega t}\right\}\right]^2. \end{aligned} \tag{2.91}$$

Furthermore, we can expand the axial component of the traction balance coupling condition using Equation (2.20) and Equation (2.89), resulting in:

$$\mu_f\left(-\frac{P}{2\mu_f}2H_i\right) - \mu_s\left(-\frac{P}{2\mu_s}2H_i + c_4\right) = 0 \quad \Rightarrow \quad c_4 = 0. \tag{2.92}$$

Finally, using the kinematic coupling condition, it can be shown that:

$$c_1 = \frac{PH_i^2}{2\mu_f} + \frac{i\omega P}{2\mu_s}(H_o^2 - H_i^2). \tag{2.93}$$

Transient fluid and quasi-static solid ($\rho_f > 0,\ \rho_s = 0$)
Similar to the preceding case, the x components of the traction and kinematic interface conditions form a system of equations with two unknowns, c_1 and c_4, which can written as follows:

$$\begin{aligned} \beta c_1 - \mu_s c_4 &= -PH_i, \\ \alpha c_1 + i\omega(H_o - H_i)c_4 &= \frac{iP}{\rho_f \omega} + \frac{i\omega P}{2\mu_s}(H_o^2 - H_i^2), \end{aligned}$$

where α and β retain the definitions from Table 2.1. The resulting solutions are:

$$c_1 = \frac{-i\omega(H_o - H_i)H_i P + \mu_s \left[\frac{iP}{\rho_f \omega} + \frac{i\omega P}{2\mu_s}(H_o^2 - H_i^2)\right]}{i\omega\beta(H_o - H_i) + \mu_s \alpha}, \tag{2.94}$$

$$c_4 = \frac{\alpha H_i P + \beta \left[\frac{iP}{\rho_f \omega} + \frac{i\omega P}{2\mu_s}(H_o^2 - H_i^2)\right]}{i\omega\beta(H_o - H_i) + \mu_s \alpha}. \tag{2.95}$$

The general solution for the quasi-static solid pressure (with $c_3 = -P$) given by Equation (2.91) applies in this case as well.

Quasi-static fluid and transient solid ($\rho_f = 0,\ \rho_s > 0$)
Analogous to the transient fluid and quasi-static solid case, the system of equations takes the form:

$$\mu_s k_s \xi_1 c_4 = \frac{\mu_s k_s \zeta_1 P}{\rho_s \omega^2} + P H_i,$$
$$c_1 + i\omega\xi_2 c_4 = \frac{P H_i^2}{2\mu_f} - \frac{i\zeta_2 P}{\rho_s \omega},$$

where for convenience we introduced a new set of parameters:

$$\xi_1 = \sin(k_s H_i) + \cot(k_s H_o)\cos(k_s H_i), \tag{2.96}$$
$$\xi_2 = \cot(k_s H_o)\sin(k_s H_i) - \cos(k_s H_i), \tag{2.97}$$
$$\zeta_1 = \csc(k_s H_o)\cos(k_s H_i), \tag{2.98}$$
$$\zeta_2 = 1 - \sin(k_s H_i)\csc(k_s H_o). \tag{2.99}$$

Solving this system of equations gives the following closed forms for the integration constants:

$$c_1 = \frac{P H_i^2}{2\mu_f} - \frac{i\zeta_2 P}{\rho_s \omega} - i\xi_2 \frac{\mu_s k_s \zeta_1 P + \rho_s \omega^2 P H_i}{\mu_s k_s \rho_s \omega \xi_1}, \tag{2.100}$$

$$c_4 = \frac{\mu_s k_s \zeta_1 P + \rho_s \omega^2 P H_i}{\mu_s k_s \rho_s \omega^2 \xi_1}. \tag{2.101}$$

The general solution for the transient solid pressure is obtained by expanding the more general form in Equation (2.87) using $c_3 = -P$ and $c_I(t) = \Re\{P[L + u_s(H_i, t)]e^{i\omega t}\}$ and the general transient solid displacement formula in Equation (2.89):

$$\begin{aligned} p_s(X,Y,t) &= [L - X + u_s(H_i,t) - u_s]\,\Re\{Pe^{i\omega t}\} - \frac{\mu_s}{2}\left[\frac{\partial u_s}{\partial r}\right]^2 \\ &= \left[L - X + \Re\left\{\frac{P}{\rho_s\omega^2}\left(1 - \sin(k_s Y)\csc(k_s H_o) - \zeta_2\right)e^{i\omega t}\right\}\right]\Re\{Pe^{i\omega t}\} \\ &\quad - \Re\left\{c_4[\cos(k_s Y) - \cot(k_s H_o)\sin(k_s Y) + \xi_2]e^{i\omega t}\right\}\Re\{Pe^{i\omega t}\} \\ &\quad - \frac{\mu_s}{2}\left[\Re\left\{\frac{P k_s}{\rho_s\omega^2}\cos(k_s Y)\csc(k_s H_o)e^{i\omega t} - c_4 k_s(\sin(k_s Y) + \cot(k_s H_o)\cos(k_s Y))e^{i\omega t}\right\}\right]^2. \end{aligned}$$

This formula also applies to the transient fluid and transient solid case provided the appropriate c_4 formulation is used.

Transient fluid and transient solid $(\rho_f, \rho_s > 0)$
The equivalent system of equations corresponding to this case can be written as follows:

$$\beta c_1 + \mu_s k_s \xi_1 c_4 = \frac{\zeta_1 P}{k_s},$$
$$\alpha c_1 + i\omega \xi_2 c_4 = \frac{iP}{\rho_f \omega} - \frac{i\zeta_2 P}{\rho_s \omega},$$

and the resulting closed form solutions are:

$$c_1 = \frac{iP}{\rho_s \rho_f \omega k_s} \frac{\omega^2 \xi_2 \zeta_1 \rho_s \rho_f - \mu_s k_s^2 \xi_1 (\rho_s - \zeta_2 \rho_f)}{i\omega \xi_2 \beta - \alpha \mu_s k_s \xi_1}, \tag{2.102}$$

$$c_4 = \frac{P}{\rho_s \rho_f \omega k_s} \frac{-\omega \alpha \zeta_1 \rho_s \rho_f + i\beta k_s (\rho_s - \zeta_2 \rho_f)}{i\omega \xi_2 \beta - \alpha \mu_s k_s \xi_1}. \tag{2.103}$$

2.2.1.4 Three space dimensions - linear case

In this section, we approach the derivation of the FSI problem described in Equations (2.1) - (2.15), but in 3D and using the assumptions about the solution behaviour which were described in Section 2.2.1.2, the 3D nonlinear case. In terms of deriving the general analytical solutions, it should be noted that in the 3D nonlinear case, we showed that advection term is constantly zero as a consequence of our assumptions. Consequently, the solutions described in Equations (2.64) and (2.65) which define the general solution for the pressure, quasi-static velocity and transient velocity, respectively, are still applicable in this context.

We begin the derivation of the solid solution, by observing that the the displacement gradient takes the form $\nabla \boldsymbol{u}_s = \frac{\partial u_s}{\partial r} \boldsymbol{e}_r \otimes \boldsymbol{e}_z$. Thus, the Cauchy stress tensor can be written as:

$$\boldsymbol{\sigma}_s = \mu_s u_s \left(\boldsymbol{e}_r \otimes \boldsymbol{e}_z + \boldsymbol{e}_z \otimes \boldsymbol{e}_r \right) - p\boldsymbol{I}.$$

This can now be used to expand the solid momentum balance equation in (2.7) into the the three cylindrical coordinates:

$$-\frac{\partial p_s}{\partial r} \boldsymbol{e}_r - \frac{1}{r} \frac{\partial p_s}{\partial \theta} \boldsymbol{e}_\theta + \left(\mu_s \frac{\partial^2 u_s}{\partial r^2} + \frac{\mu_s}{r} \frac{\partial u_s}{\partial r} - \frac{\partial p_s}{\partial z} - \rho_s \partial_{tt} u_s \right) \boldsymbol{e}_z = \boldsymbol{0}. \tag{2.104}$$

The radial and angular components show that the pressure field is invariant in these directions. Furthermore, by taking the axial derivative of the $\boldsymbol{e}_z$ component, it can be shown that $\partial^2 p_s / \partial z^2 = 0$. Based on these three observations and also taking note of our previous assumptions (i.e. spatial and temporal components are separable, the field is temporally periodic and described by a single harmonic), we can write the general solid pressure solution as:

$$p_s = \Re\{P_s(L - z)e^{i\omega t}\}, \tag{2.105}$$

where $P_s \in \mathbb{C}$ is the reference pressure over the length of the domain. Substituting this into the axial component of the momentum balance and also using the fact that $u_s = \Re\{\mathrm{u}_s e^{i\omega t}\}$, we find the following nonhomogeneous PDE for the displacement:

$$\mu_s \frac{\partial^2 u_s}{\partial r^2} + \frac{\mu_s}{r} \frac{\partial u_s}{\partial r} + \rho_s \omega^2 u_s = -P_s, \tag{2.106}$$

which is identical to Equation (2.69), with the only difference that in that equation the nonhomogenous term, i.e. $-P_s$, was unknown. Consequently, the two versions of the displacement function, quasi-static and transient, that resulted in the previous section, also apply in this case:

$$\begin{aligned} \mathrm{u}_s &= \frac{P_s}{4\mu_s}(H_o^2 - r^2) + c_3 \ln\left(\frac{r}{H_o}\right), && (\rho_s = 0), \\ \mathrm{u}_s &= -\frac{P_s}{\rho_s \omega^2}\left[1 - \frac{Y_0(-k_s r)}{Y_{0,s}^r}\right] + c_3\left[J_0(-k_s r) - \gamma Y_0(-k_s r)\right], && (\rho_s > 0). \end{aligned} \tag{2.107}$$

The last two integration constants (i.e. c_1 and c_3) are found by verifying that the coupling conditions are satisfied. In the following, we derive the closed form of these constants, while taking into account all four quasi-static / transient permutations.

Quasi-static fluid and quasi-static solid ($\rho_f = \rho_s = 0$)
First, let us expand the traction balance interface condition into its three components:

$$(-p_f + p_s)\,\boldsymbol{e}_r + \left(\mu_f \left.\frac{\partial v_f}{\partial r}\right|_{y=H_i} - \mu_s \left.\frac{\partial u_s}{\partial r}\right|_{y=H_i}\right)\boldsymbol{e}_z = \mathbf{0}. \tag{2.108}$$

Using the radial component, it is trivial to show that $P = P_f = P_s$; a result which applies to all quasi-static and transient permutations. In the case of the axial component, incorporating the results of Equations (2.65) and 2.107 yields:

$$-\mu_f \frac{PH_i}{2\mu_f} + \mu_s \frac{PH_i}{2\mu_s} + \frac{\mu_s c_3}{H_i} = 0 \quad \Rightarrow \quad c_3 = 0. \tag{2.109}$$

Finally, we expand the kinematic condition to obtain:

$$\left(-\frac{PH_i^2}{4\mu_f} + c_1\right) - i\omega \frac{P(H_o^2 - H_i^2)}{4\mu_s} = 0, \tag{2.110}$$

which can be rearranged into the closed form of the last integration constant:

$$c_1 = \frac{PH_i^2}{4\mu_f} + i\omega \frac{P(H_o^2 - H_i^2)}{4\mu_s}. \tag{2.111}$$

Transient fluid and quasi-static solid ($\rho_f > 0,\ \rho_s = 0$)
Similar to the previous case, the last two constants, c_1 and c_3, can be identified by solving the system formed by the axial components of the traction balance and kinematic interface conditions:

$$\begin{aligned} -i\mu_f J_{1,f}^* c_1 - \frac{\mu_s c_3}{H_i} &= -\frac{PH_i}{2} \\ J_{0,f}^* c_1 - i\omega \ln\left(\frac{H_i}{H_o}\right) c_3 &= \frac{i\omega P}{4\mu_s}(H_o^2 - H_i^2) - \frac{P}{\mu_f k_f^2}. \end{aligned}$$

The resulting closed forms are:

$$c_1 = \frac{i\omega \ln\left(\frac{H_i}{H_o}\right)\frac{PH_i}{2} + \frac{i\omega P}{4H_i}(H_o^2 - H_i^2) - \frac{\mu_s P}{\mu_f k_f^2 H_i}}{-\omega\mu_f J_{1,f}^* \ln(\frac{H_i}{H_o}) + J_{0,f}^* \frac{\mu_s}{H_i}}, \tag{2.112}$$

$$c_3 = \frac{J_{0,f}^* \frac{PH_i}{2} + J_{1,f}^* \frac{\mu_f \omega P}{4\mu_s}(H_o^2 - H_i^2) + J_{1,f}^* \frac{iP}{k_f^2}}{-\omega\mu_f J_{1,f}^* \ln(\frac{H_i}{H_o}) + J_{0,f}^* \frac{\mu_s}{H_i}}. \tag{2.113}$$

Quasi-static fluid and transient solid ($\rho_f = 0,\ \rho_s > 0$)
Rewriting the $\boldsymbol{e}_z$ component of the traction balance coupling condition using the general solutions for fluid velocity (2.65) and solid displacement (2.107), yields:

$$-\frac{PH_i}{2} + \mu_s \left[i\frac{P\nu_1}{\rho_s\omega^2} + i\Delta_1 c_3 \right] = 0. \tag{2.114}$$

When rearranged, we obtain the closed form of the c_3 constant:

$$c_3 = \frac{P}{i\Delta_1}\left(\frac{H_i}{2\mu_s} - \frac{i\nu_1}{\rho_s\omega^2} \right). \tag{2.115}$$

Finally, we expand the kinematic condition, which can be written as:

$$-\frac{PH_i^2}{4\mu_f} + c_1 - i\omega \left[-(1-\nu_0)\frac{P}{\rho_s\omega^2} + \Delta_0 c_3 \right] = 0. \tag{2.116}$$

Based on this, we find the closed form of the last unknown integration constant:

$$c_1 = \frac{PH_i^2}{4\mu_f} - i\omega P \left[\frac{1-\nu_0}{\rho_s\omega^2} + i\frac{\Delta_0}{\Delta_1}\left(\frac{H_i}{2\mu_s} - i\frac{\nu_1}{\rho_s\omega^2} \right) \right]. \tag{2.117}$$

Transient fluid and transient solid ($\rho_f, \rho_s > 0$)
Finally, we treat the case where both solid and fluid have transient behaviours. As in the transient fluid / quasi-static solid case, c_1 and c_3 can be found by solving the system given by the $\boldsymbol{e}_z$-components of the traction and kinematic interface conditions:

$$\begin{aligned} -i\mu_f J^*_{1,f} c_1 + i\mu_s \Delta_1 c_3 &= -i\nu_1 \frac{\mu_s P}{\rho_s\omega^2}, \\ J^*_{0,f} c_1 - i\omega\Delta_0 c_3 &= (\nu_0 - 1)\frac{iP}{\rho_s\omega} - \frac{P}{\mu_f k_f^2}. \end{aligned}$$

The resulting closed form solutions are:

$$c_1 = \frac{-\nu_1\Delta_0 \frac{\mu_s P}{\rho_s\omega} - \mu_s(1-\nu_0)\Delta_1 \frac{P}{\rho_s\omega} + i\mu_s\Delta_1 \frac{P}{\mu_f k_f^2}}{-\mu_f J^*_{1,f}\omega\Delta_0 - i\mu_s J^*_{0,f}\Delta_1}, \tag{2.118}$$

$$c_3 = \frac{i\nu_1 J^*_{0,f} \frac{\mu_s P}{\rho_s\omega^2} - \mu_f(1-\nu_0) J^*_{1,f} \frac{P}{\rho_s\omega} + i J^*_{1,f} \frac{P}{k_f^2}}{-\mu_f J^*_{1,f}\omega\Delta_0 - i\mu_s J^*_{0,f}\Delta_1}. \tag{2.119}$$

2.3 Numerical fluid-structure interaction implementation

As a first use of the benchmark and as guide to future applications and results reporting, we also carried out a validation test of our finite element implementation [67, 94]. This section presents a brief review of the structure of the weak form and discretization scheme employed to solve different model permutations (presented later in Section 2.4).

2.3.1 Finite element weak form

The fluid-structure coupling strategy used in this work is based on the technique described in [63, 67, 110]. In this approach, a Lagrange multiplier variable, $\boldsymbol{\lambda} = \boldsymbol{t}_f = -\boldsymbol{t}_s$, is introduced such that it weakly enforces both the kinematic and dynamic constraints, resulting in an FSI system which is solved monolithically. This strategy allows us to choose the domain discretization and polynomial interpolation schemes which are more appropriate for the representation of either fluid and solid solutions. In the following, we outline how this applies to our case.

The general discrete weak form including the FSI coupling conditions can be written as follows: Find $\boldsymbol{s}^n := \left(\boldsymbol{v}_f^n, \boldsymbol{v}_s^n, \boldsymbol{\lambda}^n, p_f^n, p_s^n\right) \in \boldsymbol{\mathcal{S}}_D^h := \boldsymbol{\mathcal{V}}_0^h \times \boldsymbol{\mathcal{U}}_0^h \times \boldsymbol{\mathcal{M}}_D^h \times \mathcal{W}_f^h \times \mathcal{W}_s^h$, such that for every $\boldsymbol{d} := (\boldsymbol{w}_f, \boldsymbol{w}_s, \boldsymbol{q}_\lambda, q_f, q_s) \in \boldsymbol{\mathcal{S}}_0^h := \boldsymbol{\mathcal{V}}_0^h \times \boldsymbol{\mathcal{U}}_0^h \times \boldsymbol{\mathcal{M}}_0^h \times \mathcal{W}_f^h \times \mathcal{W}_s^h$:

$$
\begin{aligned}
R(\boldsymbol{s}^n, \boldsymbol{s}^{n-1}, \boldsymbol{d}) := & \int_{\Omega_{f,h}^n} \rho_f \left[\frac{\boldsymbol{v}_f^n - \boldsymbol{v}_f^{n-1}}{\Delta_t^n} + \delta_{nl} \left(\boldsymbol{v}_f^n - \hat{\boldsymbol{v}}_f^n \right) \cdot \nabla_{\boldsymbol{x}} \boldsymbol{v}_f^n \right] \cdot \boldsymbol{w}_f \, d\Omega \\
& + \int_{\Omega_{f,h}^n} \boldsymbol{\sigma}_f^n : \nabla_{\boldsymbol{x}} \boldsymbol{w}_f + q_f \nabla_{\boldsymbol{x}} \cdot \boldsymbol{v}_f^n \, d\Omega \\
& - \int_{\Gamma_{f,h}^{I,n} \cup \Gamma_{f,h}^{O,n}} \boldsymbol{t}_f^n \cdot \boldsymbol{w}_f \, d\Omega \\
& + \int_{\Omega_{s,h}^0} \delta_{nl} \left[J_s^n \rho_s \frac{\boldsymbol{v}_s^n - \boldsymbol{v}_s^{n-1}}{\Delta_t^n} + \boldsymbol{P}_s^n : \nabla_{\boldsymbol{X}} \boldsymbol{w}_s + q_s (J_s^n - 1) \right] d\Omega \\
& + \int_{\Omega_{s,h}^0} (1 - \delta_{nl}) \left[\rho_s \frac{\boldsymbol{v}_s^n - \boldsymbol{v}_s^{n-1}}{\Delta_t^n} + \boldsymbol{\sigma}_s^n : \nabla_{\boldsymbol{X}} \boldsymbol{w}_s + q_s \nabla_{\boldsymbol{X}} \cdot \boldsymbol{v}_s^n \right] d\Omega \\
& - \int_{\Gamma_{s,h}^{I,n} \cup \Gamma_{s,h}^{O,n}} \boldsymbol{t}_s^n \cdot \boldsymbol{w}_s \, d\Omega \\
& + \int_{\Omega_{\lambda,h}^0} \boldsymbol{\lambda}^n \cdot (\boldsymbol{w}_f - \boldsymbol{w}_s) + \boldsymbol{q}_\lambda \cdot \left(\boldsymbol{v}_f^n - \boldsymbol{v}_s^n \right) \, d\Omega = 0,
\end{aligned} \tag{2.120}
$$

where the δ_{nl} term acts as a switch between the linear and nonlinear cases. The current displacement is defined as $\boldsymbol{u}_s^n = \boldsymbol{u}_s^{n-1} + \Delta_t^n \boldsymbol{v}_s^n$ with the length of the current time step Δ_t^n. Further, $\boldsymbol{t}_f^n$ and $\boldsymbol{t}_s^n$ are the discrete representations of the analytic solutions of the surface tractions acting on the inflow and outflow surfaces of the fluid and solid, respectively.

In the nonlinear cases, where the ALE term is active, the weak form is expanded to include the fluid displacement problem, which takes the form: Find $\hat{\boldsymbol{v}}_f^n \in \boldsymbol{\mathcal{W}}_D^h$ such that for every $\boldsymbol{z} \in \boldsymbol{\mathcal{W}}_0^h$:

$$
R_g(\hat{\boldsymbol{v}}_f^n, \hat{\boldsymbol{v}}_f^{n-1}, \boldsymbol{z}) := \int_{\Omega_{f,h}^0} \delta_{nl} \left[\frac{\hat{\boldsymbol{v}}_f^n - \hat{\boldsymbol{v}}_f^{n-1}}{\Delta_t^n} \cdot \boldsymbol{z} - \boldsymbol{\Phi} \nabla_{\boldsymbol{X}} \hat{\boldsymbol{v}}_f^n : \nabla_{\boldsymbol{X}} \boldsymbol{z} \right] d\Omega = 0. \tag{2.121}
$$

The newly introduced field, $\hat{\boldsymbol{v}}_f^n$, represents the artificial grid velocity and is used to account for the motion of the domain in the ALE formulation. This deformation is preferentially weighted in the axial direction by means of the diffusion field $\boldsymbol{\Phi}$ in order to avoid radial variations.

The definitions of the function spaces are:

$$
S^k(\Omega_{i,h}^0) = \{ f : \Omega_{i,h}^0 \to \mathbb{R} \mid f \in \mathcal{C}^0(\bar{\Omega}_{i,h}^0), \; f|_{\tau_e} \in \mathbb{P}^k(\tau_e), \; \forall \, \tau_e \in \mathcal{T}_i^h \}, \tag{2.122}
$$

which represent the general continuous k^{th}-order piecewise polynomial spaces defined on $\Omega^0_{i,h}$. Consequently, we can define:

$$\boldsymbol{\mathcal{V}}^h = \left[S^2(\Omega^0_{f,h})\right]^d, \qquad \boldsymbol{\mathcal{U}}^h = \left[S^2(\Omega^0_{s,h})\right]^d, \qquad \boldsymbol{\mathcal{M}}^h = \left[S^2(\Gamma^\lambda_{f,h})\right]^d,$$
$$\mathcal{W}^h_f = S^1(\Omega^0_{f,h}), \qquad \mathcal{W}^h_s = S^1(\Omega^0_{s,h}).$$

In the case of solid and fluid domain velocity, further restrictions are applied on their respective spaces in order to incorporate the Dirichlet and homogeneous boundary conditions:

$$\boldsymbol{\mathcal{V}}^h_0 = \{\boldsymbol{v} \in \boldsymbol{\mathcal{V}}^h \mid \boldsymbol{v}_\perp = \boldsymbol{0} \text{ on } \Gamma^{I,n}_{f,h} \cup \Gamma^{O,n}_{f,h}\}, \tag{2.123}$$
$$\boldsymbol{\mathcal{U}}^h_0 = \{\boldsymbol{v} \in \boldsymbol{\mathcal{U}}^h \mid \boldsymbol{v}_\perp = \boldsymbol{0} \text{ on } \Gamma^I_{s,h} \cup \Gamma^O_{s,h} \text{ and } \boldsymbol{v} = \boldsymbol{0} \text{ on } \Gamma^W_{s,h}\}, \tag{2.124}$$
$$\boldsymbol{\mathcal{W}}^h_D = \{\boldsymbol{v} \in \boldsymbol{\mathcal{V}}^h \mid \boldsymbol{v} = \boldsymbol{v}_f \text{ on } \Gamma^\lambda_h\}, \tag{2.125}$$
$$\boldsymbol{\mathcal{W}}^h_0 = \{\boldsymbol{v} \in \boldsymbol{\mathcal{V}}^h \mid \boldsymbol{v} = \boldsymbol{0} \text{ on } \Gamma^\lambda_h\}, \tag{2.126}$$

and similarly for the Lagrange multiplier,

$$\boldsymbol{\mathcal{M}}^h_D = \{\boldsymbol{\lambda} \in \boldsymbol{\mathcal{M}}^h \mid \boldsymbol{\lambda}_\perp = \boldsymbol{t}^n_{h,\perp} \text{ on } \left(\Gamma^\lambda_h \cap \Gamma^I_{f,h}\right) \cup \left(\Gamma^\lambda_h \cap \Gamma^O_{f,h}\right)\}, \tag{2.127}$$
$$\boldsymbol{\mathcal{M}}^h_0 = \{\boldsymbol{\lambda} \in \boldsymbol{\mathcal{M}}^h \mid \boldsymbol{\lambda}_\perp = \boldsymbol{0} \text{ on } \left(\Gamma^\lambda_h \cap \Gamma^I_{f,h}\right) \cup \left(\Gamma^\lambda_h \cap \Gamma^O_{f,h}\right)\}. \tag{2.128}$$

Here, we used the $\perp$ symbol to indicate more generally the components of $\boldsymbol{v}$ and $\boldsymbol{\lambda}$ which are perpendicular to the flow direction, such as $\hat{\boldsymbol{e}}_y$ in two dimensions, as well as $\hat{\boldsymbol{e}}_x$ and $\hat{\boldsymbol{e}}_y$ in the three-dimensional case.

2.4 Numerical experiments

2.4.1 Numerical validation of analytic solutions

The 16 analytic solutions presented in Section 2.2.1 were validated numerically. In the linear two dimensional case (Section 2.2.1.1), numerical differentiation was employed to verify that the analytic solutions satisfy the (strong form) momentum and mass balance (i.e. Equation (2.1) / (2.2) and Equation (2.7) / (2.8)), the boundary conditions, and the dynamic and kinematic coupling constraints (i.e. Equation (2.14) / (2.15)). All other analytic solutions were validated similarly.

2.4.2 Numerical experiments: Space-time discretization

In Section 2.5, transient and linear fluid / solid models are considered in two dimensions, and transient and nonlinear fluid / solid models are considered in three dimensions. In what follows, details about the space-time discretization of the computational domains are discussed.

2.4.2.1 Linear FSI in two dimensions

For the linear FSI case in two dimensions, the channel length and width are set to $L = H_i = 1$ and the wall thickness is set to $H_o - H_i = 0.2$, see Figure 2.1. The temporal cycle length is selected as $T = 1.024$ and the length of the time domain is set as $10T$.

The fluid and solid domains are discretized using quadrilateral elements with quadratic (and linear) interpolation for the velocity / displacement (and pressure) variables and the Lagrange multiplier domain is discretized using line elements with quadratic interpolation. Three different spatial refinement levels are considered, and referred to as *coarse*, *medium* and *fine*, see Table 2.2. The temporal domain is discretized using equidistant time points and seven different time step sizes, such that we may consider temporal refinement for a given spatial refinement level as well as three space-time refinement levels with $\delta_t/\delta_x^3 = \text{const}$.

	2D	3D
Time step size	$\delta_t \in \{0.064, 0.032, \ldots, 0.001\}$	$\delta_t \in \{0.04096, 0.02048, \ldots, 0.00064\}$
Number of time steps	$N_t \in \{160, 320, \ldots, 10240\}$	$N_t \in \{175, 350, \ldots, 11200\}$
Spatial step size	$\delta_x, \delta_y \in \{0.1, 0.05, 0.025\}$	$\delta_x, \delta_y, \delta_z \in \{0.314, 0.157, 0.079\}$
Number of spatial DOFs	$N_x \in \{1288, 4728, 18088\}$	$N_x \in \{32793, 245009, 1679243\}$

Table 2.2: *The spatial and temporal step sizes, $\delta_x, \delta_y, \delta_z$ and δ_t, and numbers of spatial and temporal degrees-of-freedom (DOFs), N_x and N_t, for the numerical experiments. For the 3D case, the spatial step size is the approximate step size at the fluid / solid interface.*

2.4.2.2 Nonlinear FSI in three dimensions

In the nonlinear FSI case in three dimensions, the fluid domain length and radius are set to $L = 1$ and $H_i = 0.7$ and the wall thickness is set to $H_o - H_i = 0.3$, see Figure 2.2. To reduce the computational cost, we only consider one quarter of the domain in numerical experiments (i.e. for $x, y \geq 0$) and set the length of the time domain to $7T$ with cycle length $T = 1.024$.

The fluid domain is discretized using tetrahedral elements (quadratic-linear interpolation for velocity and pressure), the solid domain using hexahedral elements (quadratic-linear interpolation for velocity / displacement and pressure) and the Lagrange multiplier domain using triangle elements (quadratic interpolation).

Similar to the linear FSI case in two dimensions, three different refinement levels are considered, see Figure 2.3. Since the analytic solution for the solid pressure variable (see Equation (2.68)) is now a function of the displacement (in contrast to the linear case), the mesh for the solid domain is more refined in the radial direction than in the axial and circumferential directions.

The coarse solid mesh has 5 elements in the circumferential direction, 8 elements in the radial direction and 8 elements in the axial direction. The coarse fluid mesh is selected, such that four tetrahedral element faces conform with one hexahedral element face at the interface boundary. Tetrahedral element sizes are approximately constant throughout the fluid domain. The corresponding triangular mesh for the interface domain is embedded in the interface boundary of the fluid mesh and each triangle element conforms with a tetrahedral mesh face of the fluid mesh.

The medium and fine refinement levels are achieved by using a uniform refinement of the solid domain and an approximately uniform refinement of the fluid domain. For all refinement levels, we note that all solid, interface and fluid interface nodes (including

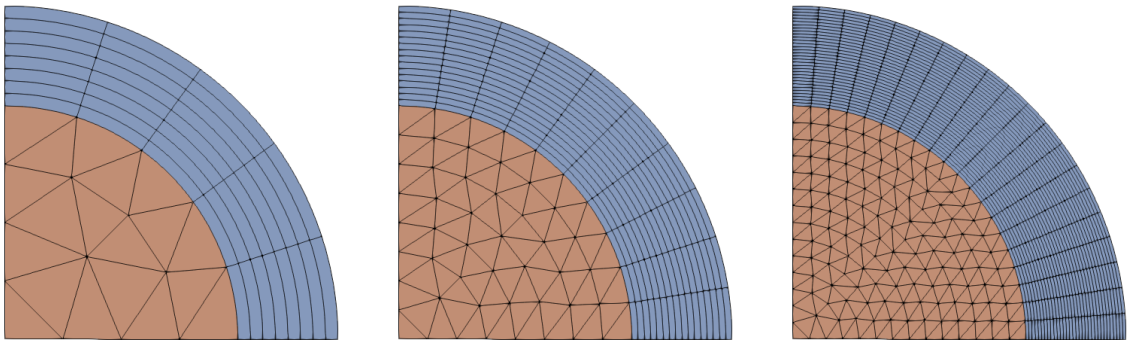


Figure 2.3: Coarse, medium *and* fine *mesh refinement levels for simulating the nonlinear transient FSI case in three dimensions.*

midpoint nodes for quadratic elements) lay exactly on circles with radius H_i around the z-axis. We select time step sizes (see Table 2.2), such that we may consider temporal refinement for a given spatial mesh, as well as three space-time refinement levels with $\delta_t/\delta_x^3 = \text{const}$.

2.4.3 Numerical solution

The linear and nonlinear transient FSI cases are used to validate the CHeart [94] implementation of the FSI method detailed in Section 2.3.1. CHeart is based on the matrix solver MUMPS [2]. A Newton-Raphson-Shimanskii solver [129] is employed to reduce the computational cost by reusing the Jacobian matrix (and its inverse) as long as the residual norm could be reduced by a factor of 3/4 [67].

Complex expression evaluators were implemented to set initial and boundary conditions, according to the settings in Equation (2.1) - (2.15) and Equation 2.49 - (2.62), respectively. We further evaluated the analytic FSI solutions in CHeart after each time step to compute the space-time error on-the-fly.

2.4.4 Space-time norm

To measure the space-time error for a particular discrete variable $\mathbf{f} \in \mathbb{R}^{n_c n_d}$ with n_c components and n_d dimensions, we compute the error over one cycle,

$$\|\mathbf{f}\|_{\Omega_i^O \times \Omega_t, 2} = \sqrt{\int_{\Omega_t} \int_{\Omega_i^O} \mathbf{f} : \mathbf{f} \;\; d\Omega_i^O \; dt}, \tag{2.129}$$

where Ω_i^O (for $i \in \{f, s, \lambda\}$) denotes the discrete spatial domain that the variable $\mathbf{f}$ is integrated over and Ω_t denotes the respective discrete time domain. In particular, $\Omega_t = [T_{final} - T, T_{final}]$ corresponds to the final simulated cycle. Here, we choose $T_{final} = 10T$ and $T_{final} = 7T$ for the linear and nonlinear FSI cases, respectively. Further, the time integral in Equation 2.129 is approximated as piecewise constant, whereas the spatial integration is approximated using the corresponding test functions of $\mathbf{f}$.

2.5 Results

This section considers the transient fluid and transient linear solid model in two spatial dimensions and the transient fluid and transient nonlinear solid model in three spatial dimensions. In the following sections, we present the velocity, displacement and pressure solutions for a parameter set that is suitable for space-time convergence analysis (Section 2.5.1 and 2.5.2) and for a physiologically relevant parameter set (Section 2.5.3). In Section 2.5.4, we demonstrate the use of the analytic solutions for validating a non-conforming monolithic FSI method (see Section 2.3.1) that exhibits first-order convergence in time.

2.5.1 Transient fluid and transient linear solid in 2D

To illustrate the linear FSI case in two spatial dimensions, we consider the following material parameters: the fluid density and viscosity are selected as $\rho_f = 1$ and $\mu_f = 0.01$, and the solid density and stiffness are chosen as $\rho_s = 1$ and $\mu_s = 0.1$. We further set the pressure amplitude as $P = 1$. Here, the Womersley number is,

$$W = H_i\sqrt{\omega\rho_f/\mu_f} = \sqrt{2\pi/(1.024 \cdot 0.01)} \approx 24.77, \tag{2.130}$$

and the Reynolds number is,

$$Re = 2\rho_f V_f H_i/\mu_f \approx \frac{0.44 \cdot 2}{0.01} \approx 88.32, \tag{2.131}$$

with $V_f = \max_{y,t} |v_f(y,t)| \approx 0.4416$.

Figure 2.4 illustrates the temporal variation of the velocity, displacement and pressure solution profiles along the y-axis. It further highlights the temporal variation of the fluid and solid pressure solutions at the inlet ($x = 0$), the midway point in the x-direction ($x = L/2$) and the outlet ($x = L$). In the linear case, the fluid and solid pressure solution is linear along the x-axis, continuous at the coupling boundary and constant along the y-axis. Figure 2.4 further highlights that $p_f(x,t) = p_s(x,t) = \Re\{P(L-x)e^{i\omega t}\} = P(L-x)cos(\omega t)$.

The velocity and displacement solutions are constant along the x-axis, but varying in y. The fluid velocity solution resembles a typical Womersley profile over time and the maximum fluid velocity occurs at the interface, where it matches the deformation rate of the solid due to the kinematic coupling condition. The solid deformation rate at the wall (i.e. at $y = H_o$) is zero, similar to the displacement, as enforced by the Dirichlet boundary condition. The maximum absolute displacement U_s is related to the maximum absolute deformation rate V_s by approximately $U_s \approx V_s/(2\pi)$.

2.5.2 Transient fluid and transient nonlinear solid in 3D

To illustrate the transient nonlinear FSI solution in three dimensions, the fluid density and viscosity were selected as $\rho_f = 2.1$ and $\mu_f = 0.03$, and the solid density and stiffness were chosen as $\rho_s = 1$ and $\mu_s = 0.1$. We further set the pressure amplitude as $P = 1$. Compared to Section 2.5.1, the Womersley number, $W \approx 14.51$, and the Reynolds number, $Re \approx 8.036$ (with $V_f = \max_{r,t} |v_f(r,t)| \approx 0.082$) are smaller.

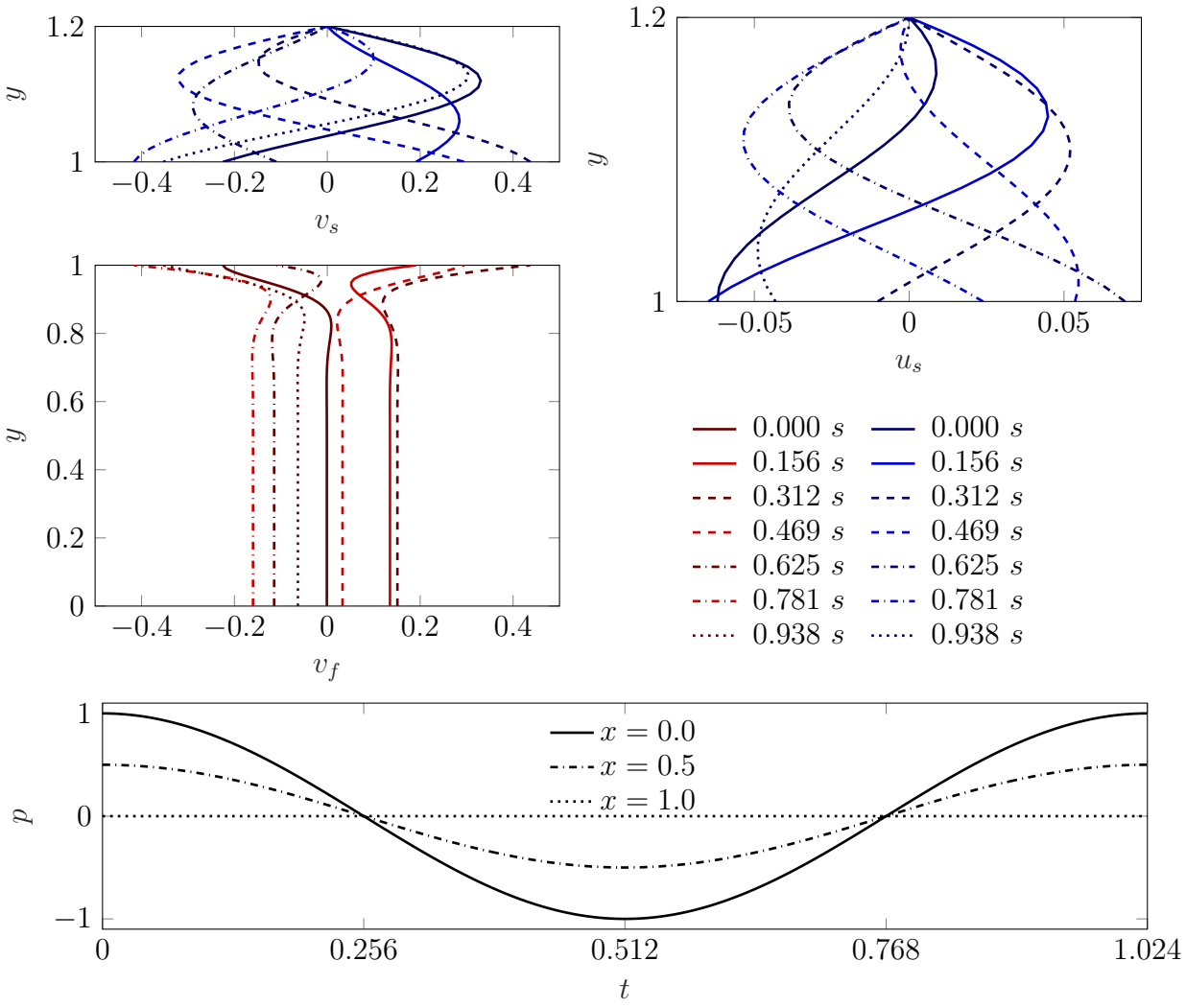


Figure 2.4: *The analytic solution for the transient linear FSI case in two dimensions with density $\rho_f = \rho_s = 1$, fluid viscosity $\mu_f = 0.01$, solid stiffness $\mu_s = 0.1$ and cycle length $T = 1.024$: Fluid and solid velocity, v_f (left) and v_s (top left), and solid displacement, u_s (top right), along the y-axis, and fluid / solid pressure, p_f and p_s (bottom), over time t at three positions along the x-axis.*

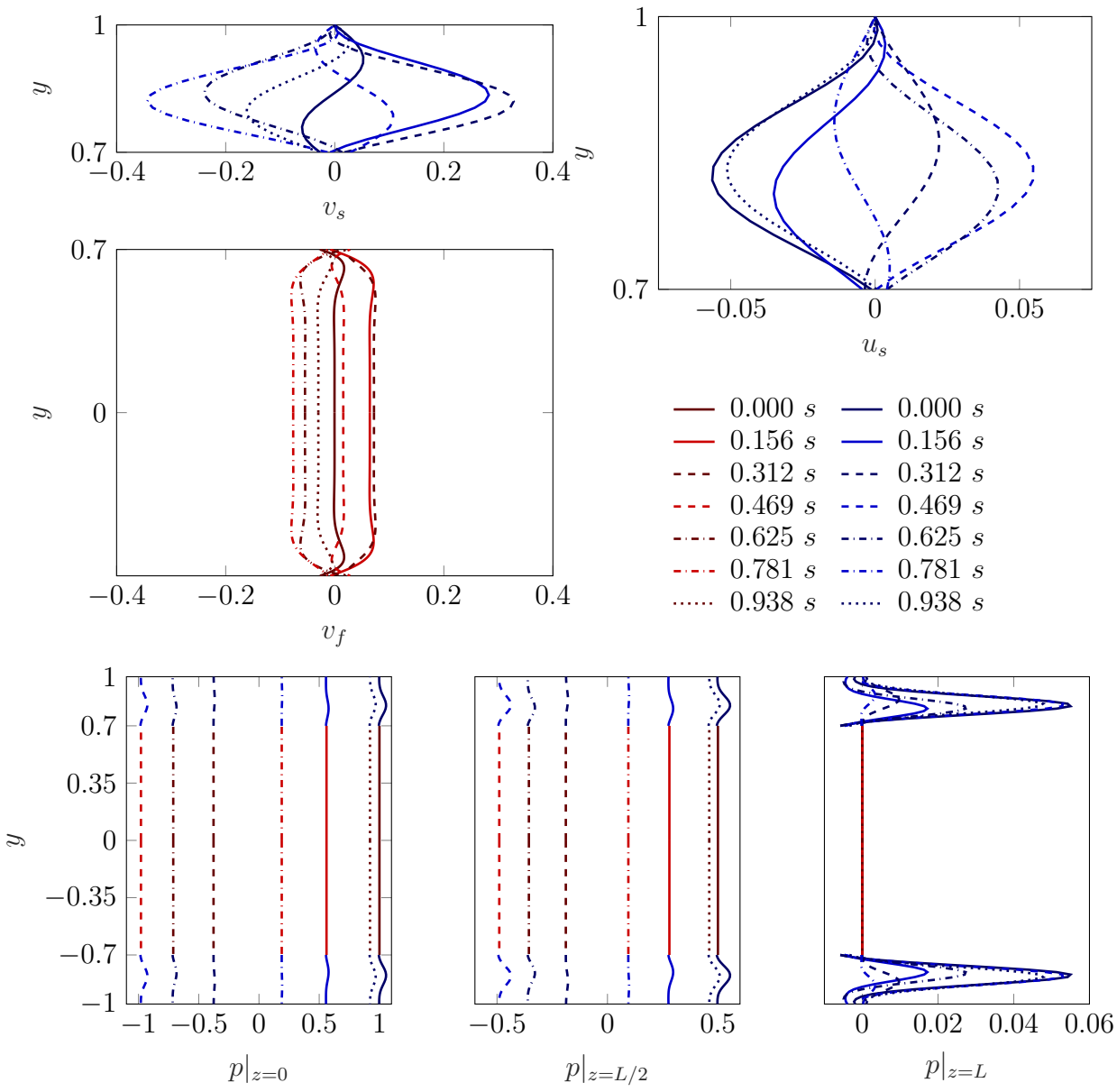


Figure 2.5: *The analytic solution for the transient nonlinear FSI case in three dimensions with fluid and solid density $\rho_f = 2.1$ and $\rho_s = 1$, fluid viscosity $\mu_f = 0.03$, solid stiffness $\mu_s = 0.1$ and cycle length $T = 1.024$: Fluid and solid velocity, v_f (left) and v_s (top left), and solid displacement, u_s (top right), along the y-axis. Further, fluid and solid pressure, p_f and p_s (bottom) over time t at three positions along the z-axis.*

Figure 2.5 illustrates the temporal variation of the velocity, displacement and pressure profiles along the y-axis. Compared to the transient linear FSI case in the previous section, the peak solid deformation rate is of similar magnitude, however, the peak fluid velocity is smaller. Furthermore, the peak values are now present in the interior of the fluid and solid domains. While the fluid velocity gradient near the wall is smaller compared to the linear FSI case, the solid deformation rate and displacement is varying along the y-axis in a similar fashion.

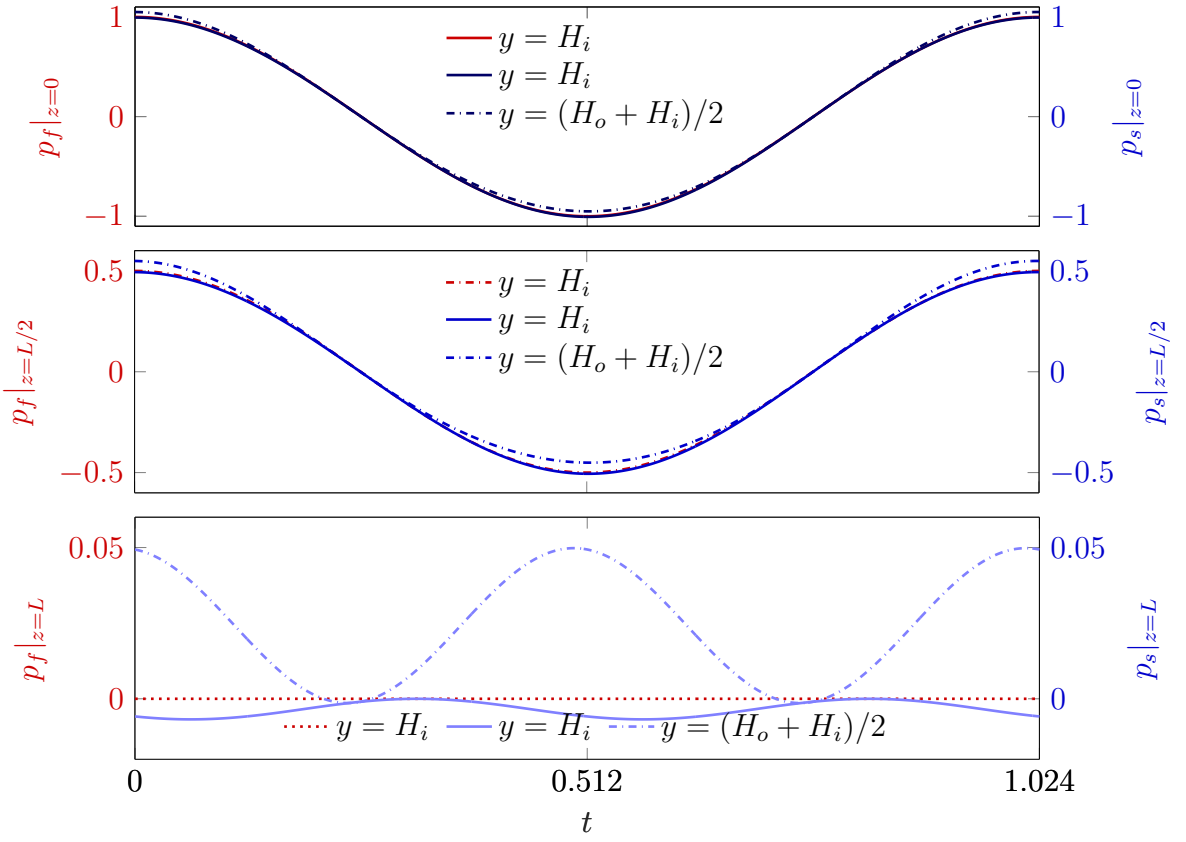


Figure 2.6: *The analytic solution for the transient nonlinear FSI case in three dimensions with fluid and solid density $\rho_f = 2.1$ and $\rho_s = 1$, fluid viscosity $\mu_f = 0.03$, solid stiffness $\mu_s = 0.1$ and cycle length $T = 1.024$: The fluid and solid pressure, p_f and p_s, over time t at three positions along the z-axis: At the inlet for $z = 0$, the midway point at $z = L/2$, and at the outlet $z = L$ (top to bottom).*

Figure 2.6 further highlights the temporal variation of the fluid and solid pressure solutions at the inlet ($x = 0$), the midway point in the x-direction ($x = L/2$) and the outlet ($x = L$). The fluid and solid pressure solutions are no longer continuous at the interface, which becomes more clear towards the outlet, see Figure 2.6 (bottom) and Figure 2.5 (bottom-right). The fact that the solid pressure solution is now a function of $[\partial u_s/\partial r]^2$ (see Equation (2.68)) results in a higher temporal frequency toward the outlet at $z = L$ as compared to the inlet at $z = 0$, and in large pressure gradients in the radial direction, see Figure 2.6.

Lastly, we emphasize that the response of the fluid and / or solid can be changed significantly by modifying even just one parameter. As an exemplifying case, we modify

the solid density and set $\rho_s = 5$. This does not significantly affect the flow (fluid velocity and pressure), however, changes the solid response, see Figure 2.7. Similar to the mode shapes of an Euler-Bernoulli beam [7], the (transient) nonlinear solid model exhibits deformation modes corresponding to the natural frequencies of the solid material.

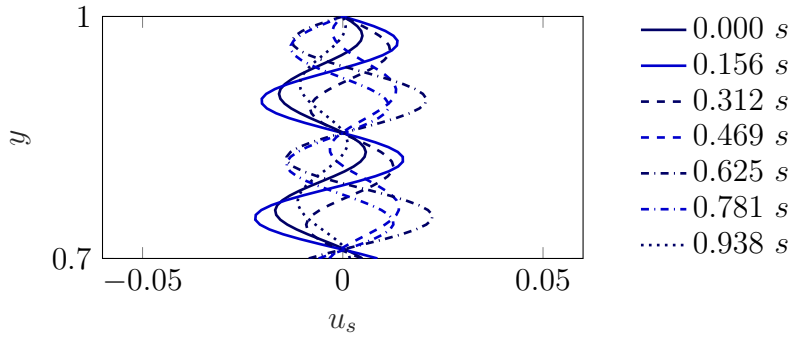


Figure 2.7: *The solid displacement for the transient nonlinear FSI case in three dimensions with fluid and solid density $\rho_f = 2.1$ and $\rho_s = 5$, fluid viscosity $\mu_f = 0.03$, solid stiffness $\mu_s = 0.1$ and cycle length $T = 1.024$: By increasing the solid density compared to Figure 2.5, the solid model exhibits higher-order deformation modes, similar to the mode shapes of an Euler-Bernoulli beam.*

2.5.3 Transient fluid and transient nonlinear solid in 3D: Physiologically relevant parameters

While the parameter choices in Section 2.5.1 and Section 2.5.2 were motivated from the viewpoint of space-time convergence analysis (see Section 2.5.4), one may select a more physiologically relevant parameter set. For example, consider parameters that are similar to those found in models of the ascending aorta (parameters taken from [5, 96, 116]), with a fluid density and viscosity of $\rho_f = 1.03\ g/cm^3$ and $\mu_f = 0.03\ g/(cm \cdot s^2)$, and a solid density and stiffness of $\rho_s = 1.03\ g/cm^3$ and $\mu_s = 2{\cdot}10^5\ g/(cm{\cdot}s^2)$. The pressure amplitude is selected as $P = 583$ to yield a maximum inflow velocity of $V_f \approx 100\ cm/s$. We further modify the dimensions of the fluid and solid domains: the fluid domain radius is $H_i = 0.7\ cm$, the solid wall thickness is $0.223\ cm$ and the fluid domain length is $5.53\ cm$. Here, the Womersley number is typical for aortic blood flow, $W \approx 10.16$, resulting in the typical Womersley flow profile (see Figure 2.8), while the Reynolds number is $Re \approx 4813.40$.

Due to the significantly increased stiffness, the deformation of the solid is small (see Figure 2.8) and the deformation rate is several orders of magnitudes smaller than the peak velocity of the fluid. This is in line with observations that the shear deformations in the ascending aorta are much smaller compared to the radial deformations (which is constrained here).

Although the deformation of the solid is small, it cannot be neglected because it affects the pressure solution. For example, similar to the previous nonlinear FSI case in this section, the fluid and solid pressure solution is discontinuous at the interface, see Figure 2.8 and Figure 2.9. This indicates that the presented analytic FSI solutions are suited for benchmarking methods that are able to capture sucht discontinuities as well as methods that assume continuity at the interface.

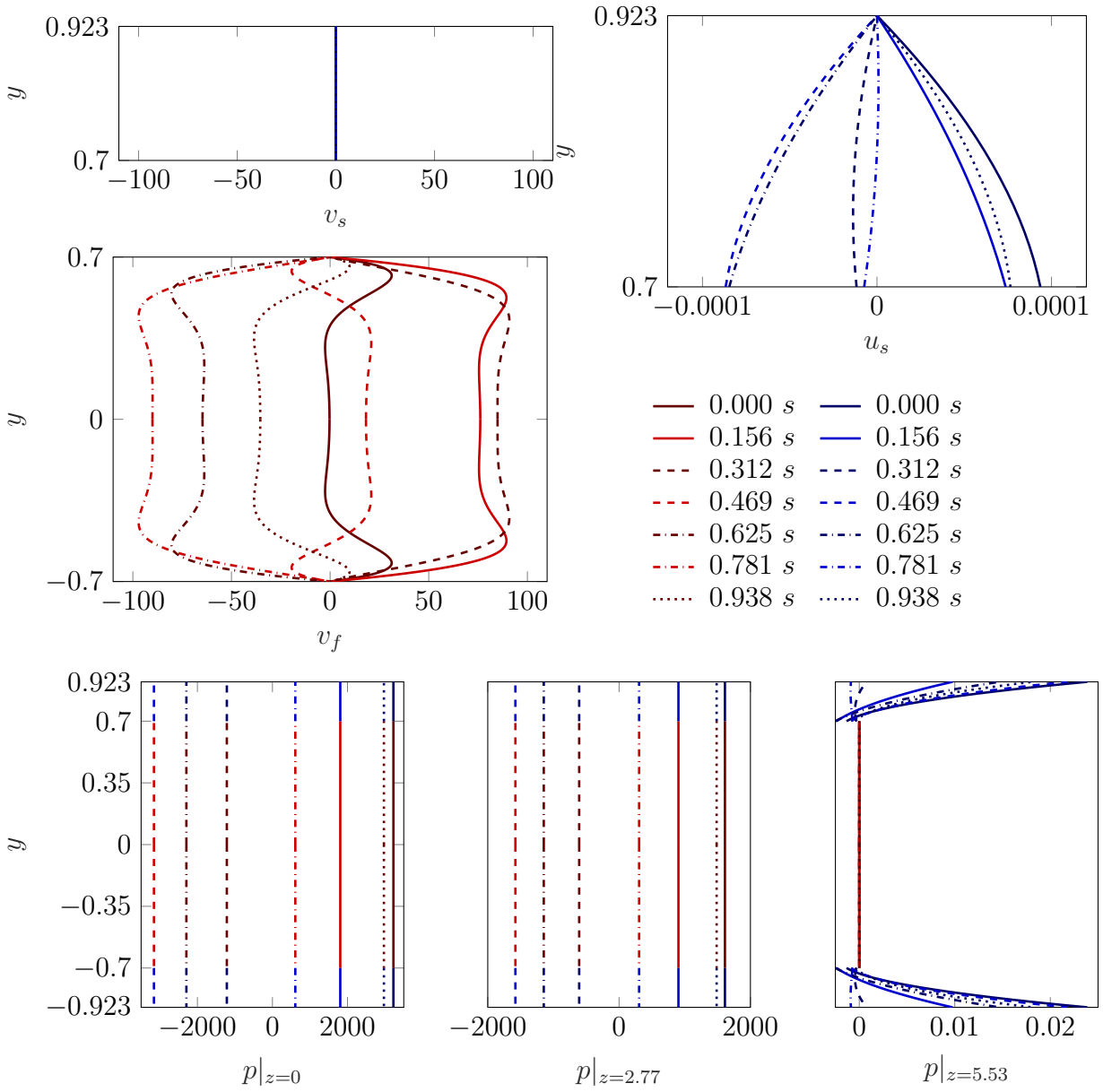


Figure 2.8: *The analytic solution for the transient nonlinear FSI case in three dimensions with fluid and solid density $\rho_f = \rho_s = 1.03\ g/cm^3$, fluid viscosity $\mu_f = 0.03\ g/(cm \cdot s^2)$, solid stiffness $\mu_s = 2 \cdot 10^5\ g/(cm \cdot s^2)$, pressure amplitude $P = 583$ and cycle length $T = 1.024\ s$: Fluid and solid velocity, v_f (left) and v_s (top left), and solid displacement, u_s (top right), along the y-axis. Further, fluid and solid pressure, p_f and p_s (bottom) over time t at three positions along the z-axis.*

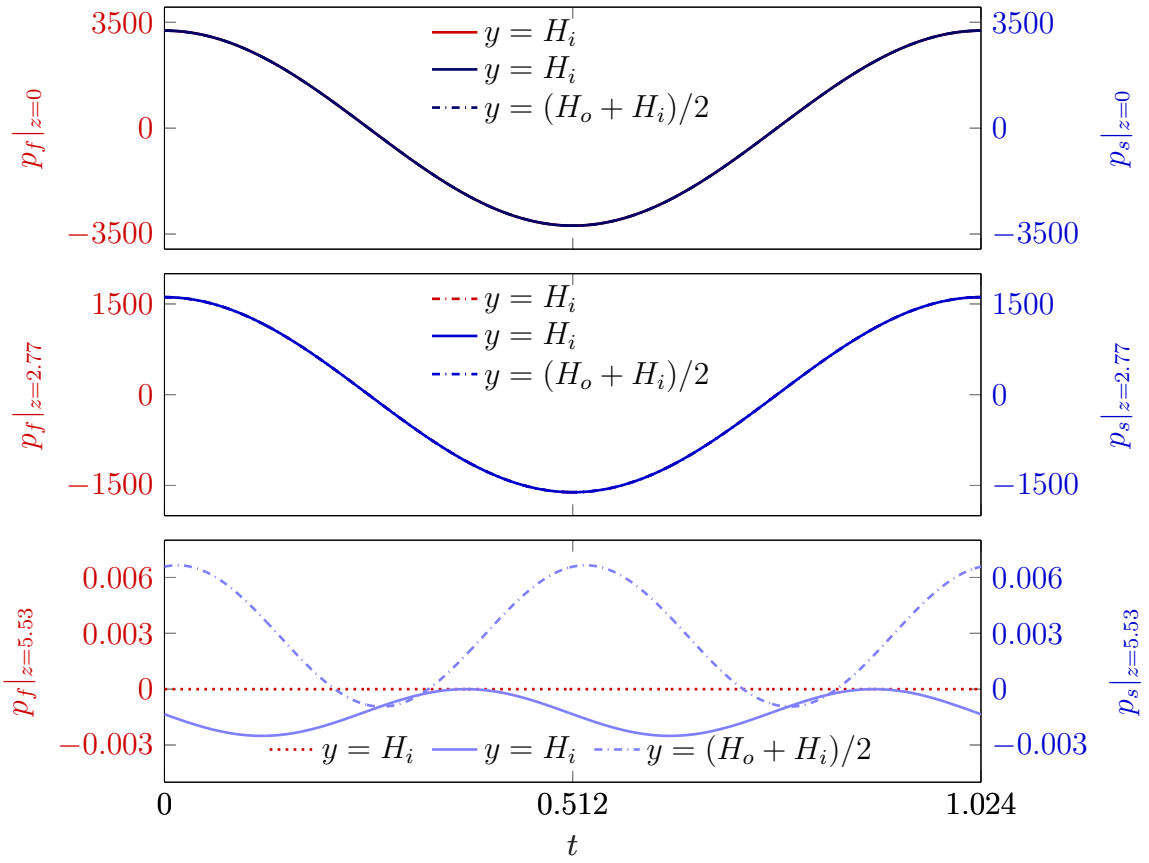


Figure 2.9: *The analytic solution for the transient nonlinear FSI case in three dimensions with fluid and solid density $\rho_f = \rho_s = 1.03\ g/cm^3$, fluid viscosity $\mu_f = 0.03\ g/(cm \cdot s^2)$, solid stiffness $\mu_s = 2 \cdot 10^5\ g/(cm \cdot s^2)$, pressure amplitude $P = 583$ and cycle length $T = 1.024\ s$: The fluid and solid pressure, p_f and p_s, over time t at three positions along the z-axis: At the inlet, at the midway point and at the outlet (top to bottom).*

2.5.4 Numerical solution and space-time convergence

In this section, results for convergence tests for the FSI method detailed in Section 2.3.1 are presented. Figure 2.10 illustrates observed temporal convergence of the algorithm applied to the linear FSI case for the fluid velocity, solid displacement and Lagrange multiplier for three different spatial resolutions. For the *coarse* refinement level, error reduction under temporal refinement is of first-order initially, then degrades as the limit of spatial discretization accuracy is approached. Such a degradation is not observed for the *medium* and *fine* spatial resolutions for all considered time step sizes. Furthermore, temporal convergence under space-time refinement (with $\delta_t/\delta_x^3 = \text{const}$) is of (optimal) rate, with $O(\delta_t)$ and $O(\delta_x^3)$ in the $L^2(\Omega)$ norm.

Space-time convergence of the FSI algorithm for the nonlinear FSI case in three dimensions is illustrated in Figure 2.11. Here, observed temporal convergence for the fluid velocity, solid displacement and solid pressure are presented for three different spatial resolutions. Similar to error reduction for the linear FSI case, temporal convergence of the non-conforming FSI algorithm is initially first-order for the *coarse* refinement level, then degrades as discretization accuracy is approached. For the *medium* refinement level, the error can be reduced further under temporal refinement. Error reduction in the fluid velocity and solid pressure solution, however, still degrades, whereas the error in the solid displacement can still be decreased. For the *fine* spatial resolution, the error in the fluid velocity, solid displacement and solid pressure variable can be reduced with optimal first-order rate for all considered time step sizes.[2] Similar to the case of two spatial dimensions, temporal convergence under space-time refinement (with $\delta_t/\delta_x^3 = \text{const}$) is of (optimal) rate, with $O(\delta_t)$ and $O(\delta_x^3)$ in the $L^2(\Omega)$ norm.

We have further performed space-time convergence studies for all other cases (2D nonlinear, 3D linear) and permutations of transient / quasi-static behavior. Similar to the cases presented here, optimal convergence rates were obtained under space-time refinement (with constant δ_t/δ_x^d in d dimensions). Deteriorating convergence under temporal refinement was only observed when discretization accuracy was reached.

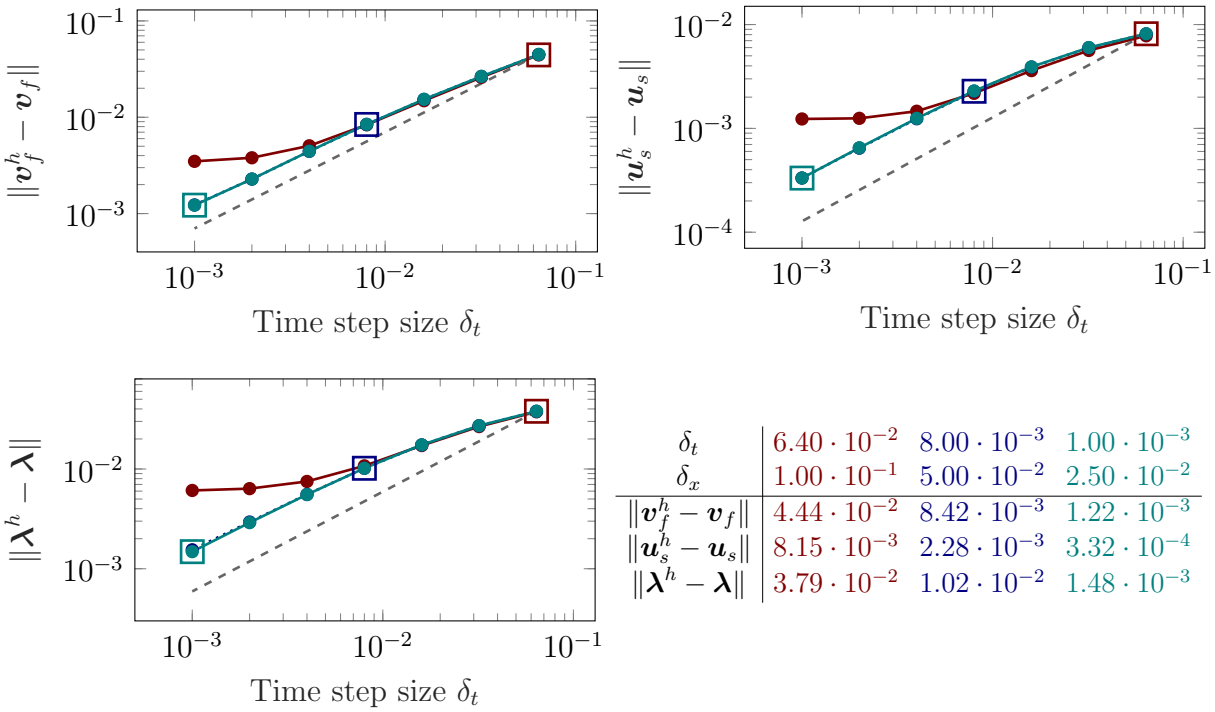
2.6 Discussion

2.6.1 Class of analytic solutions for FSI

The class of analytic FSI solutions presented in Section 2.2 provides a rich and comprehensive test bed for the validation of FSI implementations. It is comprised of permutations of linear or nonlinear, and quasi-static or transient behavior in two and three dimensions. Depending on the considered case, the velocity / displacement (pressure) solutions may exhibit simplistic quadratic (linear) dependence in one spatial dimension or more complex dependence on Bessel functions, with the pressure solution varying in multiple spatial dimensions.

The analytic solutions are functions of the material and geometric parameters and as such, the behavior and properties of the solutions can vary widely (even for the same

[2] The number of Newton iterations taken in the nonlinear solve of the combination largest time step size / finest spatial mesh increased significantly. To avoid excessive runtimes, this case was omitted in the convergence study.

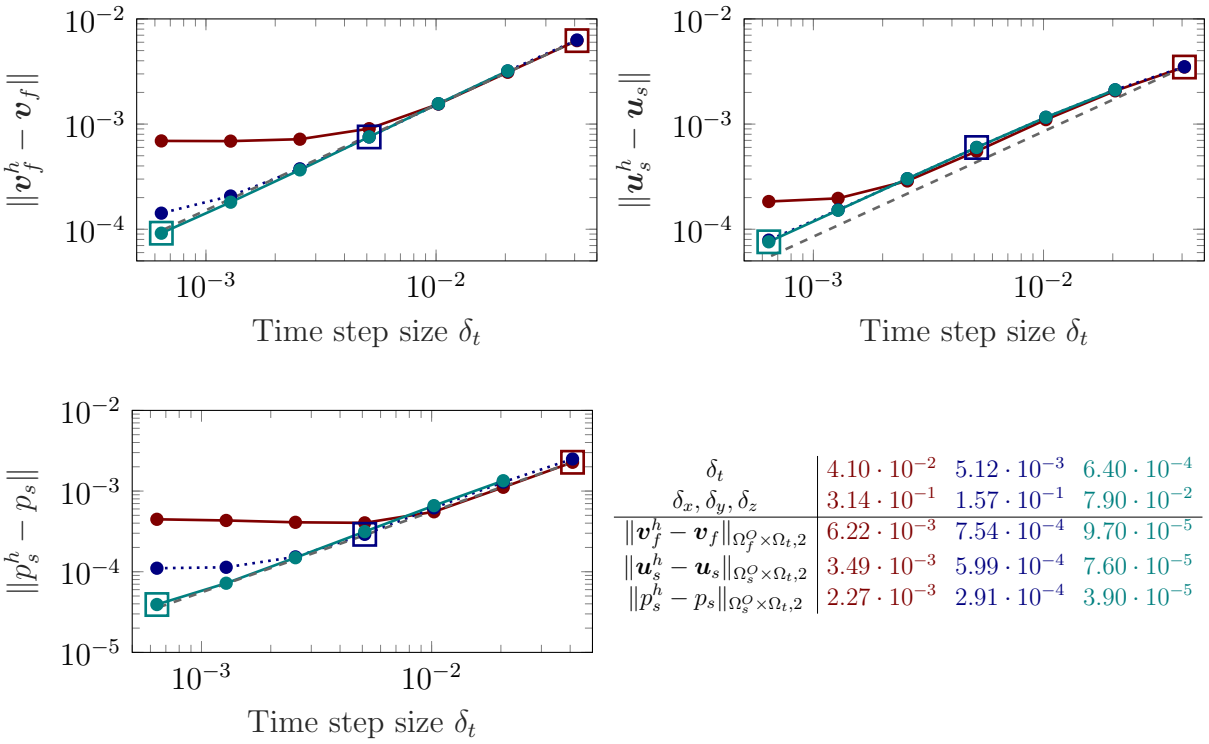


δ_t	$6.40 \cdot 10^{-2}$	$8.00 \cdot 10^{-3}$	$1.00 \cdot 10^{-3}$
δ_x	$1.00 \cdot 10^{-1}$	$5.00 \cdot 10^{-2}$	$2.50 \cdot 10^{-2}$
$\|\boldsymbol{v}_f^h - \boldsymbol{v}_f\|$	$4.44 \cdot 10^{-2}$	$8.42 \cdot 10^{-3}$	$1.22 \cdot 10^{-3}$
$\|\boldsymbol{u}_s^h - \boldsymbol{u}_s\|$	$8.15 \cdot 10^{-3}$	$2.28 \cdot 10^{-3}$	$3.32 \cdot 10^{-4}$
$\|\boldsymbol{\lambda}^h - \boldsymbol{\lambda}\|$	$3.79 \cdot 10^{-2}$	$1.02 \cdot 10^{-2}$	$1.48 \cdot 10^{-3}$

Figure 2.10: *Spatiotemporal convergence for the transient linear FSI case in two dimensions: Temporal convergence of error for cycle* 10 *(i.e.,* $\Omega_t = [9T, 10T]$*) for fluid velocity* $\boldsymbol{v}_f$*, solid displacement* $\boldsymbol{u}_s$ *and Lagrange multiplier* $\boldsymbol{\lambda}$ *(and corrsponding numerical approximations* $\boldsymbol{v}_f^h$*,* $\boldsymbol{u}_s^h$ *and* $\boldsymbol{\lambda}^h$*) for* coarse *(red),* medium *(blue) and* fine *(green) spatial resolutions. Square markers highlight optimal convergence of error for* $\delta_t/\delta_x^3 = const$ *(see Table). Dashed line illustrates optimal rate.*

FSI solution; see Section 2.5.2 and 2.5.3). For example, the flow profile can be close to a parabolic profile or resemble a typical Womersley profile; similarly, the Reynolds number can be small or (by changing, e.g., material parameters) high. On the other hand, the solid can exhibit small (e.g., as demonstrated in Section 2.5.3) or large nonlinear deformations. It is even possible to select parameters to observe higher-order deformation modes (similar to the Euler-Bernoulli mode shapes; see Figure 2.7) or to construct cases without unique solution (resonance-frequency-type solutions; e.g., see Equation (2.36)). This property of the analytic solutions accentuates one of the avenues to make the numerical solution process more challenging.

A commonly used benchmark problem is that of flow in an elastic tube (e.g., [156]); however, it is restricted to validating pulse wave propagation but not spatiotemporal behavior of the solutions in the entire domain, which is possible with the class of analytic FSI solutions. While the flow in the nonlinear FSI case is still Stokes-like (the advective term in the Navier-Stokes equations drops out because of the assumption of no radial motion), the stress response of the neo-Hooke solid material is nonlinear and does not simplify. In fact, to my best knowledge, the possibility to validate a nonlinear solid model in a fluid-structure interaction framework using an analytic solution is the first of its kind and an important contribution. Furthermore, the pressure solution varies along the axial direction and, in the nonlinear case, also in the radial direction. Thus, the analytic FSI solution cannot be reduced to (or solved in) a lower-dimensional manifold in space.



δ_t	$4.10 \cdot 10^{-2}$	$5.12 \cdot 10^{-3}$	$6.40 \cdot 10^{-4}$
$\delta_x, \delta_y, \delta_z$	$3.14 \cdot 10^{-1}$	$1.57 \cdot 10^{-1}$	$7.90 \cdot 10^{-2}$
$\|\boldsymbol{v}_f^h - \boldsymbol{v}_f\|_{\Omega_f^Q \times \Omega_t, 2}$	$6.22 \cdot 10^{-3}$	$7.54 \cdot 10^{-4}$	$9.70 \cdot 10^{-5}$
$\|\boldsymbol{u}_s^h - \boldsymbol{u}_s\|_{\Omega_s^Q \times \Omega_t, 2}$	$3.49 \cdot 10^{-3}$	$5.99 \cdot 10^{-4}$	$7.60 \cdot 10^{-5}$
$\|p_s^h - p_s\|_{\Omega_s^Q \times \Omega_t, 2}$	$2.27 \cdot 10^{-3}$	$2.91 \cdot 10^{-4}$	$3.90 \cdot 10^{-5}$

Figure 2.11: *Spatiotemporal convergence for the transient nonlinear FSI case in three dimensions: Temporal convergence of error for cycle* 7 *(i.e.,* $\Omega_t = [6T, 7T]$*) for fluid velocity* $\boldsymbol{v}_f$*, solid displacement* $\boldsymbol{u}_s$ *and solid pressure* p_s *(and corrsponding numerical approximations* $\boldsymbol{v}_f^h$*,* $\boldsymbol{u}_s^h$*, and* p_s^h*) for* coarse *(red),* medium *(blue) and* fine *(green) spatial resolutions. Square markers highlight optimal convergence of error for* $\delta_t/\delta_x^3 = const$ *(see Table). Dashed line illustrates optimal rate.*

2.6.2 Analytic solutions utility for verification of FSI algorithms

The class of analytic FSI solutions presented here enable the testing of FSI implementations at various stages: Starting from the quasi-static linear case in two dimensions, complexity can be added and the respective implementation validated step-by-step. For example, transient behavior can be added in the fluid and / or solid models, respectively, or the FSI domain changed from two dimensions to three dimensions. While the FSI geometry is relatively simple compared to other benchmarks (e.g., [64, 147]), it does not pose any challenges in setting up the computational model. In particular, the analytic solutions are smooth and discretization errors stemming from discretizing the spatial domains can be avoided (e.g., by selecting appropriate shape functions in finite element discretizations).

Another key aspect is the possibility to change Neumann-type to Dirichlet-type boundary conditions or vice-versa, complementing the validation of the implementation at the equation level to a more infrastractural aspect. The validation of the coupling constraints can further be isolated by first validating the fluid- and solid-only cases (e.g., by setting Dirichlet boundary conditions at the coupling boundary) and then adding the coupling constraint. Coupling approaches can vary widely [80, 91, 110] and affect numerical perfor-

mance of an FSI implementation. Furthermore, some FSI methods assume continuity of the pressure across the fluid / solid interface (e.g., the unified continuum method [72, 80]), whereas other methods do not make such an assumption (e.g., the non-conforming monolithic FSI method assessed in Section 2.5.4). It is an open research question how such assumptions influence the numerical solution.

A further (maybe not immediately obvious) benefit of analytic solutions is their usefulness for assessing numerical tricks such as implicit-explicit splits [91] and novel methods or time integration algorithms, such as multigrid-reduction-in-time [30, 35] (MGRIT). For example, combined with convergence theory [25, 68, 133], the class of analytic FSI solutions can be an invaluable tool for understanding the performance of MGRIT for FSI problems (see Section 5.5 and Section 5.6) and help to identify reasons of potentially slow(er) convergence, as well as support the development of remedies thereof.

2.6.3 Analytic solutions utility for spatiotemporal convergence analysis

The availability of analytic solutions further simplifies assessing the accuracy and convergence of FSI methods and implementations. Even if neglecting the complex and code-specific task of implementing forcing terms for the validation through manufactured FSI solutions, spatiotemporal errors of such forcing terms (due to their numerical inclusion) may affect the validation process. Analytic solutions make it easier to isolate potential implementation errors (as described in Section 2.6.2), conduct stability analyses under refinement (e.g., stability of an explicit method and identifying time step size limits) and perform space-time convergence studies (which is difficult with numerical benchmarks, such as [147]) as demonstrated in Section 2.5.4. For example, by conducting a thorough convergence study, it can be shown that a given FSI method was not only implemented correctly but that the theoretical best-case convergence rate can be achieved; for example, the backward Euler time-discretization with quadratic / linear finite elements in space (see Section 2.3.1), was shown to converge at the optimal first-order rate in time under space-time refinement with $\delta_t/\delta_x^3 = \text{const}$.

2.6.4 Study limitations

Although the derivation of the nonlinear FSI solutions was based on the incompressible Navier-Stokes equations as a fluid model, the advective (nonlinear) term vanishes. Thus, any potential nonlinearities in the behavior of the fluid in numerical experiments stem from sources like approximating the spatial domains, entirely. In the linear case, the velocity and displacement solutions are further constant in the axial component. While more complicated and radially varying analytic solutions would be desirable, the assumption of no radial motion (and thus, the restriction to shear deformations) was an essential ingredient to enable the derivation of the class of analytic solutions presented in this work.

2.6.5 Open-source implementation

All analytic solutions were implemented[3] in the Matlab R2018a scripting language, to avoid the need for implementing complex functions in existing simulation frameworks and to assist with the validation process. For example, the code can be used to evaluate and export the analytic solutions for a given FSI case to set appropriate initial and boundary conditions in a computational model, as well as for the comparison with the numerical solution. The material, temporal and geometric parameters can be user-defined and a user-mesh can be imported to evaluate the analytic solutions at relevant coordinates. We have further implemented functionality to visualize the analytic solutions similar to Figure 2.5 and included movies of the animated solutions for all linear / nonlinear, 2D / 3D and quasi-static / transient cases in the *Supplementary Materials* of [59] (for parameters in Section 2.5.1 and Section 2.5.2).

2.7 Conclusion

In this chapter, a class of analytic FSI solutions was presented that provides a rich and comprehensive test bed for the validation and convergence testing of linear and nonlinear FSI implementations. It enables a step-by-step validation process (two to three dimensions, linear to nonlinear, etc.) and further allows to isolate the testing of certain parts of an FSI implementation, such as the coupling conditions. It is the first analytic FSI solution that includes a nonlinear solid model. Along with a description and derivation of the class of analytic FSI solutions, their usefulness for validating numerical FSI methods was demonstrated and their value in spatiotemporal convergence testing was highlighted. A subset of the analytic FSI solutions is further used in Section 5.5 and Section 5.6 to measure error reduction of the multigrid-reduction-in-time algorithm.

[3] The code is available in the *Supplementary Materials* of [59] and as an open-source implementation. Bitbucket repository: `https://bitbucket.org/hessenthaler/fsi-analytical-solutions-matlab`.

3 Multigrid-reduction-in-time[1]

The multigrid-reduction-in-time (MGRIT) algorithm was first introduced in the works of Friedhoff et al. and Falgout et al. [30, 35] and can be interpreted as an extension or generalization of the two-grid Parareal algorithm [97].

It has been explored for various application areas, such as the numerical solution of parabolic and hyperbolic partial differential equations (PDEs) [32, 35, 66, 76], investigations of power systems [93, 127], solving adjoint and optimization problems [51, 52], and neural network training [125].

3.1 Sequential time-stepping

First, consider the general form of a time-dependent PDE in one spatial dimension:

$$f(x,t;u,\partial u/\partial x,\partial u/\partial t,\partial^2 u/\partial x^2,\ldots)=0 \qquad \text{for } (x,t)\in\Omega\times[0,T]. \tag{3.1}$$

Discretizing Equation (3.1) in space and time, sequential time-stepping can be written as,

$$\boldsymbol{u}_n=\boldsymbol{\Phi}\left(\boldsymbol{u}_n,\boldsymbol{u}_{n-1}\right)+\boldsymbol{g}_n \qquad \text{for } n=1,\ldots,N_0-1, \tag{3.2}$$

with solution $\boldsymbol{u}_n=\boldsymbol{u}(t_n)$ and forcing function $\boldsymbol{g}_n$ for N_0 time points $0=t_0<t_1<\ldots<t_{N_0-1}=T$. Here, $\boldsymbol{u}_n,\boldsymbol{g}_n\in\mathbb{R}^{N_x}$ with N_x spatial degrees-of-freedom. Further, $\boldsymbol{\Phi}$ is referred to as the *time-stepping operator*.

For linear problems, Equation (3.2) can be written as,

$$\boldsymbol{u}_n=\boldsymbol{\Phi}_n\boldsymbol{u}_{n-1}+\boldsymbol{g}_n \qquad \text{for } n=1,\ldots,N_0-1. \tag{3.3}$$

In this chapter, we assume a constant spatial discretization (e.g., no remeshing) and equidistant time points $t_n=n\cdot\delta_0$ for $n=0,\ldots,N_0-1$ with time step size $\delta_0=T/(N_0-1)$, such that $\boldsymbol{\Phi}_n=\boldsymbol{\Phi}_0$. Then, Equation (3.3) can be written in matrix form,

$$A_0\boldsymbol{u}=\begin{bmatrix} I & & & & \\ -\boldsymbol{\Phi}_0 & I & & & \\ & -\boldsymbol{\Phi}_0 & I & & \\ & & \ddots & \ddots & \\ & & & -\boldsymbol{\Phi}_0 & I \end{bmatrix}\begin{bmatrix} \boldsymbol{u}_0 \\ \boldsymbol{u}_1 \\ \boldsymbol{u}_2 \\ \vdots \\ \boldsymbol{u}_{N_0-1} \end{bmatrix}=\boldsymbol{g}, \tag{3.4}$$

with space-time matrix $A_0\in\mathbb{R}^{N_xN_0\times N_xN_0}$ and space-time solution vector $\boldsymbol{u}\in\mathbb{R}^{N_xN_0}$. Sequential time-stepping can be identified as a block-forward solve of (3.4).

[1] Parts of this chapter were submitted as [68].

3.2 Time grid hierarchy and F/C-splitting

MGRIT solves (3.4) iteratively and, similar to sequential time-stepping, is an $O(N_0)$ method; that is, the total number of (block) operations to solution is linear or near-linear with the number of time steps (assuming MGRIT is applicable / convergent).

MGRIT introduces a multilevel hierarchy of n_ℓ time grids to achieve parallelism in the temporal domain, employing a coarse-grid correction scheme based on multigrid reduction. The fine grid (referred to as level $\ell = 0$) is composed of all time points t_n (for $n = 0, \ldots, N_0 - 1$) and the coarser grids (referred to as levels $\ell = 1, \ldots, n_\ell - 1$) are derived from a uniform coarsening of the fine grid, see Figure 3.1. The temporal coarsening factors are denoted as $m_\ell \in \mathbb{N}$ (for $\ell = 0, \ldots, n_\ell - 2$),[2] such that the number of time points on each grid level is given by,

$$N_\ell = \frac{N_{\ell-1} - 1}{m_{\ell-1}} + 1, \qquad \text{for } \ell = 1, \ldots, n_\ell - 1, \tag{3.5}$$

with corresponding time step size δ_ℓ. On each grid level ℓ, time points are partitioned into F-points and C-points, and the C-points on level ℓ compose all points on the next coarser grid level $\ell + 1$. Further, let Φ_ℓ denote the time-stepping operator on level ℓ.

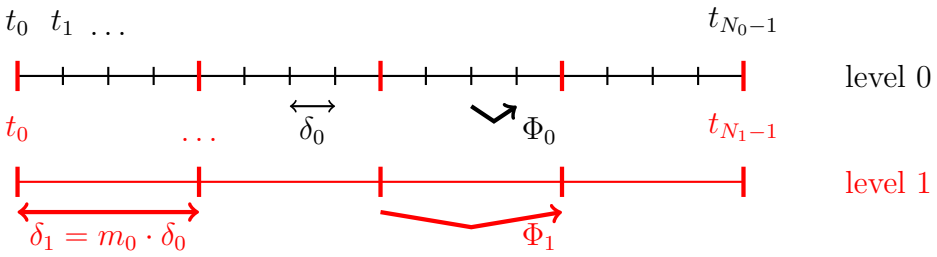


Figure 3.1: *Two-grid hierarchy: time points t_n, fine- / coarse-grid step sizes δ_0 and δ_1, temporal coarsening factor $m_0 = 4$. On level 0, F-points are denoted in black and C-points are denoted in red.*

3.3 MGRIT operators

MGRIT approximates the exact coarse-grid time-stepping operator[3] on level ℓ by introducing,

$$\Phi_\ell \approx \Phi_{\ell-1}^{m_{\ell-1}}, \qquad \text{for } \ell = 1, \ldots, n_\ell - 1, \tag{3.6}$$

and we write the space-time matrix on level ℓ,

$$A_\ell = \begin{bmatrix} I & & & \\ -\Phi_\ell & I & & \\ & -\Phi_\ell & I & \\ & & \ddots & \ddots \end{bmatrix} \in \mathbb{R}^{N_x N_\ell \times N_x N_\ell}, \qquad \text{for } \ell = 1, \ldots, n_\ell - 1. \tag{3.7}$$

[2] Note that $m_\ell = 1$ for some or all ℓ is a valid choice, e.g., for a p-multigrid-like approach.

[3] Time-stepping on the coarse-grid is referred to as exact, if it yields the same solution as sequential time-stepping on the fine-grid.

MGRIT constructs coarse-grids from a Schur complement decomposition of (3.7), relative to the F/C-splitting from Figure 3.1 [25]. The Schur complement arises from certain so-called *ideal* multigrid restriction and interpolation operators. Define ideal restriction from level ℓ to $\ell+1$ (for $\ell = 0, \ldots, n_\ell - 2$) as,

$$R_\ell = \begin{bmatrix} I & & & & & & & & & \\ & \Phi_\ell^{m_\ell-1} & \Phi_\ell^{m_\ell-2} & \cdots & \Phi_\ell & I & & & & \\ & & & & & & \ddots & & & \\ & & & & & & & \Phi_\ell^{m_\ell-1} & \Phi_\ell^{m_\ell-2} & \cdots & \Phi_\ell & I \end{bmatrix}, \tag{3.8}$$

with $R_\ell \in \mathbb{R}^{N_x N_{\ell+1} \times N_x N_\ell}$. Define ideal interpolation from level $\ell+1$ to ℓ (for $\ell = 0, \ldots, n_\ell - 2$) as,

$$P_\ell = \begin{bmatrix} I & & & & & \\ \Phi_\ell & & & & & \\ \vdots & & & & & \\ \Phi_\ell^{m_\ell-1} & & & & & \\ & I & & & & \\ & \Phi_\ell & & & & \\ & \vdots & & & & \\ & \Phi_\ell^{m_\ell-1} & & & & \\ & & & \ddots & & \\ & & & & & I \end{bmatrix}, \tag{3.9}$$

with $P_\ell \in \mathbb{R}^{N_x N_\ell \times N_x N_{\ell+1}}$. Note that the interpolation operator is not defined as the transpose of the restriction operator, since in general $\Phi_\ell \neq \Phi_\ell^T$ (for example, see [66, Equation (28)]), and thus, $P_\ell \neq R_\ell^T$.

Further, define an auxillary operator,

$$S_\ell = \begin{bmatrix} 0 & & & & & & \\ I & & & & & & \\ & I & & & & & \\ & & \ddots & & & & \\ & & & I & & & \\ & & & 0 & & & \\ & & & 0 & I & & \\ & & & & & \ddots & \\ & & & & & & I \\ & & & & & & 0 \end{bmatrix}, \tag{3.10}$$

with $S_\ell \in \mathbb{R}^{N_x N_\ell \times N_x (N_\ell - N_{\ell+1})}$ for $\ell = 1, \ldots, n_\ell - 2$. The number of block columns in S_ℓ corresponds to the total number of F-points on level ℓ. If the operator S_ℓ is applied to the right of A_ℓ, the result $A_\ell S_\ell$ is composed of all block rows in A_ℓ, and all block columns in A_ℓ that correspond to F-points on level ℓ (zeroing out the respective C-point block columns). Thus, $S_\ell^T A_\ell S_\ell$ is composed of all block rows and block columns in A_ℓ that correspond to F-points.

Using the above definitions, it is straightforward to work out that the multigrid coarse-grid operator, $R_\ell A_\ell P_\ell$, is then given by the Schur complement [25] of A_ℓ in Equation (3.7),

$$R_\ell A_\ell P_\ell = R_{I_\ell} A_\ell P_\ell = \begin{bmatrix} I & & & & \\ -\Phi_\ell^{m_\ell} & I & & & \\ & -\Phi_\ell^{m_\ell} & I & & \\ & & \ddots & \ddots & \\ & & & -\Phi_\ell^{m_\ell} & I \end{bmatrix}, \tag{3.11}$$

where restriction by injection is given by the operator,

$$R_{I_\ell} = \begin{bmatrix} I & & & & & & & & \\ & 0 & 0 & \cdots & I & & & & \\ & & & & & \ddots & & & \\ & & & & & & 0 & 0 & \cdots & 0 & I \end{bmatrix}, \tag{3.12}$$

for $\ell = 1, \ldots, n_\ell - 2$. Here, $R_{I_\ell} \in \mathbb{R}^{N_x N_{\ell+1} \times N_x N_\ell}$ has a similar block structure as R_ℓ, but with all blocks Φ_ℓ^m (for $m = 1, \ldots, m_\ell - 1$) set to zero. Thus, R_{I_ℓ} restricts the C-points from level ℓ to level $\ell + 1$, omitting the respective F-points. The number of block rows corresponds to the total number of C-points on level ℓ, i.e. $N_{\ell+1}$. Also note that the inverse of A_ℓ is given analytically [25] by,

$$A_\ell^{-1} = \begin{bmatrix} I & & & & \\ \Phi_\ell & I & & & \\ \Phi_\ell^2 & \Phi_\ell & \ddots & & \\ \vdots & \vdots & & I & \\ \Phi_\ell^{N_\ell - 1} & \Phi_\ell^{N_\ell - 2} & & \Phi_\ell & I \end{bmatrix}, \tag{3.13}$$

with $A_\ell^{-1} \in \mathbb{R}^{N_x N_\ell \times N_x N_\ell}$ for $\ell = 1, \ldots, n_\ell - 1$.

In a typical multigrid fashion, MGRIT uses a complementary relaxation process to reduce error that is not adequately reduced on coarser grids. Because MGRIT is a reduction-based method, coarse-grid correction (should) eliminate error effectively at C-points, so this is coupled with an F-relaxation scheme to eliminate error at F-points. F-relaxation can be seen as a block Jacobi-like method, where in this case each block consists of a set of contiguous F-points in the time domain (that is, F-relaxation updates all F-points based on sequential time integration from the closest (previous) C-point). Algebraically, this is equivalent to an application of the idempotent operator,

$$F_\ell = P_\ell R_{I_\ell} = I - S_\ell (S_\ell^T A_\ell S_\ell)^{-1} S_\ell^T. \tag{3.14}$$

When F-relaxation alone is insufficient, a stronger relaxation scheme can be used. Further define,

$$T_\ell = R_{I_\ell}^T. \tag{3.15}$$

Then, C-relaxation updates a C-point based on taking one time step from the previous F-point (equivalent to block Jacobi applied to the C-point block rows of A_ℓ). Algebraically,

this corresponds to an application of

$$C_\ell = \begin{bmatrix} I & & & & & \\ & \ddots & & & & \\ & & I & & & \\ & & \Phi_\ell & 0 & & \\ & & & 0 & I & \\ & & & & & \ddots \end{bmatrix} = I - T_\ell (T_\ell^T A_\ell T_\ell)^{-1} T_\ell^T A_\ell, \tag{3.16}$$

where the zero columns correspond to C-points. Using Equation (3.14) and Equation 3.16, FCF-relaxation (subsequent F-, C- and F-relaxation steps) can be written as,

$$F_\ell C_\ell F_\ell = P_\ell (I - R_\ell A_\ell P_\ell) R_{I_\ell}. \tag{3.17}$$

We further use the more general notation of rFCF-relaxation, which corresponds to F-relaxation, followed by r CF-relaxation steps.

3.4 Algorithm

Now that we have the basic ingredients for MGRIT, we can describe a standard V-cycle in the following pseudocode:

Algorithm 2 Pseudocode for MGRIT algorithm with V-cycles, adapted from [25].

1: **repeat**
2: Relax on $A_\ell \boldsymbol{u} = \boldsymbol{g}$ using rFCF-relaxation with $\boldsymbol{\Phi}_\ell$. ▷ In parallel
3: Compute coarse-grid residual $\boldsymbol{r}^\Delta = R_\ell(\boldsymbol{g} - A_\ell \boldsymbol{u})$.
4: Solve coarse grid correction problem $A_{\ell+1}\boldsymbol{e}^\Delta = \boldsymbol{r}^\Delta$ using $\boldsymbol{\Phi}_{\ell+1}$. ▷ Apply recursively
5: Correct solution at fine-grid C-points with $\boldsymbol{e}^\Delta$. ▷ In parallel
6: Update the solution at the F-points with $\boldsymbol{\Phi}_\ell$. ▷ In parallel
7: **until** norm of residual is small enough

It is clear from the pseudocode in Algorithm 2, that more work needs to be done compared to sequential time-stepping; however, more work can be done in parallel, which highlights MGRIT's potential for parallel speedup.

Multilevel V-cycles are achieved by calling the algorithm recursively in Step 4. Further, F-cycles can be achieved by performing (at least) one V-cycle as postrelaxation step at each level [146].

3.5 The XBraid library

For the work presented in this Ph.D. Thesis, the open-source implementation XBraid [158] was employed[4] and extended, e.g., to implement the time-periodic MGRIT variant described in Section 3.6.

[4] XBraid Release v2.3.0

To successfully link an existing simulation package with XBraid, the user has to implement a number of wrapper routines. The wrapper steers and calls the simulation code to perform a number of basic tasks (e.g., norm computation, data I/O, etc.) and to perform time integration. Time integration routines in the simulation tool are exposed through the so-called *step function* that represents the application of the time stepping operator $\boldsymbol{\Phi}_\ell$, see Equation (3.2) and Equation (3.3).

The *step function* can be summarized as the following pseudocode:

Algorithm 3 Pseudocode for XBraid's *step function*.

1: **function** STEP_FUNCTION($\boldsymbol{u}(\cdot,t)$, t, δ_ℓ)
2: Set initial condition $\boldsymbol{u}(\cdot,t)$
3: Set initial time t
4: Set time step size δ_ℓ
5: Compute solution $\boldsymbol{u}(\cdot,t+\delta_\ell) = \boldsymbol{\Phi}_\ell\left(\boldsymbol{u}(\cdot,t+\delta_\ell), \boldsymbol{u}(\cdot,t); t, t+\delta_\ell\right) + \boldsymbol{g}(\cdot,t+\delta_\ell)$
6: **return** $\boldsymbol{u}(\cdot,t+\delta_\ell)$

As a matter of course, a realistic *step function* may have many more steps or customizations depending on the considered setting, however, Algorithm 3 highlights the basic steps.

Overall, the required implementation overhead to enable parallel-in-time integration through MGRIT is relatively low. For example, the finite element package CHeart [18, 94] has about 98500 lines of Fortran code (includes comments), whereas the wrapper routines only amount to about 2050 lines of Fortran code (includes comments) or 2%.[5]

3.6 Extension for time-periodic problems

A significant fraction of applications in biomedical engineering and other fields exhibit time-periodic behavior similar to the time-periodic analytic solutions presented in Chapter 2. A "good" initial condition at $t = 0$, however, may not be available in practice. This means that multiple cycles have to be simulated to reach a periodic steady-state solution. Figure 3.2 highlights this by comparing the space-time error (velocity) for each cycle with the associated computational cost (i.e. wall clock time) for the 2D linear FSI solution from Section 2.5.1.

This space-time periodicity is exactly what we want to exploit with MGRIT by making the time grids periodic. That is, MGRIT is employed to approximate the solution over *one* cycle only, with $t \in [0, T]$. Convergence to the periodic steady-state is achieved by updating the initial condition $\boldsymbol{v}_f(\cdot, 0)$, based on the following approach:

> **Periodic fine-grid:** The initial condition is *updated* on the *fine grid*, whenever a new value is computed / available at $t = T$ on the *fine grid*.

To describe the communication pattern, assume that p_t processors are employed in the temporal domain. Further, consider that processor 0 owns time point $t = 0$ and processor $p_t - 1$ owns time point $t = T$. Communication is implemented as follows:

[5] Note, that additional changes or additions to the CHeart source code were made, such that these figures are not entirely accurate.

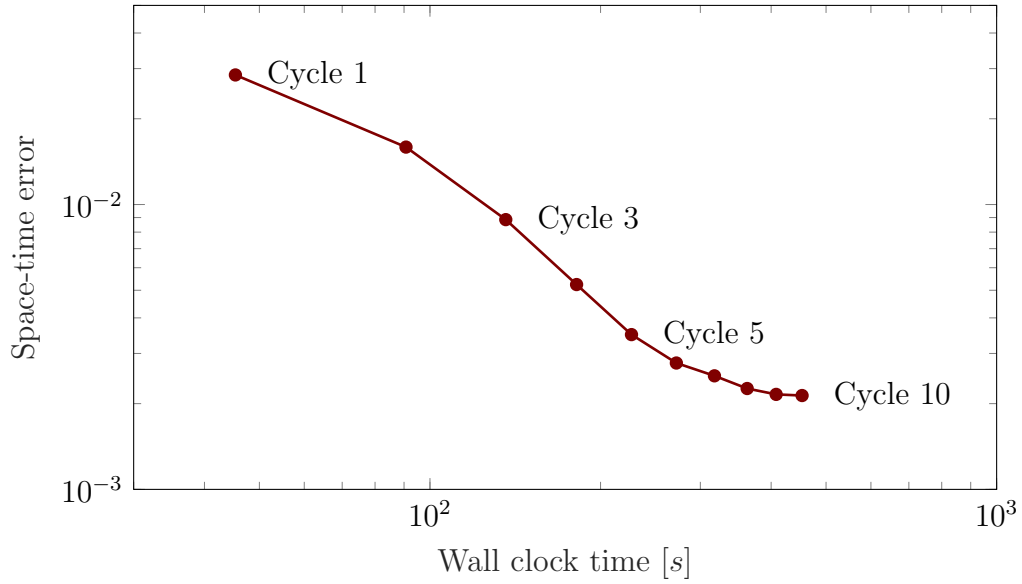


Figure 3.2: *Space-time error (velocity) for each cycle compared to wall clock time.*

- If processor $p_t - 1$ computes a new approximation at $t = T$:

  On level 0: Processor $p_t - 1$ uses MPI_iSend to send a nonblocking message to processor 0, labeled with a unique MPI tag.

  On levels 1 to $\mathbf{n_\ell - 1}$: Do not send an update

- Iteration 1: If processor 0 computes any step **on level 0** using the initial condition at $t = 0$:

  F-relaxation: Processor 0 waits for *two* messages from processor $p_t - 1$ (labeled with the MPI tag that identifies an updated initial condition) using a blocking call to MPI_Recv.

  FCF-relaxation: Processor 0 waits for *one* message from processor $p_t - 1$ (labeled with the MPI tag that identifies an updated initial condition) using a blocking call to MPI_Recv.

- Subsequent iterations: If processor 0 computes any step **on level 0** using the initial condition at $t = 0$:

  F- or FCF-relaxation: Processor 0 waits for *one* message from processor $p_t - 1$ (labeled with the MPI tag that identifies an updated initial condition) using a blocking call to MPI_Recv, unless the initial condition is already converged.

In the case that no parallelism is employed, the initial condition is directly updated from the value computed for $t = T$. Further, instead of sending the approximation of MGRIT at $t = T$, the message contains the solution of any step (e.g., relaxation) that passes $\boldsymbol{u}(\cdot, t = T)$ back to MGRIT via the step function:

Algorithm 4 Pseudo-code for XBraid's *step function* for the time-periodic case.

1: **function** STEP_FUNCTION($\boldsymbol{u}(\cdot,t)$, t, δ_ℓ)
2: **if** time $t = 0$ and iteration $i > 0$ **then**
3: **if** initial condition at $t = 0$ not converged **then**
4: Receive update as $\boldsymbol{u}(\cdot,t)$
5: **if** iteration $i = 1$ and using F-relaxation **then**
6: Discard first update and receive second update as $\boldsymbol{u}(\cdot,t)$
7: Set initial condition $\boldsymbol{u}(\cdot,t)$
8: Set initial time t
9: Set time step size δ_ℓ
10: Compute solution $\boldsymbol{u}(\cdot,t+\delta_\ell) = \boldsymbol{\Phi}_\ell\left(\boldsymbol{u}(\cdot,t+\delta_\ell), \boldsymbol{u}(\cdot,t); t, t+\delta_\ell\right) + \boldsymbol{g}(\cdot,t+\delta_\ell)$
11: **if** time $t + \delta_\ell = T$ **then**
12: **if** initial condition at $t = 0$ not converged **then**
13: **if** $\|\boldsymbol{u}(\cdot,T) - \boldsymbol{u}(\cdot,0)\|_2 < tol$ **then**
14: Set initial condition as converged
15: Send nonblocking update of $\boldsymbol{u}(\cdot,T)$
16: **return** $\boldsymbol{u}(\cdot,t+\delta_\ell)$

Remark 1. The time-periodic MGRIT algorithm starts a forward-solve on the coarsest-grid (using XBraid's so-called *skip-first-down*[6] option) and thus, no updates are available for iteration 0 in Step 2 of Algorithm 4.

Remark 2. In iteration 1 processor 0 waits for two messages in the case of F-relaxation, see Algorithm 4, Step 5. This was a requirement to achieve a robust residual reduction with a convergence factor smaller than one.

Remark 3. In general, the initial condition for the *step function* in Algorithm 4 is $\boldsymbol{u}(\cdot,t)$ for $t \in [0,T]$, but the initial condition to the time-periodic problem is $\boldsymbol{u}(\cdot,t=0)$.

Remark 4. Step 13 of Algorithm 4 allows for an approximate convergence to the time-periodic steady-state with a given tolerance *tol*, see Section 5.4.6.1. To achieve this, a *callback* function was implemented in XBraid that let's the simulation code detect whether XBraid may terminate (initial condition converged) or not (initial condition *not* converged).

Remark 5. A similar idea to exploit time-periodicity was applied for the Parareal algorithm [40, 41]. In our approach, however, we only require an *approximate* update of the initial condition. Further, the time-periodic algorithm naturally generalizes to the true multilevel case.

[6]The *skip-first-down* means that XBraid generates an initial space-time guess using sequential time-stepping on the coarsest time grid.

4 Multilevel convergence analysis of multigrid-reduction-in-time[1,2]

In this chapter, a multilevel convergence framework for multigrid-reduction-in-time (MGRIT) is derived as a generalization of previous two-grid theory, presented in [25, 133, 159] and summarized in Section 4.1. The framework provides *a priori* upper bounds on the convergence of MGRIT V- and F-cycles with different relaxation schemes by deriving the respective residual and error propagation operators, see Section 4.4. The residual and error operators are functions of the time stepping operator, analyzed directly and bounded in norm, both numerically and analytically, see Section 4.5.

Various upper bounds of different computational cost and varying sharpness are presented. These upper bounds are complemented by proposing analytic formulae for the approximate convergence factor of V-cycle algorithms that take the number of fine grid time points, the temporal coarsening factors, and the eigenvalues of the time stepping operator as parameters. Sharpness of the bounds and the quality of the approximate convergence factors are assessed in Chapter 5.

The new multilevel convergence framework lays the groundwork for future in-depth examination and understanding of MGRIT and will help guide future development and improvement of MGRIT to explore ideas such as coarsening in integration order as opposed to step size (p-MGRIT), weighted relaxation schemes, and others, see Section 7.2. A parallel MPI/C++ implementation of all derived bounds and approximate convergence factors is provided.[3]

4.1 Theoretical convergence of two-level MGRIT

Two-level convergence theory for MGRIT was developed in [25], based on the assumptions of time integration on a uniform time grid and simultaneous diagonalization of time-stepping operators (see Section 4.3) arising from linear PDEs. The theoretical bounds were shown to be quite accurate when compared with observed convergence. Southworth [133] generalized this framework, deriving necessary and sufficient conditions and tight two-level convergence bounds for general two-level MGRIT for linear PDEs on a uniform time grid. Some extensions to the case of nonuniform time grids are also provided in [133, 159]. However, no work has been done on convergence theory for the general multilevel case.

In practice, the selection of an appropriate cycling strategy and relaxation scheme is crucial for scalable multilevel performance and, ultimately, for achieving parallel speedup.

[1]Part of this work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344, LLNL-JRNL-763460.

[2]Parts of this chapter were submitted as [68, 134].

[3]Implementation publicly available at: `github.com/XBraid/XBraid-convergence-est`

A strong framework for multilevel convergence of MGRIT can help guide these decisions in a rigorous, a priori manner.

4.1.1 Why multilevel is harder

For multigrid-type algorithms, it is generally the case that two-level convergence rates provide a lower bound on attainable multilevel convergence rates; that is, a multilevel algorithm will typically observe worse convergence than its two-level counterpart. Indeed, more expensive multilevel cycling strategies such as W-cycles or F-cycles are used specifically to solve the coarse-grid operator more accurately, better approximating a two-grid method.

The non-Galerkin coarse grid used in MGRIT (see Chapter 3) makes this relationship more complicated, and it is not clear that two-grid convergence provides a lower bound on multilevel in *all* cases. However, in practice, it is consistently the case that two-level convergence is better than multilevel. To that end, a two-level method which converges every iteration is a heuristic necessary condition for multilevel convergence.

Despite strong heuristics on multilevel convergence, the connection between two-level convergence factors and multilevel is not rigorous. This motivates the rest of this chapter, where multilevel error and residual propagation operators are analytically derived, and upper bounds on ℓ^2-convergence placed directly on these operators.

4.2 Assumptions

Similar to the two-level convergence theory by Dobrev et al. [25], the following assumptions are made:

1. Analysis for the solution of linear PDEs[4] in $\mathbf{\Phi}$-form, see Equation (3.3).
2. The time-stepping operators $\{\mathbf{\Phi}_\ell\}$ are time-independent.
3. The dimension of $\{\mathbf{\Phi}_\ell\}$ is constant, i.e. $\dim(\mathbf{\Phi}_\ell) = N_x$ for all ℓ.
4. The time-stepping operators $\{\mathbf{\Phi}_\ell\}$ are simultaneously diagonalizable, see Section 4.3.
5. The time-stepping operators $\{\mathbf{\Phi}_\ell\}$ are stable, i.e. $\|\mathbf{\Phi}_\ell\| < 1$.

Here, $\{\mathbf{\Phi}_\ell\}$ denotes the family of time-stepping operators $\mathbf{\Phi}_\ell$ for $\ell = 0, \ldots, n_{\ell-1}$.

Remark 6. Assumption 4 requires that $\{\mathbf{\Phi}_\ell\}$ are simultaneously diagonalizable from a mathematical point-of-view and, in addition, that this diagonalization is computable with reasonable computational cost.

Remark 7. Assumption 5 relates to the notion of strong stability. It is also possible to have a stable time integration scheme with $\|\mathbf{\Phi}_\ell\| > 1$, if $\|\mathbf{\Phi}_\ell^i\| < 1$ for some i (see [92] and [95, Section 9.5, Equation (9.22)]), but such schemes are not considered in this work.

[4] In Section 5.4.8, possible extensions to the nonlinear PDE case are explored.

4.3 Simultaneous diagonalization of $\{\Phi_\ell\}$

Let Φ_ℓ denote the time-stepping operator on level ℓ. Primary results in this chapter rest on the assumption that $\{\Phi_\ell\}$ are diagonalizable with the same set of eigenvectors, for all levels $\ell = 0, ..., n_\ell - 1$. This is equivalent to saying that the set $\{\Phi_\ell\}$ commutes, that is, $\Phi_i\Phi_j = \Phi_j\Phi_i$ for all i, j, and that Φ_ℓ is diagonalizable for all ℓ. The concept of simultaneous diagonalization (albeit, in the Fourier basis) was introduced in [36], and was modified and used as the basis for the improved two-grid convergence bounds developed in [25].

In terms of when such an assumption is valid, let $\mathcal{L}$ be a time-independent operator (such as a spatial discretization) that is propagated through time by operators $\{\Phi_\ell\}$. Note that all rational functions of $\mathcal{L}$ commute and that if $\mathcal{L}$ is diagonalizable, so is any rational function of $\mathcal{L}$. Indeed, nearly all standard time-integration routines, including all single-step Runge-Kutta-type methods, consist of some rational function of $\mathcal{L}$, and all such schemes are simultaneously diagonalizable with the eigenvectors of $\mathcal{L}$. To that end, let $\mathcal{L} = UDU^{-1}$, where $D_{kk} = \xi_k$ is a diagonal matrix containing the eigenvalues of $\mathcal{L}$ and columns of U are the corresponding eigenvectors. Denote the Butcher tableau of a general s-stage Runge-Kutta method as

$$\begin{array}{c|c} \mathfrak{c} & \mathfrak{A} \\ \hline & \mathfrak{b}^T \end{array}.$$

With some algebra, one can show that Φ_ℓ corresponding to a given Butcher tableau is exactly given by the Runge-Kutta stability function applied to $\mathcal{L}$ in block form,

$$\begin{aligned} \Phi_\ell &= I + \delta_\ell(\mathfrak{b}^T \otimes I)\left(I - \delta_\ell \mathfrak{A} \otimes \mathcal{L}\right)^{-1}\left(\mathbf{1}_s \otimes \mathcal{L}\right) \\ &= U\Big(I + \delta_\ell\left(\mathfrak{b}^T \otimes I\right)\left(I - \delta_\ell \mathfrak{A} \otimes D\right)^{-1}\left(\mathbf{1}_s \otimes D\right)\Big)U^{-1} \\ &= U\Lambda_\ell U^{-1}, \end{aligned} \tag{4.1}$$

where $(\Lambda_\ell)_{kk} = \lambda_{\ell,k}$, for $k = 1, \ldots, N_x$, are the eigenvalues of Φ_ℓ, given by

$$\lambda_{\ell,k} = 1 + \delta_\ell \xi_k \mathfrak{b}^T (I - \delta_\ell \xi_k \mathfrak{A})^{-1}\mathbf{1}. \tag{4.2}$$

Note that Equation (4.2) is exactly the stability function for a Runge-Kutta scheme applied to $\delta_\ell\xi_k$, for time step δ_ℓ and spatial eigenvalue $\{\xi_k\}$ [89, Sec. 2.1, §.4]. This highlights the fact that solving the spatial eigenvalue problem also provides the eigenvalues of all Φ_ℓ for arbitrary Runge-Kutta schemes and time-step sizes.

Now suppose $\mathcal{A}$ is some matrix operator, where each entry is a rational function of time-stepping operators in $\{\Phi_\ell\}$. In particular, this applies to error and residual propagation operators of MGRIT that are derived in Section 4.4. Let $\widetilde{U}$ denote a block-diagonal matrix with diagonal blocks given by U. Then, as in [133],

$$\|\mathcal{A}(\Phi_0, \ldots, \Phi_{n_\ell - 1})\|_{(\widetilde{U}\widetilde{U}^*)^{-1}} = \sup_k \|\mathcal{A}(\lambda_{0,k}, \ldots, \lambda_{n_\ell-1,k})\|_2. \tag{4.3}$$

Thus, the $(\widetilde{U}\widetilde{U}^*)^{-1}$-norm of $\mathcal{A}(\Phi_0, ..., \Phi_{n_\ell-1})$ can be computed by maximizing the ℓ^2-norm of $\mathcal{A}$ over eigenvalues of $\{\Phi_\ell\}$. In the case that $\{\Phi_\ell\}$ are normal matrices, U is

unitary and the $(\widetilde{U}\widetilde{U}^*)^{-1}$-norm reduces to the standard Euclidean 2-norm. More generally, we have bounds on the ℓ^2-norm of $\mathcal{A}(\Phi_0, ..., \Phi_{n_\ell-1})$,

$$\begin{aligned}\frac{1}{\kappa(U)}\left(\sup_k \|\mathcal{A}(\lambda_{0,k},\ldots,\lambda_{n_\ell-1,k})^i\|_2\right)\\ \leq \|\mathcal{A}(\Phi_0,\ldots,\Phi_{n_\ell-1})^i\|_2 \leq \kappa(U)\left(\sup_k \|\mathcal{A}(\lambda_{0,k},\ldots,\lambda_{n_\ell-1,k})^i\|_2\right),\end{aligned} \tag{4.4}$$

for $i = 1, 2, \ldots$ applications of $\mathcal{A}$, where $\kappa(U)$ denotes the matrix condition number of U.[5]

Here, we are interested in $\mathcal{A}$ corresponding to the residual and error propagation operators of n_l-level MGRIT, denoted as $\mathcal{E}^{n_l}$ and $\mathcal{R}^{n_l}$, respectively. Convergence of MGRIT requires,

$$\|\,(\mathcal{E}^{n_l})^i\,\|, \|\,(\mathcal{R}^{n_l})^i\,\| \to 0, \tag{4.5}$$

as iteration i increases. To that end, bounding $\sup_k \|\,(\mathcal{E}^{n_l}(\lambda_{0,k},\ldots,\lambda_{n_\ell-1,k}))^i\,\| < 1$ for all k provides necessary and sufficient conditions for $\|\,(\mathcal{E}^{n_l}(\Phi_0,\ldots,\Phi_{n_\ell-1}))^i\,\| \to 0$ with i (eventually), and similarly for $\mathcal{R}^{n_l}(\Phi_0,\ldots,\Phi_{n_\ell-1})$. Here, we have focused on the ℓ^2-norm. It is worth pointing out that, in some cases, people are interested in an ℓ^∞-norm. However, because $\|\cdot\|_\infty \leq \|\cdot\|_2$, conditions developed here apply to the ℓ^∞-norm as well.

4.4 Multilevel residual and error propagation

As noted in [133], residual and error propagation are formally similar, that is,

$$\mathcal{R}^{n_\ell} = A_0\mathcal{E}^{n_\ell}A_0^{-1} = A_0(I - M^{-1}A_0)A_0^{-1} = I - A_0M^{-1}, \tag{4.6}$$

where M^{-1} denotes the MGRIT preconditioner for A_0^{-1}. Noting that there is a closed form for A_ℓ^{-1}, it follows that if the error propagation operator of a particular MGRIT algorithm is known, the residual propagation operator can be easily found by the relation in (4.6), and vice-versa. In this section, we derive the error propagation operator for generalized two-level MGRIT and multilevel (V-cycle) MGRIT with F- and FCF-relaxation, using the following convention for sums and products: $\sum_{i=a}^b f_i = 0$ and $\prod_{i=a}^b f_i = 1$ for $b < a$. We further write, e.g., $\mathcal{E}^{n_\ell=2}$ to refer to the two-grid error propagation operator and distinguish the notation from a power of $\mathcal{E}$, and similarly for other numbers of levels n_ℓ.

4.4.1 Two-level MGRIT with rFCF-relaxation

Here, we generalize the two-level error propagator, as given in [25], to two-level MGRIT with rFCF-relaxation. rFCF-relaxation refers to F-relaxation followed by r CF-relaxation steps. A similar result can be found in [42], where MGRIT was interpreted as Parareal with overlap in time.

The error propagator for an exact iterative two-grid method with rFCF-relaxation and $r \geq 0$ is given as,

$$\begin{aligned}0 = I - A_0^{-1}A_0 &= (I - P_0(R_0A_0P_0)^{-1}R_0A_0)(F_0C_0)^rF_0\\ &= (I - P_0(R_0A_0P_0)^{-1}R_0A_0)P_0(I - R_0A_0P_0)^rR_{I_0},\end{aligned} \tag{4.7}$$

and MGRIT approximates the coarse-grid operator as $A_1 \approx R_0A_0P_0$.

[5] A similar modified norm also occurs in the case of integrating in time with a mass matrix [25].

Lemma 1. *The error propagator of two-level MGRIT with rFCF-relaxation and $r \geq 0$ is given as,*

$$\mathcal{E}_{rFCF}^{n_\ell=2} = (I - P_0 A_1^{-1} R_0 A_0) P_0 (I - R_0 A_0 P_0)^r R_{I_0}. \tag{4.8}$$

Proof. This follows by substituting the coarse-grid operator $A_1 \approx R_0 A_0 P_0$ in Equation (4.7). □

4.4.2 Multilevel V-cycles with F-relaxation

The error propagator of a multilevel V-cycle method with F-relaxation can be derived from the error propagator of the exact two-level method on level ℓ,

$$\begin{aligned} 0 &= I - A_\ell^{-1} A_\ell = (I - P_\ell (R_\ell A_\ell P_\ell)^{-1} R_\ell A_\ell)(I - S_\ell (S_\ell^T A_\ell S_\ell)^{-1} S_\ell^T A_\ell) \\ &= I - (P_\ell (R_\ell A_\ell P_\ell)^{-1} R_\ell + S_\ell (S_\ell^T A_\ell S_\ell)^{-1} S_\ell^T) A_\ell \end{aligned}$$

and noting that,

$$A_\ell^{-1} = P_\ell (R_\ell A_\ell P_\ell)^{-1} R_\ell + S_\ell (S_\ell^T A_\ell S_\ell)^{-1} S_\ell^T. \tag{4.9}$$

Lemma 2. *The error propagator of a multilevel V-cycle method with F-relaxation is given as,*

$$\begin{aligned} \mathcal{E}_F^{n_\ell} = P_0 R_{I_0} &- \left(\prod_{k=0}^{n_\ell - 2} P_k \right) A_{n_\ell - 1}^{-1} \left(\prod_{k=n_\ell - 2}^{0} R_k \right) A_0 P_0 R_{I_0} \\ &- \sum_{i=0}^{n_\ell - 3} \left(\prod_{k=0}^{i} P_k \right) S_{i+1} (S_{i+1}^T A_{i+1} S_{i+1})^{-1} S_{i+1}^T \left(\prod_{k=i}^{0} R_k \right) A_0 P_0 R_{I_0}, \end{aligned} \tag{4.10}$$

for $n_\ell \geq 2$ levels.

Proof. For $n_\ell = 2$, we have,

$$\mathcal{E}_F^{n_\ell=2} = P_0 R_{I_0} - P_0 A_1^{-1} R_0 A_0 P_0 R_{I_0} = (I - P_0 A_1^{-1} R_0 A_0) P_0 R_{I_0},$$

which is equivalent with (4.8) for $r = 0$. Now, assume it is true for $n_\ell = n$ levels. Substituting an exact two-level method on the coarse grid, that is (4.9), yields,

$$\begin{aligned} \mathcal{E}_F^{n_\ell=n} &= P_0 R_{I_0} - \left(\prod_{k=0}^{n-2} P_k \right) \Big[P_{n-1} (R_{n-1} A_{n-1} P_{n-1})^{-1} R_{n-1} \\ &\qquad + S_{n-1} (S_{n-1}^T A_{n-1} S_{n-1})^{-1} S_{n-1}^T \Big] \left(\prod_{k=n-2}^{0} R_k \right) A_0 P_0 R_{I_0} \\ &\qquad - \sum_{i=0}^{n-3} \left(\prod_{k=0}^{i} P_k \right) S_{i+1} (S_{i+1}^T A_{i+1} S_{i+1})^{-1} S_{i+1}^T \left(\prod_{k=i}^{0} R_k \right) A_0 P_0 R_{I_0} \\ &= P_0 R_{I_0} - \left(\prod_{k=0}^{n-1} P_k \right) (R_{n-1} A_{n-1} P_{n-1})^{-1} \left(\prod_{k=n-1}^{0} R_k \right) A_0 P_0 R_{I_0} \\ &\qquad - \sum_{i=0}^{n-2} \left(\prod_{k=0}^{i} P_k \right) S_{i+1} (S_{i+1}^T A_{i+1} S_{i+1})^{-1} S_{i+1}^T \left(\prod_{k=i}^{0} R_k \right) A_0 P_0 R_{I_0}. \end{aligned}$$

Approximating the exact coarse grid operator on level $n+1$ by $A_n \approx R_{n-1}A_{n-1}P_{n-1}$ concludes the proof. □

4.4.3 Multilevel V-cycles with FCF-relaxation

The error propagator of a multilevel V-cycle method with FCF-relaxation can be derived from the error propagator of the exact two-level method on level ℓ,

$$\begin{aligned}0 =& I - A_\ell^{-1}A_\ell = (I - P_\ell(R_\ell A_\ell P_\ell)^{-1}R_\ell A_\ell)F_\ell C_\ell F_\ell \\ =& I - P_\ell(R_\ell A_\ell P_\ell)^{-1}R_\ell A_\ell - S_\ell(S_\ell^T A_\ell S_\ell)^{-1}S_\ell^T A_\ell - T_\ell(T_\ell^T A_\ell T_\ell)^{-1}T_\ell^T A_\ell \\ &+ S_\ell(S_\ell^T A_\ell S_\ell)^{-1}S_\ell^T A_\ell T_\ell(T_\ell^T A_\ell T_\ell)^{-1}T_\ell^T A_\ell + P_\ell(R_\ell A_\ell P_\ell)^{-1}R_\ell A_\ell T_\ell(T_\ell^T A_\ell T_\ell)^{-1}T_\ell^T A_\ell,\end{aligned}$$

and noting that,

$$A_\ell^{-1} = T_\ell(T_\ell^T A_\ell T_\ell)^{-1}T_\ell^T + \left[S_\ell(S_\ell^T A_\ell S_\ell)^{-1}S_\ell^T + P_\ell(R_\ell A_\ell P_\ell)^{-1}R_\ell\right]\left[I - A_\ell T_\ell(T_\ell^T A_\ell T_\ell)^{-1}T_\ell^T\right].$$

Lemma 3. *The error propagator of a multilevel V-cycle method with FCF-relaxation is given as,*

$$\begin{aligned}\mathcal{E}_{FCF}^{n_\ell} =& P_0(I - (T_0^T A_0 T_0)^{-1}R_{I_0}A_0P_0)R_{I_0} \\ &- \left(\prod_{k=0}^{n_\ell-2}P_k\right)A_{n_\ell-1}^{-1}\left(\prod_{k=n_\ell-2}^{0}R_k\left[I - A_kT_k(T_k^TA_kT_k)^{-1}T_k^T\right]\right)A_0P_0R_{I_0} \\ &- \sum_{i=1}^{n_\ell-2}\left(\prod_{k=0}^{i-1}P_k\right)\Big[S_i(S_i^TA_iS_i)^{-1}S_i^T\left[I - A_iT_i(T_i^TA_iT_i)^{-1}T_i^T\right] \\ &+ T_i(T_i^TA_iT_i)^{-1}T_i^T\Big]\left(\prod_{k=i-1}^{0}R_k\left[I - A_kT_k(T_k^TA_kT_k)^{-1}T_k^T\right]\right)A_0P_0R_{I_0},\end{aligned} \tag{4.11}$$

with $n_\ell \geq 2$ levels.

Proof. The proof is analogous to the proof of Lemma 2. □

4.4.4 Multilevel F-cycles with rFCF-relaxation

Following [53, p. 53], error propagation of MGRIT for a multilevel F-cycle[6] with rFCF-relaxation can be defined recursively,

$$\mathcal{F}_{rFCF}^{n_\ell} = M_{rFCF,0}^F, \qquad \text{for } n_\ell \geq 2, \tag{4.12}$$

with

$$\begin{aligned}M_{rFCF,\ell-1}^F &= P_{\ell-1}\left(I - \left(I - M_\ell^V M_\ell^F\right)A_\ell^{-1}R_{\ell-1}A_{\ell-1}P_{\ell-1}\right)(I - R_{\ell-1}A_{\ell-1}P_{\ell-1})^r R_{I_{\ell-1}}, \\ M_{rFCF,\ell-1}^V &= P_{\ell-1}\left(I - \left(I - M_\ell^V\right)A_\ell^{-1}R_{\ell-1}A_{\ell-1}P_{\ell-1}\right)(I - R_{\ell-1}A_{\ell-1}P_{\ell-1})^r R_{I_{\ell-1}},\end{aligned}$$

[6] To refer to F-cycle error propagation, we use the notation $\mathcal{F}^{n_\ell}$ instead of $\mathcal{E}^{n_\ell}$.

for $l = 1, \ldots, n_\ell - 2$, and,

$$\begin{aligned} M^F_{rFCF,n_\ell-2} &= M^V_{rFCF,n_\ell-2} \\ &= P_{n_\ell-2}\left(I - A^{-1}_{n_\ell-1}R_{n_\ell-2}A_{n_\ell-2}P_{n_\ell-2}\right)\left(I - R_{n_\ell-2}A_{n_\ell-2}P_{n_\ell-2}\right)^r R_{I_{n_\ell-2}}. \end{aligned}$$

It is easy to verify, that for $n_\ell = 2$, the recursive formulae result in $\mathcal{F}^{n_\ell=2}_{rFCF} = \mathcal{E}^{n_\ell=2}_{rFCF}$. For $n_\ell = 3$ and $r = 0$, we can write,

$$\mathcal{F}^{n_\ell=3}_F = \mathcal{E}^{n_\ell=2}_F + P_0P_1\left(I - A_2^{-1}R_1A_1P_1\right)^2 R_{I_1}A_1^{-1}R_0A_0P_0R_{I_0}.$$

However, it is not straightforward to convert the recursive definition in (4.12) into a summation similar to (4.10) or (4.11), for arbitrary n_ℓ. Yet, this formula is still useful for numerically and analytically computing bounds of $\mathcal{F}^{n_\ell}_{rFCF}$, hence, we include it.

4.5 Bounds for MGRIT residual and error propagation

Following the work in [25], we assume that operators Φ_ℓ, $\ell = 0, \ldots, n_\ell - 1$, can be diagonalized by the same set of eigenvectors (see Equation (4.1)), with eigenvalues denoted $\{\lambda_{\ell,k}\}$, for $1 \leq k \leq N_x$. We also assume that $\{\Phi_\ell\}$ are strongly stable time stepping operators, that is, $\|\Phi_\ell\| < 1$, which implies $|\lambda_{\ell,k}| < 1$ for all $\ell = 0, \ldots, n_\ell - 1$ and $k = 1, \ldots, N_x$, see Section 4.2, Assumption 5. To simplify notation in the following derivations, we use Φ_ℓ to denote the diagonalized time stepping operator moving forward. Results then follow in a $(\widetilde{U}\widetilde{U}^*)^{-1}$-norm, which (as discussed in Section 4.3) is equivalent to the ℓ^2-norm if Φ_ℓ is normal (and, thus, U is unitary).

4.5.1 Residual and error on level 0 and level 1

It is typically difficult or impossible in practical applications to precisely measure the error propagation of an iterative method or the error itself. It is, however, possible to measure the residual, and stopping criteria for iterative methods are often based on a residual tolerance. In the case of MGRIT, there is a nice relation between residual and error propagation. From [133], the norm of residual and error propagation operators are equal in the $(\widetilde{U}\widetilde{U}^*)^{-1}$-norm (recall, $\widetilde{U}$ is a block diagonal matrix of eigenvectors, U).[7] If $\{\Phi_\ell\}$ are normal operators, they are diagonalizable by unitary transformation, in which case $\widetilde{U}\widetilde{U}^* = I$, and residual and error propagation are equal in the ℓ^2-norm.

Similar to Section 4.4, let $\mathcal{E}^{n_\ell}_{rFCF}$ be the error propagator on level 0. We further refer to $\mathcal{E}^{n_\ell,\Delta}_{rFCF}$ as the error propagator on level 1, that denotes the operator that acts on the error at C-points on level 0, i.e., *all* points on level 1. In the two-grid setting, we also refer to $\mathcal{E}^{n_\ell,\Delta}_{rFCF}$ as the coarse-grid error propagator.

To quantify how *fast* MGRIT converges in the *worst case*, we can bound the convergence factor of the fine grid residual [25] $\mathbf{r}_{i+1}$ at iteration $i+1$, $i \in \mathbb{N}_0$, by the norm of the error propagator on level 1 (in the unitary case),

$$\|\mathbf{r}_{i+1}\|_2/\|\mathbf{r}_i\|_2 = \|A_1\mathbf{e}^\Delta_{i+1}\|_2/\|A_1\mathbf{e}^\Delta_i\|_2 \leq \|A_1\mathcal{E}^{n_\ell,\Delta}_{rFCF}A_1^{-1}\|_2 = \|\mathcal{E}^{n_\ell,\Delta}_{rFCF}\|_2, \tag{4.13}$$

[7] Although [133] specifically addresses two-grid bounds, equality of residual and error propagation in the $(\widetilde{U}\widetilde{U}^*)^{-1}$-norm follows if Φ_ℓ is simultaneously diagonalizable for all levels ℓ.

where $\mathbf{e}_{i+1}^{\Delta}$ is the error on level 1 or equivalently, error at C-points on level 0. With,

$$\mathbf{e}_{i+1}^{\Delta} = \mathcal{E}_{rFCF}^{n_\ell,\Delta}\mathbf{e}_i^{\Delta} = \mathcal{E}_{rFCF}^{n_\ell,\Delta}R_{I_0}\mathbf{e}_i, \quad \Leftrightarrow \quad P_0\mathbf{e}_{i+1}^{\Delta} = P_0\mathcal{E}_{rFCF}^{n_\ell,\Delta}\mathbf{e}_i^{\Delta} = P_0\mathcal{E}_{rFCF}^{n_\ell,\Delta}R_{I_0}\mathbf{e}_i,$$

we can identify, $\mathcal{E}_{rFCF}^{n_\ell,\Delta} = R_{I_0}\mathcal{E}_{rFCF}^{n_\ell}P_0$, which is a generalization of the approach in [25], where the operators P_0 and R_{I_0} are pulled out to the left and right of the error propagator. Thus, in general we analyze the error propagator on level 1 to bound residual propagation on level 0, as given in (4.13).

This raises the question of how the error develops at the F-points on the fine grid. Considering error propagation on level 0 over i iterations,

$$\mathbf{e}_{i+1} = \mathcal{E}_{rFCF}^{n_\ell}\mathbf{e}_i = \ldots = \left(\mathcal{E}_{rFCF}^{n_\ell}\right)^{i+1}\mathbf{e}_0 = \left(P_0\mathcal{E}_{rFCF}^{n_\ell,\Delta}R_{I_0}\right)^{i+1}\mathbf{e}_0 = P_0\left(\mathcal{E}_{rFCF}^{n_\ell,\Delta}\right)^{i+1}R_{I_0}\mathbf{e}_0,$$

we find that error propagation at the F-points of the fine grid can be bounded by error propagation at the respective C-points times a constant.

Lemma 4. *Error propagation on level* 0 *for an MGRIT V-cycle method can be bounded by error propagation on level* 1*,*

$$\|\mathcal{E}_{rFCF}^{n_\ell}\|_2 \leq \sqrt{m_0}\|\mathcal{E}_{rFCF}^{n_\ell,\Delta}\|_2,$$

with temporal coarsening factor m_0 *on level* 0*.*

Proof. This follows from,

$$\begin{aligned}\|\mathcal{E}_{rFCF}^{n_\ell}\|_2 &= \|P_0\mathcal{E}_{rFCF}^{n_\ell,\Delta}R_{I_0}\|_2 \leq \|P_0\|_2\|\mathcal{E}_{rFCF}^{n_\ell,\Delta}\|_2\|R_{I_0}\|_2 \\ &\leq \sqrt{\|P_0\|_1\|P_0\|_\infty}\|\mathcal{E}_{rFCF}^{n_\ell,\Delta}\|_2\sqrt{\|R_{I_0}\|_1\|R_{I_0}\|_\infty} \leq \sqrt{m_0}\|\mathcal{E}_{rFCF}^{n_\ell,\Delta}\|_2,\end{aligned}$$

where submultiplicativity and the inequality $\|D\|_2 \leq \sqrt{\|D\|_1\|D\|_\infty}$ (see [69]) were used. □

Remark 8. It is clear from Lemma 4, that if the error at C-points on level 0 converges, then the error at F-points on level 0 converges as well. This is the basis for the theory developed in the rest of this chapter, where convergence is attained by bounding $\mathcal{E}_{rFCF}^{n_\ell,\Delta}$ in norm.

Lemma 4 is intuitive in the sense that the fine grid error propagation is a direct result of the level 1 error propagation; it is simply ideal interpolation applied to the level 1 error; that is, the operator P_0 propagates the error at the C-points on level 0 to the subsequent F-points. A similar result was presented in [133] for two-level convergence of Parareal and MGRIT.

Based on the formulae derived in Section 4.4, we can construct residual and error propagators numerically and bound the *worst case* convergence factor of MGRIT from above via

$$c_f = \max_i \|\mathbf{r}_{i+1}\|_2/\|\mathbf{r}_i\|_2 \leq \|\mathcal{E}_{rFCF}^{n_\ell,\Delta}\|_2, \tag{4.14}$$

which corresponds to the maximum singular value of $\mathcal{E}_{rFCF}^{n_\ell,\Delta}$. In practice, the dimension of $\mathcal{E}_{rFCF}^{n_\ell}$ grows with the problem size in space and time, N_x and N_0. Similarly, $\mathcal{E}_{rFCF}^{n_\ell,\Delta}$

grows with N_x and N_1. Depending on the available resources, numerical construction and investigation of these operators may be limited by memory consumption and/or compute time. To that end, it is desirable to derive further *cheaper* upper bounds that enable fast assessment of MGRIT convergence for larger space-time problem sizes. In the following, we present several bounds and approximate convergence factors for fine-grid residual propagation and error propagation on level 1.

4.5.2 Upper bound using inequality

One straightforward way to reduce computational cost by roughly one order of magnitude is bounding the ℓ^2-norm of the error propagator on level 1 using the well-known inequality [69],

$$\|\mathcal{E}_{rFCF}^{n_\ell,\Delta}\|_2^2 \leq \|\mathcal{E}_{rFCF}^{n_\ell,\Delta}\|_1 \|\mathcal{E}_{rFCF}^{n_\ell,\Delta}\|_\infty. \tag{4.15}$$

In [25], this was used to develop an upper-bound on two-grid convergence, which was proven to be sharp in [133]. This section extends this approach to three and four grid levels based on analytic formulae. Although the sharpness of (4.15) suffers as the number of levels increases (see Chapter 5), it is shown that it is still reasonably sharp and provides a useful tool to analyze MGRIT convergence a priori.

4.5.2.1 Two-level MGRIT with rFCF-relaxation

The coarse-grid error propagator follows from Equation (4.8) for $n_\ell = 2$,

$$\mathcal{E}_{rFCF}^{n_\ell=2,\Delta} = R_{I_0}\mathcal{E}_{rFCF}^{n_\ell=2}P_0 = (I - A_1^{-1}R_0A_0P_0)(I - R_0A_0P_0)^r.$$

And with,

$$R_0A_0P_0 = \begin{bmatrix} I & & & \\ -\Phi_0^{m_0} & I & & \\ & -\Phi_0^{m_0} & I & \\ & & \ddots & \ddots \end{bmatrix}, \tag{4.16}$$

and,

$$\begin{aligned} I - A_1^{-1}R_0A_0P_0 &= I - \begin{bmatrix} I & & & \\ \Phi_1 & I & & \\ \Phi_1^2 & \Phi_1 & I & \\ \vdots & & & \ddots \end{bmatrix} \begin{bmatrix} I & & & \\ -\Phi_0^{m_0} & I & & \\ & -\Phi_0^{m_0} & I & \\ & & \ddots & \ddots \end{bmatrix} \\ &= -\begin{bmatrix} 0 & & & \\ \Phi_1 - \Phi_0^{m_0} & 0 & & \\ \Phi_1\left(\Phi_1 - \Phi_0^{m_0}\right) & \Phi_1 - \Phi_0^{m_0} & 0 & \\ \vdots & & & \ddots \\ \Phi_1^{N_1-2}\left(\Phi_1 - \Phi_0^{m_0}\right) & \cdots & & 0 \end{bmatrix}, \end{aligned} \tag{4.17}$$

and

$$
\begin{aligned}
(I - R_0A_0P_0)^r &= \begin{bmatrix} 0 & & & & \\ \Phi_0^{m_0} & 0 & & & \\ & \Phi_0^{m_0} & 0 & & \\ & & \ddots & \ddots & \end{bmatrix} (I - R_0A_0P_0)^{r-1} \\
&= \begin{bmatrix} 0 & & & & \\ 0 & & & & \\ \Phi_0^{2m_0} & 0 & & & \\ & \Phi_0^{2m_0} & 0 & & \\ & & \ddots & \ddots & \end{bmatrix} (I - R_0A_0P_0)^{r-2} \\
&= \ldots = \begin{bmatrix} 0 & & & & \\ \vdots & & & & \\ 0 & & & & \\ \Phi_0^{rm_0} & 0 & & & \\ & \Phi_0^{rm_0} & 0 & & \\ & & \ddots & \ddots & \end{bmatrix}. \qquad (4.18)
\end{aligned}
$$

the coarse-grid error propagator follows as,

$$
\mathcal{E}_{rFCF}^{n_\ell=2,\Delta} = \begin{bmatrix} 0 & & & & & & & \\ \vdots & & & & & & & \\ 0 & & & & & & & \\ (\Phi_0^{m_0} - \Phi_1)\Phi_0^{rm_0} & & & & & & & \\ \Phi_1(\Phi_0^{m_0} - \Phi_1)\Phi_0^{rm_0} & & (\Phi_0^{m_0} - \Phi_1)\Phi_0^{rm_0} & & & & & \\ \vdots & & & & & & & \\ \Phi_1^{N_1-r-2}(\Phi_0^{m_0} - \Phi_1)\Phi_0^{rm_0} & \Phi_1^{N_1-r-3}(\Phi_0^{m_0} - \Phi_1)\Phi_0^{rm_0} & \cdots & (\Phi_0^{m_0} - \Phi_1)\Phi_0^{rm_0} & 0 & \cdots & 0 \end{bmatrix}, \qquad (4.19)
$$

where the first $r+1$ rows and last $r+1$ columns are zero.

Theorem 1. *Let $\{\Phi_\ell\}$ be simultaneously diagonalizable by the same unitary transformation X, with eigenvalues $\{\lambda_{\ell,k}\}$, $|\lambda_{\ell,k}| < 1$. Then, the* worst case *convergence factor of the fine-grid residual of two-level MGRIT with rFCF-relaxation is bounded by,*

$$
c_f \leq \max_{1\leq k\leq N_x} |\lambda_{0,k}^{m_0} - \lambda_{1,k}||\lambda_{0,k}|^{rm_0} \frac{1 - |\lambda_{1,k}|^{N_1-1-r}}{1 - |\lambda_{1,k}|}.
$$

Proof. This follows from Equation (4.19) and inequality (4.15),

$$
\begin{aligned}
\|\mathcal{E}_{rFCF}^{n_\ell=2,\Delta}\|_2 &\leq \sqrt{\|\mathcal{E}_{rFCF}^{n_\ell=2,\Delta}\|_1 \|\mathcal{E}_{rFCF}^{n_\ell=2,\Delta}\|_\infty} = \|\mathcal{E}_{rFCF}^{n_\ell=2}\|_1 \\
&= \max_{1\leq k\leq N_x} |\lambda_{0,k}^{m_0} - \lambda_{1,k}||\lambda_{0,k}|^{rm_0} \frac{1 - |\lambda_{1,k}|^{N_1-1-r}}{1 - |\lambda_{1,k}|}.
\end{aligned}
$$

In the last step, the relationship $\sum_{i=0}^{N-1} a^i = (1-a^N)/(1-a)$ was used. □

Remark 9. The cases of F- and FCF-relaxation (i.e., $r = 0$ and $r = 1$), yield the result in [25],

$$\|\mathcal{E}_F^{n_\ell=2,\Delta}\|_2 \leq \max_{1\leq k\leq N_x} |\lambda_{0,k}^{m_0} - \lambda_{1,k}| \frac{1-|\lambda_{1,k}|^{N_1-1}}{1-|\lambda_{1,k}|},$$
$$\|\mathcal{E}_{FCF}^{n_\ell=2,\Delta}\|_2 \leq \max_{1\leq k\leq N_x} |\lambda_{0,k}^{m_0} - \lambda_{1,k}||\lambda_{0,k}|^{m_0} \frac{1-|\lambda_{1,k}|^{N_1-2}}{1-|\lambda_{1,k}|}.$$

Remark 10. In [133], it was shown that the bound in Theorem 1 is exact to $O(1/N_1)$ for F- and FCF-relaxation.

An interesting observation of (4.19) is the fact that the coarse-grid error propagator is nilpotent and that each block can be diagonalized by the same unitary transformation. This implies that we can reorder the rows and columns of the coarse-grid error propagator, yielding a block diagonal form with lower triangular nilpotent blocks.

Lemma 5. *Let Φ_0 and Φ_1 be simultaneously diagonalizable by the same unitary transformation X, with eigenvalues $\{\lambda_{\ell,k}\}$, $|\lambda_{\ell,k}| < 1$. Then, the ℓ^2-norm of the coarse-grid error propagator of two-level MGRIT with rFCF-relaxation can be computed as,*

$$\|\mathcal{E}_{rFCF}^{n_\ell=2,\Delta}\|_2 = \sup_{1\leq k\leq N_x} \|\tilde{\mathcal{E}}_{rFCF}^{n_\ell=2,\Delta}(k)\|_2,$$

with the coarse-grid error propagator $\tilde{\mathcal{E}}_{rFCF}^{n_\ell=2,\Delta}(k)$ for a single spatial mode k with $1 \leq k \leq N_x$.

Proof. This follows from the discussion above and the fact that the spectral norm of a block diagonal operator with lower triangular blocks can be computed as the supremum of the spectral norm of all lower triangular blocks. See also [25], Remark 3.1. □

Remark 11. Lemma 5 implies that computing a bound of the form of (4.14) can be parallelized over the number of spatial modes. Thus, the time complexity of evaluating (4.14) is $O(N_x N_1^3/p)$ with $1 \leq p \leq N_x$ parallel processors.

Remark 12. Lemma 5 formalizes and generalizes the discussion for two-level MGRIT with F- and FCF-relaxation in [36, Section 4.2].

Remark 13. The result in Lemma 5 is not limited to $n_\ell = 2$ and can be applied to all subsequent convergence results.

4.5.2.2 Three-level V-cycles with F-relaxation

Evaluating the error propagator in Equation (4.10) for a three-level V-cycle with F-relaxation on the coarse-grid yields,

$$\mathcal{E}_F^{n_\ell=3,\Delta} = R_{I_0}\mathcal{E}_F^{n_\ell=3}P_0 = I - \left[P_1 A_2^{-1} R_1 + S_1 (S_1^T A_1 S_1)^{-1} S_1^T\right] R_0 A_0 P_0,$$

where,

$$S_1(S_1^T A_1 S_1)^{-1} S_1^T = \begin{bmatrix} 0 & & & & & & & & & & \\ & I & & & & & & & & & \\ & \Phi_1 & I & & & & & & & & \\ & \Phi_1^2 & \Phi_1 & I & & & & & & & \\ & \vdots & & & \ddots & & & & & & \\ & \Phi_1^{m_1-2} & \Phi_1^{m_1-3} & \cdots & \Phi_1 & I & & & & & \\ & & & & & 0 & & & & & \\ & & & & & & I & & & & \\ & & & & & & \Phi_1 & I & & & \\ & & & & & & \Phi_1^2 & \Phi_1 & I & & \\ & & & & & & \vdots & & & \ddots & \\ & & & & & & \Phi_1^{m_1-2} & \Phi_1^{m_1-3} & \cdots & \Phi_1 & I \\ & & & & & & & & & & 0 \\ & & & & & & & & & & & \ddots \\ & & & & & & & & & & & & 0 \end{bmatrix} \tag{4.20}$$

and

$$P_1 A_2^{-1} R_1 = \begin{bmatrix} I & & & & \\ \Phi_1 & & & & \\ \Phi_1^2 & & & & \\ \vdots & & & & \\ \Phi_1^{m_1-1} & & & & \\ \Phi_2 & \Phi_1^{m_1-1} & \Phi_1^{m_1-2} & \cdots & I \\ \Phi_1\Phi_2 & \Phi_1\Phi_1^{m_1-1} & \Phi_1\Phi_1^{m_1-2} & \cdots & \Phi_1 \\ \vdots & \vdots & \vdots & & \vdots \\ \Phi_1^{m_1-1}\Phi_2 & \Phi_1^{m_1-1}\Phi_1^{m_1-1} & \Phi_1^{m_1-1} I \Phi_1^{m_1-2} & \cdots & \Phi_1^{m_1-1} \\ \vdots & \vdots & \vdots & & \vdots \\ \vdots & \vdots & \vdots & & \vdots \\ \Phi_2^{N_2-2} & \Phi_2^{N_2-3}\Phi_1^{m_1-1} & \Phi_2^{N_2-3}\Phi_1^{m_1-2} & \cdots & \Phi_2^{N_2-3} \\ \Phi_1\Phi_2^{N_2-2} & \Phi_1\Phi_2^{N_2-3}\Phi_1^{m_1-1} & \Phi_1\Phi_2^{N_2-3}\Phi_1^{m_1-2} & \cdots & \Phi_1\Phi_2^{N_2-3} \\ \vdots & \vdots & \vdots & & \vdots \\ \Phi_1^{m_1-1}\Phi_2^{N_2-2} & \Phi_1^{m_1-1}\Phi_2^{N_2-3}\Phi_1^{m_1-1} & \Phi_1^{m_1-1}\Phi_2^{N_2-3}\Phi_1^{m_1-2} & \cdots & \Phi_1^{m_1-1}\Phi_2^{N_2-3} \\ \Phi_2^{N_2-1} & \Phi_2^{N_2-2}\Phi_1^{m_1-1} & \Phi_2^{N_2-2}\Phi_1^{m_1-2} & \cdots & \Phi_2^{N_2-2} \end{bmatrix} \begin{matrix} \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{matrix} \ddots \tag{4.21}$$

With (4.16), (4.20) and (4.21) follows the nilpotent operator,

$$\mathcal{E}_F^{n_\ell=3,\Delta} = \begin{bmatrix} 0 & & \\ \Phi_0^{m_0} - \Phi_1 & 0 & \\ \Phi_1(\Phi_0^{m_0} - \Phi_1) & \Phi_0^{m_0} - \Phi_1 & 0 \\ \Phi_1^2(\Phi_0^{m_0} - \Phi_1) & \Phi_1(\Phi_0^{m_0} - \Phi_1) & \Phi_0^{m_0} - \Phi_1 \\ \vdots & \vdots & \vdots \\ \Phi_1^{m_1-2}(\Phi_0^{m_0} - \Phi_1) & \Phi_1^{m_1-3}(\Phi_0^{m_0} - \Phi_1) & \Phi_1^{m_1-4}(\Phi_0^{m_0} - \Phi_1) \\ \Phi_1^{m_1-1}\Phi_0^{m_0} - \Phi_2 & \Phi_1^{m_1-2}(\Phi_0^{m_0} - \Phi_1) & \Phi_1^{m_1-3}(\Phi_0^{m_0} - \Phi_1) \\ \Phi_1(\Phi_1^{m_1-1}\Phi_0^{m_0} - \Phi_2) & \Phi_1\Phi_1^{m_1-2}(\Phi_0^{m_0} - \Phi_1) & \Phi_1\Phi_1^{m_1-3}(\Phi_0^{m_0} - \Phi_1) \\ \vdots & \vdots & \vdots \\ \Phi_1^{m_1-1}(\Phi_1^{m_1-1}\Phi_0^{m_0} - \Phi_2) & \Phi_1^{m_1-1}\Phi_1^{m_1-2}(\Phi_0^{m_0} - \Phi_1) & \Phi_1^{m_1-1}\Phi_1^{m_1-3}(\Phi_0^{m_0} - \Phi_1) \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ \Phi_2^{N_2-3}(\Phi_1^{m_1-1}\Phi_0^{m_0} - \Phi_2) & \Phi_2^{N_2-3}\Phi_1^{m_1-2}(\Phi_0^{m_0} - \Phi_1) & \Phi_2^{N_2-3}\Phi_1^{m_1-3}(\Phi_0^{m_0} - \Phi_1) \\ \Phi_1\Phi_2^{N_2-3}(\Phi_1^{m_1-1}\Phi_0^{m_0} - \Phi_2) & \Phi_1\Phi_2^{N_2-3}\Phi_1^{m_1-2}(\Phi_0^{m_0} - \Phi_1) & \Phi_1\Phi_2^{N_2-3}\Phi_1^{m_1-3}(\Phi_0^{m_0} - \Phi_1) \\ \vdots & \vdots & \vdots \\ \Phi_1^{m_1-1}\Phi_2^{N_2-3}(\Phi_1^{m_1-1}\Phi_0^{m_0} - \Phi_2) & \Phi_1^{m_1-1}\Phi_2^{N_2-3}\Phi_1^{m_1-2}(\Phi_0^{m_0} - \Phi_1) & \Phi_1^{m_1-1}\Phi_2^{N_2-3}\Phi_1^{m_1-3}(\Phi_0^{m_0} - \Phi_1) \\ \Phi_2^{N_2-2}(\Phi_1^{m_1-1}\Phi_0^{m_0} - \Phi_2) & \Phi_2^{N_2-2}\Phi_1^{m_1-2}(\Phi_0^{m_0} - \Phi_1) & \Phi_2^{N_2-2}\Phi_1^{m_1-3}(\Phi_0^{m_0} - \Phi_1) \end{bmatrix} \begin{matrix} \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{matrix} \ddots \tag{4.22}$$

Comparing the error propagator of a three-level V-cycle with F-relaxation on level 1 (see Equation (4.22)) with the two-level error propagator for F-relaxation in [25] highlights a slight complication: In general, the maximum absolute column sum (and similarly, for the maximum absolute row sum) is no longer given by the first column. Instead, the maximum absolute column sum is given by the maximum of the first m_1 absolute column sums, corresponding to the first CF-interval (first C-point and first $m_1 - 1$ F-points) on level 1. This structure arises because of the recursive partitioning of time points into F- and C-points on each level.

Theorem 2. *Let $\{\Phi_\ell\}$ be simultaneously diagonalizable by the same unitary transformation X, with eigenvalues $\{\lambda_{\ell,k}\}$, $|\lambda_{\ell,k}| < 1$. Then, the* worst case *convergence factor of three-level V-cycles with F-relaxation is bounded by,*

$$c_f \leq \sqrt{\|\mathcal{E}_F^{n_\ell=3,\Delta}\|_1 \|\mathcal{E}_F^{n_\ell=3,\Delta}\|_\infty}.$$

and $\|\mathcal{E}_F^{n_\ell=3,\Delta}\|_1$ and $\|\mathcal{E}_F^{n_\ell=3,\Delta}\|_\infty$ are given analytically as,

$$\|\mathcal{E}_F^{n_\ell=3,\Delta}\|_1 = \max_{1\leq k\leq N_x} \begin{cases} |\lambda_{2,k} - \lambda_{0,k}^{m_0}\lambda_{1,k}^{m_1-1}| \left(|\lambda_{2,k}|^{N_2-2} + \frac{1-|\lambda_{2,k}|^{N_2-2}}{1-|\lambda_{2,k}|}\frac{1-|\lambda_{1,k}|^{m_1}}{1-|\lambda_{1,k}|}\right) \\ \qquad + |\lambda_{1,k} - \lambda_{0,k}^{m_0}|\frac{1-|\lambda_{1,k}|^{m_1-1}}{1-|\lambda_{1,k}|} \\ \hdashline |\lambda_{1,k}|^{j-1}|\lambda_{1,k} - \lambda_{0,k}^{m_0}| \left[|\lambda_{2,k}|^{N_2-2} + \frac{1-|\lambda_{2,k}|^{N_2-2}}{1-|\lambda_{2,k}|}\frac{1-|\lambda_{1,k}|^{m_1}}{1-|\lambda_{1,k}|}\right] \\ \qquad + |\lambda_{1,k} - \lambda_{0,k}^{m_0}|\frac{1-|\lambda_{1,k}|^{m_1-2}}{1-|\lambda_{1,k}|} \qquad \textit{for } j = 1,\ldots,m_1-1, \end{cases} \tag{4.23}$$

and

$$\|\mathcal{E}_F^{n_\ell=3,\Delta}\|_\infty = \max_{1\leq k\leq N_x} \begin{cases} |\lambda_{2,k} - \lambda_{0,k}^{m_0}\lambda_{1,k}^{m_1-1}|\frac{1-|\lambda_{2,k}|^{N_2-1}}{1-|\lambda_{2,k}|} \\ \qquad + |\lambda_{1,k} - \lambda_{0,k}^{m_0}|\frac{1-|\lambda_{2,k}|^{N_2-1}}{1-|\lambda_{2,k}|}\frac{1-|\lambda_{1,k}|^{m_1-1}}{1-|\lambda_{1,k}|} \\ \hdashline |\lambda_{1,k} - \lambda_{0,k}^{m_0}|\frac{1-|\lambda_{1,k}|^{j}}{1-|\lambda_{1,k}|} + |\lambda_{1,k}|^j\frac{1-|\lambda_{2,k}|^{N_2-2}}{1-|\lambda_{2,k}|}|\lambda_{2,k} - \lambda_{0,k}^{m_0}\lambda_{1,k}^{m_1-1}| \\ \qquad + |\lambda_{1,k}|^j\frac{1-|\lambda_{2,k}|^{N_2-2}}{1-|\lambda_{2,k}|}|\lambda_{1,k} - \lambda_{0,k}^{m_0}|\frac{1-|\lambda_{1,k}|^{m_1-1}}{1-|\lambda_{1,k}|} \\ \qquad \textit{for } j = 1,\ldots,m_1-1. \end{cases} \tag{4.24}$$

Proof. The proof is analogous to Theorem 1. □

4.5.2.3 Four-level V-cycles with F-relaxation

Evaluating the error propagator in Equation (4.10) for a four-level V-cycle with F-relaxation on level 1 yields,

$$\begin{aligned} \mathcal{E}_F^{n_\ell=4,\Delta} &= R_{I_0}\mathcal{E}_F^{n_\ell=4}P_0 \\ &= I - \left[P_1P_2A_3^{-1}R_2R_1 + S_1(S_1^TA_1S_1)^{-1}S_1^T + P_1S_2(S_2^TA_2S_2)^{-1}S_2^TR_1\right]R_0A_0P_0, \end{aligned}$$

where,

$$
P_1 S_2 (S_2^T A_2 S_2)^{-1} S_2^T R_1 =
\begin{bmatrix}
0 \\ \vdots \\ 0
\end{bmatrix}
\begin{bmatrix}
\Phi_1^{m_1-1} & \cdots & I & & & & & & & & & & & & \\
\vdots & \cdots & \vdots & & & & & & & & & & & & \\
\Phi_1^{m_1-1}\Phi_1^{m_1-1} & \cdots & \Phi_1^{m_1-1} & & & & & & & & & & & & \\
\Phi_2\Phi_1^{m_1-1} & \cdots & \Phi_2 & & \Phi_1^{m_1-1} & \cdots & I & & & & & & & & \\
\vdots & & \vdots & & \vdots & & \vdots & & & & & & & & \\
\Phi_1^{m_1-1}\Phi_2\Phi_1^{m_1-1} & \cdots & \Phi_1^{m_1-1}\Phi_2 & & \Phi_1^{m_1-1}\Phi_1^{m_1-1} & \cdots & \Phi_1^{m_1-1} & & & & & & & & \\
\vdots & & \vdots & & \vdots & & & \ddots & & & & & & & \\
\Phi_2^{m_2-2}\Phi_1^{m_1-1} & \cdots & \Phi_2^{m_2-2} & & \Phi_2^{m_2-3}\Phi_1^{m_1-1} & \cdots & \Phi_2^{m_2-3} & \cdots & \cdots & I & & & & & \\
\vdots & & \vdots & & \vdots & & \vdots & & & \vdots & & & & & \\
\Phi_1^{m_1-1}\Phi_2^{m_2-2}\Phi_1^{m_1-1} & \cdots & \Phi_1^{m_1-1}\Phi_2^{m_2-2} & & \Phi_1^{m_1-1}\Phi_2^{m_2-3}\Phi_1^{m_1-1} & \cdots & \Phi_1^{m_1-1}\Phi_2^{m_2-3} & \cdots & \cdots & \Phi_1^{m_1-1} & & & & & \\
 & & & & & & & & & & 0 & & & & \\
 & & & & & & & & & & & \ddots & & & \\
 & & & & & & & & & & & & & & 0
\end{bmatrix}
\tag{4.25}
$$

and,

$$P_1P_2A_3^{-1}R_2R_1 = \begin{bmatrix} I & & & & & & \\ \vdots & & & & & & \\ \Phi_1^{m_1-1} & & & & & & \\ \Phi_2 & & & & & & \\ \vdots & & & & & & \\ \Phi_1^{m_1-1}\Phi_2^{m_2-1} & & & & & & \\ \Phi_3 & \Phi_2^{m_2-1}\Phi_1^{m_1-1} & \cdots & \Phi_2^{m_2-1} & \Phi_2^{m_2-2}\Phi_1^{m_1-1} & \cdots & I \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ \Phi_1^{m_1-1}\Phi_3 & \Phi_1^{m_1-1}\Phi_2^{m_2-1}\Phi_1^{m_1-1} & \cdots & \Phi_1^{m_1-1}\Phi_2^{m_2-1} & \Phi_1^{m_1-1}\Phi_2^{m_2-2}\Phi_1^{m_1-1} & \cdots & \Phi_1^{m_1-1} \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ \Phi_2^{m_2-1}\Phi_3 & \Phi_2^{m_2-1}\Phi_2^{m_2-1}\Phi_1^{m_1-1} & \cdots & \Phi_2^{m_2-1}\Phi_2^{m_2-1} & \Phi_2^{m_2-1}\Phi_2^{m_2-2}\Phi_1^{m_1-1} & \cdots & \Phi_2^{m_2-1} \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ \Phi_1^{m_1-1}\Phi_2^{m_2-1}\Phi_3 & \Phi_1^{m_1-1}\Phi_2^{m_2-1}\Phi_2^{m_2-1}\Phi_1^{m_1-1} & \cdots & \Phi_1^{m_1-1}\Phi_2^{m_2-1}\Phi_2^{m_2-1} & \Phi_1^{m_1-1}\Phi_2^{m_2-1}\Phi_2^{m_2-2}\Phi_1^{m_1-1} & \cdots & \Phi_1^{m_1-1}\Phi_2^{m_2-1} & \ddots \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ \Phi_1^{m_1-1}\Phi_3^{N_3-2} & \Phi_1^{m_1-1}\Phi_3^{N_3-3}\Phi_2^{m_2-1}\Phi_1^{m_1-1} & \cdots & \Phi_1^{m_1-1}\Phi_3^{N_3-3}\Phi_2^{m_2-1} & \Phi_1^{m_1-1}\Phi_3^{N_3-3}\Phi_2^{m_2-2}\Phi_1^{m_1-1} & \cdots & \Phi_1^{m_1-1}\Phi_3^{N_3-3} \\ \Phi_2\Phi_3^{N_3-2} & \Phi_2\Phi_3^{N_3-3}\Phi_2^{m_2-1}\Phi_1^{m_1-1} & \cdots & \Phi_2\Phi_3^{N_3-3}\Phi_2^{m_2-1} & \Phi_2\Phi_3^{N_3-3}\Phi_2^{m_2-2}\Phi_1^{m_1-1} & \cdots & \Phi_2\Phi_3^{N_3-3} \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ \Phi_1^{m_1-1}\Phi_2\Phi_3^{N_3-2} & \Phi_1^{m_1-1}\Phi_2\Phi_3^{N_3-3}\Phi_2^{m_2-1}\Phi_1^{m_1-1} & \cdots & \Phi_1^{m_1-1}\Phi_2\Phi_3^{N_3-3}\Phi_2^{m_2-1} & \Phi_1^{m_1-1}\Phi_2\Phi_3^{N_3-3}\Phi_2^{m_2-2}\Phi_1^{m_1-1} & \cdots & \Phi_1^{m_1-1}\Phi_2\Phi_3^{N_3-3} \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ \Phi_2^{m_2-1}\Phi_3^{N_3-2} & \Phi_2^{m_2-1}\Phi_3^{N_3-3}\Phi_2^{m_2-1}\Phi_1^{m_1-1} & \cdots & \Phi_2^{m_2-1}\Phi_3^{N_3-3}\Phi_2^{m_2-1} & \Phi_2^{m_2-1}\Phi_3^{N_3-3}\Phi_2^{m_2-2}\Phi_1^{m_1-1} & \cdots & \Phi_2^{m_2-1}\Phi_3^{N_3-3} \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ \Phi_1^{m_1-1}\Phi_2^{m_2-1}\Phi_3^{N_3-2} & \Phi_1^{m_1-1}\Phi_2^{m_2-1}\Phi_3^{N_3-3}\Phi_2^{m_2-1}\Phi_1^{m_1-1} & \cdots & \Phi_1^{m_1-1}\Phi_2^{m_2-1}\Phi_3^{N_3-3}\Phi_2^{m_2-1} & \Phi_1^{m_1-1}\Phi_2^{m_2-1}\Phi_3^{N_3-3}\Phi_2^{m_2-2}\Phi_1^{m_1-1} & \cdots & \Phi_1^{m_1-1}\Phi_2^{m_2-1}\Phi_3^{N_3-3} \\ \Phi_3^{N_3-1} & \Phi_3^{N_3-2}\Phi_2^{m_2-1}\Phi_1^{m_1-1} & \cdots & \Phi_3^{N_3-2}\Phi_2^{m_2-1} & \Phi_3^{N_3-2}\Phi_2^{m_2-2}\Phi_1^{m_1-1} & \cdots & \Phi_3^{N_3-2} \end{bmatrix}. \tag{4.26}$$

The sum of (4.20), (4.25) and (4.26) yields,

$$P_1P_2A_3^{-1}R_2R_1 + S_1(S_1^TA_1S_1)^{-1}S_1^T + P_1S_2(S_2^TA_2S_2)^{-1}S_2^TR_1$$

$$= \begin{bmatrix} I & & & & & & & & \\ \Phi_1 & I & & & & & & & \\ \vdots & \vdots & \ddots & & & & & & \\ \Phi_1^{m_1-1} & \Phi_1^{m_1-2} & & & & & & & \\ \Phi_2 & \Phi_1^{m_1-1} & \cdots & I & & & & & \\ \vdots & \vdots & & \vdots & \ddots & & & & \\ \vdots & \vdots & & \vdots & & & & & \\ \Phi_1^{m_1-1}\Phi_2^{m_2-1} & \Phi_1^{m_1-1}\Phi_2^{m_2-2}\Phi_1^{m_1-1} & \cdots & \Phi_1^{m_1-1}\Phi_2^{m_2-2} & \Phi_1^{m_1-1}\Phi_2^{m_2-3}\Phi_1^{m_1-1} & \ddots & & & \\ \Phi_3 & \Phi_2^{m_2-1}\Phi_1^{m_1-1} & \cdots & \Phi_2^{m_2-1} & \Phi_2^{m_2-2}\Phi_1^{m_1-1} & \cdots & I & & \\ \vdots & \vdots & & \vdots & \vdots & & \vdots & & \\ \Phi_1^{m_1-1}\Phi_3 & \Phi_1^{m_1-1}\Phi_2^{m_2-1}\Phi_1^{m_1-1} & \cdots & \Phi_1^{m_1-1}\Phi_2^{m_2-1} & \Phi_1^{m_1-1}\Phi_2^{m_2-2}\Phi_1^{m_1-1} & \cdots & \Phi_1^{m_1-1} & & \\ \vdots & \vdots & & \vdots & \vdots & & \vdots & & \\ \vdots & \vdots & & \vdots & \vdots & & \vdots & & \\ \Phi_2^{m_2-1}\Phi_3 & \Phi_2^{m_2-1}\Phi_2^{m_2-1}\Phi_1^{m_1-1} & \cdots & \Phi_2^{m_2-1}\Phi_2^{m_2-1} & \Phi_2^{m_2-1}\Phi_2^{m_2-2}\Phi_1^{m_1-1} & \cdots & \Phi_2^{m_2-1} & & \\ \vdots & \vdots & & \vdots & \vdots & & \vdots & & \\ \Phi_1^{m_1-1}\Phi_2^{m_2-1}\Phi_3 & \Phi_1^{m_1-1}\Phi_2^{m_2-1}\Phi_2^{m_2-1}\Phi_1^{m_1-1} & \cdots & \Phi_1^{m_1-1}\Phi_2^{m_2-1}\Phi_2^{m_2-1} & \Phi_1^{m_1-1}\Phi_2^{m_2-1}\Phi_2^{m_2-2}\Phi_1^{m_1-1} & \cdots & \Phi_1^{m_1-1}\Phi_2^{m_2-1} & \ddots & \\ \vdots & \vdots & & \vdots & \vdots & & \vdots & & \\ \vdots & \vdots & & \vdots & \vdots & & \vdots & & \\ \Phi_1^{m_1-1}\Phi_3^{N_3-2} & \Phi_1^{m_1-1}\Phi_3^{N_3-3}\Phi_2^{m_2-1}\Phi_1^{m_1-1} & \cdots & \Phi_1^{m_1-1}\Phi_3^{N_3-3}\Phi_2^{m_2-1} & \Phi_1^{m_1-1}\Phi_3^{N_3-3}\Phi_2^{m_2-2}\Phi_1^{m_1-1} & \cdots & \Phi_1^{m_1-1}\Phi_3^{N_3-3} & & \\ \Phi_2\Phi_3^{N_3-2} & \Phi_2\Phi_3^{N_3-3}\Phi_2^{m_2-1}\Phi_1^{m_1-1} & \cdots & \Phi_2\Phi_3^{N_3-3}\Phi_2^{m_2-1} & \Phi_2\Phi_3^{N_3-3}\Phi_2^{m_2-2}\Phi_1^{m_1-1} & \cdots & \Phi_2\Phi_3^{N_3-3} & & \\ \vdots & \vdots & & \vdots & \vdots & & \vdots & & \\ \Phi_1^{m_1-1}\Phi_2\Phi_3^{N_3-2} & \Phi_1^{m_1-1}\Phi_2\Phi_3^{N_3-3}\Phi_2^{m_2-1}\Phi_1^{m_1-1} & \cdots & \Phi_1^{m_1-1}\Phi_2\Phi_3^{N_3-3}\Phi_2^{m_2-1} & \Phi_1^{m_1-1}\Phi_2\Phi_3^{N_3-3}\Phi_2^{m_2-2}\Phi_1^{m_1-1} & \cdots & \Phi_1^{m_1-1}\Phi_2\Phi_3^{N_3-3} & & \\ \vdots & \vdots & & \vdots & \vdots & & \vdots & & \\ \vdots & \vdots & & \vdots & \vdots & & \vdots & & \\ \Phi_2^{m_2-1}\Phi_3^{N_3-2} & \Phi_2^{m_2-1}\Phi_3^{N_3-3}\Phi_2^{m_2-1}\Phi_1^{m_1-1} & \cdots & \Phi_2^{m_2-1}\Phi_3^{N_3-3}\Phi_2^{m_2-1} & \Phi_2^{m_2-1}\Phi_3^{N_3-3}\Phi_2^{m_2-2}\Phi_1^{m_1-1} & \cdots & \Phi_2^{m_2-1}\Phi_3^{N_3-3} & & \\ \vdots & \vdots & & \vdots & \vdots & & \vdots & & \\ \Phi_1^{m_1-1}\Phi_2^{m_2-1}\Phi_3^{N_3-2} & \Phi_1^{m_1-1}\Phi_2^{m_2-1}\Phi_3^{N_3-3}\Phi_2^{m_2-1}\Phi_1^{m_1-1} & \cdots & \Phi_1^{m_1-1}\Phi_2^{m_2-1}\Phi_3^{N_3-3}\Phi_2^{m_2-1} & \Phi_1^{m_1-1}\Phi_2^{m_2-1}\Phi_3^{N_3-3}\Phi_2^{m_2-2}\Phi_1^{m_1-1} & \cdots & \Phi_1^{m_1-1}\Phi_2^{m_2-1}\Phi_3^{N_3-3} & & \\ \Phi_3^{N_3-1} & \Phi_3^{N_3-2}\Phi_2^{m_2-1}\Phi_1^{m_1-1} & \cdots & \Phi_3^{N_3-2}\Phi_2^{m_2-1} & \Phi_3^{N_3-2}\Phi_2^{m_2-2}\Phi_1^{m_1-1} & \cdots & \Phi_3^{N_3-2} & & \end{bmatrix},$$

where we notice that the sparsity patterns are nonoverlapping. Then, the error propagator is given as,

$$\mathcal{E}_F^{n_t=4,\Delta} = -\begin{bmatrix}
0 & 0 & 0 & \cdots & 0 & \cdots & 0 & 0 \\
\Phi_1 - \Phi_0^{m_0} & 0 & 0 & \cdots & 0 & \cdots & 0 & 0 \\
\Phi_1(\Phi_1 - \Phi_0^{m_0}) & \Phi_1 - \Phi_0^{m_0} & 0 & \cdots & 0 & \cdots & 0 & 0 \\
\vdots & \vdots & \ddots & \ddots & \vdots & \ddots & \vdots & \vdots \\
\Phi_1^{m_1-2}(\Phi_1 - \Phi_0^{m_0}) & \Phi_1^{m_1-3}(\Phi_1 - \Phi_0^{m_0}) & \cdots & 0 & \cdots & 0 & 0 & 0 \\
\Phi_2 - \Phi_1^{m_1-1}\Phi_0^{m_0} & \Phi_1^{m_1-2}(\Phi_1 - \Phi_0^{m_0}) & \cdots & 0 & \cdots & 0 & 0 & 0 \\
\cdots & \cdots & \cdots & \cdots & \ddots & \cdots & \cdots & \cdots \\
\Phi_1^{m_1-1}\Phi_2^{m_2-2}(\Phi_2 - \Phi_1^{m_1-1}\Phi_0^{m_0}) & \Phi_1^{m_1-1}\Phi_2^{m_2-3}\Phi_1^{m_1-2}(\Phi_1 - \Phi_0^{m_0}) & \cdots & \Phi_1^{m_1-1}\Phi_2^{m_2-3}(\Phi_2 - \Phi_1^{m_1-1}\Phi_0^{m_0}) & \cdots & \ddots & \cdots & \cdots \\
\Phi_3 - \Phi_2^{m_2-1}\Phi_1^{m_1-1}\Phi_0^{m_0} & \Phi_2^{m_2-1}\Phi_1^{m_1-2}(\Phi_1 - \Phi_0^{m_0}) & \cdots & \Phi_2^{m_2-2}(\Phi_2 - \Phi_1^{m_1-1}\Phi_0^{m_0}) & \cdots & 0 & \cdots & \cdots \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \ddots & \cdots \\
\Phi_1^{m_1-1}(\Phi_3 - \Phi_2^{m_2-1}\Phi_1^{m_1-1}\Phi_0^{m_0}) & \Phi_1^{m_1-1}\Phi_2^{m_2-1}\Phi_1^{m_1-2}(\Phi_1 - \Phi_0^{m_0}) & \cdots & \Phi_1^{m_1-1}\Phi_2^{m_2-2}(\Phi_2 - \Phi_1^{m_1-1}\Phi_0^{m_0}) & \cdots & \Phi_1^{m_1-2}(\Phi_1 - \Phi_0^{m_0}) & \cdots & \cdots \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
\Phi_2^{m_2-1}(\Phi_3 - \Phi_2^{m_2-1}\Phi_1^{m_1-1}\Phi_0^{m_0}) & \Phi_2^{m_2-1}\Phi_1^{m_1-2}(\Phi_1 - \Phi_0^{m_0}) & \cdots & \Phi_2^{m_2-2}(\Phi_2 - \Phi_1^{m_1-1}\Phi_0^{m_0}) & \cdots & \Phi_2^{m_2-2}(\Phi_2 - \Phi_1^{m_1-1}\Phi_0^{m_0}) & \cdots & \cdots \\
\Phi_1^{m_1-1}\Phi_2^{m_2-1}(\Phi_3 - \Phi_2^{m_2-1}\Phi_1^{m_1-1}\Phi_0^{m_0}) & \Phi_1^{m_1-1}\Phi_2^{m_2-1}\Phi_1^{m_1-2}(\Phi_1 - \Phi_0^{m_0}) & \cdots & \Phi_1^{m_1-1}\Phi_2^{m_2-2}(\Phi_2 - \Phi_1^{m_1-1}\Phi_0^{m_0}) & \cdots & \Phi_1^{m_1-1}\Phi_2^{m_2-3}(\Phi_2 - \Phi_1^{m_1-1}\Phi_0^{m_0}) & \cdots & \cdots \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
\Phi_1^{m_1-1}\Phi_2^{N_3-3}(\Phi_3 - \Phi_2^{m_2-1}\Phi_1^{m_1-1}\Phi_0^{m_0}) & \Phi_1^{m_1-1}\Phi_2^{N_3-3}\Phi_1^{m_2-1}\Phi_1^{m_1-2}(\Phi_1 - \Phi_0^{m_0}) & \cdots & \Phi_1^{m_1-1}\Phi_2^{N_3-3}\Phi_2^{m_2-2}(\Phi_2 - \Phi_1^{m_1-1}\Phi_0^{m_0}) & \cdots & \Phi_1^{m_1-1}\Phi_2^{N_3-4}(\Phi_3 - \Phi_2^{m_2-1}\Phi_1^{m_1-1}\Phi_0^{m_0}) & \cdots & \cdots \\
\Phi_2\Phi_3^{N_3-3}(\Phi_3 - \Phi_2^{m_2-1}\Phi_1^{m_1-1}\Phi_0^{m_0}) & \Phi_2\Phi_3^{N_3-3}\Phi_2^{m_2-1}\Phi_1^{m_1-2}(\Phi_1 - \Phi_0^{m_0}) & \cdots & \Phi_2\Phi_3^{N_3-3}\Phi_2^{m_2-2}(\Phi_2 - \Phi_1^{m_1-1}\Phi_0^{m_0}) & \cdots & \Phi_2\Phi_3^{N_3-4}(\Phi_3 - \Phi_2^{m_2-1}\Phi_1^{m_1-1}\Phi_0^{m_0}) & \cdots & \cdots \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
\Phi_1^{m_1-1}\Phi_2\Phi_3^{N_3-3}(\Phi_3 - \Phi_2^{m_2-1}\Phi_1^{m_1-1}\Phi_0^{m_0}) & \Phi_1^{m_1-1}\Phi_2\Phi_3^{N_3-3}\Phi_2^{m_2-1}\Phi_1^{m_1-2}(\Phi_1 - \Phi_0^{m_0}) & \cdots & \Phi_1^{m_1-1}\Phi_2\Phi_3^{N_3-3}\Phi_2^{m_2-2}(\Phi_2 - \Phi_1^{m_1-1}\Phi_0^{m_0}) & \cdots & \Phi_1^{m_1-1}\Phi_2\Phi_3^{N_3-4}(\Phi_3 - \Phi_2^{m_2-1}\Phi_1^{m_1-1}\Phi_0^{m_0}) & \cdots & \cdots \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
\Phi_2^{m_2-1}\Phi_3^{N_3-3}(\Phi_3 - \Phi_2^{m_2-1}\Phi_1^{m_1-1}\Phi_0^{m_0}) & \Phi_2^{m_2-1}\Phi_3^{N_3-3}\Phi_2^{m_2-1}\Phi_1^{m_1-2}(\Phi_1 - \Phi_0^{m_0}) & \cdots & \Phi_2^{m_2-1}\Phi_3^{N_3-3}\Phi_2^{m_2-2}(\Phi_2 - \Phi_1^{m_1-1}\Phi_0^{m_0}) & \cdots & \Phi_2^{m_2-1}\Phi_3^{N_3-4}(\Phi_3 - \Phi_2^{m_2-1}\Phi_1^{m_1-1}\Phi_0^{m_0}) & \cdots & \cdots \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
\Phi_1^{m_1-1}\Phi_2^{m_2-1}\Phi_3^{N_3-3}(\Phi_3 - \Phi_2^{m_2-1}\Phi_1^{m_1-1}\Phi_0^{m_0}) & \Phi_1^{m_1-1}\Phi_2^{m_2-1}\Phi_3^{N_3-3}\Phi_2^{m_2-1}\Phi_1^{m_1-2}(\Phi_1 - \Phi_0^{m_0}) & \cdots & \Phi_1^{m_1-1}\Phi_2^{m_2-1}\Phi_3^{N_3-3}\Phi_2^{m_2-2}(\Phi_2 - \Phi_1^{m_1-1}\Phi_0^{m_0}) & \cdots & \Phi_1^{m_1-1}\Phi_2^{m_2-1}\Phi_3^{N_3-4}(\Phi_3 - \Phi_2^{m_2-1}\Phi_1^{m_1-1}\Phi_0^{m_0}) & \cdots & \cdots \\
\Phi_3^{N_3-2}(\Phi_3 - \Phi_2^{m_2-1}\Phi_1^{m_1-1}\Phi_0^{m_0}) & \Phi_3^{N_3-2}\Phi_2^{m_2-1}\Phi_1^{m_1-2}(\Phi_1 - \Phi_0^{m_0}) & \cdots & \Phi_3^{N_3-2}\Phi_2^{m_2-2}(\Phi_2 - \Phi_1^{m_1-1}\Phi_0^{m_0}) & \cdots & \Phi_3^{N_3-3}(\Phi_3 - \Phi_2^{m_2-1}\Phi_1^{m_1-1}\Phi_0^{m_0}) & \cdots & \Phi_1 - \Phi_0^{m_0}
\end{bmatrix} \tag{4.27}$$

This yields the following result.

Theorem 3. *Let $\{\Phi_\ell\}$ be simultaneously diagonalizable by the same unitary transformation X, with eigenvalues $\{\lambda_{\ell,k}\}$, $|\lambda_{\ell,k}| < 1$. Then, the* worst case *convergence factor of four-level V-cycles with F-relaxation is bounded by,*

$$c_f \leq \sqrt{\|\mathcal{E}_F^{n_\ell=4,\Delta}\|_1 \|\mathcal{E}_F^{n_\ell=4,\Delta}\|_\infty}, \tag{4.28}$$

and $\|\mathcal{E}_F^{n_\ell=4,\Delta}\|_1$ and $\|\mathcal{E}_F^{n_\ell=4,\Delta}\|_\infty$ are given analytically as,

$$\|\mathcal{E}_F^{n_\ell=4,\Delta}\|_1 = \max_{\substack{1\leq k\leq N_x \\ 0\leq d\leq m_1m_2-1}} s_d^{col}(k), \qquad \|\mathcal{E}_F^{n_\ell=4,\Delta}\|_\infty = \max_{\substack{1\leq k\leq N_x \\ 0\leq d\leq m_1m_2-1}} s_d^{row}(k),$$

where the column and row sums, s_d^{col} and s_d^{row} (row and column subscripts d), are defined as follows. The absolute column sums of the first CF-interval on level 1 *are given as,*

$$\begin{aligned}
s_0^{col}(k) &= |\lambda_{3,k}|^{N_3-2}|\lambda_{3,k} - \lambda_{0,k}^{m_0}\lambda_{1,k}^{m_1-1}\lambda_{2,k}^{m_2-1}| \\
&\quad + |\lambda_{3,k} - \lambda_{0,k}^{m_0}\lambda_{1,k}^{m_1-1}\lambda_{2,k}^{m_2-1}|\frac{1-|\lambda_{3,k}|^{N_3-2}}{1-|\lambda_{3,k}|}\frac{1-|\lambda_{2,k}|^{m_2}}{1-|\lambda_{2,k}|}\frac{1-|\lambda_{1,k}|^{m_1}}{1-|\lambda_{1,k}|} \\
&\quad + |\lambda_{2,k} - \lambda_{0,k}^{m_0}\lambda_{1,k}^{m_1-1}|\frac{1-|\lambda_{2,k}|^{m_2-1}}{1-|\lambda_{2,k}|}\frac{1-|\lambda_{1,k}|^{m_1}}{1-|\lambda_{1,k}|} + |\lambda_{1,k} - \lambda_{0,k}^{m_0}|\frac{1-|\lambda_{1,k}|^{m_1-1}}{1-|\lambda_{1,k}|},
\end{aligned}$$

corresponding to the first C-point on level 1*. Next,*

$$\begin{aligned}
s_{m_1(m_2-j)}^{col}(k) &= |\lambda_{2,k} - \lambda_{0,k}^{m_0}\lambda_{1,k}^{m_1-1}|\left(\textstyle\sum_{p=0}^{j-2}|\lambda_{2,k}|^p\right)\frac{1-|\lambda_{1,k}|^{m_1}}{1-|\lambda_{1,k}|} + |\lambda_{1,k} - \lambda_{0,k}^{m_0}|\frac{1-|\lambda_{1,k}|^{m_1-1}}{1-|\lambda_{1,k}|} \\
&\quad + |\lambda_{2,k}|^{j-1}|\lambda_{2,k} - \lambda_{0,k}^{m_0}\lambda_{1,k}^{m_1-1}|\left[|\lambda_{3,k}|^{N_3-2} + \frac{1-|\lambda_{3,k}|^{N_3-2}}{1-|\lambda_{3,k}|}\frac{1-|\lambda_{2,k}|^{m_2}}{1-|\lambda_{2,k}|}\frac{1-|\lambda_{1,k}|^{m_1-1}}{1-|\lambda_{1,k}|}\right],
\end{aligned}$$

for $j = 1, \ldots, m_2-1$, corresponding to the interior level 2 *C-points of the first CF-interval on level* 1*. Lastly,*

$$\begin{aligned}
s_{m_1(m_2-j)-r-1}^{col}(k) &= |\lambda_{1,k} - \lambda_{0,k}^{m_0}|\left[\left(\sum_{q=0}^{m_1-1}|\lambda_{1,k}|^q\right)\left(\sum_{p=0}^{j-1}|\lambda_{2,k}|^p\right) + \left(\sum_{q=0}^{r-1}|\lambda_{1,k}|^q\right)\right] \\
&\quad + |\lambda_{2,k}|^j|\lambda_{1,k}|^r|\lambda_{1,k} - \lambda_{0,k}^{m_0}|\left[|\lambda_{3,k}|^{N_3-2} + \frac{1-|\lambda_{3,k}|^{N_3-2}}{1-|\lambda_{3,k}|}\frac{1-|\lambda_{2,k}|^{m_2}}{1-|\lambda_{2,k}|}\frac{1-|\lambda_{1,k}|^{m_1}}{1-|\lambda_{1,k}|}\right],
\end{aligned}$$

for $j = 0, \ldots, m_2-1$ and $r = 0, \ldots, m_1-2$, corresponding to the level 2 *F-points of the first CF-interval on level* 1*.*

The absolute row sums of the last FC-interval on level 1 *are given as,*

$$\begin{aligned}
s_{N_1-1}^{row}(k) = \frac{1-|\lambda_{3,k}|^{N_3-1}}{1-|\lambda_{3,k}|}\Bigg[&|\lambda_{1,k} - \lambda_{0,k}^{m_0}|\frac{1-|\lambda_{2,k}|^{m_2}}{1-|\lambda_{2,k}|}\frac{1-|\lambda_{1,k}|^{m_1-1}}{1-|\lambda_{1,k}|} \\
&+ |\lambda_{2,k} - \lambda_{0,k}^{m_0}\lambda_{1,k}^{m_1-1}|\frac{1-|\lambda_{2,k}|^{m_2-1}}{1-|\lambda_{2,k}|} + |\lambda_{3,k} - \lambda_{0,k}^{m_0}\lambda_{1,k}^{m_1-1}\lambda_{2,k}^{m_2-1}|\Bigg],
\end{aligned}$$

corresponding to the last C-point on level 1. *Next,*

$$
\begin{aligned}
s^{row}_{N_1-1-m_1m_2+jm_1}(k) = \frac{1-|\lambda_{2,k}|^j}{1-|\lambda_{2,k}|}\left[|\lambda_{1,k}-\lambda_{0,k}^{m_0}|\frac{1-|\lambda_{1,k}|^{m_1-1}}{1-|\lambda_{1,k}|}+|\lambda_{2,k}-\lambda_{0,k}^{m_0}\lambda_{1,k}^{m_1-1}|\right] \\
+|\lambda_{2,k}|^j\frac{1-|\lambda_{3,k}|^{N_3-2}}{1-|\lambda_{3,k}|}\Bigg[|\lambda_{1,k}-\lambda_{0,k}^{m_0}|\frac{1-|\lambda_{2,k}|^{m_2}}{1-|\lambda_{2,k}|}\frac{1-|\lambda_{1,k}|^{m_1-1}}{1-|\lambda_{1,k}|} \\
+|\lambda_{2,k}-\lambda_{0,k}^{m_0}\lambda_{1,k}^{m_1-1}|\frac{1-|\lambda_{2,k}|^{m_2-1}}{1-|\lambda_{2,k}|}+|\lambda_{3,k}-\lambda_{0,k}^{m_0}\lambda_{1,k}^{m_1-1}\lambda_{2,k}^{m_2-1}|\Bigg],
\end{aligned}
$$

for $j = 1, \ldots, m_2 - 1$, *corresponding to the interior C-points of the last FC-interval on level* 1. *Lastly,*

$$
\begin{aligned}
s^{row}_{N_1-1-m_1m_2+r+jm_1}(k) = |\lambda_{1,k}|^r|\lambda_{2,k}|^j\frac{1-|\lambda_{3,k}|^{N_3-2}}{1-|\lambda_{3,k}|} \\
\left[|\lambda_{2,k}-\lambda_{0,k}^{m_0}\lambda_{1,k}^{m_1-1}|\frac{1-|\lambda_{2,k}|^{m_2-1}}{1-|\lambda_{2,k}|}+|\lambda_{3,k}-\lambda_{0,k}^{m_0}\lambda_{1,k}^{m_1-1}\lambda_{2,k}^{m_2-1}|\right] \\
+|\lambda_{1,k}|^r\left(\sum_{q=0}^{j-1}|\lambda_{2,k}|^q\right)\left[|\lambda_{2,k}-\lambda_{0,k}^{m_0}\lambda_{1,k}^{m_1-1}|+|\lambda_{1,k}-\lambda_{0,k}^{m_0}|\frac{1-|\lambda_{1,k}|^{m_1-1}}{1-|\lambda_{1,k}|}\right] \\
+|\lambda_{1,k}|^r|\lambda_{1,k}-\lambda_{0,k}^{m_0}|\frac{1-|\lambda_{3,k}|^{N_3-2}}{1-|\lambda_{3,k}|}\frac{1-|\lambda_{2,k}|^{m_2}}{1-|\lambda_{2,k}|}\frac{1-|\lambda_{1,k}|^{m_1-1}}{1-|\lambda_{1,k}|} \\
+|\lambda_{1,k}-\lambda_{0,k}^{m_0}|\frac{1-|\lambda_{1,k}|^r}{1-|\lambda_{1,k}|},
\end{aligned}
$$

for $j = 0, \ldots, m_2 - 1$ *and* $r = 1, \ldots, m_1 - 1$, *corresponding to the F-points of the last FC-interval on level* 1.

Proof. The proof is analogous to Theorem 2. □

Remark 14. We note, that evaluating the $2m_1m_2$ analytic formulae in Theorem 3 significantly reduce the time complexity of evaluating Equation (4.28) compared to constructing $\mathcal{E}_F^{n_\ell=4,\Delta}$ numerically and computing $\|\mathcal{E}_F^{n_\ell=4,\Delta}\|_1$ and $\|\mathcal{E}_F^{n_\ell=4,\Delta}\|_\infty$.

4.5.2.4 Three-level V-cycles with FCF-relaxation

Following the same approach as in the previous sections, we can find the following result for three-level V-cycles with FCF-relaxation.

Theorem 4. *Let* $\{\Phi_\ell\}$ *be simultaneously diagonalizable by the same unitary transformation* X, *with eigenvalues* $\{\lambda_{\ell,k}\}$, $|\lambda_{\ell,k}| < 1$. *Then, the* worst case *convergence factor of three-level V-cycles with FCF-relaxation is bounded by,*

$$
c_f \leq \sqrt{\|\mathcal{E}_{FCF}^{n_\ell=3,\Delta}\|_1\|\mathcal{E}_{FCF}^{n_\ell=3,\Delta}\|_\infty},
$$

and $\|\mathcal{E}_F^{n_\ell=3,\Delta}\|_1$ *and* $\|\mathcal{E}_F^{n_\ell=3,\Delta}\|_\infty$ *are given analytically as,*

$$
\|\mathcal{E}_{FCF}^{n_\ell=3,\Delta}\|_1 = \max_{\substack{1\leq k\leq N_x \\ 0\leq d\leq m_1m_2-1}} s_d^{col}(k), \qquad \|\mathcal{E}_{FCF}^{n_\ell=3,\Delta}\|_\infty = \max_{\substack{1\leq k\leq N_x \\ 0\leq d\leq m_1m_2-1}} s_d^{row}(k).
$$

The absolute column sums of the first CF-interval on level 1 *are given as,*

$$s_0^{col}(k) = |\lambda_{0,k}|^{m_0}|\lambda_{1,k} - \lambda_{0,k}^{m_0}|\left[|\lambda_{1,k}|^{m_1-1}\left(|\lambda_{2,k}|^{N_2-3} + \tfrac{1-|\lambda_{2,k}|^{N_2-3}}{1-|\lambda_{2,k}|}\tfrac{1-|\lambda_{1,k}|^{m_1}}{1-|\lambda_{1,k}|}\right) + \tfrac{1-|\lambda_{1,k}|^{m_1-1}}{1-|\lambda_{1,k}|}\right],$$

corresponding to the last F-point on level 1*. Next,*

$$\begin{aligned} s_{m_1-2}^{col}(k) = |\lambda_{0,k}|^{m_0}\Bigg[&|\lambda_{1,k} - \lambda_{0,k}^{m_0}|\frac{1-|\lambda_{1,k}|^{m_1}}{1-|\lambda_{1,k}|} \\ &+ |\lambda_{1,k}||\lambda_{2,k} - \lambda_{0,k}^{m_0}\lambda_{1,k}^{m_1-1}|\left(|\lambda_{2,k}|^{N_2-3} + \frac{1-|\lambda_{2,k}|^{N_2-3}}{1-|\lambda_{2,k}|}\frac{1-|\lambda_{1,k}|^{m_1}}{1-|\lambda_{1,k}|}\right)\Bigg], \end{aligned}$$

corresponding to the first C-point on level 1 *if* $m_1 = 2$*, or the penultimate F-point on level* 1 *if* $m_1 > 2$*. Lastly, if* $m_1 > 2$*,*

$$s_{m_1-2-j}^{col}(k) = |\lambda_{0,k}|^{m_0}|\lambda_{1,k}|^j|\lambda_{1,k} - \lambda_{0,k}^{m_0}|\left[|\lambda_{2,k}|^{N_2-2} + \tfrac{1-|\lambda_{1,k}|^j}{1-|\lambda_{1,k}|} + \tfrac{1-|\lambda_{2,k}|^{N_2-2}}{1-|\lambda_{2,k}|}\tfrac{1-|\lambda_{1,k}|^{m_1}}{1-|\lambda_{1,k}|}\right],$$

for $j = 1, \ldots, m_1 - 2$*, corresponding to the first C-point and the following F-points on level* 1*.*

The absolute row sums of the last FC-interval on level 1 *are given as,*

$$\begin{aligned} s_{N_1-1}^{row}(k) = &|\lambda_{0,k}|^{m_0}|\lambda_{1,k}||\lambda_{2,k} - \lambda_{0,k}^{m_0}\lambda_{1,k}^{m_1-1}|\frac{1-|\lambda_{2,k}|^{N_2-2}}{1-|\lambda_{2,k}|} \\ &+|\lambda_{0,k}|^{m_0}\left[|\lambda_{1,k} - \lambda_{0,k}^{m_0}|\left(1 + |\lambda_{1,k}|\tfrac{1-|\lambda_{2,k}|^{N_2-2}}{1-|\lambda_{2,k}|}\tfrac{1-|\lambda_{1,k}|^{m_1-1}}{1-|\lambda_{1,k}|} + |\lambda_{1,k}||\lambda_{2,k}|^{N_2-2}\left(\textstyle\sum_{q=0}^{m_1-3}|\lambda_{1,k}|^q\right)\right)\right], \end{aligned}$$

corresponding to the last C-point on level 1*, and,*

$$\begin{aligned} s_{N_1-m_1+j}^{row}(k) = |\lambda_{0,k}|^{m_0}\Bigg[&|\lambda_{1,k} - \lambda_{0,k}^{m_0}|\frac{1-|\lambda_{1,k}|^{j+2}}{1-|\lambda_{1,k}|} \\ &+ |\lambda_{1,k}|^{j+2}\frac{1-|\lambda_{2,k}|^{N_2-3}}{1-|\lambda_{2,k}|}\left(|\lambda_{2,k} - \lambda_{0,k}^{m_0}\lambda_{1,k}^{m_1-1}| + |\lambda_{1,k} - \lambda_{0,k}^{m_0}|\frac{1-|\lambda_{1,k}|^{m_1-1}}{1-|\lambda_{1,k}|}\right)\Bigg], \end{aligned}$$

for $j = 0, \ldots, m_1 - 2$*,corresponding to the preceding F-points on level* 1*.*

Proof. The proof is analogous to Theorem 2. □

4.5.2.5 Three-level F-cycles with F-relaxation

Following the same approach as in the previous sections, we can find the following result for three-level F-cycles with F-relaxation.

Theorem 5. *Let* $\{\Phi_\ell\}$ *be simultaneously diagonalizable by the same unitary transformation* X*, with eigenvalues* $\{\lambda_{\ell,k}\}$*,* $|\lambda_{\ell,k}| < 1$*. Then, the* worst case *convergence factor of three-level F-cycles with F-relaxation is bounded by,*

$$c_f \leq \sqrt{\|\mathcal{F}_F^{n_\ell=3,\Delta}\|_1\|\mathcal{F}_F^{n_\ell=3,\Delta}\|_\infty},$$

and $\|\mathcal{F}_F^{n_\ell=3,\Delta}\|_1$ *and* $\|\mathcal{F}_F^{n_\ell=3,\Delta}\|_\infty$ *are given analytically as,*

$$\|\mathcal{F}_F^{\Delta,n_l=3}\|_1 = \max_{1\le k\le N_x} \begin{cases} |\lambda_{1,k}-\lambda_{0,k}^{m_0}|\sum_{r=0}^{b}|\lambda_{1,k}|^r \\ \quad +\left(\sum_{r=0}^{m_1-1}|\lambda_{1,k}|^r\right)\left(\sum_{q=1}^{N_2-3}|\lambda_{2,k}|^{q-1}|q\lambda_{0,k}^{m_0}\lambda_{1,k}^{b+1}-(q+1)\lambda_{1,k}^{m_1}\lambda_{2,k}-(q-1)\lambda_{0,k}^{m_0}\lambda_{1,k}^{m_1-1}\lambda_{2,k}+q\lambda_{2,k}^2|\right) \\ \quad +|\lambda_{2,k}|^{N_2-3}|(N_2-2)\lambda_{0,k}^{m_0}\lambda_{1,k}^{b+1}-(N_2-1)\lambda_{1,k}^{m_1}\lambda_{2,k}-(N_2-3)\lambda_{0,k}^{m_0}\lambda_{1,k}^{m_1-1}\lambda_{2,k}+(N_2-2)\lambda_{2,k}^2|, \\ \hline |\lambda_{1,k}-\lambda_{0,k}^{m_0}|\Bigg[\left(\sum_{r=0}^{b+m_1-j}|\lambda_{1,k}|^r\right) \\ \quad +\left(\sum_{r=0}^{m_1-1}|\lambda_{1,k}|^{r+m_1-1-j}\right)\left(\sum_{q=1}^{N_2-4}|\lambda_{2,k}|^q|q\lambda_{2,k}-(q+1)\lambda_{1,k}^{m_1}|\right) \\ \quad +|\lambda_{1,k}|^{m_1-1-j}|\lambda_{2,k}|^{N_2-3}|(N_2-3)\lambda_{2,k}-(N_2-2)\lambda_{1,k}^{m_1}|\Bigg] \qquad \text{for } j=1,\dots,m_1-1, \end{cases}$$

with $b = N_1 - m_1N_2 + 3m_1 - 3$*, and,*

$$\|\mathcal{F}_F^{\Delta,n_l=3}\|_1 = \max_{1\le k\le N_x} \begin{cases} |\lambda_{1,k}-\lambda_{0,k}^{m_0}|\sum_{r=0}^{b}|\lambda_{1,k}|^r+\left(\sum_{q=1}^{N_2-2}|\lambda_{2,k}|^{q-1}|q\lambda_{0,k}^{m_0}\lambda_{1,k}^{b+1}-(q+1)\lambda_{1,k}^{m_1}\lambda_{2,k}-(q-1)\lambda_{0,k}^{m_0}\lambda_{1,k}^{m_1-1}\lambda_{2,k}+q\lambda_{2,k}^2|\right) \\ \quad +\left(\sum_{r=0}^{m_1-2}|\lambda_{1,k}|^r\right)\left(\sum_{q=1}^{N_2-3}|\lambda_{2,k}|^q|\lambda_{1,k}-\lambda_{0,k}^{m_0}||q\lambda_{2,k}-(q+1)\lambda_{1,k}^{m_1}|\right), \\ \hline |\lambda_{1,k}-\lambda_{0,k}^{m_0}|\left(\sum_{r=0}^{b+m_1+j-N_1}|\lambda_{1,k}|^r\right) \\ \quad +\left(\sum_{q=1}^{N_2-3}|\lambda_{1,k}|^{m_1-1+j-(N_1-1)}|\lambda_{2,k}|^{q-1}|q\lambda_{0,k}^{m_0}\lambda_{1,k}^{b+1}-(q+1)\lambda_{1,k}^{m_1}\lambda_{2,k}-(q-1)\lambda_{0,k}^{m_0}\lambda_{1,k}^{m_1-1}\lambda_{2,k}+q\lambda_{2,k}^2|\right) \\ \quad +\left(\sum_{r=b-1-m_1+2}^{b-1}|\lambda_{1,k}|^{r-(N_1-1)+j}\right)\left(\sum_{q=1}^{N_2-4}|\lambda_{2,k}|^q|\lambda_{1,k}-\lambda_{0,k}^{m_0}||q\lambda_{2,k}-(q+1)\lambda_{1,k}^{m_1}|\right) \\ \quad \text{for } j=N_1-m_1+1,\dots,N_1-1. \end{cases}$$

Proof. The proof is analogous to Theorem 2. □

4.5.3 Approximate convergence factor

In Section 4.5.2, we have presented analytic formulae as exact representations of the inequality bound (4.15). These *a priori* convergence bounds reduce memory consumption and computational cost significantly. It is, however, increasingly difficult to derive such analytic formulae for larger numbers of levels. Thus, we propose an analytic approximate convergence factor for,

- multilevel V-cycles with F- and FCF-relaxation,
- multilevel F-cycles with F-relaxation,

that takes the eigenvalues $\lambda_{\ell,k}$ of the time stepping operator Φ_ℓ, the number of time points N_ℓ, and the temporal coarsening factors m_ℓ for each level as parameters. This yields approximate *a priori* convergence factors with linear memory and time complexity.[8] The proposed approximate convergence factors are based on approximating the inequality bound (4.15), and therefore, are expected to be a conservative upper bound in a large number of cases. More specifically, the multilevel formulae generalize the analytic formulae in Section 4.5.2 in an approximate manner.

First, we present the approximate convergence factor for multilevel V-cycles with F-relaxation.

[8]The generalization of Lemma 5 implies that time complexity is in fact $O(N_x/p)$ with $1 \le p \le N_x$ parallel processors.

Approximation 1. *Let $\{\Phi_\ell\}$ be simultaneously diagonalizable by the same unitary transformation X, with eigenvalues $\{\lambda_{\ell,k}\}$, $|\lambda_{\ell,k}| < 1$. Then, the* approximate *convergence factor of multilevel MGRIT V-cycles with F-relaxation is given as,*

$$\tilde{c}_f \approx \max_{1\leq k\leq N_x} \sqrt{s_0^{row}(k,n_\ell)\ s_{N_1-1}^{col}(k,n_\ell)} \approx \sqrt{\|\mathcal{E}_F^{n_\ell,\Delta}\|_1 \|\mathcal{E}_F^{n_\ell,\Delta}\|_\infty}, \tag{4.29}$$

with approximate *maximum absolute column and row sum,*

$$\begin{aligned} s_0^{col}(k,n_\ell) \approx & \sum_{\ell=1}^{n_\ell-1} \left|\lambda_{\ell,k} - \lambda_{0,k}^{m_0}\left(\prod_{p=1}^{l-1}\lambda_{p,k}^{\tilde{m}_p-1}\right)\right| \left(\prod_{q=1}^{l}\frac{1-|\lambda_{q,k}|^{\tilde{m}_q-1}}{1-|\lambda_{q,k}|}\right) \\ & + (n_\ell > 2)\cdot |\lambda_{n_\ell-1,k}|^{\tilde{m}_{n_\ell-1}-1}\left|\lambda_{n_\ell-1,k} - \lambda_{0,k}^{m_0}\left(\prod_{p=1}^{n_\ell-2}\lambda_{p,k}^{\tilde{m}_p-1}\right)\right|, \\ s_{N_1-1}^{row}(k,n_\ell) \approx & \sum_{\ell=1}^{n_\ell-1} \left|\lambda_{\ell,k} - \lambda_{0,k}^{m_0}\left(\prod_{p=1}^{l-1}\lambda_{p,k}^{\tilde{m}_p-1}\right)\right| \left(\prod_{q=l}^{n_\ell-1}\frac{1-|\lambda_{q,k}|^{\tilde{m}_q}}{1-|\lambda_{q,k}|}\right), \end{aligned}$$

with $\tilde{m}_\ell = [m_0,\ldots,m_{n_\ell-2},N_{n_\ell-1}-1]^T$. Furthermore, in many cases, $\|\mathcal{E}_F^{n_\ell,\Delta}\|_2 \leq \tilde{c}_f$, because the analytic formulae approximate the right-hand-side of Equation (4.15) *(compare Equation* (4.29)*).*

A similar result can be formulated for multilevel V-cycles with FCF-relaxation.

Approximation 2. *Let $\{\Phi_\ell\}$ be simultaneously diagonalizable by the same unitary transformation X, with eigenvalues $\{\lambda_{\ell,k}\}$, $|\lambda_{\ell,k}| < 1$. Then, the* approximate *convergence factor of multilevel MGRIT V-cycles with FCF-relaxation is given as,*

$$\tilde{c}_f \approx \max_{1\leq k\leq N_x} \sqrt{s_0^{row}(k,n_\ell)\ s_{N_1-1}^{col}(k,n_\ell)} \approx \sqrt{\|\mathcal{E}_{FCF}^{n_\ell,\Delta}\|_1 \|\mathcal{E}_{FCF}^{n_\ell,\Delta}\|_\infty}, \tag{4.30}$$

with approximate *maximum absolute column sum,*

$$\begin{aligned} s_0^{col}(k,n_\ell) \approx\ & (n_\ell>2)\cdot|\lambda_{0,k}|^{m_0}|\lambda_{1,k}-\lambda_{0,k}^{m_0}|\frac{1-|\lambda_{1,k}|^{m_1}}{1-|\lambda_{1,k}|} \\ & + \tfrac{1}{n_\ell-1}|\lambda_{0,k}|^{m_0}\left[\textstyle\sum_{p=2}^{n_\ell-2}\left(\prod_{j=1}^{p-1}|\lambda_{j,k}|\right)\left|\lambda_{p,k}-\lambda_{0,k}^{m_0}\left(\prod_{j=1}^{p-1}\lambda_{j,k}^{m_j-1}\right)\right|\left(\prod_{j=1}^{p}\tfrac{1-|\lambda_{j,k}|^{m_j}}{1-|\lambda_{j,k}|}\right)\right] \\ & + \tfrac{1}{n_\ell-1}\tfrac{1-|\lambda_{n_\ell-1,k}|^{N_{n_\ell-1}-1}}{1-|\lambda_{n_\ell-1,k}|}|\lambda_{0,k}|\left(\textstyle\prod_{j=0}^{n_\ell-2}|\lambda_{j,k}|^{m_j-1}\right)\left(\prod_{j=1}^{n_\ell-2}\tfrac{1-|\lambda_{j,k}|^{m_j}}{1-|\lambda_{j,k}|}\right)|\lambda_{1,k}-\lambda_{0,k}^{m_0}| \\ & + \tfrac{1}{n_\ell-1}\tfrac{1-|\lambda_{n_\ell-1,k}|^{N_{n_\ell-1}-1}}{1-|\lambda_{n_\ell-1,k}|}|\lambda_{0,k}|^{m_0}\left(\textstyle\prod_{j=1}^{n_\ell-2}|\lambda_{j,k}|\right)\left(\sum_{p=2}^{n_\ell-1}\left|\lambda_{p,k}-\lambda_{0,k}^{m_0}\left(\prod_{j=1}^{p-1}\lambda_{j,k}^{m_j-1}\right)\right|\right)\left(\prod_{j=1}^{n_\ell-2}\tfrac{1-|\lambda_{j,k}|^{m_j}}{1-|\lambda_{j,k}|}\right), \end{aligned}$$

and approximate *maximum absolute row sum,*

$$\begin{aligned} s_{N_1-1}^{row}(k,n_\ell) \approx & |\lambda_{0,k}|^{m_0}\frac{1-|\lambda_{n_\ell-1,k}|^{N_{n_\ell-1}-1}}{1-|\lambda_{n_\ell-1,k}|} \\ & \left[\sum_{p=1}^{n_\ell-1}\left(\prod_{j=1}^{p-1}|\lambda_{j,k}|\right)\left|\lambda_{p,k}-\lambda_{0,k}^{m_0}\left(\prod_{j=1}^{p-1}\lambda_{j,k}^{m_j-1}\right)\right|\left(\prod_{j=p}^{n_\ell-2}\frac{1-|\lambda_{j,k}|^{m_j}}{1-|\lambda_{j,k}|}\right)\right]. \end{aligned}$$

Furthermore, in many cases, $\|\mathcal{E}_{FCF}^{n_\ell,\Delta}\|_2 \leq \tilde{c}_f$, because the analytic formulae approximate the right-hand-side of Equation (4.15) *(compare Equation* (4.30)*).*

For F-cycles, we propose taking the analytic formulae for a three-level F-cycle in Theorem 5 and *collapsing* the coarse-grids into level 1.

Approximation 3. *Let* $\{\Phi_\ell\}$ *be simultaneously diagonalizable by the same unitary transformation* X*, with eigenvalues* $\{\lambda_{\ell,k}\}$*,* $|\lambda_{\ell,k}| < 1$*. Then, the* approximate *convergence factor of multilevel MGRIT F-cycles with F-relaxation is given as,*

$$\tilde{c}_f \approx \sqrt{\|\mathcal{F}_F^{n_\ell=3,\Delta}\|_1 \|\mathcal{F}_F^{n_\ell=3,\Delta}\|_\infty}, \tag{4.31}$$

where the error propagation operator $\mathcal{F}_F^{n_\ell=3,\Delta}$ *is constructed with eigenvalues* $[\lambda_{0,k}, \lambda_{1,k}, \lambda_{2,k}]^T$ *(for* $k = 1, \ldots, N_x$*), number of time points* $[N_0, N_1, N_{n_\ell-1}]^T$ *and coarsening factors* $[m_0, m_1, m_2 m_3 \cdots m_{n_\ell-2}]^T$*.*

Remark 15. In the three-level case, the approximate convergence factor in Approximation 3 coincides with the analytic upper bound in Theorem 5.

4.6 Open-source implementation of analytic and numerical bounds

An MPI/C++ implementation of all derived bounds and approximate convergence factors was written and released as open-source software.[9] The code takes the (complex or real) eigenvalues of the family of Φ_ℓ as input along with a definition of the desired MGRIT algorithm (V- or F-cycles, relaxation scheme, number of levels, coarsening factors, etc.) and computes the bound or approximate convergence factor values.

Functionality was implemented, such that the user may supply the eigenvalues of a spatial operator to compute the respective eigenvalues of Φ_ℓ based on the stability function of a given Runge-Kutta time integration scheme and its Butcher tableau.

Source code level documentation is provided using Doxygen [150].

[9] Github repository: `github.com/XBraid/XBraid-convergence-est`

5 Numerical results

In this section, we evaluate the derived bounds and approximate convergence factors for various model problems that are important for, or components of, FSI problems:

- Section 5.1: Diffusion equation
- Section 5.2: Wave equation
- Section 5.3: Linear elasticity equation
- Section 5.4: Stokes and Navier-Stokes equation
- Section 5.5 and Section 5.6: Fluid-structure interaction problem

In particular, we assess how sharp the various upper bounds are and how much sharpness is sacrificed by employing a bound that is cheaper to compute numerically. Derived bounds and approximate convergence factors are compared with the maximum observed convergence factor in numerical simulations, based on the ℓ^2-norm of the residual at iteration i (see Equation (4.13)):

$$\max_i \|\boldsymbol{r}_{i+1}\|_2/\|\boldsymbol{r}_i\|_2.$$

In [25], it was noted that in the two-level setting, L-stable schemes seem to be better suited for parallel-in-time integration than A-stable schemes. In Section 5.1 and Section 5.2, this observation is reviewed in the multilevel setting. Furthermore, the difference between V- and F-cycle convergence is investigated, as well as the effect of F- and FCF-relaxation.

Section 5.3 and Section 5.4 investigate for linear PDEs, whether time-independent constraint equations (here, incompressibility constraint) are relevant to observed convergence behavior. Section 5.4 – Section 5.6 assess the new time-periodic MGRIT algorithm from Section 3.6 with regards to exploiting the time-periodic component of linear and nonlinear (coupled) PDE problems and whether consistent and significant speedup can be achieved.

5.1 Diffusion equation[1]

Consider the general time-dependent diffusion equation in two spatial dimensions $\boldsymbol{x} \in \Omega = (0, 2\pi) \times (0, 2\pi)$ with time $t \in (0, 2\pi]$,

$$\partial_t u = \nabla \cdot [\boldsymbol{K} \nabla u], \tag{5.1}$$

[1]Results in this section have previously appeared in [60, 68].

with homogeneous boundary and discontinuous initial condition (see Figure 5.1),

$$\begin{aligned} u(\boldsymbol{x},\cdot) &= 0 \qquad\qquad \text{for } \boldsymbol{x} \in \partial\Omega, \\ u(\cdot,0) &= 1 - \max\Big\{\text{sign}\Big((4-(x_1-\pi+1)^2-4(x_2-\pi)^2)^2 + 1.2(1+\pi-x_1)^3 - 10\Big), 1\Big\} \\ &\qquad\qquad \text{for } \boldsymbol{x} \in \Omega \cup \partial\Omega, \end{aligned}$$

for a scalar solution $u(\boldsymbol{x},t)$ and boundary $\partial\Omega$. Here, $\boldsymbol{K} = \text{diag}(k_1,k_2) = \text{const}$ is the grid-aligned conductivity tensor. If $k_1 = k_2$, the problem is isotropic, while if $k_1 \ll k_2$ or $k_2 \ll k_1$, the problem is anisotropic.

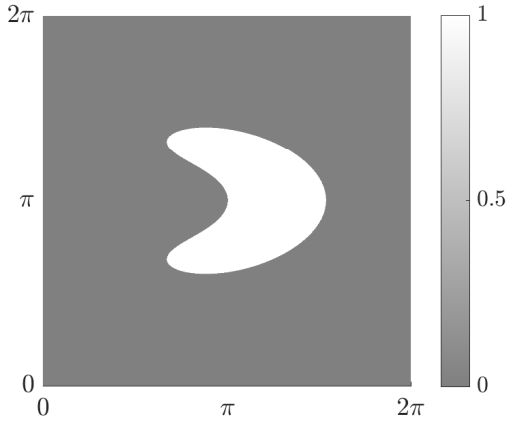


Figure 5.1: *Initial condition for 2D diffusion equation.*

The spatial problem is discretized using second-order centered finite differences and singly-diagonal implicit Runge-Kutta (SDIRK) time-integration schemes [54, 55] of orders 1-4 (Butcher tableaux provided in Appendix B) are considered. Then, the time-stepping operators $\{\Phi_\ell\}$ are unitarily diagonalizable. In [25], it was noted that in the two-level setting, L-stable schemes seem to be better suited for parallel-in-time integration than A-stable schemes. Here, we review this observation in the multilevel setting.

We further investigate the difference between V- and F-cycle convergence, as well as the effect of F- and FCF-relaxation. The number of time grids varies between two and six levels. The fine grid is composed of $N_0 = 1025$ time points and the temporal coarsening factor is $m_\ell = 2$ between all levels. The spatial domain is two-dimensional and discretized using 11 nodes in each coordinate direction (grid spacing δ_x).

All test cases are implemented in C++, using the open-source libraries Armadillo [121, 122] and XBraid [158], a non-intrusive implementation of the MGRIT algorithm. The absolute stopping tolerance for MGRIT is selected as $\|\boldsymbol{r}_i\|_2 < 10^{-11}$ and the initial global space-time guess is random.

5.1.1 Isotropic diffusion

First, we consider the isotropic case with $k_1 = k_2 = 10$. The CFL number on each level,

$$\text{CFL}_\ell = 2\pi/(N_\ell - 1)\left(k_1/\delta_x^2 + k_2/\delta_x^2\right) = 4\pi k_1/[\delta_x^2(N_\ell - 1)],$$

ranges between $\text{CFL}_0 \approx 0.376$ on level 0 and $\text{CFL}_5 \approx 12.036$ on level 5. Results for F-relaxation are shown in Figure 5.2 and FCF relaxation in Figure 5.3 (note the difference in y-axis limits).

In the case of F-relaxation, there is a considerable difference in convergence behavior between the A-stable and L-stable Runge-Kutta schemes. For A-stable schemes, convergence of MGRIT deteriorates with a growing number of time grid levels, which corresponds to a growing CFL number on the coarse grid, and eventually diverges. On the other hand, L-stable schemes show a less dramatic increase in the convergence factor. In fact, the estimated and observed convergence factors plateau for V-cycle algorithms with L-stable time integration. For F-cycle algorithms with F-relaxation and L-stable schemes, observed convergence is flat for all considered time grid hierarchies and only a slight increase can be observed in the upper bound values and approximate convergence factor.

In the case of FCF-relaxation, all observed convergence factors for SDIRK orders 2-4 are constant with respect to number of levels, and only a slight increase in convergence factor occurs for SDIRK1. FCF-relaxation was shown to be a critical ingredient for a scalable multilevel solver in [30]. An important observation for F-cycle convergence is that all upper bounds predict constant convergence factors, suggesting that an MGRIT algorithm with F-cycles and FCF-relaxation yields a robust and scalable multilevel solver for the isotropic diffusion equation.

In general, all upper bounds and approximate convergence factors provide good qualitative a priori estimates of the observed convergence. These estimates become less sharp for larger numbers of time grid levels, but the estimates do appear to be robust across changes in time integration order. Furthermore, note that Approximations 1 – 3 estimate observed convergence as well or better than more expensive upper bounds, demonstrating their applicability and efficacy. Overall, results in this section demonstrate that theoretical results presented in this work provide a valuable tool for designing robust and scalable multilevel solvers. It further provides guidance to avoid less optimal parameter choices for MGRIT, such as F-relaxation with A-stable RK schemes.

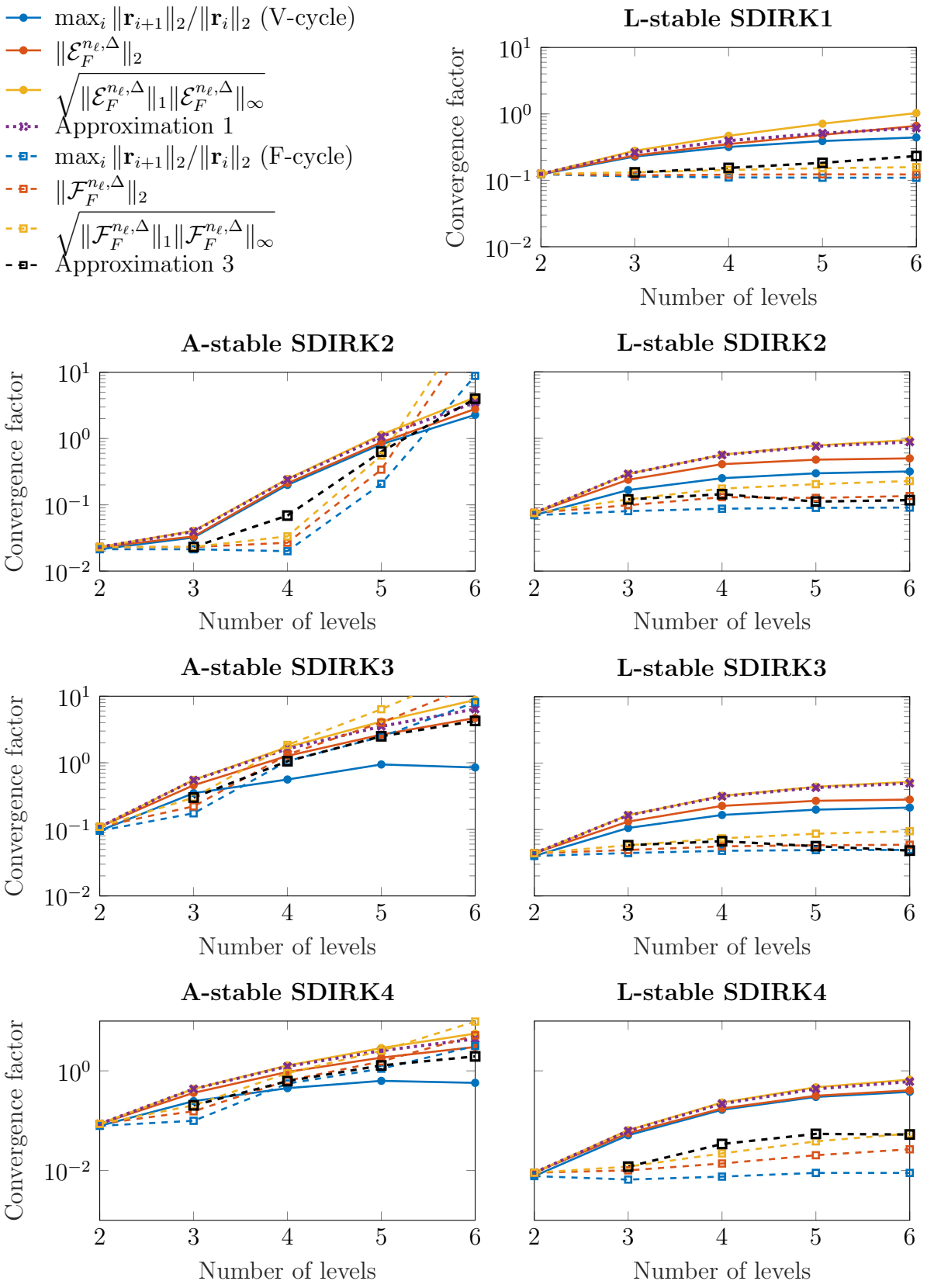


Figure 5.2: *Isotropic diffusion: Comparison of V- and F-cycle MGRIT with F-relaxation. Convergence of A-stable schemes deteriorates much quicker with a growing number of time grid levels and V-cycle MGRIT than for L-stable schemes and V-cycle MGRIT. The convergence factor for L-stable schemes and F-cycle MGRIT is almost constant.*

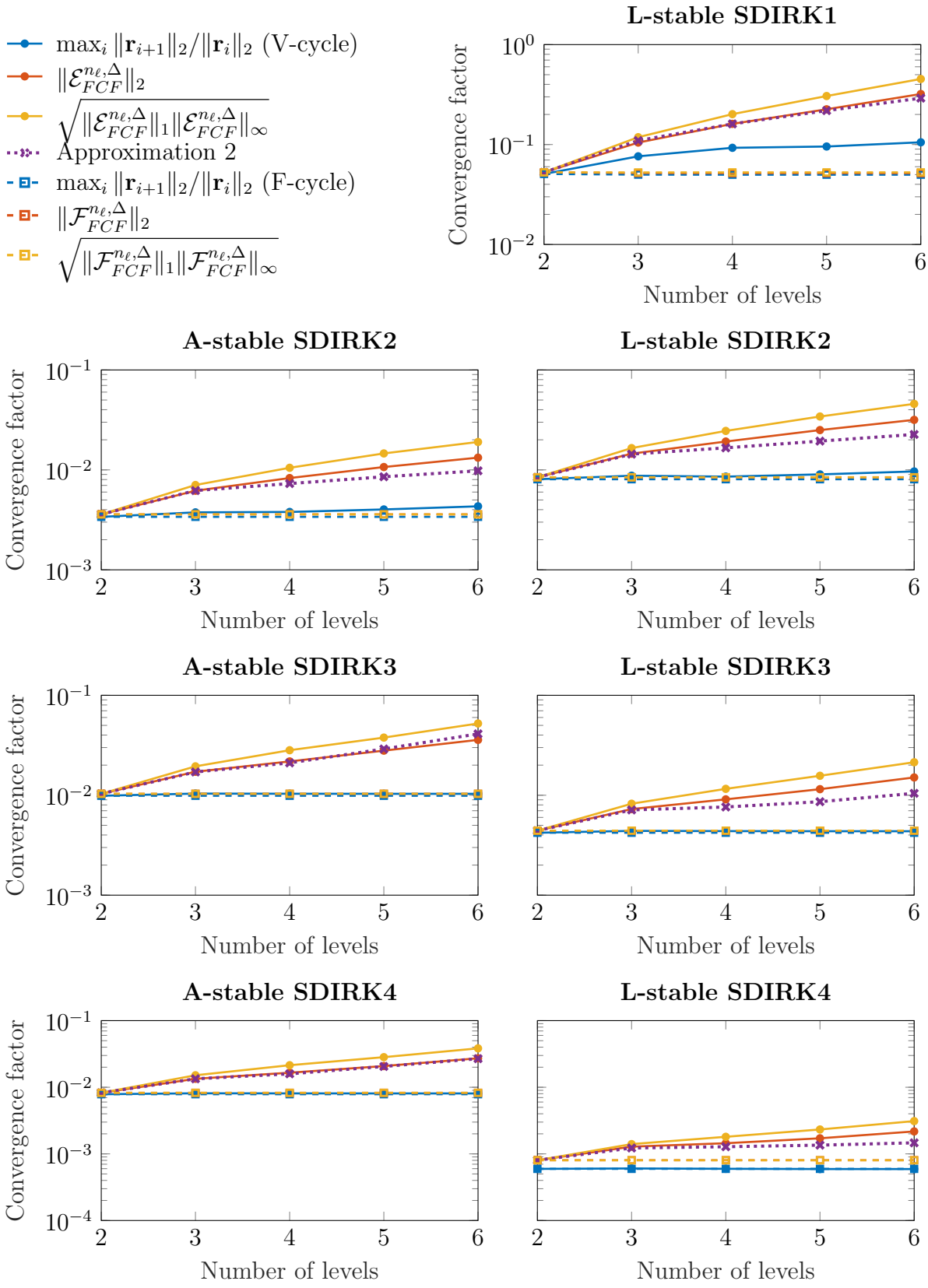


Figure 5.3: *Isotropic diffusion: Comparison of V- and F-cycle MGRIT with FCF-relaxation. Convergence of A-stable and L-stable schemes deteriorates only slightly for an MGRIT V-cycle algorithm. On the other hand, the convergence factor for an F-cycle MGRIT algorithm is constant for all considered RK schemes and cases.*

5.1.2 Anisotropic diffusion

In this section, we investigate the anisotropic diffusion case for the L-stable SDIRK1 scheme, backward Euler, to assess how sensitive the estimates are with respect to conductivity parameters. Conductivity parameters are given as $k_1 = 0.5$ and $k_2 = 0.001$, and the CFL number on each level, $\text{CFL}_\ell = 2\pi(k_1+k_2)/[\delta_x^2(N_\ell-1)]$, ranges between $\text{CFL}_0 \approx 0.009$ on level 0 and $\text{CFL}_5 \approx 0.302$ on level 5. Results are presented in Figure 5.4.

For V-cycle algorithms with F- and FCF-relaxation, the estimated and observed convergence factors grow with the number of grid levels, similar to the isotropic case. Again, FCF-relaxation yields a quicker plateauing of the observed convergence factor. On the other hand, for F-cycle algorithms with F- and FCF-relaxtion, observed and estimated convergence are effectively constant. This is indicative that convergence in the case of V-cycles is limited by solving the coarse-grid problem well, but that the non-Galerkin coarse-grid operator (that is, taking larger time steps on the coarse grid using the same integration scheme) is indeed an effective preconditioner. This behavior differs compared with using multigrid to solve anisotropic diffusion discretizations in the spatial setting, where stronger cycles such as F- and W-cycles often do not improve convergence [100].

The approximate bounds on convergence of F-cycles are fairly sharp for F- and FCF-relaxation and all numbers of levels tested. In the case of V-cycles, the bounds and approximate convergence factors loose sharpness as the number of time grid levels increases (similar to Section 5.1.1) but still provide reasonable estimates on convergence. Indeed, for V-cycles with F-relaxation, Approximation 1 is quite sharp for all tested number of levels.

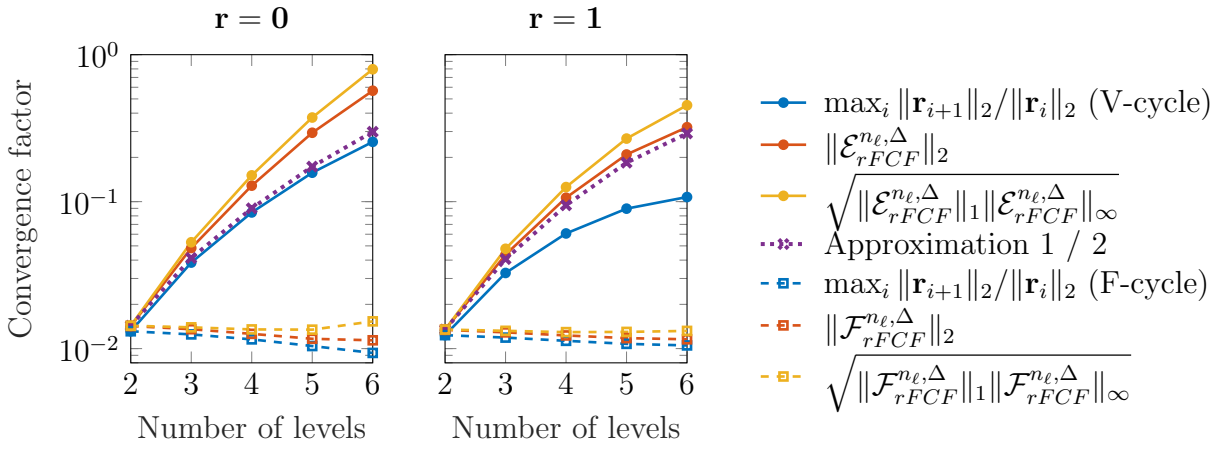


Figure 5.4: *Anisotropic diffusion: Comparison of V- and F-cycle MGRIT with F-relaxation ($r = 0$) and FCF-relaxation ($r = 1$). With a growing number of time grids, the convergence factor increases relatively quickly for V-cycle MGRIT. On the other hand, F-cycle MGRIT yields a nearly constant convergence factor, and thus, a more robust algorithm.*

5.2 Wave equation[2]

Consider the wave equation in two spatial dimensions over domain $\Omega = (0, 2\pi) \times (0, 2\pi)$,

$$\partial_{tt} u = c^2 \nabla \cdot \nabla u \qquad \text{for } \boldsymbol{x} \in \Omega, t \in (0, 2\pi], \tag{5.2}$$

with scalar solution $u(\boldsymbol{x}, t)$ and wave speed $c = \sqrt{10}$. We transform Equation (5.2) into a system of PDEs that is first-order in time,

$$\partial_t u = v, \qquad \partial_t v = c^2 \nabla \cdot \nabla u, \qquad \text{for } \boldsymbol{x} \in \Omega, t \in (0, 2\pi], \tag{5.3}$$

with initial condition (see Figure 5.5) and boundary conditions,

$$u(\cdot, 0) = \sin(x)\sin(y), \quad v(\cdot, 0) = 0, \qquad \text{for } \boldsymbol{x} \in \Omega \cup \partial\Omega, \tag{5.4}$$

$$u(\boldsymbol{x}, \cdot) = v(\boldsymbol{x}, \cdot) = 0, \qquad \text{for } \boldsymbol{x} \in \partial\Omega. \tag{5.5}$$

This problem corresponds to a 2D membrane with imposed non-zero initial displacement u and zero initial velocity v. The membrane enters an oscillatory motion pattern due to initial stresses in the material. Thus, it is a simplified representative of a hyperbolic model that shares characteristic behavior with PDEs in solid dynamics research, such as linear elasticity [66]. Similar to Section 5.1, we use second-order centered finite differences to discretize the spatial operator in Equation (5.3) and A- and L-stable Runge-Kutta schemes for time integration. The same spatial and temporal step sizes are employed and the initial space-time guess and convergence criteria are selected analogously. Here, the time stepping operators $\{\Phi_\ell\}$ are simultaneously diagonalizable and the Courant number on each level is given by $\mathcal{C}_\ell = 2c\pi/[\delta_x(N_\ell - 1)]$, ranging between $\mathcal{C}_0 \approx 0.034$ on level 0 and $\mathcal{C}_5 \approx 1.087$ on level 5.

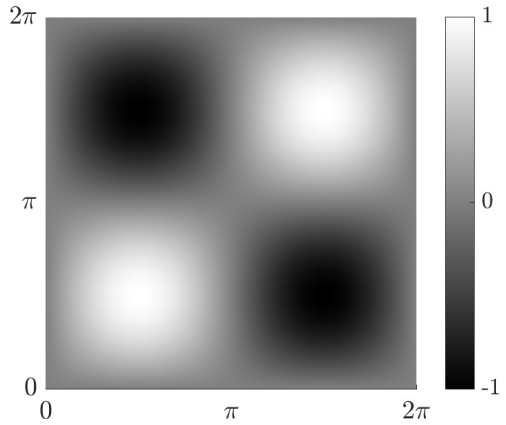


Figure 5.5: *Initial condition for 2D wave equation.*

An MGRIT V-cycle algorithm with FCF-relaxation shows quickly increasing convergence factors with a growing number of time grid levels (Figure 5.7). The worst-case convergence factors quickly exceed 1, and thus diverge, which is correctly predicted by all upper bounds and Approximation 2. Similarly, using an F-cycle algorithm results in a

[2]Results in this section have previously appeared in [60, 68].

less dramatic, but still significant increase in observed and predicted convergence factors with respect to the number of levels. For some schemes, particularly L-stable ones, an F-cycle is able to retain convergence up to the six levels in time considered here, but the bounds and approximations developed here do not predict these results.

In general, the upper bounds on convergence applied to the wave equation are significantly less sharp compared to the diffusion equation (see Section 5.1), but they are still able to accurately represent correct trends. For example, convergence factors are initially constant in most cases, then increase almost linearly with the number of levels, such as the case of L-stable SDIRK 3 in Figure 5.9. This highlights the fact that designing robust and convergent parallel-in-time algorithms for hyperbolic problems is generally perceived as difficult, and emphasizes the benefit of the presented upper bounds for F-cycle algorithms. For example, the convergence factor can be estimated a priori to select a time grid hierarchy that is likely to yield a significant speedup. In combination with performance modeling [38], such a priori estimates can provide valuable guidance.

It is noted that in the investigated cases, a similarly strong benefit of FCF-relaxation over F-relaxation as for the diffusion equation cannot be observed for the wave equation, see Figure 5.6 and Figure 5.8. However, in some cases FCF-relaxation increases the maximum number of time grid levels for which convergence can be achieved. Thus, in practice one would prefer F-relaxation over FCF-relaxation to reduce the computational cost of a given algorithm. The fact that FCF-relaxation is not sufficient to design a scalable multilevel solver for the wave equation is the major difference to the observations made for the diffusion equation.

We further note, that the observed convergence factors and upper bound values are smaller with higher time integration order, especially when L-stable SDIRK schemes are employed. For example, the theory suggests to use five-level MGRIT with F-cycles and L-stable SDIRK4 with an estimated upper bound on the convergence factor of $O(10^{-3})$. This results in a very fast algorithm for hyperbolic PDEs.

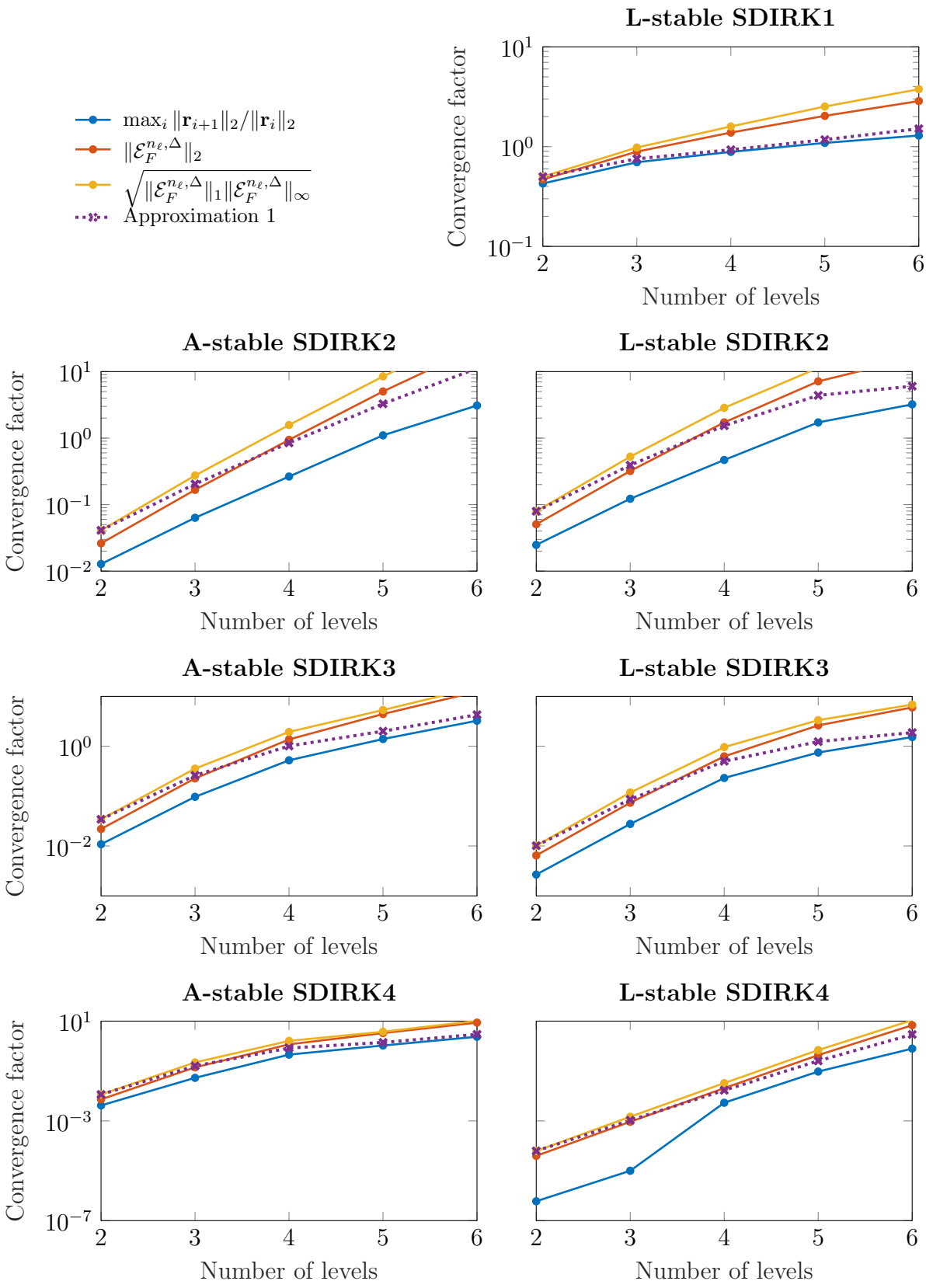


Figure 5.6: *Wave equation: Observed convergence and predicted upper bounds on convergence of MGRIT with V-cycles and F-relaxation shows very similar trends as for MGRIT with V-cycles and FCF-relaxation, see Figure 5.7. This shows that switching from F-relaxation to FCF-relaxation alone is not sufficient to yield a robust MGRIT algorithm for the wave equation (and likely, other hyperbolic PDEs).*

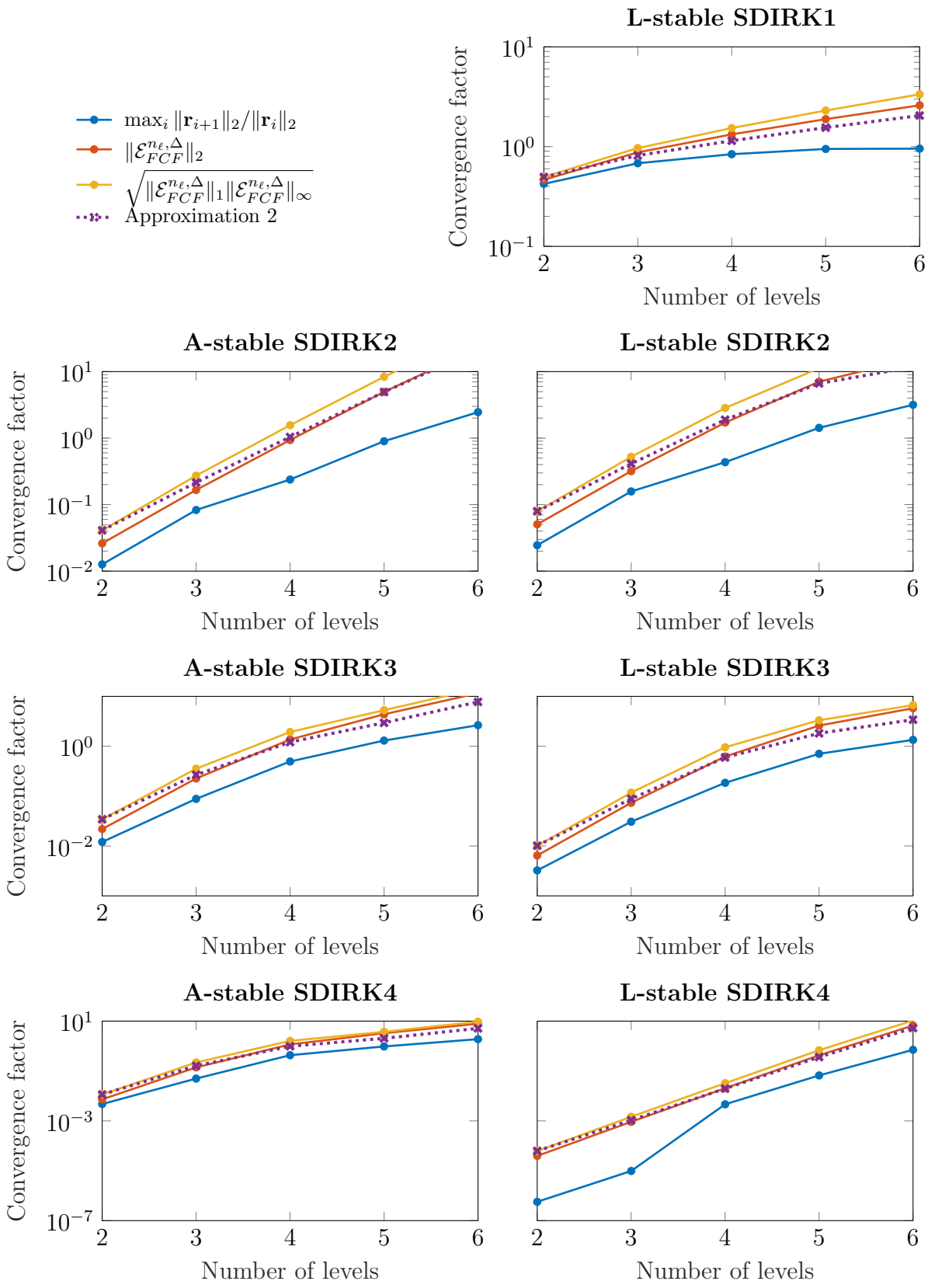


Figure 5.7: *Wave equation: The convergence factor of MGRIT with V-cycles and FCF-relaxation increases substantially with a growing number of time grid levels and eventually exceeds* 1*. This means that in the worst case, MGRIT V-cycles yields a divergent algorithm, which is in line with observations for hyperbolic PDEs in the literature [33, 34, 66].*

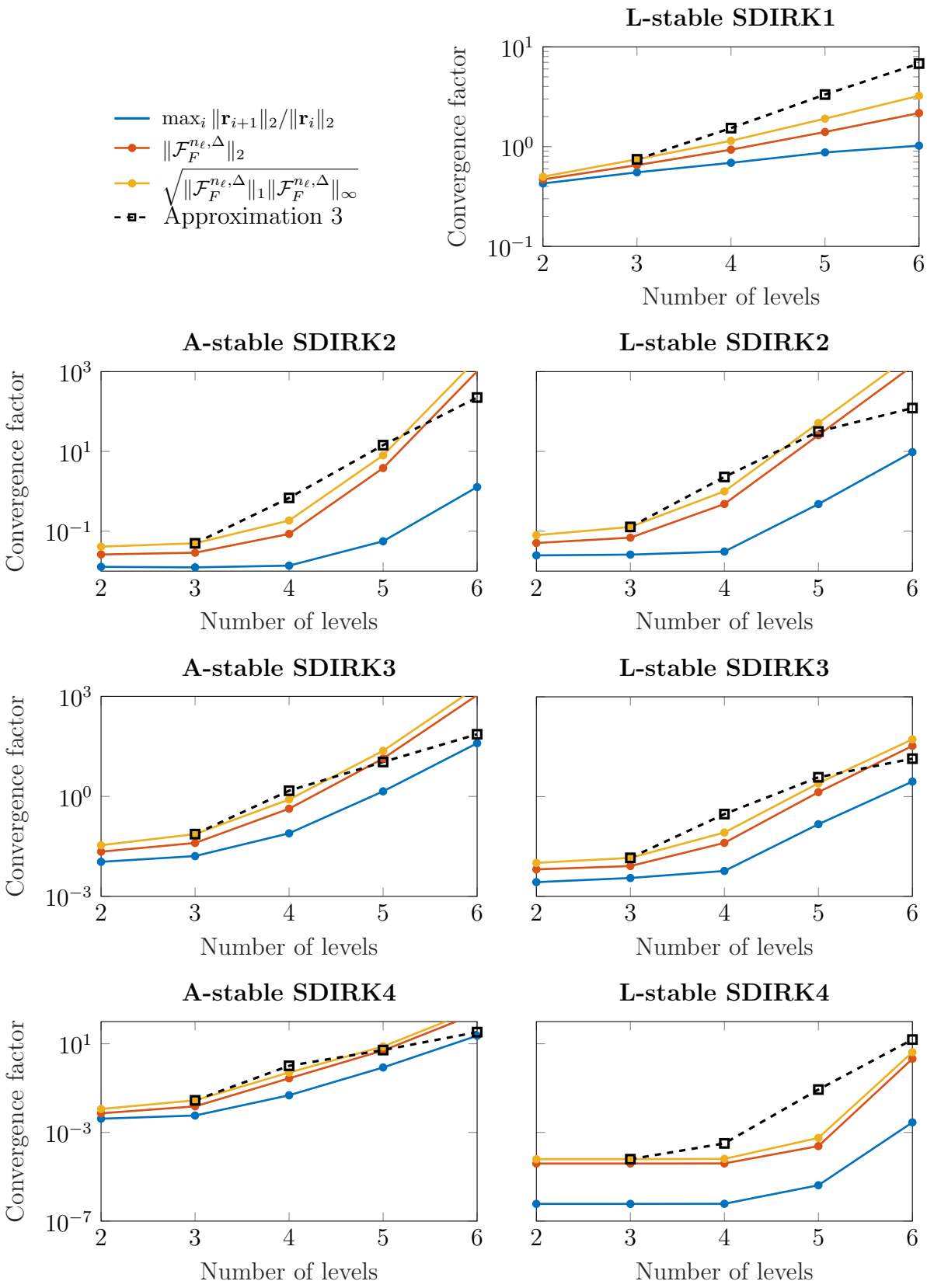


Figure 5.8: *Wave equation: Observed convergence and predicted upper bounds on convergence of MGRIT with F-cycles and F-relaxation shows very similar trends as for MGRIT with F-cycles and FCF-relaxation, see Figure 5.9. This shows that switching from F-relaxation to FCF-relaxation alone is not sufficient to yield a robust MGRIT algorithm for the wave equation (and likely, other hyperbolic PDEs).*

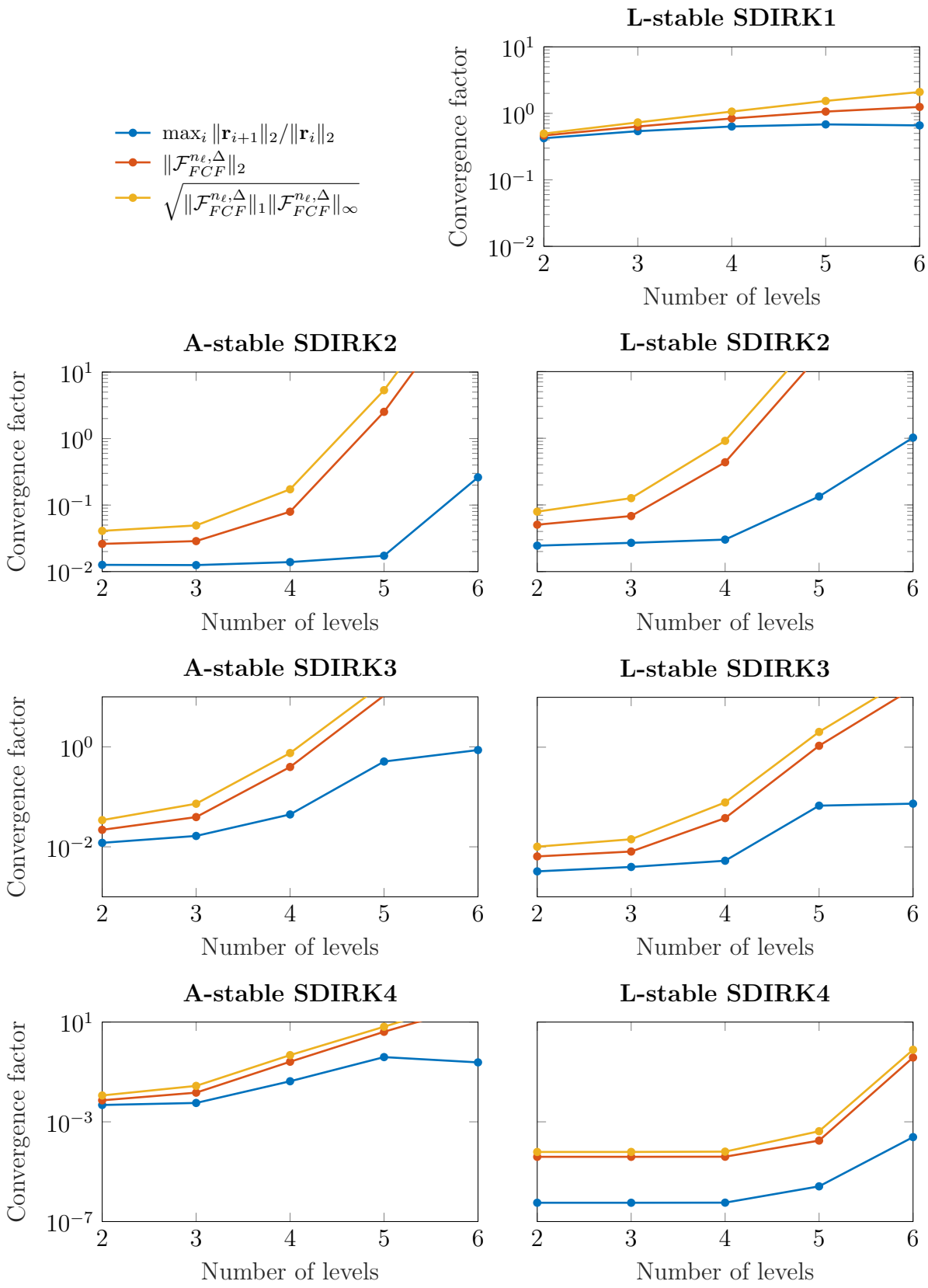


Figure 5.9: *Wave equation: The convergence of MGRIT with F-cycles and FCF-relaxation deteriorates with a larger number of time grid levels compared to MGRIT with V-cycles, see Figure 5.7. Generally, convergent algorithms are given for a larger range of time grid levels and observed convergence is better than the predictions from the upper bounds. This shows that the choice of F-cycles over V-cycles is one likely ingredient for future improvements of MGRIT for hyperbolic-type PDEs.*

5.3 Linear elasticity[3,4]

As described in literature and in the previous sections, parallel-in-time methods, such as Parareal and MGRIT, work well for parabolic PDEs and can achieve significant speedups over sequential time-stepping. Even for hyperbolic PDEs, fast convergence can be achieved, if care is taken when selecting an appropriate time-integration scheme, see Section 5.2. In general, however, literature on successfully solving hyperbolic PDEs with parallel-in-time integration methods is sparse and, depending on the application, instabilities or other complications may occur. For example, the authors of [22, 33, 34] report limitations of their parallel implicit time-integrator (PITA) for second-order solid dynamics systems, where the accuracy of PITA can be affected by the presence of natural frequencies of the structure. In the study in [33], a sufficiently small time step size was required to avoid responses of the system that were described as the *beating phenomenon*), which can lead to slow (and time step-dependent) convergence, and thus, poor performance.

In this section, the theory developed in [25] and Chapter 4 is employed to further the understanding of why poor performance is observed for solid dynamics problems. For a linear elasticity model problem, some remedies and solutions are proposed using features of MGRIT (e.g., slow temporal coarsening and FCF-relaxation) and more importantly, a different formulation of the problem that is more amenable to parallel-in-time methods. The modified scheme results in a practical algorithm that enjoys speedup benefits over the sequential algorithm.

In the following, two different backward Euler time discretization schemes are presented and investigated within the parallel-in-time context. The model problem is the incompressible second-order hyperbolic elasticity equation.

5.3.1 Model problem

Consider an incompressible linear-elastic beam in two spatial dimensions with domain $\Omega = \Omega(t) \subset \mathbb{R}^2 \times [0, T]$ and Dirichlet boundary Γ^D. The initially undeformed domain is given as $\Omega(0) = [0, 8] \times [0, 1]$, see Figure 5.11. The final time is $T = 1024$.

Further, $\boldsymbol{X} \in \Omega(0)$ and $\boldsymbol{x} \in \Omega$ denote the reference and current position of a material point and $\nabla_{\boldsymbol{X}}$ and $\nabla_{\boldsymbol{x}}$ denote the respective Lagrangian and Eulerian gradient operators. The deformation gradient is defined as,

$$\boldsymbol{F} = \nabla_{\boldsymbol{X}} \boldsymbol{x} = \nabla_{\boldsymbol{X}} \boldsymbol{u} + \boldsymbol{I}, \tag{5.6}$$

with the displacement of a material point with respect to its position in the reference configuration,

$$\boldsymbol{u} = \boldsymbol{u}(\boldsymbol{x}, t) = \boldsymbol{x} - \boldsymbol{X}. \tag{5.7}$$

The partial derivative operator with respect to time is denoted as ∂_t and $\partial_{tt} = \partial_t \partial_t$.

[3] Results in this section have previously appeared in [66].

[4] The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP/2007–2013)/ERC Grant Agreement No. 306757 (LEAD) and from the Engineering and Physical Sciences Research Council (EP/N011554/1 and EP/R003866/1).

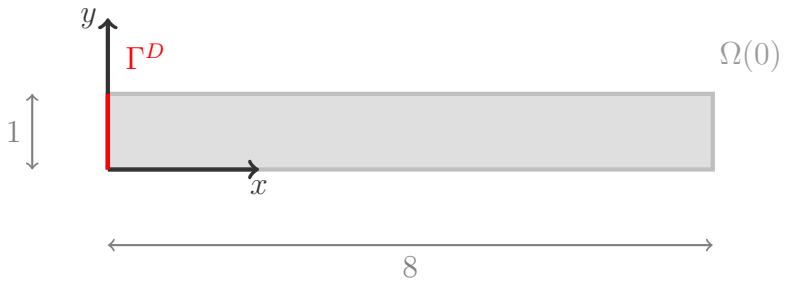


Figure 5.10: *Undeformed domain $\Omega(0)$ and Dirichlet boundary $\Gamma^D = \Omega|_{x=0}$ for linear-elastic beam problem in two dimensions. Zero displacement boundary conditions are prescribed at $x = 0$.*

The governing equations for the dynamic and linear-elastic response of an incompressible solid structure with given initial data and Dirichlet boundary condition data are given as,

$$\rho\partial_{tt}\boldsymbol{u} - \nabla_{\boldsymbol{x}} \cdot \boldsymbol{\sigma} = \boldsymbol{0} \qquad \text{in } \Omega, \tag{5.8}$$

$$\nabla_{\boldsymbol{x}} \cdot \partial_t \boldsymbol{u} = 0 \qquad \text{in } \Omega, \tag{5.9}$$

$$\boldsymbol{u}(\cdot, t) = \boldsymbol{0} \qquad \text{on } \Gamma^D, \tag{5.10}$$

$$\boldsymbol{u}(\cdot, 0) = \boldsymbol{0}, \quad \partial_t \boldsymbol{u}(\cdot, 0) = \boldsymbol{v}(\cdot, 0) = \hat{\boldsymbol{v}}_0 \qquad \text{in } \Omega(0), \tag{5.11}$$

with density ρ, Cauchy stress tensor $\boldsymbol{\sigma}(\boldsymbol{u}, p) = \mu(\boldsymbol{F} - \boldsymbol{I}) - p\boldsymbol{I}$, material stiffness parameter μ, the hydrostatic pressure variable p and initial velocity data $\hat{\boldsymbol{v}}_0$. Here, the initial velocity is given as,

$$\hat{\boldsymbol{v}}_0 = \begin{bmatrix} -x^2/640 \\ x^2(8-x)/640 \end{bmatrix}, \tag{5.12}$$

see Figure 5.11. Equation (5.8) can be transformed to a system of first-order equations,

$$\partial_t \boldsymbol{u} = \boldsymbol{v} \qquad \text{in } \Omega, \tag{5.13}$$

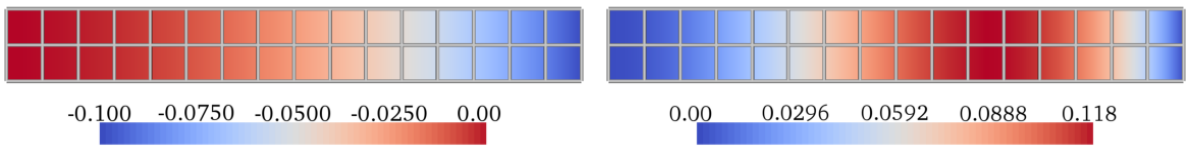
$$\rho\partial_t \boldsymbol{v} = \nabla_{\boldsymbol{x}} \cdot \boldsymbol{\sigma} \qquad \text{in } \Omega, \tag{5.14}$$

$$\nabla_{\boldsymbol{x}} \cdot \boldsymbol{v} = 0 \qquad \text{in } \Omega, \tag{5.15}$$

$$\boldsymbol{u}(\cdot, t) = \boldsymbol{0} \qquad \text{on } \Gamma^D, \tag{5.16}$$

$$\boldsymbol{u}(\cdot, 0) = \boldsymbol{0}, \quad \boldsymbol{v}(\cdot, 0) = \hat{\boldsymbol{v}}_0 \qquad \text{in } \Omega(0), \tag{5.17}$$

with velocity $\boldsymbol{v}$. For simplicity, the material parameters are selected as $\mu = \rho = 1$.



(a) Initial velocity in x-direction. (b) Initial velocity in y-direction.

Figure 5.11: Discretization using 16 x 2 quadrilateral elements and initial velocity.

Dynamic response of the linear-elastic beam

The initial velocity distribution over the cantilever beam length (see Figure 5.11) causes the free end to first deflect in the negative x- and positive y-directions. The elastic stresses cause the beam to decelerate and move back downward, passing its initial position and deflecting in negative y-direction. The beam deformation then follows an up-and-down deflection pattern.

5.3.2 Discrete temporal domain

The temporal domain $[0, T]$ is decomposed into N_0 equidistant time points, such that,

$$t_n = n \cdot \delta_0, \quad \text{with } n = 0, \dots, N_0 - 1, \tag{5.18}$$

with time step size $\delta_0 = T/(N_0 - 1)$, initial time $t_0 = 0$ and final time $t_{N_0-1} = T$. Then, velocity, pressure and displacement at time point t_n are denoted as $[\boldsymbol{v}_n, p_n, \boldsymbol{u}_n]^T = [\boldsymbol{v}(\cdot, t_n), p(\cdot, t_n), \boldsymbol{u}(\cdot, t_n)]^T$.

In the following, two different time discretization schemes (referred to as Scheme I and Scheme II) are introduced and investigated. Scheme I is considered as the default scheme in the application code CHeart [94] and is motivated by better conserving the energy in the system for large time step sizes (see Section 5.3.6.1; for more details, see [130, 131, 157]). On the other hand, Scheme II is proposed as an improvement for parallel-in-time methods with the capability of predicting amplitudes of oscillation with comparable quality for practical time step sizes.

5.3.3 Scheme I

To reduce the complexity of the computational model, Equation (5.13) is eliminated from the system of equations in the following by including it implicitly. That is, we solve for velocity $\boldsymbol{v}$ and hydrostatic pressure p and update the displacement variable $\boldsymbol{u}$ based on the solution for the velocity variable. Further, we note that all quantities are computed on the reference domain $\Omega(0)$. That is, linear-elastic response is assumed and higher-order effects of the deforming domain are neglected.

We approximate the partial derivative operator ∂_t in Equation (5.13) using the midpoint rule,

$$\boldsymbol{u}_n = \boldsymbol{u}_{n-1} + \delta_0 \frac{\boldsymbol{v}_n + \boldsymbol{v}_{n-1}}{2} \qquad \text{in } \Omega_0. \tag{5.19}$$

The partial derivative operator in Equation (5.14) is discretized using the backward Euler scheme where we substitute with Equation (5.19). Thus, we search $[\boldsymbol{v}_n, p_n]^T$ for all $n = 1, \dots, N_0 - 1$, such that,

$$\rho \boldsymbol{v}_n - \frac{\mu \delta_0^2}{2} \nabla_{\boldsymbol{X}}^2 \boldsymbol{v}_n + \delta_0 \nabla_{\boldsymbol{X}} p_n = \rho \boldsymbol{v}_{n-1} + \frac{\mu \delta_0^2}{2} \nabla_{\boldsymbol{X}}^2 \boldsymbol{v}_{n-1} + \mu \delta_0 \nabla_{\boldsymbol{X}}^2 \boldsymbol{u}_{n-1} \quad \text{in } \Omega_0, \tag{5.20}$$

$$\nabla_{\boldsymbol{X}} \cdot \boldsymbol{v}_n = -\nabla_{\boldsymbol{X}} \cdot \boldsymbol{v}_{n-1} \qquad \text{in } \Omega_0, \tag{5.21}$$

and compute the displacement $\boldsymbol{u}_n$ according to Equation (5.19) after each solve.

The domain Ω_n is discretized using 16 x 2 quadrilateral elements, Ω_n^h, with mesh size $\delta_x = \delta_y = 0.5$, see Figure 5.11. Finite element discretizations are constructed using inf-sup stable $\mathbb{Q}^2 - \mathbb{Q}^1$ Taylor-Hood elements for velocity and pressure, $\boldsymbol{v}_n^h$ and p_n^h, and $\mathbb{Q}^2$ elements for displacement, $\boldsymbol{u}_n^h$. The superscript h denotes the space-discretized version of the domain and state variables. As we do not consider spatial refinement or coarsening, we omit the superscript h for the remainder of Section 5.3.

Equation (5.20) and Equation (5.21) are then discretized in space, which leads to the following problem: find the space-time discrete solution vector $[\boldsymbol{v}_n, \boldsymbol{p}_n]^T$, such that for each $n = 1, \ldots, N_0 - 1$,

$$\left[\rho \boldsymbol{M} + \frac{\mu\delta_0^2}{2}\boldsymbol{K}\right]\boldsymbol{v}_n + \delta_0 \boldsymbol{B}^T \boldsymbol{p}_n = \left[\rho \boldsymbol{M} - \frac{\mu\delta_0^2}{2}\boldsymbol{K}\right]\boldsymbol{v}_{n-1} - \mu\delta_0 \boldsymbol{K}\boldsymbol{u}_{n-1} \quad \text{in } \Omega_0, \tag{5.22}$$

$$\boldsymbol{B}\boldsymbol{v}_n = -\boldsymbol{B}\boldsymbol{v}_{n-1} \quad \text{in } \Omega_0, \tag{5.23}$$

where $\boldsymbol{M}$ is the mass matrix and $\boldsymbol{K}$ and $\boldsymbol{B}$ refer to the discretized weak form Laplacian and divergence operators $\nabla^2_{\boldsymbol{X}}()$ and $\nabla_{\boldsymbol{X}} \cdot ()$, respectively. Note, after solving for a given $[\boldsymbol{v}_n, \boldsymbol{p}_n]^T$, we can update the displacement $\boldsymbol{u}_n$ from Equation (5.19). Writing Equation (5.22) and Equation (5.23) in matrix form and including the update given in Equation (5.19) yields the following linear system,

$$\begin{bmatrix} \rho\boldsymbol{M} + \frac{\mu\delta_0^2}{2}\boldsymbol{K} & \delta_0\boldsymbol{B}^T & \boldsymbol{0} \\ \boldsymbol{B} & \boldsymbol{0} & \boldsymbol{0} \\ -\frac{\delta_0}{2}\boldsymbol{I} & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_n \\ \boldsymbol{p}_n \\ \boldsymbol{u}_n \end{bmatrix} = \begin{bmatrix} \rho\boldsymbol{M} - \frac{\mu\delta_0^2}{2}\boldsymbol{K} & \boldsymbol{0} & -\mu\delta_0\boldsymbol{K} \\ -\boldsymbol{B} & \boldsymbol{0} & \boldsymbol{0} \\ \frac{\delta_0}{2}\boldsymbol{I} & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_{n-1} \\ \boldsymbol{p}_{n-1} \\ \boldsymbol{u}_{n-1} \end{bmatrix}. \tag{5.24}$$

Denoting the linear operators on the left and right hand sides by,

$$\boldsymbol{D}^I := \begin{bmatrix} \rho\boldsymbol{M} + \frac{\mu\delta_0^2}{2}\boldsymbol{K} & \delta_0\boldsymbol{B}^T & \boldsymbol{0} \\ \boldsymbol{B} & \boldsymbol{0} & \boldsymbol{0} \\ -\frac{\delta_0}{2}\boldsymbol{I} & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}, \quad \boldsymbol{C}^I := \begin{bmatrix} \rho\boldsymbol{M} - \frac{\mu\delta_0^2}{2}\boldsymbol{K} & \boldsymbol{0} & -\mu\delta_0\boldsymbol{K} \\ -\boldsymbol{B} & \boldsymbol{0} & \boldsymbol{0} \\ \frac{\delta_0}{2}\boldsymbol{I} & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}, \tag{5.25}$$

with $\boldsymbol{\Phi}^I := [\boldsymbol{D}^I]^{-1}\boldsymbol{C}^I$ and $\boldsymbol{s}_0^I = \hat{\boldsymbol{s}}_0 := [\hat{\boldsymbol{v}}_0, \boldsymbol{0}, \boldsymbol{0}]^T$, the state variables $\boldsymbol{s}_n^I := [\boldsymbol{v}_n, \boldsymbol{p}_n, \boldsymbol{u}_n]^T$ can be computed by the following equation,

$$\boldsymbol{s}_n^I = \boldsymbol{\Phi}^I \boldsymbol{s}_{n-1}^I \qquad \text{for } n = 1, \ldots, N_0 - 1. \tag{5.26}$$

Note, that for the considered model (linear-elastic, incompressible) and for fixed spatial resolution, the operator $\boldsymbol{\Phi}^I$ is only dependent on time step size δ_0. That is, the operator $\boldsymbol{\Phi}^I$ only needs to be computed once per time step size.

It is further noted that $\boldsymbol{\Phi}^I$ has the sparsity pattern,

$$\begin{bmatrix} [\boldsymbol{\Phi}^I]_{11} & [\boldsymbol{\Phi}^I]_{12} & [\boldsymbol{\Phi}^I]_{13} \\ [\boldsymbol{\Phi}^I]_{21} & [\boldsymbol{\Phi}^I]_{22} & [\boldsymbol{\Phi}^I]_{23} \\ [\boldsymbol{\Phi}^I]_{31} & [\boldsymbol{\Phi}^I]_{32} & [\boldsymbol{\Phi}^I]_{33} \end{bmatrix} = \begin{bmatrix} [\boldsymbol{\Phi}^I]_{11} & \boldsymbol{0} & [\boldsymbol{\Phi}^I]_{13} \\ [\boldsymbol{\Phi}^I]_{21} & \boldsymbol{0} & [\boldsymbol{\Phi}^I]_{23} \\ [\boldsymbol{\Phi}^I]_{31} & \boldsymbol{0} & [\boldsymbol{\Phi}^I]_{33} \end{bmatrix}, \tag{5.27}$$

which highlights that the current state vector $\boldsymbol{s}_i^I$ is not dependent on the previous pressure value (i.e. the pressure variable and its associated rows and columns in $\boldsymbol{\Phi}^I$ can be ignored without affecting $\boldsymbol{u}$ or $\boldsymbol{v}$.) Thus, we eliminate the rows and columns related to the pressure variable and proceed with the analysis by simultaneously diagonalizing,

$$\boldsymbol{U}^{-1} \begin{bmatrix} [\boldsymbol{\Phi}_\ell^I]_{11} & [\boldsymbol{\Phi}_\ell^I]_{13} \\ [\boldsymbol{\Phi}_\ell^I]_{31} & [\boldsymbol{\Phi}_\ell^I]_{33} \end{bmatrix} \boldsymbol{U} = \text{diag}(\lambda_{\ell,1}, \lambda_{\ell,2}, \ldots, \lambda_{\ell,N_x}) \qquad \text{for } \ell = 0, \ldots, n_{\ell-1}. \tag{5.28}$$

5.3.4 Multigrid-reduction-in-time (MGRIT)

In this section, the following two-level and multilevel variants of MGRIT (see Chapter 3) are investigated for the linear-elastic beam problem:

- Two-level MGRIT with rFCF-relaxation for $r \in \{0, 1, 2\}$ and temporal coarsening factor $m_\ell \in \{2, 4, 8, 16, 32\}$.
- Multilevel MGRIT FMG-cycles with $n_l \in \{2, 3, 4, 5, 6\}$ levels, temporal coarsening factor $m_\ell = 2$ between all levels and $\{1, 2, 3, 4\}$ V-cycles per FMG-level.

Fine grid time step sizes of $\delta_0 \in \{1, 0.1, 0.01, 0.001\}$ are considered, resulting in $N_0 \in \{1025, 10241, 102401, 1024001\}$ time steps on level 0. The initial condition (see Equation (5.11) and Equation (5.12)) is used as initial guess at all time steps for MGRIT.

The stopping criterion on the residual norm is selected as $\|\boldsymbol{r}\|_{\mathfrak{M}} \leq 5{\cdot}10^{-9}/\sqrt{\delta_x\delta_y\delta_0}$, with the maximum number of MGRIT iterations set to 60 iterations. The standard Euclidean norm of the MGRIT residual $\boldsymbol{r}$ on level 1 is given as,

$$\begin{aligned} \|\boldsymbol{r}\|_{2,w}^2 &= w_0\|\boldsymbol{r}^{\boldsymbol{v}}\|_2^2 + w_1\|\boldsymbol{r}^p\|_2^2 + w_2\|\boldsymbol{r}^{\boldsymbol{u}}\|_2^2 \\ &= w_0(\boldsymbol{r}^{\boldsymbol{v}})^T\boldsymbol{r}^{\boldsymbol{v}} + w_1(\boldsymbol{r}^p)^T\boldsymbol{r}^p + w_2(\boldsymbol{r}^{\boldsymbol{u}})^T\boldsymbol{r}^{\boldsymbol{u}} \end{aligned} \tag{5.29}$$

with weights w_0, w_1 and w_3 equal to 1, where $\boldsymbol{r}^{\boldsymbol{v}}$, $\boldsymbol{r}^p$ and $\boldsymbol{r}^{\boldsymbol{u}}$ denote components in the residual vector corresponding to velocity, pressure and displacement contributions. Here, the weights are modified to approximately measure residual reduction in the mass matrix norm,

$$\|\boldsymbol{r}\|_{\mathfrak{M}}^2 \approx (\boldsymbol{r}^{\boldsymbol{v}})^T\boldsymbol{r}^{\boldsymbol{v}} + \frac{1}{m_0}(\boldsymbol{r}^{\boldsymbol{u}})^T\boldsymbol{r}^{\boldsymbol{u}} \tag{5.30}$$

to measure residual reduction in the following numerical experiments as close to the two-level theory (see [25]) as possible, however, without the additional expense of computing the exact mass matrix norm. It is important to note that this approach only changes how the solution progress (i.e. reduction of residual norm) is measured. But it does neither affect the coarse-grid update nor change the numerical solution.

For the numerical experiments, the finite element software tool CHeart [94] is employed. In CHeart, Scheme I was available as the default scheme for linear elasticity. Wrapper routines were written to incorporate the MGRIT algorithm into CHeart using the open-source library XBraid [158], a nonintrusive implementation of the MGRIT algorithm. Separate MPI groups and communicators were introduced in space and time to maintain the capability of CHeart to parallelize in the spatial domain by using domain decomposition methods while enabling independent parallelization in the temporal domain. That is, one can parallelize in space, time or in space-time.

Here, the XBraid option to skip work on the first down-cycle is used. Note, due to the linearity of the problem, the operator $\boldsymbol{\Phi}_\ell^I$ is only computed once for each time step size (i.e. time grid level). This significantly reduces computational work compared to recomputing the operator for each time step.

5.3.5 Two-level MGRIT and Scheme I

If Scheme I is employed as a one-step integrator with two-level MGRIT, divergence is observe in the numerical experiments for all considered test cases.

To qualitatively investigate what leads to the divergence of MGRIT in conjunction with Scheme I, we track the current position of the tip of the cantilever beam (initial coordinate $[8, 0.5]^T$ at $t = 0$) over time with the time horizon $t \in [0, 64]$. The fine grid time step size is $\delta_0 = 1$ with coarsening factor $m_0 = 2$. FCF-relaxation is employed. Figure 5.12 shows the current approximation of the tip's displacement with respect to the initial position over time for a number of algorithmic steps, for example, after FCF-relaxation, after restriction, after the coarse-grid solve, *etc.* (where we extract the values on return from applying $\boldsymbol{\Phi}_\ell^I$). The data in Figure 5.12 highlight how the current approximation of the cantilever's tip first improves. Though, already during the first MGRIT iteration an instability is introduced by the coarse-grid update which is then amplified in subsequent steps. The observed phenomena are neither physical nor part of the mathematical model, however, they are in line with observations in the literature, see [33, 34]. Previously this has been a limiting factor for parallel-in-time integration and the dynamic elasticity equation.

On the other hand, the divergence of the numerical algorithm is reflected by the very large theoretical convergence bounds, i.e. $c_f^F, c_f^{FCF} >> 1$, confirming experimental observations.

5.3.6 Scheme II

The instability of MGRIT with Scheme I motivates developing a modified scheme that does not suffer from the same instability. For example, a slightly different scheme can be obtained by approximating the partial derivative operator in Equation (5.13) as,

$$\boldsymbol{u}_n = \boldsymbol{u}_{n-1} + \delta_0 \boldsymbol{v}_n \qquad \text{in } \Omega_0. \tag{5.31}$$

Thus, in Scheme II we search the time-discrete $(\boldsymbol{v}_n, p_n)$ for all $n = 1, \ldots, N_0 - 1$, such that,

$$\rho \boldsymbol{v}_n - \mu \delta_0^2 \nabla_{\boldsymbol{X}}^2 \boldsymbol{v}_n + \delta_0 \nabla_{\boldsymbol{X}} p_n = \rho \boldsymbol{v}_{n-1} + \mu \delta_0 \nabla_{\boldsymbol{X}}^2 \boldsymbol{u}_{n-1} \qquad \text{in } \Omega_0, \tag{5.32}$$

$$\nabla_{\boldsymbol{X}} \cdot \boldsymbol{v}_n = 0 \qquad \text{in } \Omega_0, \tag{5.33}$$

Similar to Section 5.3.3, Scheme II can be written as: for each $n = 1, \ldots, N_0 - 1$, we seek the space-time discrete $[\boldsymbol{v}_n, \boldsymbol{p}_n]^T$, such that,

$$\left[\rho \boldsymbol{M} + \mu \delta_0^2 \boldsymbol{K}\right] \boldsymbol{v}_n + \delta_0 \boldsymbol{B}^T \boldsymbol{p}_n = \rho \boldsymbol{M} \boldsymbol{v}_{n-1} - \mu \delta_0 \boldsymbol{K} \boldsymbol{u}_{n-1} \qquad \text{on } \Omega_0, \tag{5.34}$$

$$\boldsymbol{B} \boldsymbol{v}_n = 0 \qquad \text{on } \Omega_0, \tag{5.35}$$

and update the displacement $\boldsymbol{u}_n$ from Equation (5.31) after each solve. In matrix notation, we can write,

$$\begin{bmatrix} \rho \boldsymbol{M} + \mu \delta_0^2 \boldsymbol{K} & \delta_0 \boldsymbol{B}^T & \boldsymbol{0} \\ \boldsymbol{B} & \boldsymbol{0} & \boldsymbol{0} \\ -\delta_0 \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_n \\ \boldsymbol{p}_n \\ \boldsymbol{u}_n \end{bmatrix} = \begin{bmatrix} \rho \boldsymbol{M} & \boldsymbol{0} & -\mu \delta_0 \boldsymbol{K} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_{n-1} \\ \boldsymbol{p}_{n-1} \\ \boldsymbol{u}_{n-1} \end{bmatrix}. \tag{5.36}$$

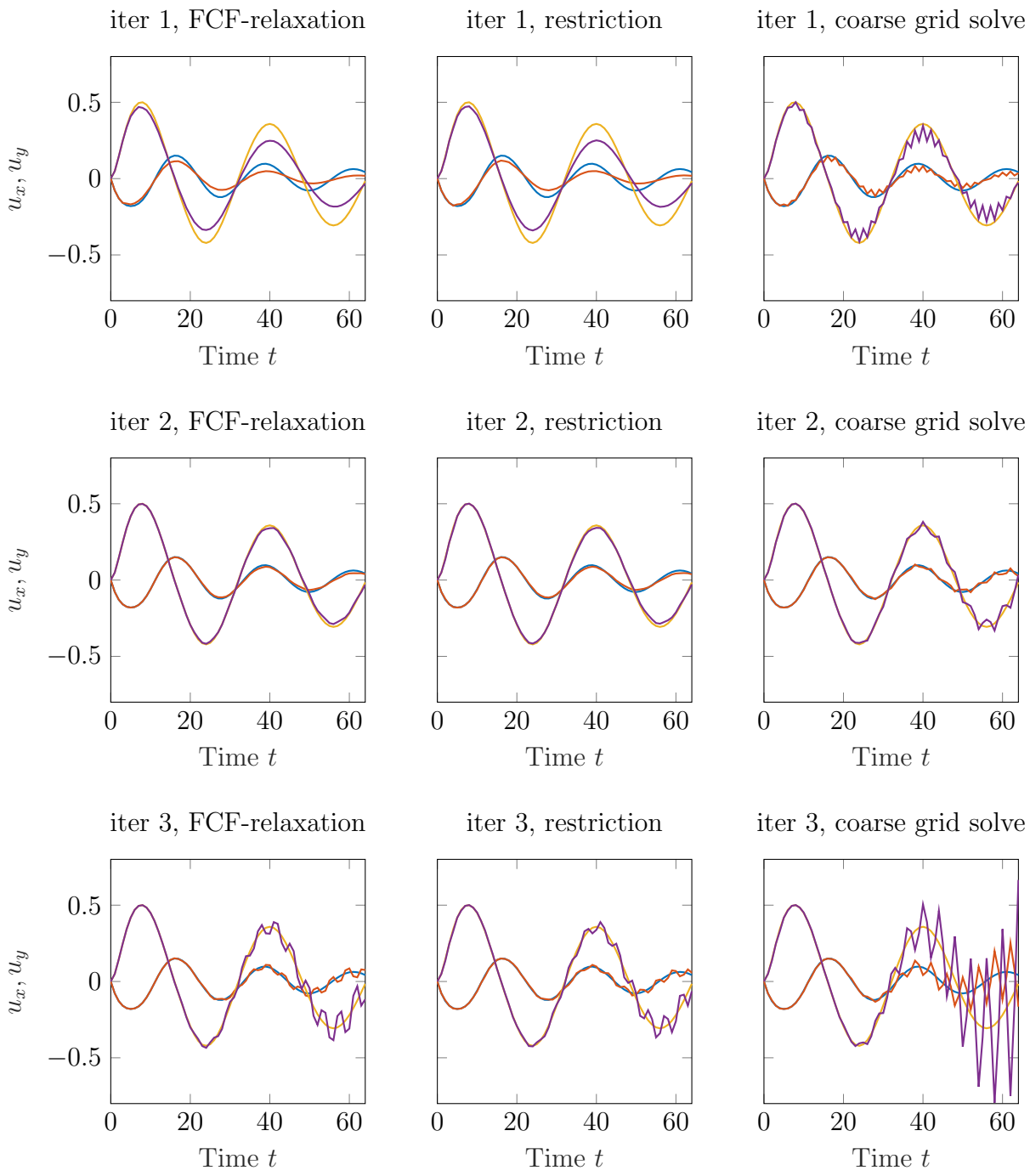


Figure 5.12: *Current approximation of the tip displacement for two-grid MGRIT using Scheme I with temporal coarsening factor $m_0 = 2$, time step size $\delta_0 = 1$, temporal domain $t \in [0, 64]$ and FCF-relaxation compared with a reference solution from sequential time stepping. Note, how the coarse grid update introduces an instability which is amplified in subsequent steps.*

With

$$\boldsymbol{D}^{II} := \begin{bmatrix} \rho\boldsymbol{M} + \mu\delta_0^2\boldsymbol{K} & \delta_0\boldsymbol{B}^T & \boldsymbol{0} \\ \boldsymbol{B} & \boldsymbol{0} & \boldsymbol{0} \\ -\delta_0\boldsymbol{I} & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \quad \text{and} \quad \boldsymbol{C}^{II} := \begin{bmatrix} \rho\boldsymbol{M} & \boldsymbol{0} & -\mu\delta_0\boldsymbol{K} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}, \tag{5.37}$$

Equation (5.36) can be written as,

$$\boldsymbol{s}_n^{II} = \boldsymbol{\Phi}^{II}\boldsymbol{s}_{n-1}^{II} \qquad \text{for } n = 1, \ldots, N_0 - 1, \tag{5.38}$$

where $\boldsymbol{\Phi}^{II} := [\boldsymbol{D}^{II}]^{-1}\boldsymbol{C}^{II}$ and $\boldsymbol{s}_n^{II} := [\boldsymbol{v}_n, \boldsymbol{p}_n, \boldsymbol{u}_n]^T$ with $\boldsymbol{s}_0^{II} = \hat{\boldsymbol{s}}_0 := [\hat{\boldsymbol{v}}_0, 0, \boldsymbol{0}]^T$. Again, the operator $\boldsymbol{\Phi}^{II}$ only depends on time step size δ_0. Similar to Equation (5.28), the convergence analysis is performed on the reduced system without the pressure contributions.

5.3.6.1 Effect of time step size on amplitude of oscillation

Because selecting a different time integration scheme can affect the accuracy of the numerical solution, the sequential time stepping solution obtained with Scheme II is first compared with the sequential time stepping solution obtained with Scheme I, before studying Scheme II when using MGRIT.

Backward Euler time integration schemes introduce artificial numerical diffusion. Its effect on the sum of kinematic and potential energy of the system over time depends on the time step size, where we expect energy conservation in the asymptotic limit $\delta_0 \to 0$. The artificial damping of the system causes the amplitude of oscillation to become smaller over time, see Figure 5.13. The backward Euler time integration scheme exhibits quick damping for $\delta_0 = 1$ irrespectively of the chosen scheme (Scheme I and Scheme II). The effect of numerical damping becomes smaller for smaller δ_0, where we note that both schemes reproduce the amplitudes of the beam oscillation with comparable quality for $\delta_0 = 0.001$. Further, considering a second-order symplectic Verlet scheme (which maintains the amplitude of oscillation) with $\delta_0 = 0.001$ as reference, we observe how both schemes tend toward the same solution with $O(\delta_0)$, see Figure 5.14, such that the mismatch between both schemes becomes negligible. This is expected to be true for time step sizes in practical applications.

It is important to note that MGRIT converges to the same solution (within the selected solver tolerance) as sequential time stepping on the fine grid. Thus, the converged numerical solution obtained with MGRIT suffers from the same amount of numerical damping as the numerical solution from sequential time stepping on the fine grid.

5.3.7 Two-level MGRIT and Scheme II

Using Scheme II as the one-step integrator (that is, $\boldsymbol{\Phi}_\ell^{II}$) in a two-level algorithm, we observe worst-case convergence factors[5] of smaller than 1 (i.e. residual norm is decreased for all iterations) for a range of different coarsening factors m_0 and for all considered time step sizes, see Figure 5.15.

We note that for $\delta_0 = 1$, both experimental and predicted convergence factors are in excellent agreement and that the predicted values of c_f^F and c_f^{FCF} are a sharp upper bound

[5]Unless noted otherwise, reported experimental convergence factors are the global maximum values.

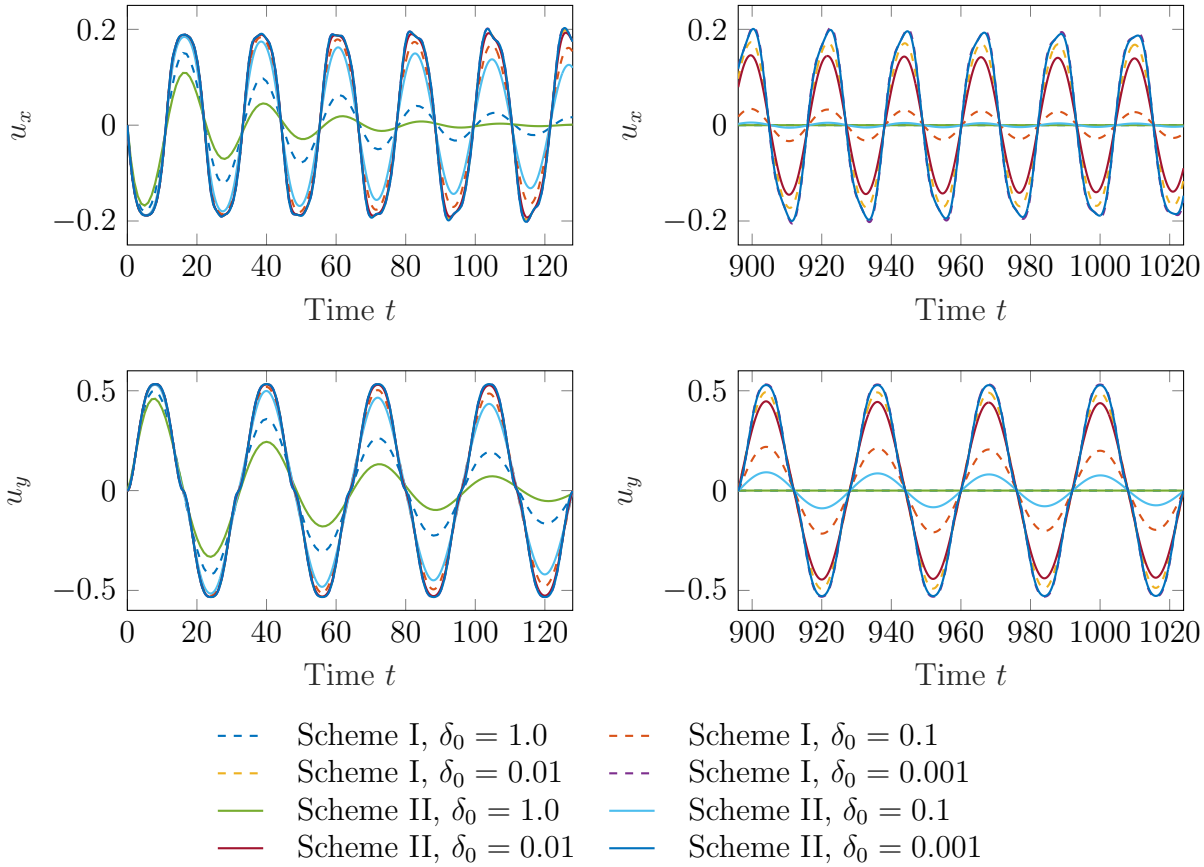


Figure 5.13: *Displacement $\boldsymbol{u}$ of the tip of the cantilever beam (initial position $[8, 0.5]^T$) in the x- and y-directions for $t \in [0, 128]$ and $[896, 1024]$ for Scheme I and Scheme II and time step sizes $\delta_0 \in \{1, 0.1, 0.01, 0.001\}$. Note that numerical damping is reduced with $\delta_0 \to 0$.*

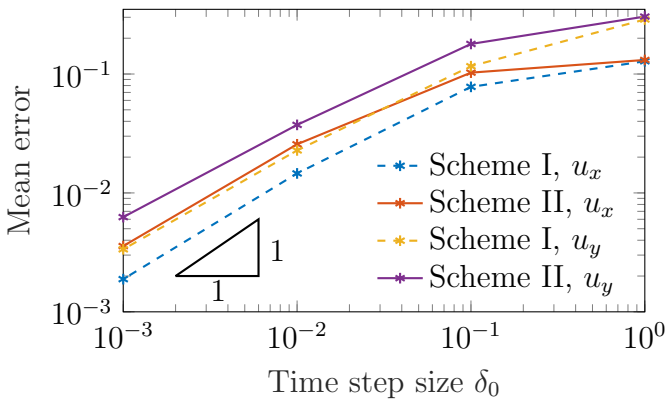


Figure 5.14: *Mean absolute error of the displacement of the beam's tip for $t \in [0, 1024]$. Here, a second-order symplectic Verlet scheme with $\delta_0 = 0.001$ was used as reference.*

despite the approximation of the computed residual norm, see Section 5.3.4. Here, the convergence factors first increase as the coarsening factor m_0 increases, before decreasing due to the small coarse-grid size $N_{n_\ell - 1}$.

On the other hand, for $\delta_0 \in \{0.1, 0.01, 0.001\}$ we observe an increase in predicted and

observed convergence factors as the coarsening factor m_0 increases. Again, predicted and observed convergence factors are in excellent agreement for almost all considered cases. Only for $\delta_0 = 0.01$ is the maximum observed convergence factor larger than the predicted upper bound. The observed rate is 1% larger for both F-relaxation with $m_0 = 32$ and FCF-relaxation with $m_0 \in \{16, 32\}$. Again, this is likely due to the approximate residual norm computation.

Further, we note that additional relaxation steps can be beneficial for large fine grid time step sizes, whereas the effect is negligible for $\delta_0 = 0.001$, thus suggesting that relaxation can be omitted for small fine grid time step sizes to reduce computational work without sacrificing convergence.

Finally, we note that in all considered cases we do not observe any instability as described in Section 5.3.5 and the previous work [33, 34]. For example, Figure 5.16 illustrates the position of the tip of the cantilever after the first three MGRIT iterations with $t \in [0, 64]$, $\delta_0 = 0.1$, and $m_0 = 2$. Here, no artificial amplification of the amplitude of oscillation is observed, in contrast to the case of MGRIT with Scheme I, see Figure 5.12.

The results in this section highlight the benefit of using theoretical upper bounds, as given in [25], as a tool to estimate experimental convergence *a priori* and to design convergent MGRIT algorithms with guaranteed worst-case convergence factors.

5.3.8 Multilevel MGRIT with Scheme II

Here, we generalize the convergent MGRIT algorithm with Scheme II to the multilevel case. Multilevel hierarchies with $n_\ell \in \{3, 4, 5\}$ levels are considered for $\delta_0 \in \{1, 0.1, 0.01\}$; further, $n_\ell \in \{3, 4, 5, 6\}$ levels for $\delta_0 = 0.001$. Total coarsening factors of $\delta_{n_\ell - 1}/\delta_0 \in \{4, 8, 16, 32, 64\}$ are studied, where only m_0 is varied but $m_\ell = 2$ is selected for $\ell = 1, \ldots, n_\ell - 2$. The MGRIT algorithm is started with a forward solve on the coarsest time grid (i.e. use of the XBraid skip-first-down option), performs an initial V-cycle and full multigrid (FMG) cycles for all following MGRIT iterations. Scheme II is employed as one-step integrator on all grid levels and one V-cycle is performed as post-relaxation step at each FMG level. To provide a better indicator for overall performance of the MGRIT algorithm, the mean experimental convergence factor over all MGRIT iterations is reported.

Figure 5.17 reports mean experimental convergence factors over the total coarsening factor $\delta_{n_\ell - 1}/\delta_0$. As illustrated, the best convergence factor for a particular combined coarsening factor is available through the use of FMG-cycles with more levels and slower coarsening between levels, as opposed to FMG-cycles with more aggressive coarsening between levels. Generally, observed convergence factors are significantly smaller than in the two-grid case. Thus, the use of FMG-cycles allows more aggressive coarsening and yields more potential for parallelism in the temporal domain.

If the fine grid size and total coarsening factor are kept fixed, the mean convergence factor improves with growing numbers of levels n_ℓ. For example, for $\delta_0 = 0.1$ and $\prod_{\ell=0}^{n_\ell - 2} = 64$ the mean convergence factor is approximately 0.92 for $n_\ell = 5$ but 1.07 for $n_\ell = 4$ and 1.22 for $n_\ell = 3$.[6] Thus, one can obtain a moderately convergent instead of a slowly

[6] In particular, a measured mean convergence factor of larger than 1 indicates that the residual for the considered problem and a given algorithm cannot satisfy its convergence criteria within the per-

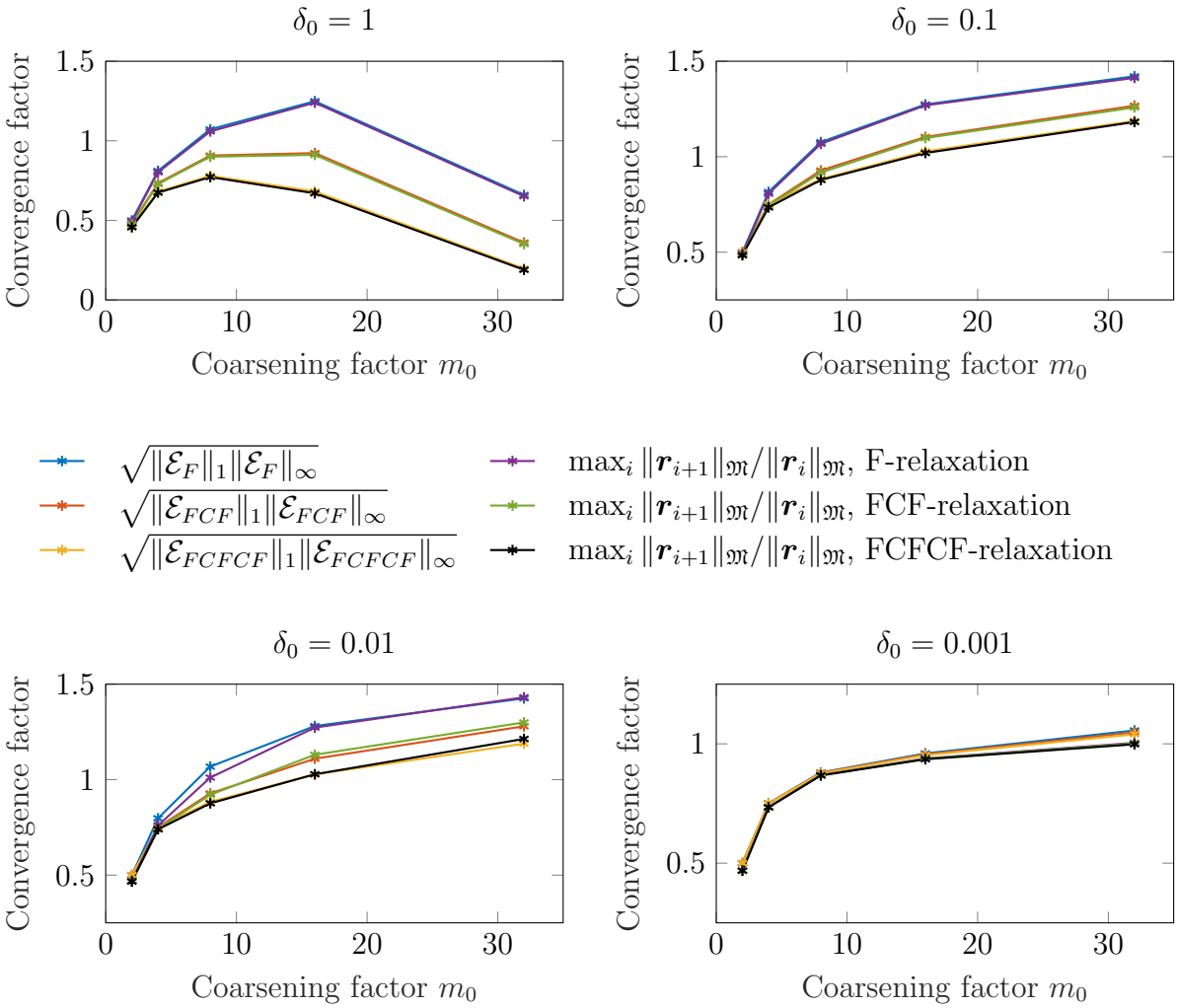


Figure 5.15: *Predicted and measured convergence factor for a two-grid algorithm with various fine grid time step sizes* $\delta_{N_t^0} \in \{1.0, 0.1, 0.01, 0.001\}$ *and coarsening factors* $m_0 \in \{2, 4, 8, 16, 32\}$.

divergent algorithm simply by introducing an additional intermediate time grid level but with the same fine and coarsest grid size.

Timing results

In this section, we present speedup results for an MGRIT algorithm that employs Scheme II and $n_\ell \in \{4, 5, 6\}$ time grid levels with a coarsening factor of two between all time grid levels. Here, we also investigate the effect of using up to four V-cycles as post-relaxation at each FMG-level. The time step size is selected as $\delta_0 = 0.0005$ with $T = 64$. The Euclidean norm with $w_0 = 1$, $w_1 = 0$, $w_2 = 1/m_0$ (see Equation (5.29)) is used to measure solution progress. To investigate the dependency of the wall clock time for MGRIT on the convergence criterion, we employ *tight*, *medium* and *loose* tolerances

formed 60 iterations.

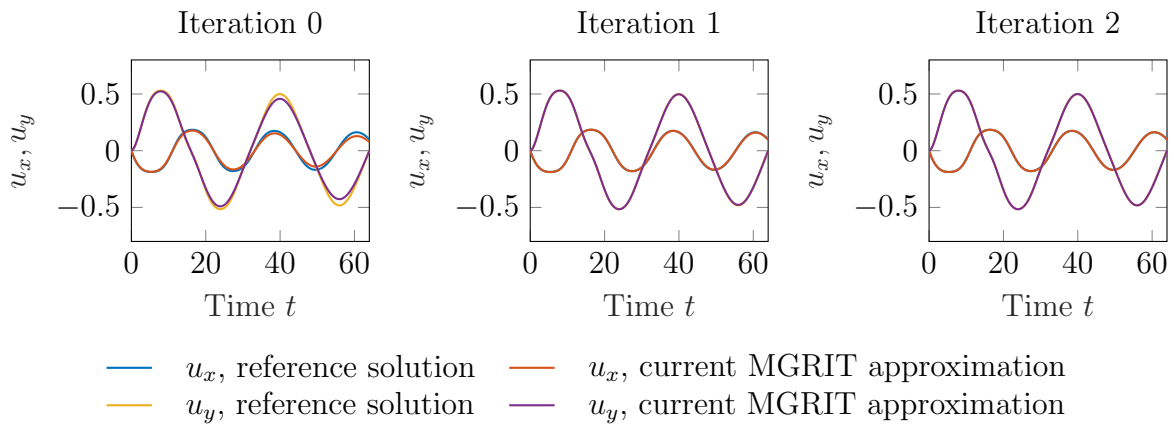


Figure 5.16: *Current approximation of the tip displacement for two-grid MGRIT using Scheme II with coarsening factor $m_0 = 2$, fine grid time step size $\delta_0 = 0.1$, $t \in [0, 64]$ and FCF-relaxation compared with reference solution from sequential time stepping. No instability is observed with Scheme II in contrast to using Scheme I, see Figure 5.12.*

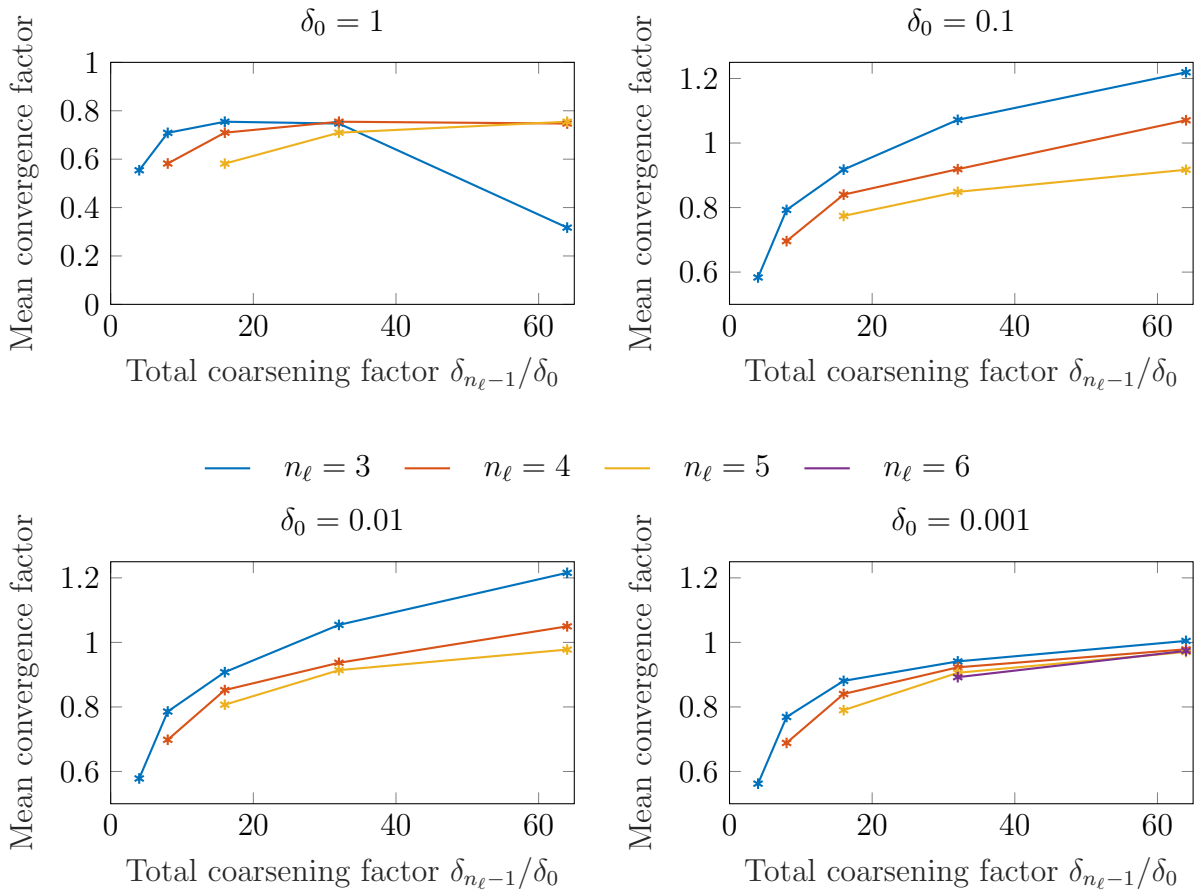


Figure 5.17: *Measured mean convergence factor for m_l-grid MGRIT algorithm with F-cycles and Scheme II for fine grid time step sizes $\delta_0 \in \{1, 0.1, 0.01, 0.001\}$. Note, that the x-axis corresponds to the combined coarsening factor $\prod_{\ell=0}^{n_\ell-2}$.*

of $4.472 \cdot 10^{-7}$, $4.472 \cdot 10^{-6}$ and $4.472 \cdot 10^{-5}$, respectively.

All reported timing results were obtained on TheoSim (see Table A) with $\{16, 32, 48, \ldots, 192\}$ processors using 16 processors per compute node.[7] Data export was switched off and status messages were restricted to a bare minimum to ensure that reported wall clock times are almost entirely dominated by computation and communication costs but not data I/O tasks. The elapsed wall clock time for the sequential time stepping algorithm (using Scheme II) was 263 seconds, which is established as baseline.

Firstly, we note that the MGRIT solution with a loose tolerance of $4.472 \cdot 10^{-5}$ is a good approximation of the sequential time stepping solution. For example, Figure 5.18 illustrates the position of the cantilever's tip (initial position $[8, 0.5]^T$) and its velocity over time. Here, one can appreciate that all fine scale details in the solution are governed by the sequential time stepping solution as well as the MGRIT solution.

As Figure 5.19 illustrates, a 4-level MGRIT solver converges to its tolerance in less iterations than a 6-level MGRIT solver for a given number of V-cycles per FMG-level. On the other hand, performing more V-cycles per FMG-level improves convergence factors significantly. Thus, the required number of iterations to solve the problem to solver tolerance decreases. With $n_\ell = 6$ levels, for example, MGRIT takes 37 iterations to satisfy the tight tolerance when using one V-cycle per FMG-level, however, it takes 25, 20 or 17 iterations when performing two, three or four V-cycles per FMG-level. We further note, that the residual norm is always decreased in subsequent iterations.

As observed in the previous paragraph, the required number of iterations drops when performing additional V-cycles at each FMG-level. Despite the additional *per-iteration* cost; however, the measured wall clock time of the algorithm decreases due to the significantly smaller convergence factors. Figure 5.20 to Figure 5.22 highlight this behavior, where each additional V-cycle per FMG-level yields a reduction in wall clock time for all $n_\ell \in \{4, 5, 6\}$. Here, the best speedups are observed for a 4-level MGRIT solver with four V-cycles per FMG-level. The measured speedup factor is 5.25x, 2.12x and 1.33x for the three considered solver tolerances, see Table 5.1a - Table 5.1c. Further, the speedup is 5.25x, 4.72x or 2.91x for the loose solver tolerance and a 4-, 5- or 6-level MGRIT solver and four V-cycles per FMG-level. Note, that the slope of the curves in Figure 5.20 and Figure 5.22 show better scaling of a 6-level MGRIT solver compared to a 4-level MGRIT solver due to the smaller coarse grid size and larger potential for parallelism. At processor counts of up to 192, no crossover point can be reached when performing only one V-cycle per FMG-level and imposing the tight tolerance. For all other combinations of tolerances and cycling strategies considered in this section, however, a crossover point is reached.

Lastly, we compare the performance of the FMG-cycle algorithm to a V-cycle algorithm with F- or FCF-relaxation. At the loose solver tolerance, a speedup of 1.95x is achieved using 192 processors with $n_\ell = 4$ and F-relaxation, see Table 5.1d. Additional FC-relaxation steps, however, result in a more expensive V-cycle algorithm. No speedups are observed for the medium and tight tolerances (not included in Table 5.1). Further, we note the benefit of using FMG-cycles with additional V-cycles per FMG-level over a V-cycle algorithm. For example, for $n_\ell = 4$ the FMG-cycle algorithm can solve the problem to a tighter solver tolerance compared to the V-cycle algorithm with a similar speedup. The best measured speedups are 5.25x for a FMG-cycle algorithm and 1.95x for

[7] Employing one compute node, a preliminary experiment showed best performance of our implementation at 16 processors per node.

a V-cycle algorithm.

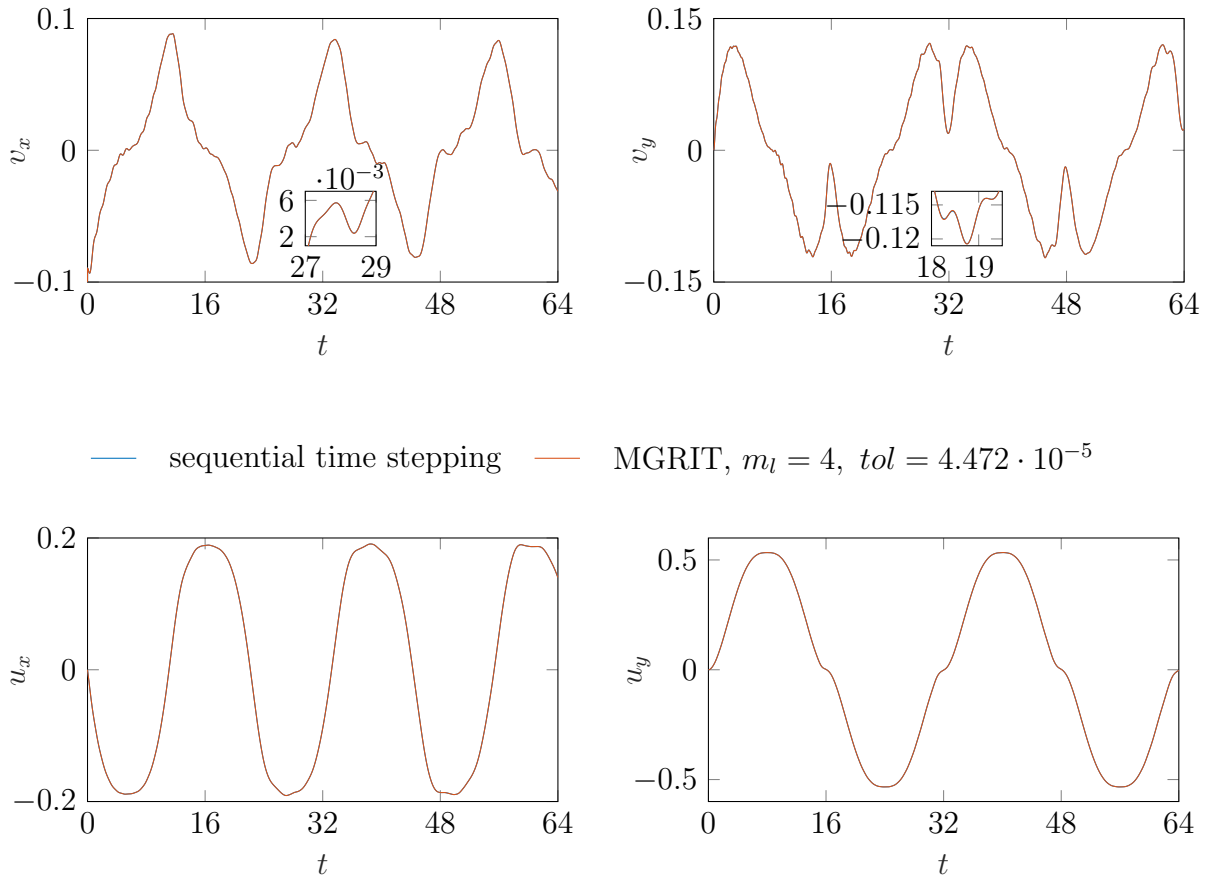


Figure 5.18: *Comparison of velocity and displacement at the tip of the cantilever beam (initial position* $[8, 0.5]^T$*) for sequential time stepping and MGRIT with Scheme II (*$n_\ell = 4$*,* $m_\ell = 2$ *for* $\ell = 0, \ldots, n_\ell - 1$*), time step size* $\delta_0 = 0.0005$ *and MGRIT tolerance* $tol = 4.472 \cdot 10^{-5}$*. Note, how fine time scale variations are resolved despite the relatively loose MGRIT convergence tolerance.*

5.3.9 Discussion

In the previous sections, two different backward Euler time discretization schemes were presented and investigated. Scheme I was considered as the default scheme in the application code CHeart [94], whereas Scheme II was proposed as an improvement for parallel-in-time integration methods. It was shown, that MGRIT with Scheme I exhibits strong instabilities for the linear elasticity model problem. The observations are in line with previous work in this field [33, 34]. Although MGRIT with Scheme II uses only a slightly different time discretization, a convergent scheme can be derived for a range of coarsening factors, both for two-grid and multilevel algorithms. Scheme II not only results in a stable algorithm when used with MGRIT but also yields a speedup over sequential time stepping while maintaining the accuracy of Scheme I for practical time step sizes.

In the light of observed differences between A-stable and L-stable time discretization schemes in Section 5.1 and Section 5.2, it seems likely that the stability properties of

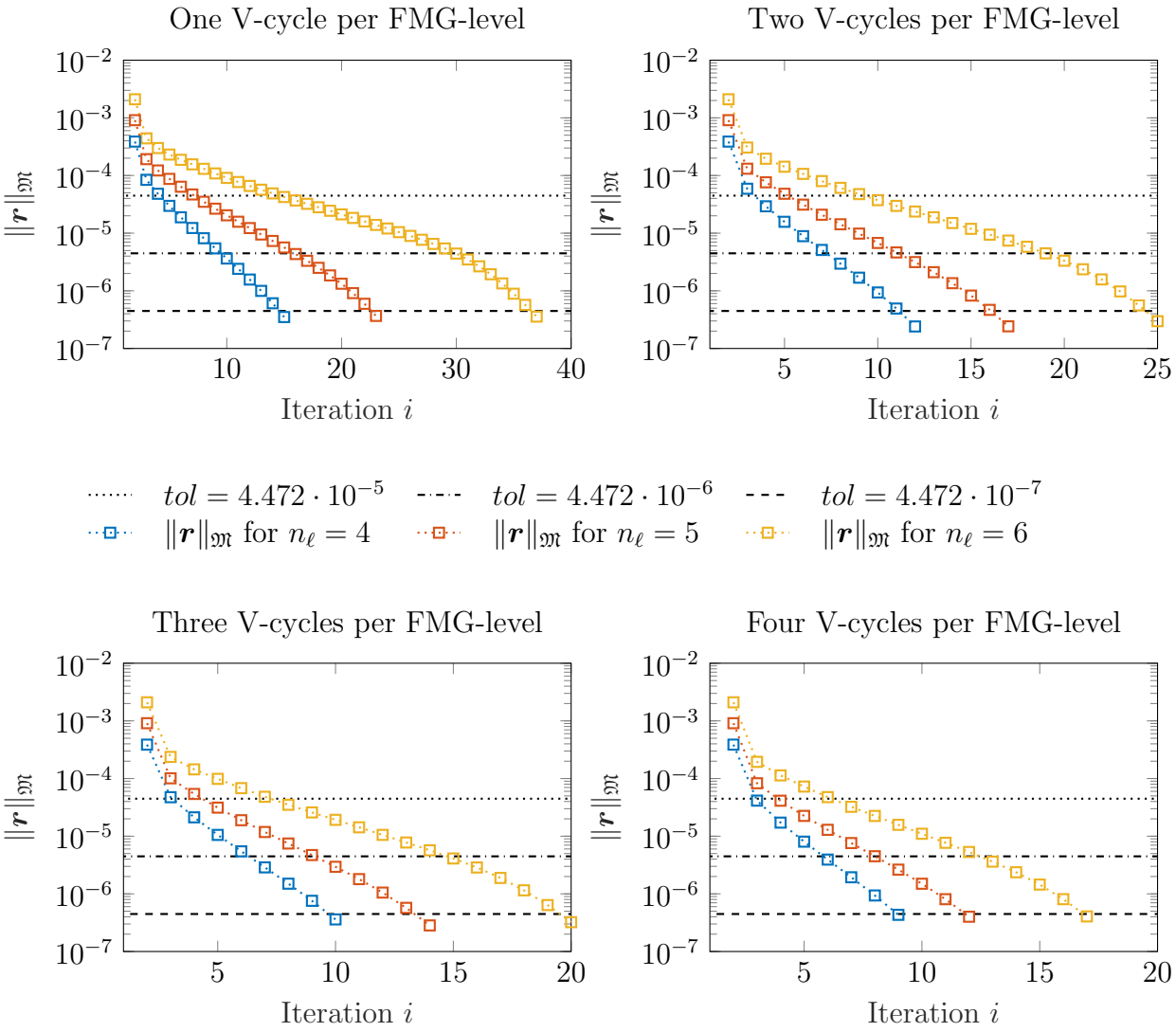


Figure 5.19: *Decrease of MGRIT residual norm $\|\boldsymbol{r}\|_{\mathfrak{M}}$ (with $w_0 = 1$, $w_1 = 0$, $w_2 = 1/m_0$; see Equation (5.29)) for $n_\ell \in \{4, 5, 6\}$ levels and $\{1, 2, 3, 4\}$ V-cycles per FMG-level. Note how the additional V-cycle per FMG-level yields a faster decrease in the residual norm. Thus, the required number of iterations to solve the problem to solver tolerance decreases. This causes a drop in wall clock time, as seen in Figure 5.20 - Figure 5.22, despite the higher per-cycle cost.*

Scheme I and Scheme II are the key drivers of divergence and convergence, respectively. For example, Scheme II is a standard backward Euler scheme, which is L-stable. On the other hand, Scheme I resembles a Crank-Nicolson method, which is only A-stable.

In the case of using MGRIT with Scheme II, convergence was predicted for two-grid algorithms with F- and FCF-relaxation and a range of coarsening factors. The predictions matched quite closely with observations in the numerical experiments, supporting the use of the analysis presented in [25] as a powerful tool to design convergent algorithms *a priori*.

Although the analysis presented in [25] is restricted to the two-grid case, the convergent and efficient multilevel algorithm was a straightforward generalization of its two-grid

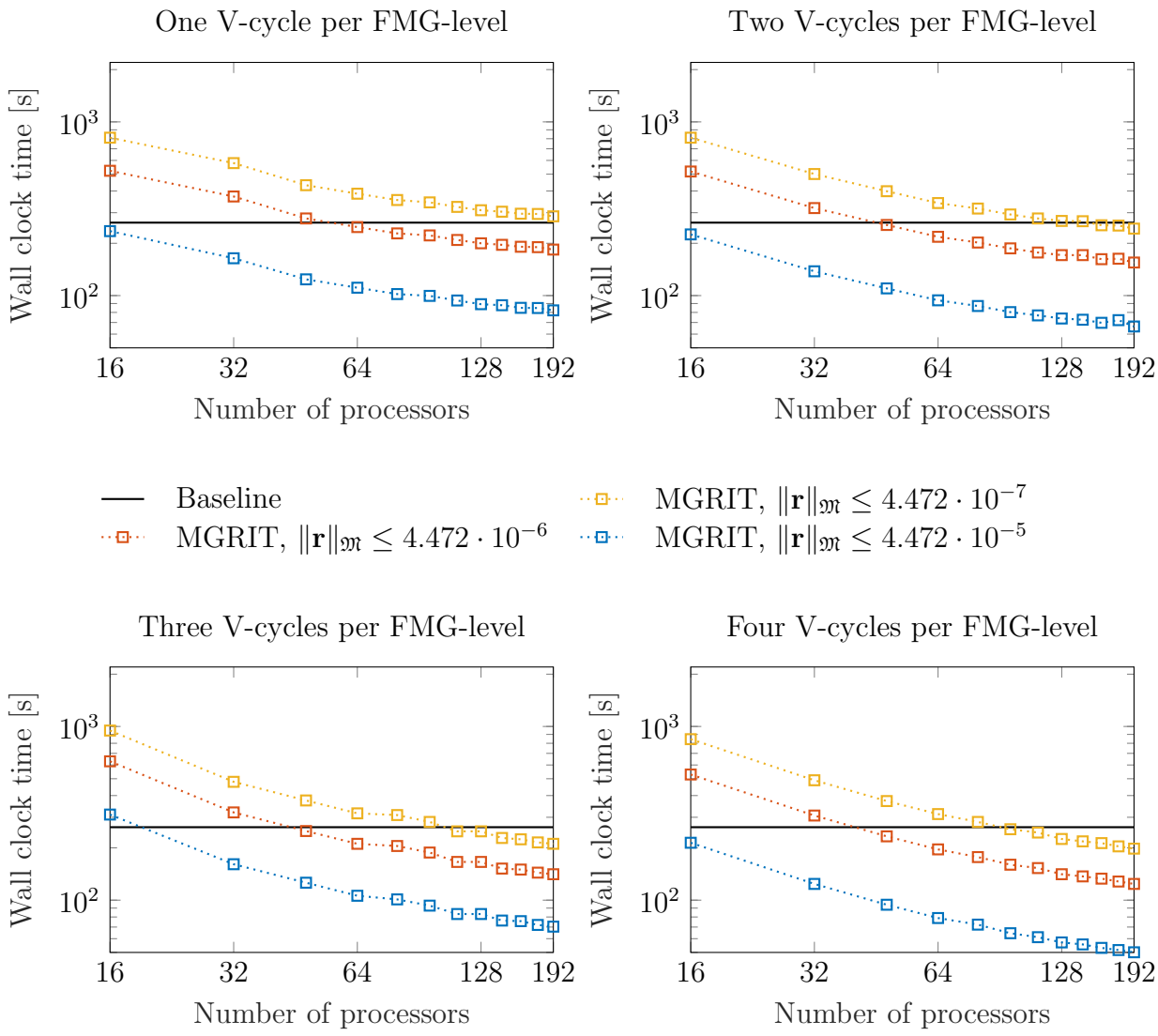


Figure 5.20: *Wall clock time for MGRIT with Scheme II and FMG-cycles with* $n_\ell = 4$ *and* $m_\ell = 2$ *for all* ℓ.

counterpart. Here, the use of FMG-cycles was beneficial to accelerate convergence in the true multilevel case, enabling larger total coarsening factors $\delta_{n_\ell-1}/\delta_0$ compared to the two-grid case. It was also shown that, for a given total coarsening factor $\delta_{n_\ell-1}/\delta_0$, slow temporal coarsening (and thus, more time grids) can improve convergence over faster temporal coarsening with less time grids. Furthermore, performing additional V-cycles at each FMG-level makes the coarse-grid solve more powerful and improves convergence significantly.

Further, timing results for the time-parallel algorithm were presented in Section 5.3.8. Using 192 processors, a speedup of 5.25x was achieved for a 4-level algorithm with four V-cycles per FMG-level. Similarly, speedups of 4.72x and 2.91x were shown for 5- and 6-level algorithms. It was also demonstrated, that the use of FMG-cycles resulted in a better speedup than the use of V-cycles with F- and FCF-relaxation. The reported wall clock times are almost entirely dominated by computation and communication, however, data export, for example, is a completely serial process for sequential time-stepping while

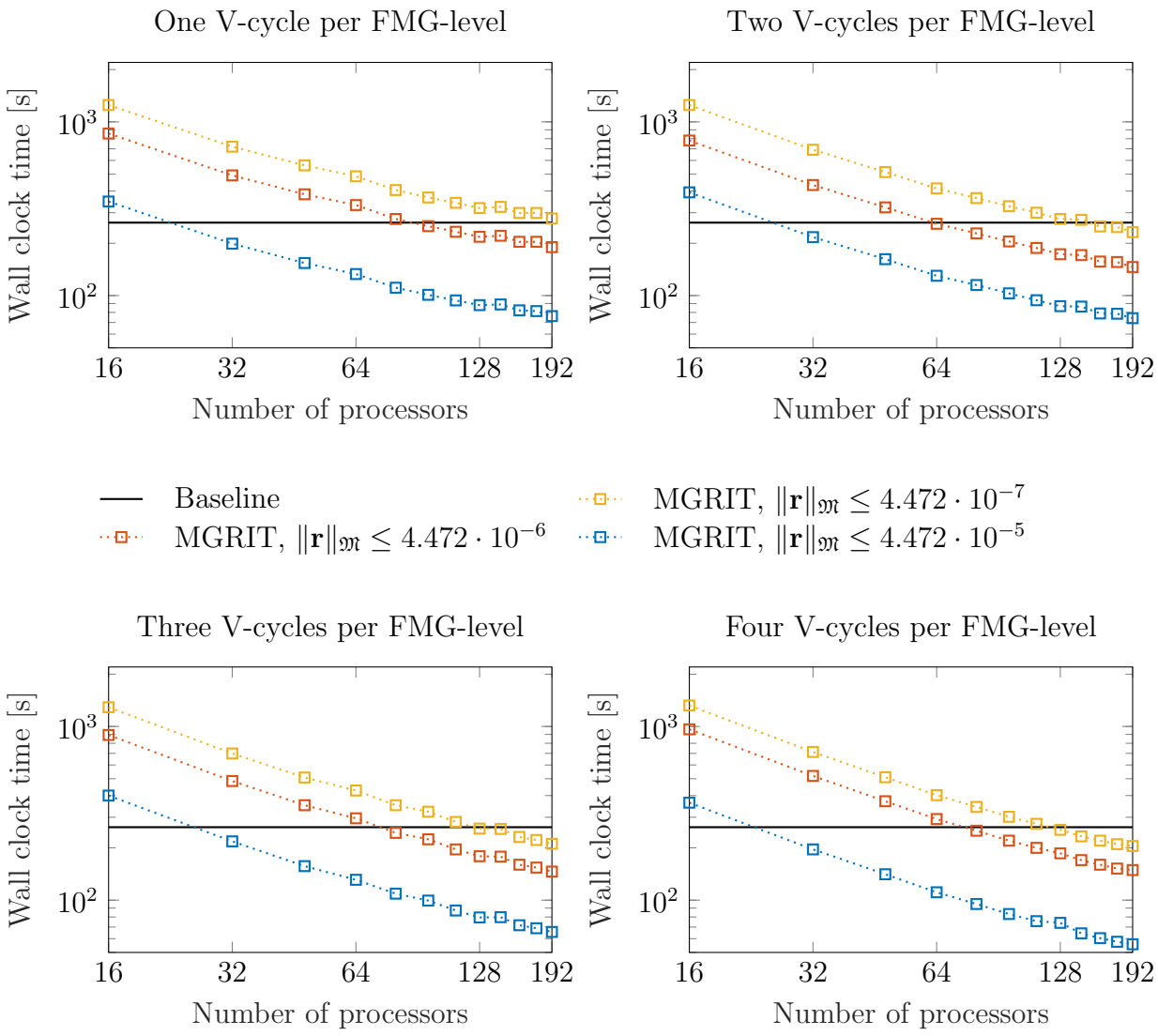


Figure 5.21: *Wall clock time for MGRIT with Scheme II and FMG-cycles with* $n_\ell = 5$ *and* $m_\ell = 2$ *for all* ℓ.

it is parallel for the MGRIT algorithm. Thus, for practical applications that include data I/O tasks, larger speedups can be expected. Lastly, Figure 5.22 (for example, 6-level algorithm with four V-cycles per FMG-level) shows scaling of the wall clock time at fixed spatial problem size until 192 processors, which we expect to continue beyond the number of processors employed in this study.

No refinement of the spatial problem or spatial parallelism was considered as the performance of MGRIT and the expected speedup are mainly dictated by the temporal dimension size. Spatial parallelism will degrade strong scaling, because communication overhead will be relatively larger. Larger spatial problems, however, will help to improve strong scaling because each time step will be more expensive, that is, communication overhead will be relatively smaller.

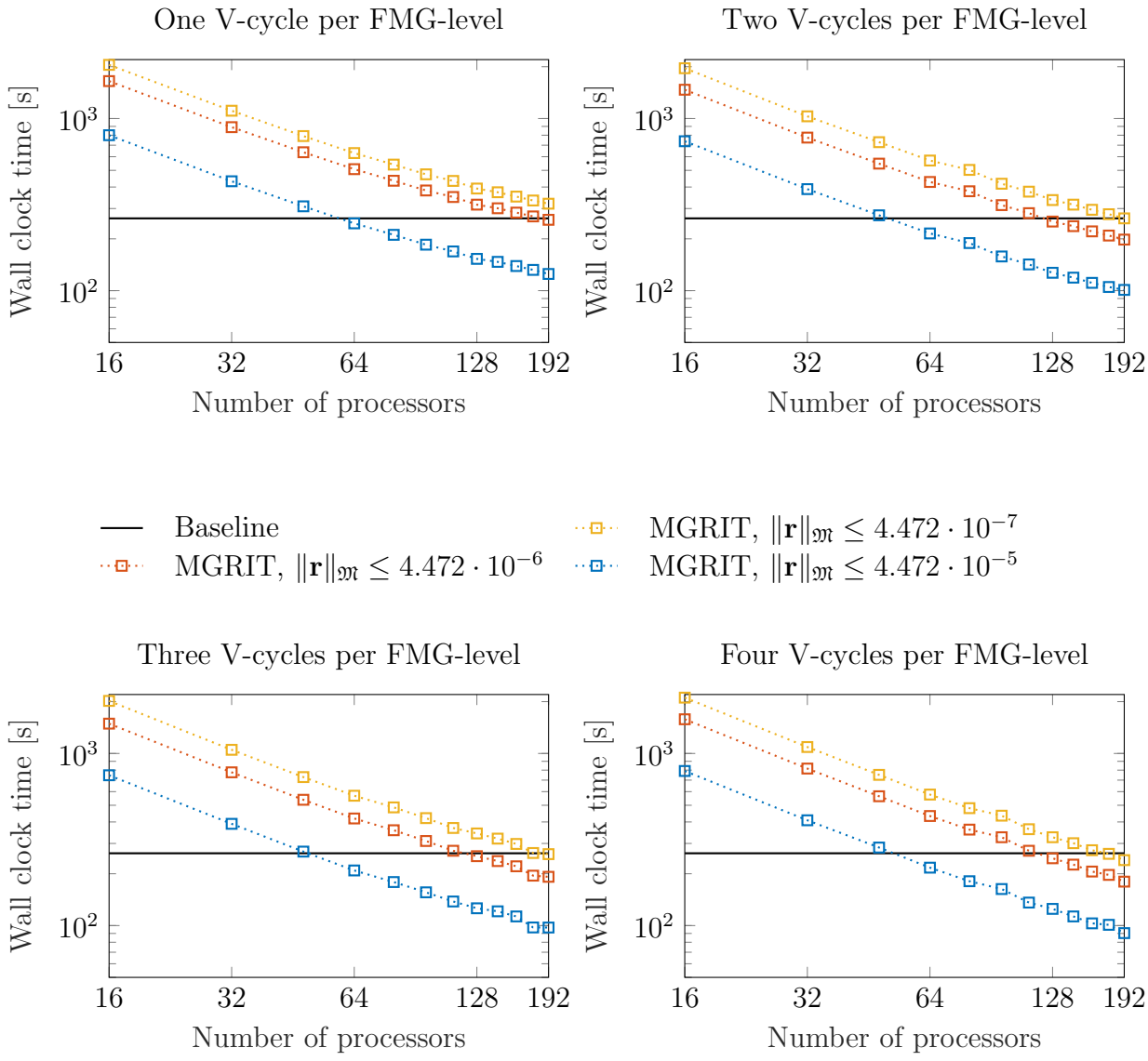


Figure 5.22: *Wall clock time for MGRIT with Scheme II and FMG-cycles with* $n_\ell = 6$ *and* $m_\ell = 2$ *for all* ℓ.

# V-cycles	$n_\ell = 4$	$n_\ell = 5$	$n_\ell = 6$
1	0.92x	0.95x	0.82x
2	1.08x	1.13x	1.00x
3	1.25x	1.25x	1.01x
4	1.33x	1.28x	1.10x

(a) *MGRIT tolerance* $tol = 4.472 \cdot 10^{-7}$.

# V-cycles	$n_\ell = 4$	$n_\ell = 5$	$n_\ell = 6$
1	1.43x	1.38x	1.02x
2	1.70x	1.80x	1.33x
3	1.87x	1.80x	1.37x
4	2.12x	1.77x	1.46x

(b) *MGRIT tolerance* $tol = 4.472 \cdot 10^{-6}$.

# V-cycles	$n_\ell = 4$	$n_\ell = 5$	$n_\ell = 6$
1	3.20x	3.45x	2.10x
2	3.96x	3.55x	2.60x
3	3.74x	4.01x	2.70x
4	5.25x	4.72x	2.91x

(c) *MGRIT tolerance* $tol = 4.472 \cdot 10^{-5}$.

# FC-relaxations	$n_\ell = 4$	$n_\ell = 5$	$n_\ell = 6$
0	1.95x	1.28x	0.61x
1	1.83x	1.13x	0.54x

(d) *MGRIT tolerance* $tol = 4.472 \cdot 10^{-5}$.

Table 5.1: *Measured speedup using* 192 *processors with* $n_\ell \in \{4, 5, 6\}$ *levels. Table 5.1a - Table 5.1c with FMG-cycles and* $\{1, 2, 3, 4\}$ *V-cycles at each FMG-level; Table 5.1d with V-cycles and F- or FCF-relaxation.*

5.4 Flow through simplified stenosed valve in 2D[8]

This section considers flow in the ascending aorta (see Figure 5.23) in a simplified two-dimensional geometry. Figure 5.24 illustrates the setting: Pulsatile blood flow is entering the domain from the left ventricle. The blood flows past two stenosed valves (approximated as rigid tissue), the aortic sinus (anatomic dilations of the ascending aorta) and enters the ascending aorta. The outflow is slightly extended where the aortic arch would be located.

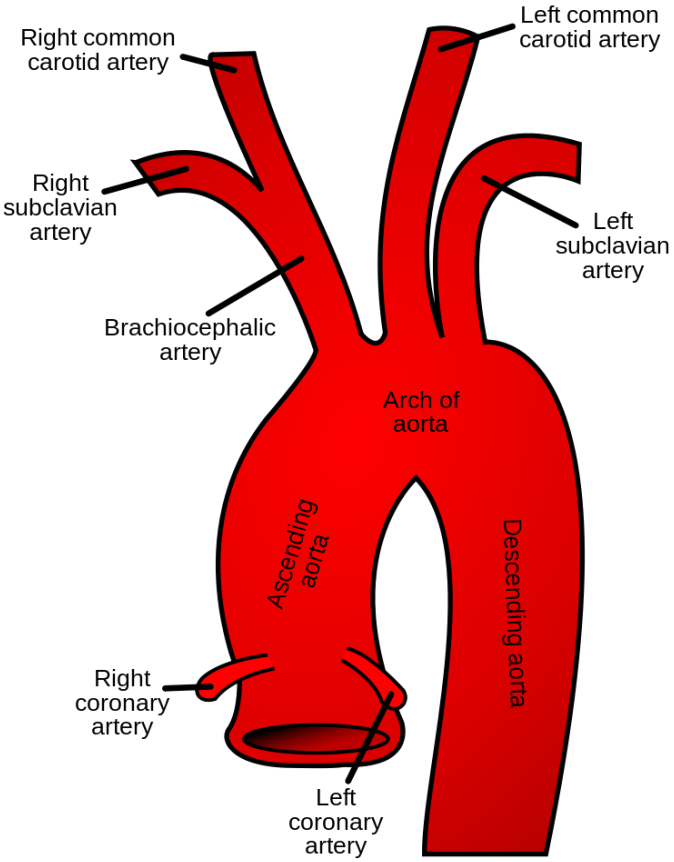


Figure 5.23: *Schematic of the ascending aorta with coronary arteries, the aortic arch with three branches and the descending aorta. Image[9] from [50].*

The Navier-Stokes equations for incompressible Newtonian flow are considered as a general model for the blood flow in the ascending aorta:

$$\rho_f \partial_t \boldsymbol{v}_f + \rho_f \boldsymbol{v}_f \cdot \nabla \boldsymbol{v}_f - \nabla \cdot \left[\mu_f \left(\nabla \boldsymbol{v}_f + \nabla^T \boldsymbol{v}_f - p_f \boldsymbol{I}\right)\right] = 0 \qquad \text{in } \Omega_f, \tag{5.39}$$

$$\nabla \cdot \boldsymbol{v}_f = 0 \qquad \text{in } \Omega_f, \tag{5.40}$$

with velocity $\boldsymbol{v}_f$, pressure p_f, density $\rho_f = 10^{-3}\ g/mm^3$ and viscosity $0.04\ g/(mm \cdot s)$.

[8] Results in this section have previously appeared [61]

[9] Public Domain image: `https://commons.wikimedia.org/w/index.php?curid=1617151`

Pulsatile parabolic inflow with cycle length $T = 1.024\ s$ is prescribed at the inflow,

$$\boldsymbol{v}_f = \begin{bmatrix} 200 \\ 0 \end{bmatrix} \frac{mm}{s} \left(0.5 + 0.5 \cdot \cos\left(\frac{2\pi t - \pi \cdot 1\ s}{T} \right) \right) \left[1 - \left(\frac{y - 0.685\ mm}{14.415\ mm} \right)^2 \right], \quad (5.41)$$

and pulsatile outflow is prescribed at the left and right coronary arteries,

$$\boldsymbol{v}_f = \begin{bmatrix} 0 \\ 10 \end{bmatrix} \frac{mm}{s} \left(0.5 + 0.5 \cdot \cos\left(\frac{2\pi t - \pi \cdot 1\ s}{T} \right) \right) \begin{cases} 1 & \text{for } y > 0, \\ -1 & \text{for } y < 0. \end{cases} \quad (5.42)$$

Zero Neumann conditions are prescribed at the outflow and zero Dirichlet conditions are prescribed at all other boundaries. The initial condition is set as zero.

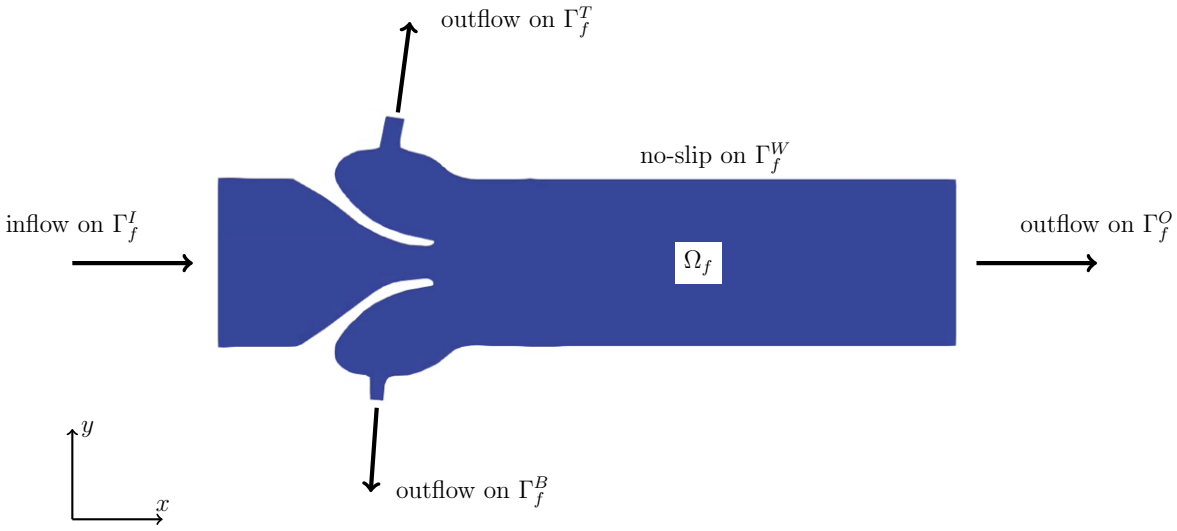


Figure 5.24: *Simplified stenosed valve geometry in two dimensions with maximally dilated valves. Inflow occurs from the left ventricle on the left. Minor outflows are observed toward the left and right coronary arteries at the top and bottom, and main outflow toward the aortic arch on the right.*

5.4.1 Space-time discretization

Space is discretized using finite elements with 1388 inf-sup stable $\mathrm{P}^2 - \mathrm{P}^1$ Taylor-Hood elements and $N_x = 6879$ degrees-of-freedom, see Figure 5.25. The length of the temporal domain corresponds to 10 cardiac cycles with time $t \in [0, 10T]$. A backward Euler time discretization scheme is employed to discretize the temporal domain with equidistant time steps and a time step size of $\delta_t = 1\ ms$.

5.4.2 Comparison of models for Stokes flow and Navier-Stokes flow

Because the multilevel convergence theory for MGRIT in Section 4 was developed under the assumption of linear PDEs, let's first consider neglecting the nonlinear term $\boldsymbol{v}_f \cdot \nabla \boldsymbol{v}_f$ in Equation (5.55), which yields the Stokes flow model for an incompressible Newtonian fluid. Figure 5.26 illustrates how the flow (magnitude) develops over cycle 10. The

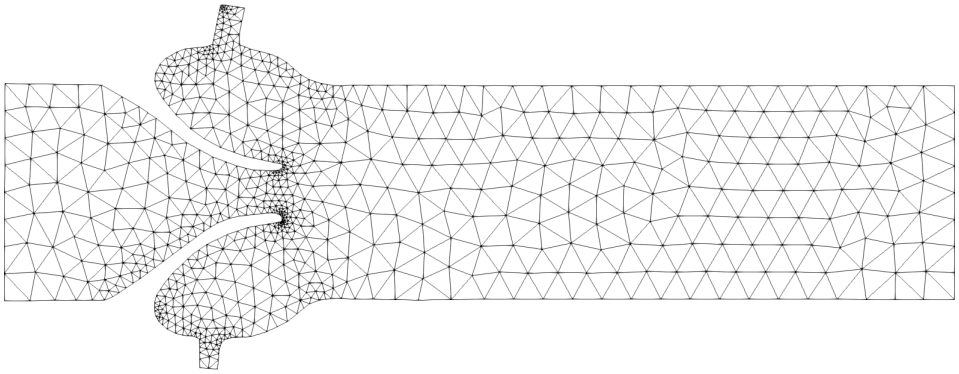


Figure 5.25: *Mesh for the stenosed valve problem consisting of* 1388 *triangular finite elements. The mesh is refined near small features and boundaries with high curvature.*

parabolic inflow (see Equation (5.41)) is constricted between the valves to form a jet. The jet quickly widens to an approximately parabolic flow profile, similar to Poiseuille flow in a channel; for example, note the (close to) parallel streamlines at $t = 9.6\ s$ in Figure 5.26. Further note the small reflow regions between the valves and the coronary arteries.

In contrast, Figure 5.28 illustrates the flow (magnitude) over cycle 10, as obtained from the Navier-Stokes flow model. Here, the jet (due to the flow constriction from the valves) extends farther downstream of the valves. This yields more pronounced reflow regions (e.g., compare the streamlines in Figure 5.26 and Figure 5.28 for $t = 9.6\ s$) and creates significantly different flow patterns. Further note that the y-component of the velocity permits the identification of the front of the jet, see Figure 5.27.

Thus, the observed physical flow field varies significantly depending on the underlying model (Stokes flow or Navier-Stokes flow). In later parts of Section 5.4, it will be investigated whether convergence analysis for the linear PDE case can still yield reasonable *a priori* estimates of the observed convergence of multilevel MGRIT applied to the nonlinear PDE case.

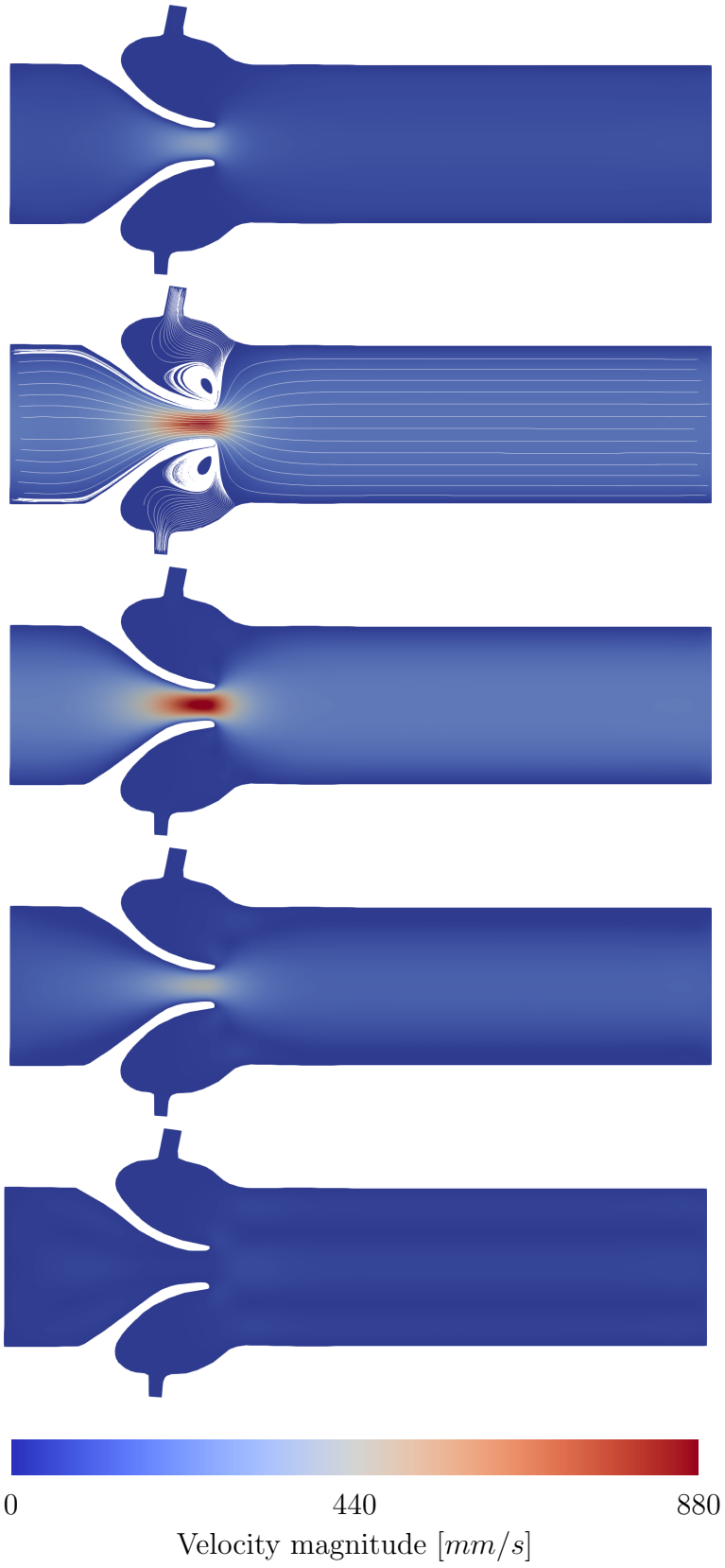


Figure 5.26: *Stokes flow: Velocity magnitude at multiple time points over the* 10^{th} *cycle:* $t \in \{9.4, 9.6, 9.8, 10.0, 10.2\}$ s. *The parabolic inflow becomes a jet between the valves. The jet quickly widens to an approximately parabolic flow profile, similar to Poiseuille flow in a channel.*

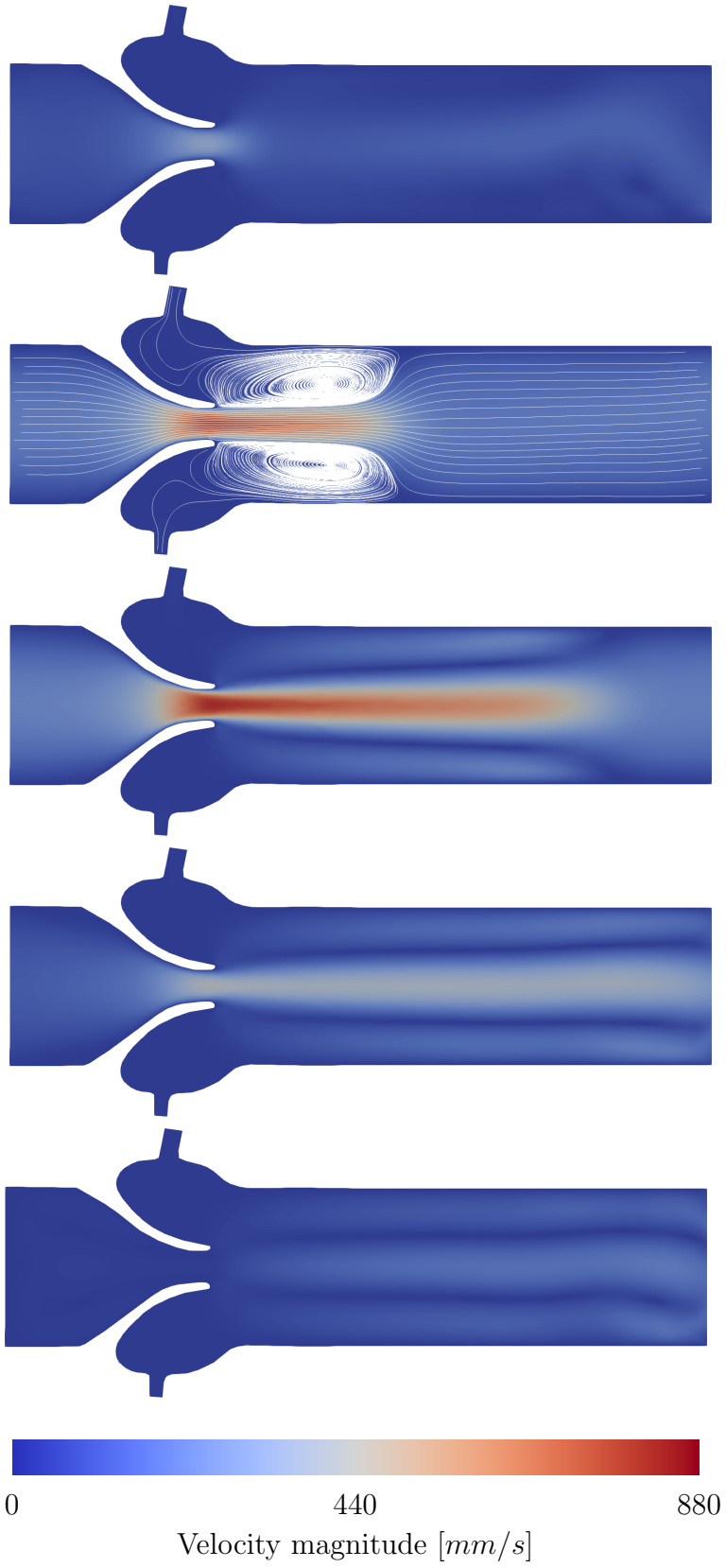


Figure 5.27: *Navier-Stokes flow: Velocity magnitude at multiple time points over the* 10^{th} *cycle:* $t \in \{9.4, 9.6, 9.8, 10.0, 10.2\}$ *s. The parabolic inflow becomes a jet between the valves. In contrast to the Stokes flow case (see Figure 5.26), the jet is clearly visible downstream and the y-component of the flow is nonzero, see Figure 5.28.*

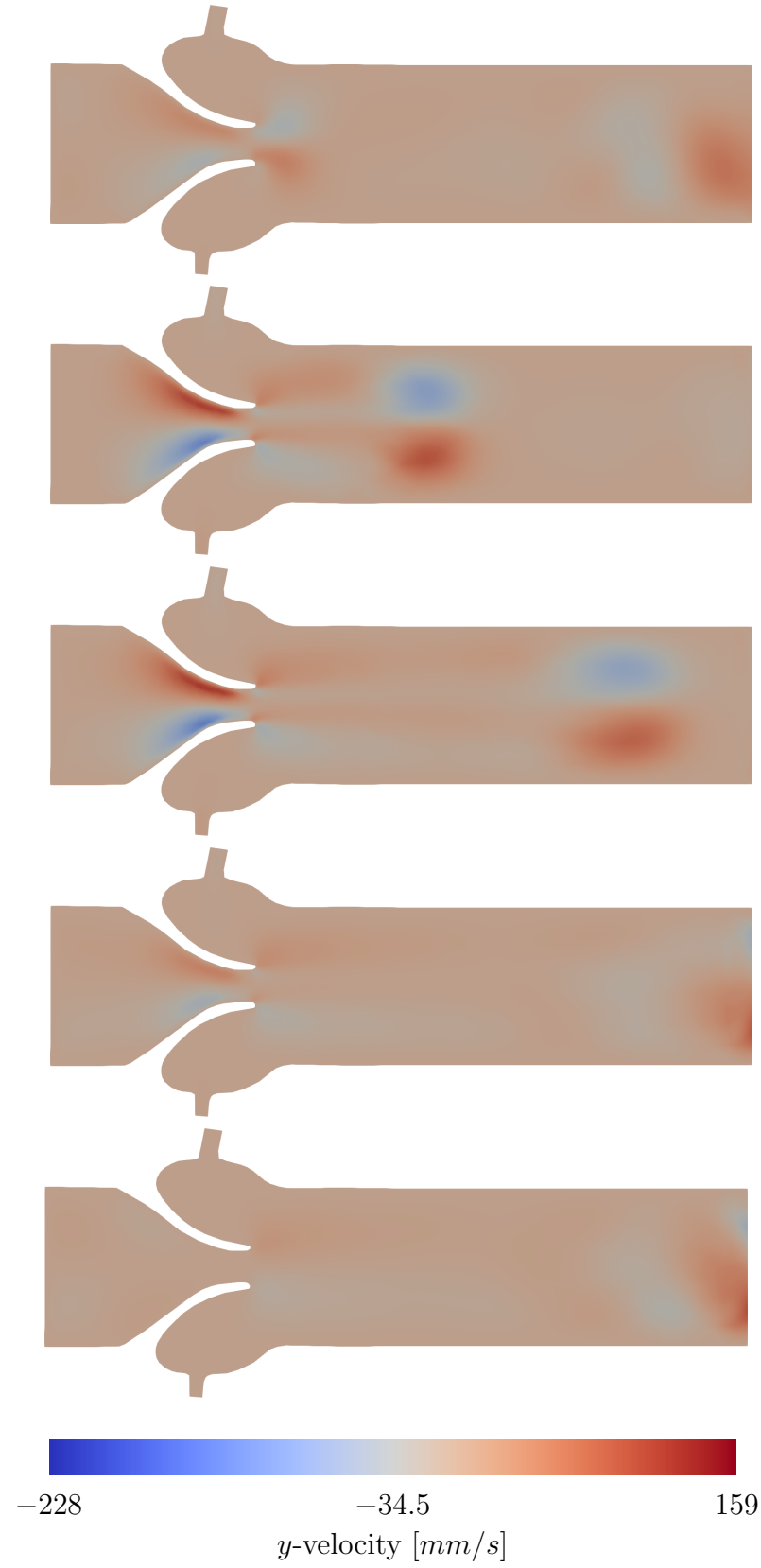


Figure 5.28: *Navier-Stokes flow: Nonzero y-component of the velocity at multiple time points over the 10^{th} cycle: $t \in \{9.4, 9.6, 9.8, 10.0, 10.2\}$ s. The traveling wave front due to the pulsatile inflow is clearly noticeable in the y-component at the tip of the jet (compare with Figure 5.26).*

5.4.3 Periodic steady-state solution

A common theme in many fields is the presence of temporal periodicity in the investigated model. For the stenosed valve problem, the periodicity is boundary-driven due to the pulsatile inflow. In practice, the boundary conditions might be known from medical imaging data, while the flow field in the interior of the domain might not, which is why simulations come into play. For the numerical model, however, an initial condition is required, which is often unknown as well. In general, a zero initial condition is as good as any other (reasonable) initial condition like in our case. The numerical model is then driven to a *periodic steady-state*, which means that cycle-to-cycle variations become small or negligible with respect to some error measure. That is, the solution becomes periodic in time:

$$\|\boldsymbol{v}_f(\cdot, t+qT) - \boldsymbol{v}_f(\cdot, t+(q-1)T)\| \to^{q\to\infty} 0 \qquad \forall\ t \in [0, T). \tag{5.43}$$

In practice, one can anticipate reaching the periodic steady-state quicker (i.e. for smaller $q \in \mathbb{Z}^+$) if a better initial condition is employed.

Figure 5.29 illustrates convergence of the solution at the beginning of each cycle $\boldsymbol{v}_f(\cdot, (q-1)T)$ to the solution at the end of cycle 10, i.e. $\boldsymbol{v}_f(\cdot, 10T)$, for the Stokes flow model. Here, the average error reduction is approximately one order of magnitude for each simulated cycle. This highlights the fact that the accuracy of the periodic steady-state solution is directly linked to the number of simulated cycles. As a matter of course, each reduction of the cycle-to-cycle error (at $t = (q-1)T$ for $q = 2, 3, \ldots$) by approximately one order of magnitude increases the total computational cost by the computational cost of adding one cycle length to the temporal domain. In the context of using parallel-in-time integration with MGRIT, this motivates reducing the computational cost by simulating only one cardiac cycle but making the time grid periodic, see Section 3.6. That is, convergence to the periodic steady-state is achieved by updating the initial condition.

5.4.4 Weak form of the Stokes flow problem

Consider the Stokes problem,

$$\rho_f \partial_t \boldsymbol{v}_f - \nabla \cdot \left[\mu_f \left(\nabla \boldsymbol{v}_f + \nabla^T \boldsymbol{v}_f - p_f \boldsymbol{I}\right)\right] = 0 \qquad \text{in } \Omega_f, \tag{5.44}$$

$$\nabla \cdot \boldsymbol{v}_f = 0 \qquad \text{in } \Omega_f, \tag{5.45}$$

which is discretized using finite elements in space and backward Euler in time, see Section 5.4.1. The general discrete weak form can be written as follows:

Find $\boldsymbol{s}^{n+1} := (\boldsymbol{v}_f^{n+1}, p_f^{n+1}) \in \boldsymbol{\mathcal{V}}_D^h \times \mathcal{W}_f^h$, such that for every $\boldsymbol{d} := (\boldsymbol{w}_f, q_f) \in \boldsymbol{\mathcal{V}}_0^h \times \mathcal{W}_f^h$:

$$\begin{aligned} R(\boldsymbol{s}^{n+1}, \boldsymbol{s}^n, \boldsymbol{d}) = & \int_{\Omega_f^h} \rho_f \frac{\boldsymbol{v}_f^{n+1} - \boldsymbol{v}_f^n}{\delta_t} \cdot \boldsymbol{w}_f \, d\Omega_f^h \\ & + \int_{\Omega_f^h} \mu_f \left[\nabla \boldsymbol{v}_f^{n+1} + \nabla^T \boldsymbol{v}_f^{n+1} - p_f^{n+1} \boldsymbol{I}\right] : \nabla \boldsymbol{w}_f + q_f \nabla \cdot \boldsymbol{v}_f^{n+1} \, d\Omega_f^h = 0. \end{aligned} \tag{5.46}$$

The definition of the function spaces are:

$$S^p(\Omega_f^h) = \{f : \Omega_f^h \to \mathbb{R} \mid f \in \mathcal{C}^0(\bar{\Omega}_f^h), f|_{\tau_e} \in \mathbb{P}^p(\tau_e), \forall\ \tau_e \in T_f^h\}, \tag{5.47}$$

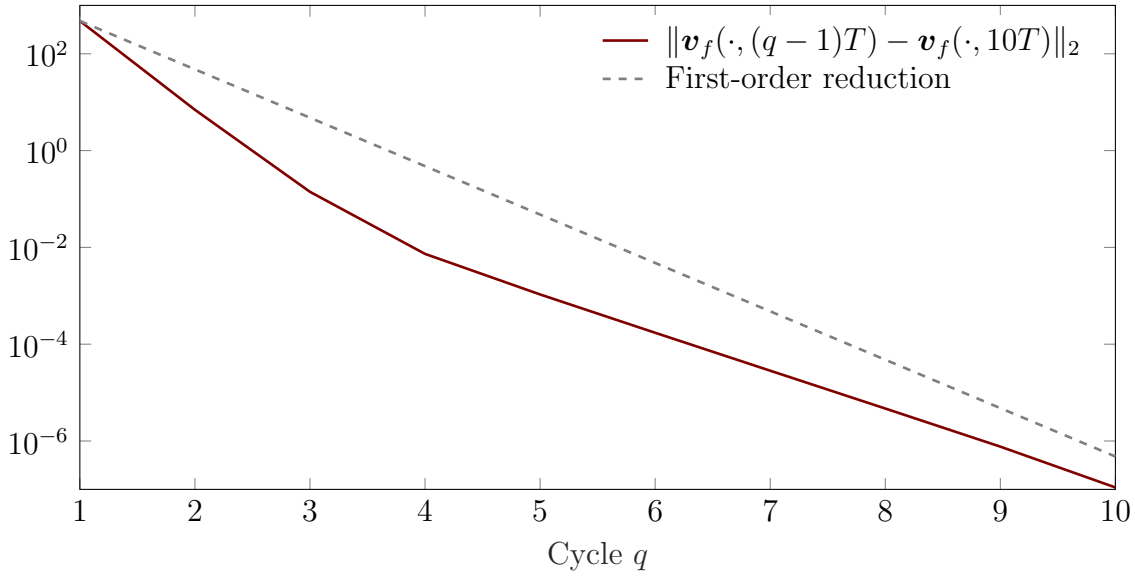


Figure 5.29: *Convergence of the solution at the beginning of each cycle* $\boldsymbol{v}_f(\cdot,(q-1)T)$ *to the solution at the end of cycle* 10*, i.e.* $\boldsymbol{v}_f(\cdot,10T)$*, for the Stokes flow model with cycle length* T*.*

which represent the general continuous p^{th}-order piecewise polynomial spaces defined on Ω_f^h. Consequently, we define:

$$\boldsymbol{\mathcal{V}}^h = [S^2(\Omega_f^h)]^2 \qquad \text{and} \qquad \mathcal{W}_f = S^1(\Omega_f^h), \tag{5.48}$$

and incorporating the boundary conditions:

$$\boldsymbol{\mathcal{V}}_0^h = \{\boldsymbol{v} \in \boldsymbol{\mathcal{V}}^h \mid [\boldsymbol{v}]_y = 0 \text{ on } \Gamma_f^O\}, \tag{5.49}$$

$$\boldsymbol{\mathcal{V}}_D^h = \{\boldsymbol{v} \in \boldsymbol{\mathcal{V}}^h \mid \boldsymbol{v} = \boldsymbol{v}_f \text{ according to (5.41) and (5.42)}\}. \tag{5.50}$$

Rewriting Equation (5.46) in operator form yields,

$$\begin{bmatrix} \rho_f \boldsymbol{M} + \mu_f \delta_t \boldsymbol{K} & \delta_t \boldsymbol{B}^T \\ \delta_t \boldsymbol{B} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_f^{n+1} \\ p_f^{n+1} \end{bmatrix} = \begin{bmatrix} \rho_f \boldsymbol{M} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_f^{n} \\ p_f^{n} \end{bmatrix}, \tag{5.51}$$

where $\boldsymbol{M}$ is the mass matrix, and $\boldsymbol{K}$ and $\boldsymbol{B}$ refer to the discretized weak form Laplacian and divergence operators $\nabla^2()$ and $\nabla \cdot ()$, respectively. Rewriting (5.51) in $\boldsymbol{\Phi}$-form,

$$\begin{bmatrix} \boldsymbol{v}_f^{n+1} \\ p_f^{n+1} \end{bmatrix} = \underbrace{\begin{bmatrix} \rho_f \boldsymbol{M} + \mu_f \delta_t \boldsymbol{K} & \delta_t \boldsymbol{B}^T \\ \delta_t \boldsymbol{B} & \boldsymbol{0} \end{bmatrix}^{-1} \begin{bmatrix} \rho_f \boldsymbol{M} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}}_{\boldsymbol{\Phi}} \begin{bmatrix} \boldsymbol{v}_f^{n} \\ p_f^{n} \end{bmatrix}, \tag{5.52}$$

highlights the sparsity:

$$\boldsymbol{\Phi} = \begin{bmatrix} [\boldsymbol{\Phi}]_{11} & \boldsymbol{0} \\ [\boldsymbol{\Phi}]_{21} & \boldsymbol{0} \end{bmatrix}. \tag{5.53}$$

The fact that $\boldsymbol{v}_f^{n+1}$ and p_f^{n+1} do not depend on p_f^n makes sense in that p_f is used as a Lagrange multiplier to enforce the incompressibility constraint (5.45) at each time point.

Equation (5.45) is further a time-independent linear PDE, however, time-independent PDEs can be solved in just one MGRIT iteration because the coarse-grid solve is exact. These two observations provide arguments to restrict the convergence analysis to the velocity block $[\boldsymbol{\Phi}]_{11}$ similar to the analysis for the linear elasticity problem in Section 5.3.

5.4.5 Multilevel MGRIT for the Stokes flow problem

In this section, consider MGRIT convergence for the temporal domain $[0, T]$ and the following parameter choices:

- Cycling strategy: V- and F-cycles
- Relaxation scheme: F- and FCF-relaxation
- Number of levels: $n_\ell \in \{2, 3, 4, 5, 6\}$
- Temporal coarsening factor: $m_\ell = 2$ for all ℓ
- Absolute tolerance: 10^{-8}
- Zero initial space-time guess
- Skip first down-cycle

To simplify notation, let $\boldsymbol{\Phi}_\ell$ denote the velocity block $[\boldsymbol{\Phi}]_{11}$ for all grid levels $\ell = 0, 1, \ldots, n_{\ell-1}$. The analysis (see Chapter 4) proceeds by simultaneously diagonalizing $\{\boldsymbol{\Phi}_\ell\}$,

$$\boldsymbol{U}^{-1}\boldsymbol{\Phi}_\ell\boldsymbol{U} = \operatorname{diag}(\lambda_{\ell,1}, \lambda_{\ell,2}, \ldots, \lambda_{\ell,N_x}) \qquad \text{for } \ell = 0, 1, \ldots, n_{\ell-1}, \tag{5.54}$$

with eigenvalues $\{\lambda_{\ell,k}\}$.

Section 5.1 and 5.2 showed that the inequality bounds and approximate convergence factors provide reasonably sharp estimates of observed worst-case convergence at reduced computational cost compared to computing the ℓ^2-norm of the respective error propagation operators. Thus, the analysis in this section is restricted to these cheaper estimates. Furthermore, the additional scaling due to the condition number of the eigenvector matrix is ignored since Section 5.2 showed that the quality of the estimates is good even without the scaling factor. In that sense, the estimates are no longer guaranteed upper bounds but rather a means to estimate observed worst-case convergence *a priori*.

Figure 5.30 illustrates multilevel convergence of MGRIT V-cycles with F- and FCF-relaxation for the Stokes problem. Observed worst-case convergence factors increase with increasing numbers of levels in the MGRIT hierarchy, as is usually observed for MGRIT V-cycles. While the sharpness of the estimates based on the inequality bound deteriorates with larger n_ℓ, they still provide estimates of the observed worst-case convergence factor of reasonable quality; especially, when considering that these estimates can be computed *a priori*. For the wave equation, it was observed that additional relaxation steps (e.g., switching from F- to FCF-relaxation) cannot improve MGRIT convergence significantly, however, for the Stokes problem there is a significant benefit.

On the other hand, Figure 5.31 illustrates multilevel convergence of MGRIT F-cycles with F- and FCF-relaxation. While the observed worst-case convergence factor of MGRIT

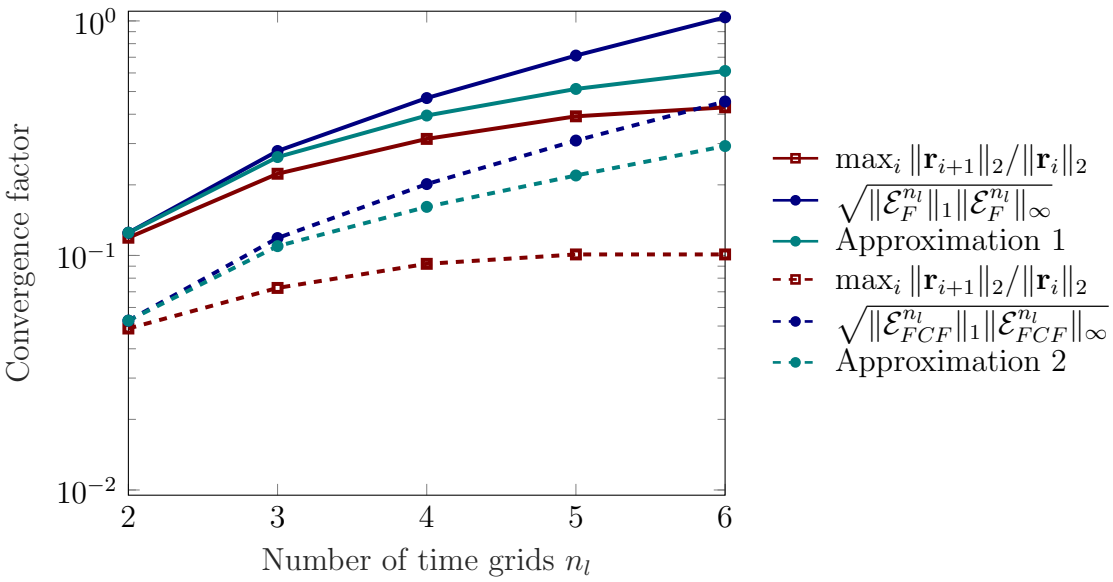


Figure 5.30: *Multilevel convergence of MGRIT V-cycles with F- and FCF-relaxation for the Stokes problem: Observed worst-case convergence factor compared with a priori estimates (no p-contribution).*

V-cycles increases with a growing number of levels, such an increase is not observed for MGRIT F-cycles, irrespective of the choice of F- or FCF-relaxation. In most cases, two-level convergence provides a heuristic upper bound on multilevel convergence (see Section 4), and thus, constant or almost constant convergence factors for an increasing number of levels can be considered ideal / optimal. The estimates based on the inequality bound predict this desirable property of MGRIT F-cycles. Approximation 3, however, predicts a slight increase in the convergence factor. This is plausible since the approximation is based on various additional simplifications and, overall, it is the cheapest estimate available.

Overall, convergence of MGRIT for the Stokes problem is fairly similar to the diffusion equation, see Section 5.1. This is surprising from the point-of-view that the Stokes equations are a simplification of the Navier-Stokes equations, which have some hyperbolic component. It is, however, less surprising if one considers that the Stokes problem in (5.44) and (5.45) infact looks quite similar to the diffusion equation with additional constraints (i.e. incompressibility).

5.4.6 Multilevel MGRIT for the time-periodic Stokes flow problem

Here, the setting from Section 5.4.5 is considered, however, the time-grids are now made periodic. That is, MGRIT is employed to approximate the solution over one cycle only, with $t \in [0, T]$, and convergence to the periodic steady-state is achieved by updating the initial condition $\boldsymbol{v}_f(\cdot, 0)$, based on the approach described in Section 3.6.

5.4.6.1 Two-level MGRIT with F- and FCF-relaxation

First, consider two-level MGRIT with FCF-relaxation and a temporal coarsening factor of 2. The two-level algorithm converges in 12 MGRIT iterations and the error in the

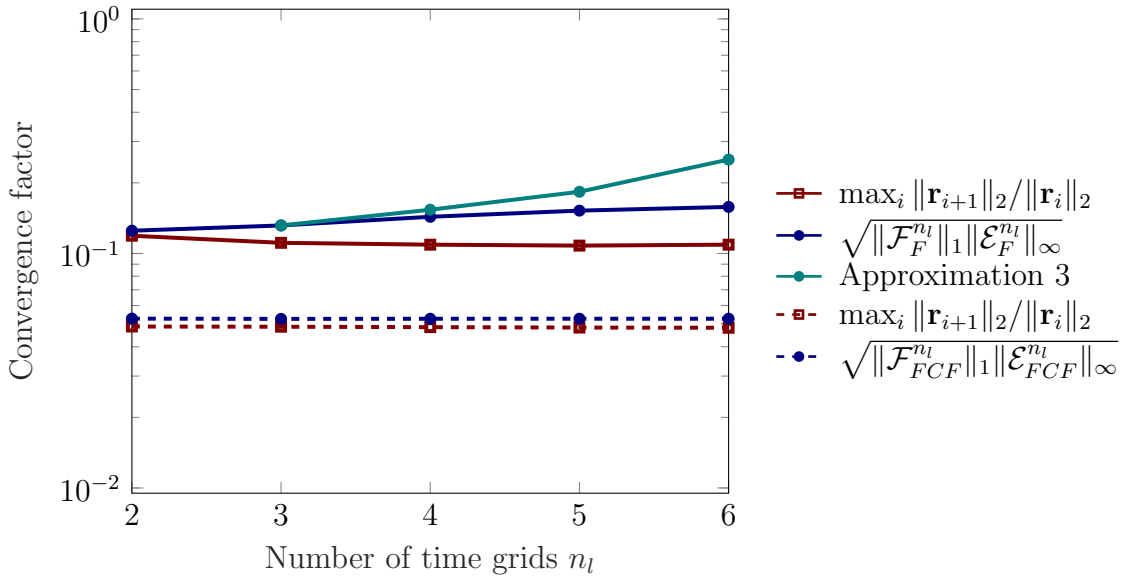


Figure 5.31: *Multilevel convergence of MGRIT F-cycles with F- and FCF-relaxation for the Stokes problem: Observed worst-case convergence factor compared with a priori estimates (no p-contribution).*

initial condition converges at a similar rate, see Figure 5.32. On the other hand, consider the residual norm at each time point on the coarse grid, see Figure 5.33. In contrast to traditional MGRIT for the nonperiodic case, the largest residual norm is not observed towards the end of the temporal domain, but at the beginning. This makes sense as the largest mismatch / discontinuity in the space-time solution will occur at $t = 0$ due to the continuously updated initial condition; it is this mismatch that is then propagated in time and appears as a *spike*. Looking closer at Figure 5.33, it is seen that the spikes occur at t_4 for FCF-relaxation (and at t_2 for F-relaxation; plot omitted), which is due to the fact that MGRIT propagates the exact solution across two coarse-grid intervals for FCF-relaxation (and one coarse-grid interval for F-relaxation).

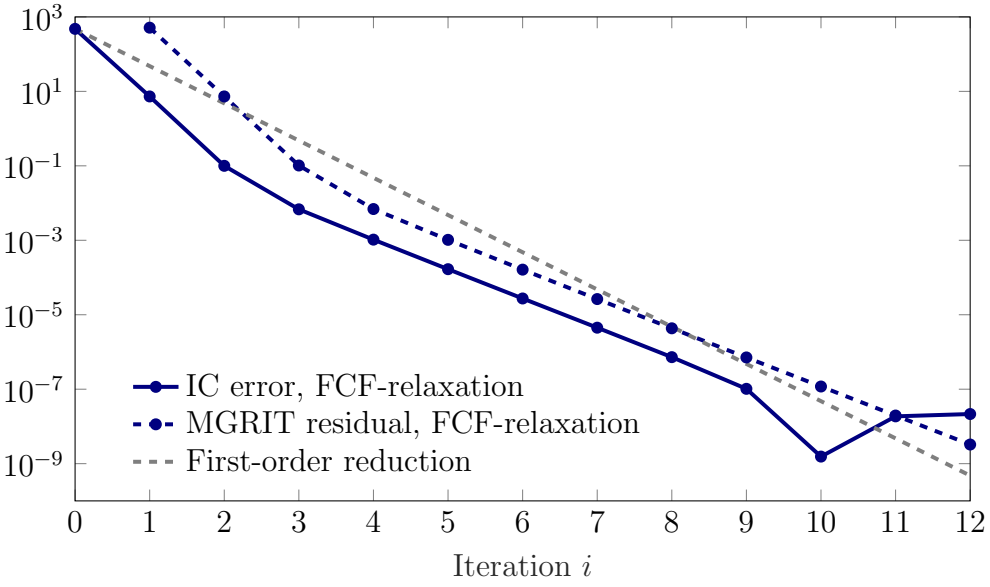


Figure 5.32: *Two-level MGRIT with F- and FCF-relaxation and temporal coarsening factor* 2 *for the time-periodic Stokes problem: Reduction of MGRIT residual and initial condition (IC) error.*

Since the spike only occurs at a single time point on the coarse grid and right at the beginning of the temporal domain, one can consider skipping the updates of the initial condition once a reasonable tolerance is achieved. For example, consider a tolerance of 10^{-10} for the initial condition (compare Figure 5.29), which is used to skip further updates once the ℓ^2-norm of the changes in the initial condition are smaller than said tolerance. For FCF-relaxation, this means that the tolerance for the initial condition is reached faster than the MGRIT tolerance. Thus, the property that MGRIT propagates the exact solution (here, this would be the *converged* initial condition) means that the spike in Figure 5.33 can be smoothed out and MGRIT can be terminated earlier, see Figure 5.34.

In preliminary experiments for MGRIT with F-relaxation, it was found that the MGRIT tolerance was satisfied before the initial condition could be considered converged. Thus, the time-periodic MGRIT algorithm from Section 3.6 checks convergence of the initial condition after each MGRIT iteration. The time-periodic MGRIT algorithm is only permitted to terminate if both convergence criteria for MGRIT and the initial condition are satisfied. This is achieved by means of a *callback* function that was implemented in XBraid. This *callback* function permits the simulation code (here: CHeart) to detect convergence of the initial condition to a time-periodic steady-state. If the intial condition is converged, it lets XBraid terminate, otherwise it requests more MGRIT iterations if this is not the case.

Convergence estimates

Now, consider two-level MGRIT with F- and FCF-relaxation and temporal coarsening factor $m_0 \in \{2, 4, 8, 16, 32\}$ for the time-periodic Stokes problem. It is then assessed how well the inequality bounds and approximate convergence factors (i.e. Approximation 1 and Approximation 2) can capture the convergence behavior despite the fact that these

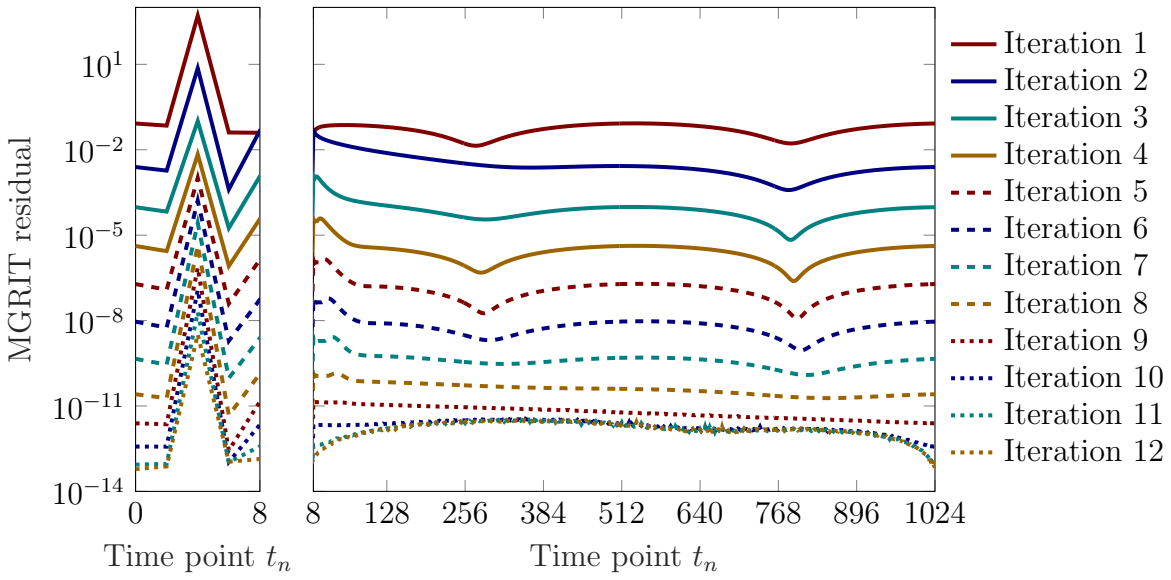


Figure 5.33: *Two-level convergence of MGRIT with FCF-relaxation and temporal coarsening factor 2 for the time-periodic Stokes problem: MGRIT residual at C-points for multiple iterations if no IC tolerance is used.*

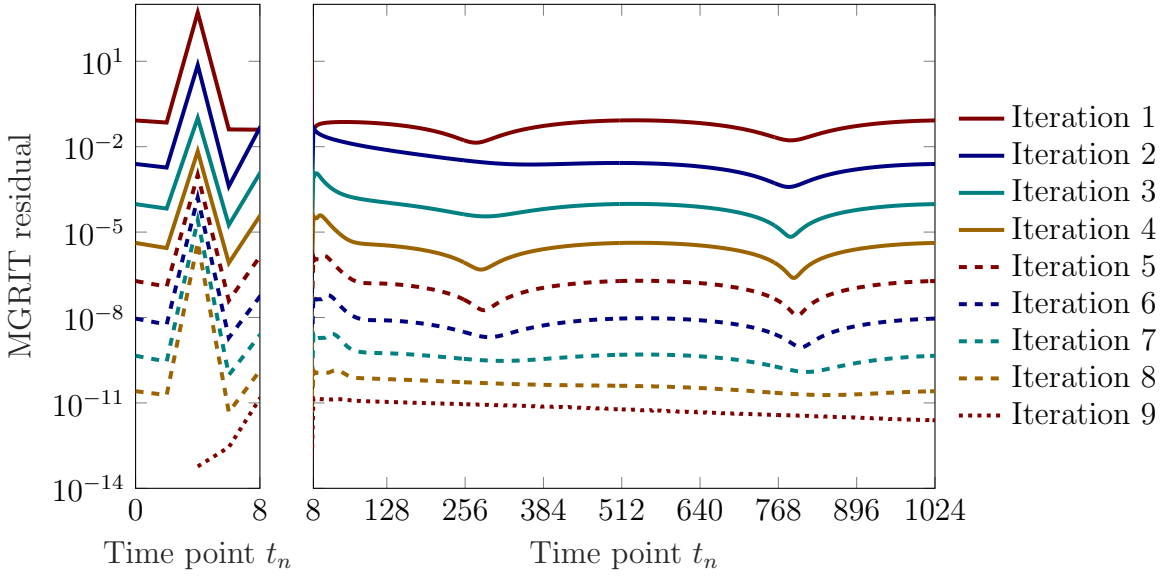


Figure 5.34: *Two-level convergence of MGRIT with FCF-relaxation and temporal coarsening factor 2 for the time-periodic Stokes problem: MGRIT residual at C-points for multiple iterations if IC tolerance 10^{-10} is used.*

theoretical results were not derived for the time-periodic case.

Figure 5.35 highlights that the estimates (of the worst-case convergence factor) based on the inequality bound and the approximate convergence factors are effectively the same. For two-level MGRIT with F-relaxation, the theoretical estimates predict the worst-case convergence factor for $m_0 = 2$ well, and almost perfectly for $m_0 > 2$. For FCF-relaxation, however, the theoretical estimates underestimate the worst-case convergence factor observed for the time-periodic MGRIT algorithm, which is not desirable. In particular, the

experiments show a decreasing worst-case convergence factor with an increasing temporal coarsening factor, whereas the theoretical estimates predict an increasing worst-case convergence factor. In fact, it is quite surprising that one observes better (worst-case) convergence for two-level MGRIT with FCF-relaxation when the temporal coarsening increases. This is quite unusual but very beneficial, since it yields a better converging algorithm with cheaper per-iteration cost.

One should emphasize that the benefits of FCF-relaxation only arise when the temporal coarsening factor is large, see Figure 5.35. That is, in time-parallel runs with small temporal coarsening factors it is likely that MGRIT with F-relaxation will have a shorter time-to-solution than MGRIT with FCF-relaxation, whereas the opposite can be anticipated when the temporal coarsening factor is large.

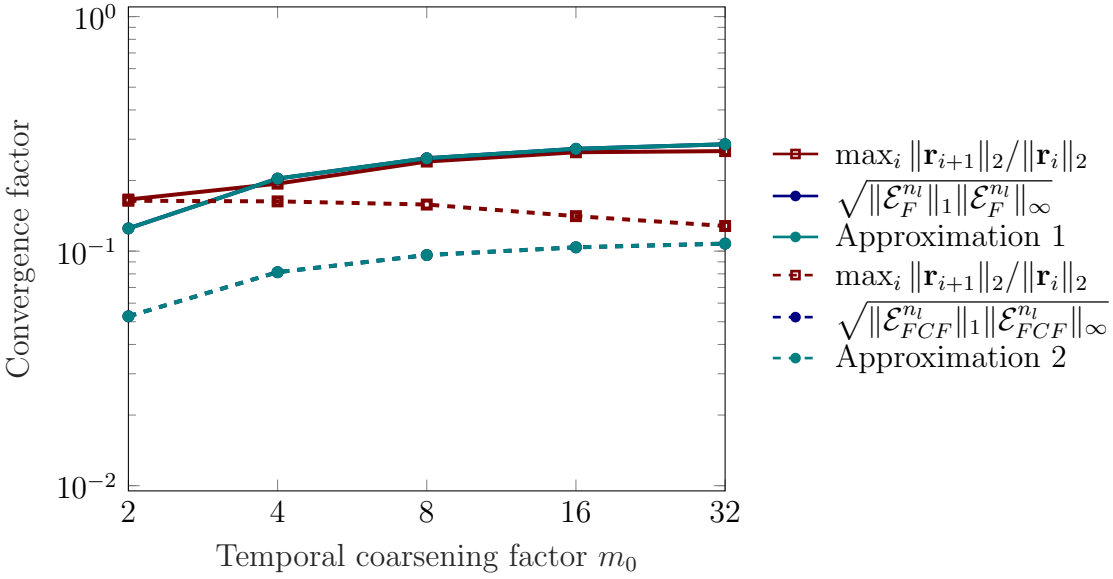


Figure 5.35: *Two-level MGRIT with F- and FCF-relaxation for the Stokes problem: Observed worst-case convergence factor (for time-periodic problem) compared with a priori estimates (for one period; no p-contribution).*

Parallel performance on LEAD and TheoSim

Let's now consider parallelism applied to the spatial domain, the temporal domain and the spatiotemporal domain. Runtimes reported here were obtained using two small-scale clusters: LEAD and TheoSim (see Section A). The reported runtime of the serial algorithm is the best of 5 runs and of the parallel algorithms it is the best of 1 run.

If no parallelism is employed, the algorithm takes 523.05 s on LEAD and 332.16 s on TheoSim (referred to as *baseline*). Employing spatial parallelism, the time-to-solution can be reduced (see Figure 5.36 and Figure 5.37) to 55.89 s on LEAD (speedup: 9.36x) and 42.82 s on TheoSim (speedup: 7.76x), using 16 processors.

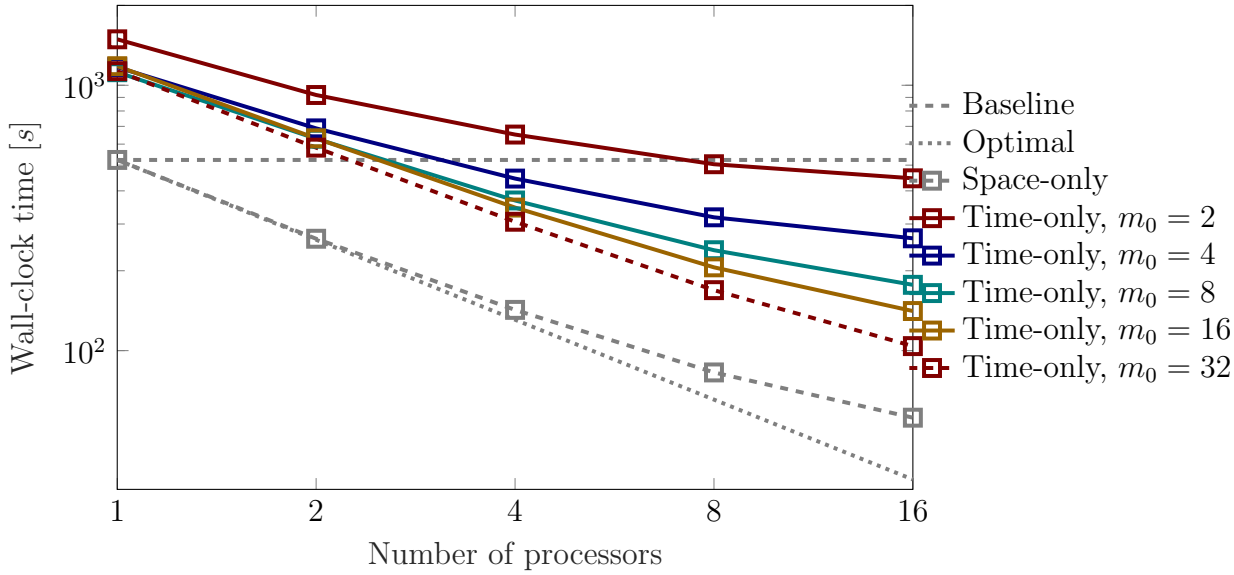


Figure 5.36: *Runtimes on LEAD for two-level convergence of MGRIT with F-relaxation and temporal coarsening factor m_0 for the time-periodic Stokes problem: IC tol is 10^{-10}. Speedup with time-only parallelism is* 5.03 *with* 16 *processors.*

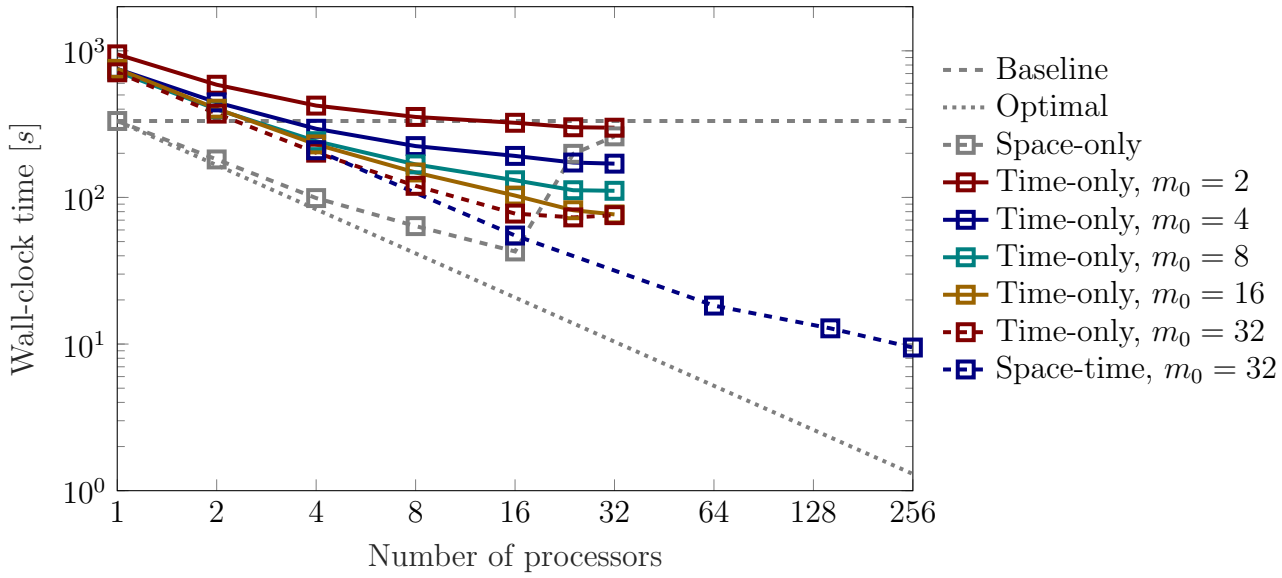


Figure 5.37: *Runtimes on TheoSim for two-level convergence of MGRIT with F-relaxation and temporal coarsening factor m_0 for the time-periodic Stokes problem: IC tol is 10^{-10}. Speedup with time-only parallelism is* 4.54 *with* 24 *processors. Speedup with space-time parallelism is* 35.07 *with* 256 *processors.*

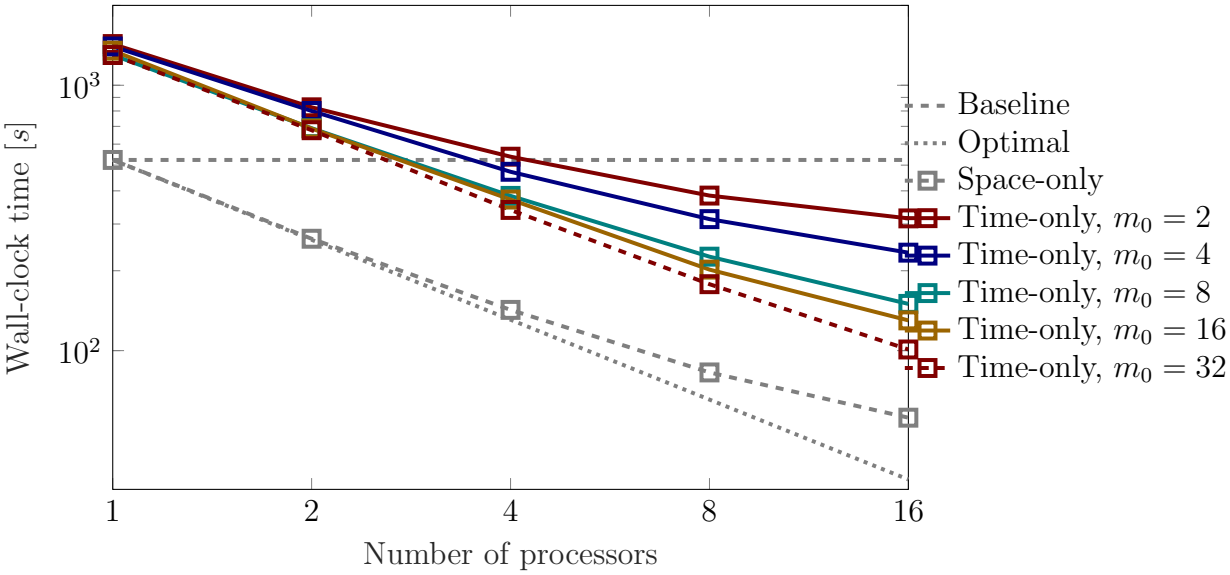


Figure 5.38: *Runtimes on LEAD for two-level convergence of MGRIT with FCF-relaxation and temporal coarsening factor m_0 for the time-periodic Stokes problem: IC tol is 10^{-10}. Speedup with time-only parallelism is* 5.18 *with* 16 *processors.*

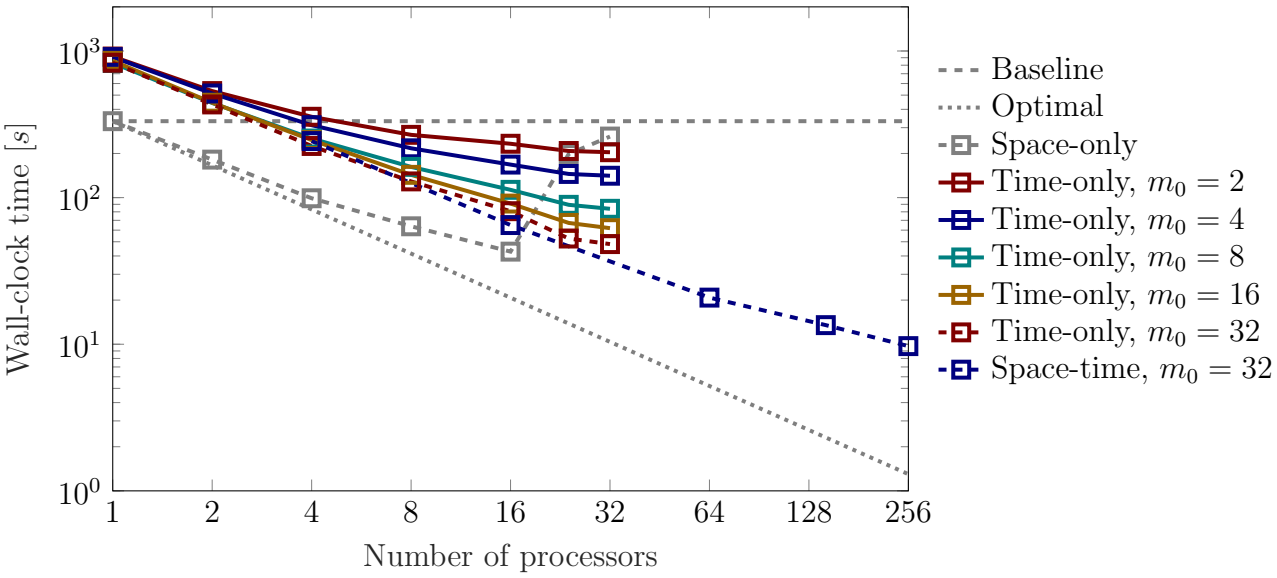


Figure 5.39: *Runtimes on TheoSim for two-level convergence of MGRIT with FCF-relaxation and temporal coarsening factor m_0 for the time-periodic Stokes problem: IC tol is 10^{-10}. Speedup with time-only parallelism is* 6.89 *with* 32 *processors. Speedup with space-time parallelism is* 34.21 *with* 256 *processors.*

Using two-level MGRIT without parallelism yields a more expensive algorithm, e.g. see Figure 5.36. The benefit of MGRIT is exploited by using more processors and by coarsening more aggressively. For example, the wall-clock time of MGRIT with F-relaxation and time-parallelism on LEAD is:

- 446 s; using a temporal coarsening factor of 2 and 16 processors yields a speedup of 1.17x.

- 169 s; using a temporal coarsening factor of 32 and 4 processors yields a speedup of 3.09x.
- 104 s; using a temporal coarsening factor of 32 and 16 processors yields a speedup of 5.03x.

With time-parallelism, the largest speedups achieved are:

- F-relaxation:

  LEAD: 5.03x with a wall-clock time of 104 s using 16 processors.

  TheoSim: 4.54x with a wall-clock time of 73.1 s using 24 processors.

- FCF-relaxation:

  LEAD: 5.18x with a wall-clock time of 101 s using 16 processors.

  TheoSim: 6.89x with a wall-clock time of 48.2 s using 32 processors.

Thanks to the nonintrusiveness of MGRIT, speedups from space- and time-parallelism can be combined. Again, the best speedups are achieved when using more processors and when coarsening more aggressively, e.g., see Figure 5.37. The best achieved speedups using space-time parallelism on TheoSim are:

- F-relaxation: 35.07x with a wall-clock time of 9.47 s using 256 processors.
- FCF-relaxation: 34.21x with a wall-clock time of 9.71 s using 256 processors.

From the data reported here, it can be concluded that for time-only parallelism, FCF-relaxation gives a better speedup than F-relaxation, however, F-relaxation gives a better speedup if space-time parallelism is employed. It seems likely, though, that FCF-relaxation should give the best possible speedup. For example, for F-relaxation the combined space-only speedup (7.76x) and time-only speedup (4.54x) of approximately 35.23x is very close to the observed speedup of 35.07x. On the other hand, for FCF-relaxation there is some mismatch between the combined space-only speedup (7.76x) and time-only speedup (6.89x) of approximately 53.47x and the observed speedup of 34.21x. As the employed nodes of the TheoSim cluster share the network connection with other compute nodes, other production runs may have slowed down MGRIT's MPI messages.

Overall, the achieved speedups for two-level MGRIT are quite large compared to results reported in literature. For example, for a 2D compressible fluid dynamics application Falgout et al. [31] reported a speedup of 7.53x using time-only parallelism on 4096 processors, i.e. on a considerably larger number of processors. While the setting is slightly different (e.g., different Re number, different space-time resolution, etc.), the increased parallel efficiency is arguably achieved through exploiting the periodicity of the solution in time, and thus, to the development of the new time-periodic MGRIT variant.

5.4.6.2 Multilevel MGRIT with F- and FCF-relaxation

In the previous section, the time-periodic MGRIT algorithm from Section 3.6 was explored in the two-level setting. The largest observed speedup was 6.89x with time-parallelism and 35.07x with space-time parallelism, using 16 and 256 processors, respectively. Motivated

from these results, it is investigated if the time-periodic two-level MGRIT algorithm generalizes to the multilevel setting without further modifications, i.e. if it converges, how fast it converges and if the theoretical estimates for the nonperiodic case are useful to estimate convergence in the time-periodic case. Here, only time-parallelism is considered with a fixed temporal coarsening factor of $m_\ell = 2$ for $\ell = 0, \ldots, n_\ell - 2$. Results reported were obtained on the TheoSim cluster, see Section A.

Convergence estimates

First, convergence estimates for the nonperiodic case are compared with observed worst-case convergence of time-periodic multilevel MGRIT V- and F-cycle algorithms. Figure 5.40 and Figure 5.41 illustrate that in the case of multilevel V- and F-cycle algorithms with F-relaxation, the theoretical estimates (inequality bound and Approximation 1) capture the correct magnitude and trend of the observed worst-case convergence factors. In the case of FCF-relaxation, however, the observed worst-case convergence factors cannot be predicted (V-cycle) or only poorly (correct trend but incorrect magnitude for multilevel F-cycle).

In fact, it is quite surprising that one observes better (worst-case) convergence for two-level MGRIT with FCF-relaxation when the temporal coarsening increases. This is quite unusual but very beneficial, since it yields a better converging algorithm with cheaper per-iteration cost. In Section 5.4.6.1, it was observed that FCF-relaxation in combination with a large temporal coarsening factor can indeed yield faster (worst-case) convergence than with a small temporal coarsening factor. Figure 5.40 illustrates a similar observation for the multilevel case: the worst-case convergence factor for multilevel V-cycles with FCF-relaxation decreases with an increasing number of time grid levels; thus, at the same time yielding a cheaper per-iteration cost, more potential for parallelism and a smaller coarse-grid size (i.e. smaller sequential component).

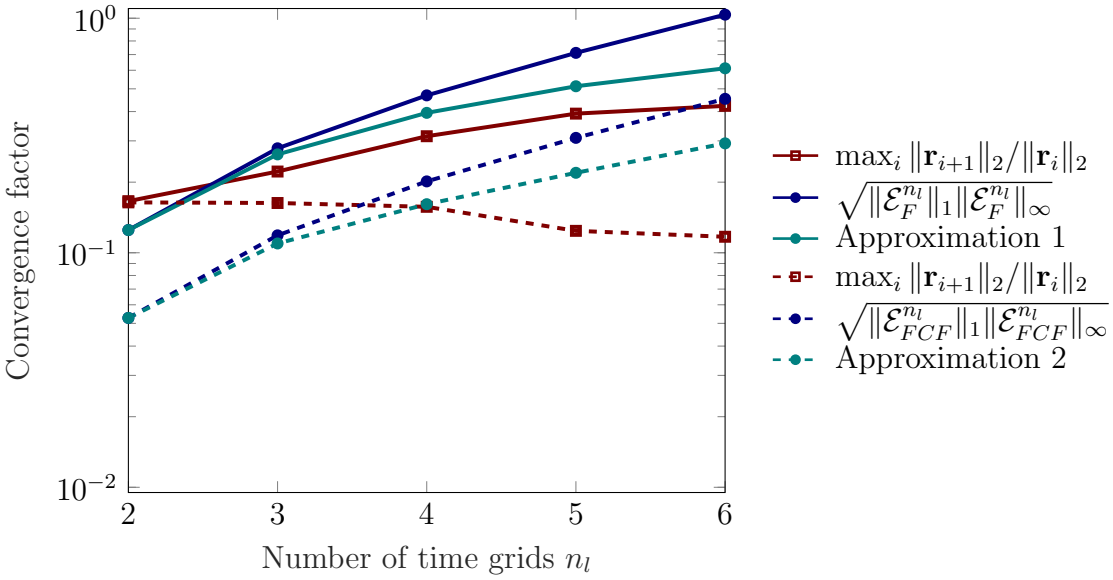


Figure 5.40: *Multilevel convergence of MGRIT V-cycles with F- and FCF-relaxation for the Stokes problem: Observed worst-case convergence factor (for time-periodic problem) compared with a priori estimates (for one period; no p-contribution).*

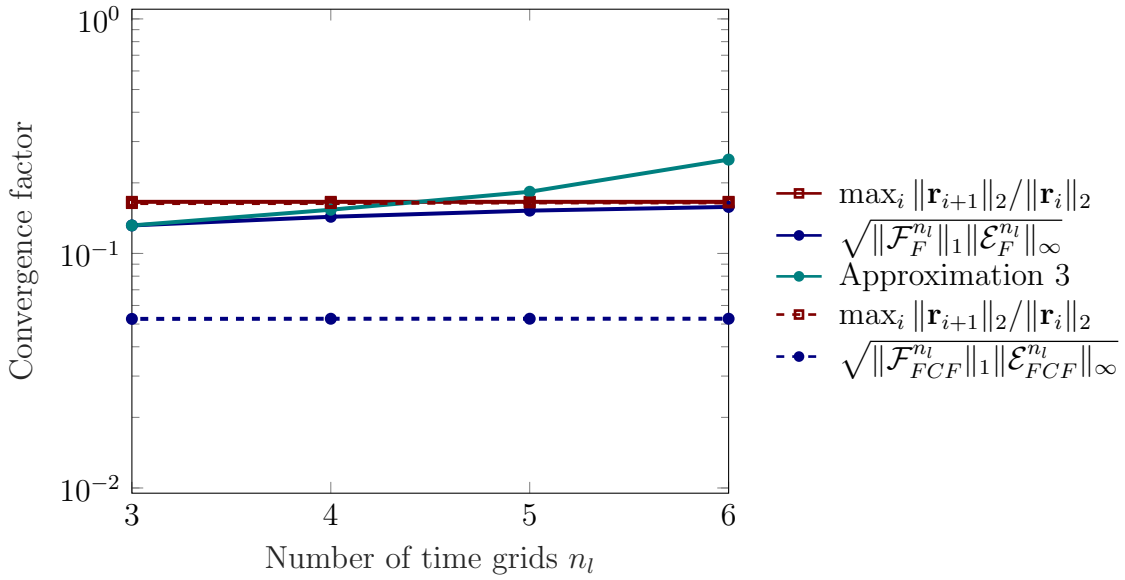


Figure 5.41: *Multilevel convergence of MGRIT F-cycles with F- and FCF-relaxation for the Stokes problem: Observed worst-case convergence factor (for time-periodic problem) compared with a priori estimates (for one period; no p-contribution).*

Parallel performance on TheoSim

Starting with the two-level MGRIT algorithm, Figure 5.42 and Figure 5.43 highlight that the degree of parallelism that can be exploited through MGRIT eventually saturates, giving only negligible speedups (for both F- and FCF-relaxation). On the other hand, using more time grid levels yields a more expensive algorithm than serial time stepping if only one or few processors are used to parallelize the time component. However, the benefit of using time-periodic multilevel MGRIT with F- or FCF-relaxation is bigger when employing more processors. For example, the slope for $n_\ell = 6$ time grid levels for MGRIT with FCF-relaxation is much steeper than for two-level MGRIT with FCF-relaxation (see Figure 5.42). The best speedups are achieved for multilevel MGRIT with $n_\ell = 6$ time grid levels using 32 processors: the speedup is 1.90x for F-relaxation and 3.48x for FCF-relaxation.

The fact that the wall-clock time of six-level MGRIT with FCF-relaxation is smaller than six-level MGRIT with F-relaxation reflects and confirms the observations in Figure 5.40: the cheaper per-iteration cost and smaller coarse grid size yield a larger speedup.

5.4.7 Weak form of the Navier-Stokes flow problem

Now, let's switch from the linear PDE to the nonlinear PDE case: Consider the Navier-Stokes problem,

$$\rho_f \partial_t \boldsymbol{v}_f + \rho_f \boldsymbol{v}_f \cdot \nabla \boldsymbol{v}_f - \nabla \cdot \left[\mu_f \left(\nabla \boldsymbol{v}_f + \nabla^T \boldsymbol{v}_f - p_f \boldsymbol{I}\right)\right] = 0 \qquad \text{in } \Omega_f, \tag{5.55}$$

$$\nabla \cdot \boldsymbol{v}_f = 0 \qquad \text{in } \Omega_f, \tag{5.56}$$

which is discretized using finite elements in space and backward Euler in time, see Section 5.4.1. The general dxiscrete weak form can be written as follows:

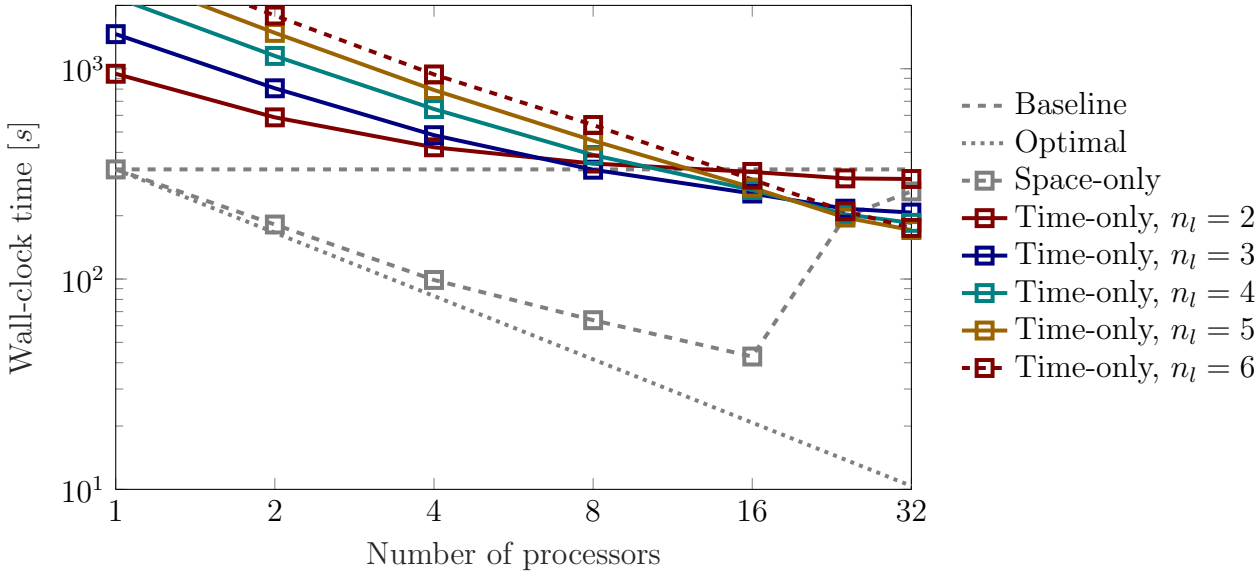


Figure 5.42: *Runtimes on TheoSim for n_l-level convergence of MGRIT with F-relaxation and temporal coarsening factor $m_l = 2$ for the time-periodic Stokes problem: IC tol is 10^{-10}. Speedup with time-only parallelism is 1.90 with $n_l = 6$ levels and 32 processors.*

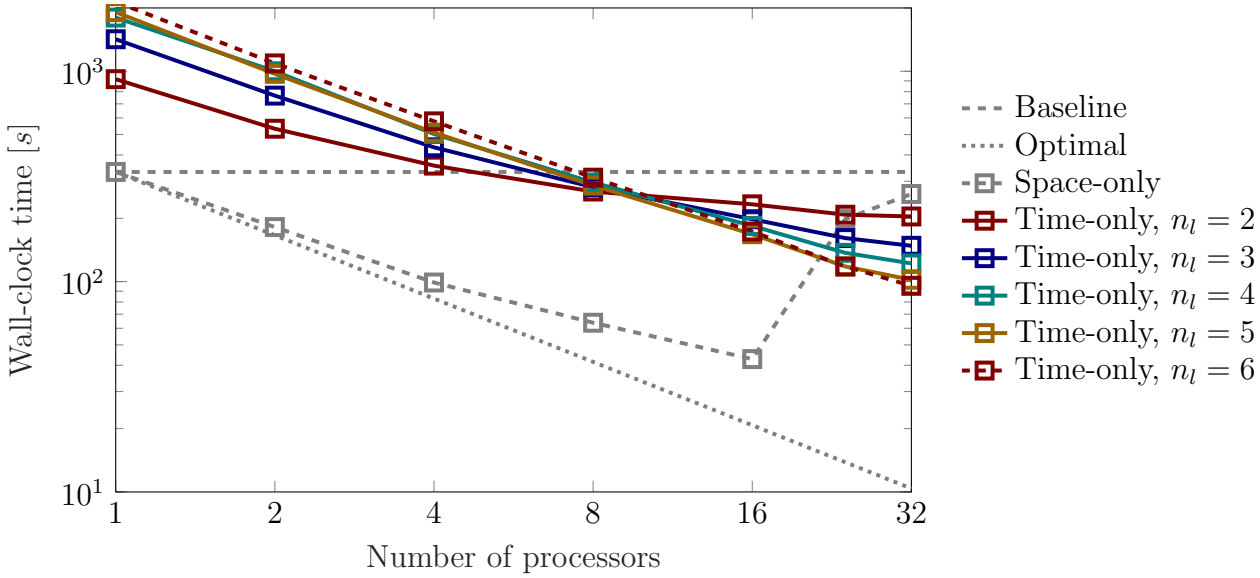


Figure 5.43: *Runtimes on TheoSim for n_l-level convergence of MGRIT with FCF-relaxation and temporal coarsening factor $m_l = 2$ for the time-periodic Stokes problem: IC tol is 10^{-10}. Speedup with time-only parallelism is 3.48 with $n_l = 6$ levels and 32 processors.*

Find $\boldsymbol{s}^{n+1} := (\boldsymbol{v}_f^{n+1}, p_f^{n+1}) \in \boldsymbol{\mathcal{V}}_D^h \times \mathcal{W}_f^h$, such that for every $\boldsymbol{d} := (\boldsymbol{w}_f, q_f) \in \boldsymbol{\mathcal{V}}_0^h \times \mathcal{W}_f^h$:

$$\begin{aligned} R(\boldsymbol{s}^{n+1}, \boldsymbol{s}^n, \boldsymbol{d}) = &\int_{\Omega_f^h} \rho_f \left[\frac{\boldsymbol{v}_f^{n+1} - \boldsymbol{v}_f^n}{\delta_t} + \boldsymbol{v}_f^{n+1} \cdot \nabla \boldsymbol{v}_f^{n+1} \right] \cdot \boldsymbol{w}_f \, d\Omega_f^h \\ &+ \int_{\Omega_f^h} \mu_f \left[\nabla \boldsymbol{v}_f^{n+1} + \nabla^T \boldsymbol{v}_f^{n+1} - p_f^{n+1} \boldsymbol{I} \right] : \nabla \boldsymbol{w}_f + q_f \nabla \cdot \boldsymbol{v}_f^{n+1} \, d\Omega_f^h = 0. \end{aligned} \tag{5.57}$$

The definition of the function spaces are:

$$S^p(\Omega_f^h) = \{f : \Omega_f^h \to \mathbb{R} \mid f \in \mathcal{C}^0(\bar{\Omega}_f^h), f|_{\tau_e} \in \mathbb{P}^p(\tau_e), \forall\ \tau_e \in T_f^h\}, \tag{5.58}$$

which represent the general continuous p^{th}-order piecewise polynomial spaces defined on Ω_f^h. Consequently, we define:

$$\boldsymbol{\mathcal{V}}^h = [S^2(\Omega_f^h)]^2 \qquad \text{and} \qquad \mathcal{W}_f = S^1(\Omega_f^h), \tag{5.59}$$

and incorporating the boundary conditions:

$$\boldsymbol{\mathcal{V}}_0^h = \{\boldsymbol{v} \in \boldsymbol{\mathcal{V}}^h \mid [\boldsymbol{v}]_y = 0 \text{ on } \Gamma_f^O\}, \tag{5.60}$$

$$\boldsymbol{\mathcal{V}}_D^h = \{\boldsymbol{v} \in \boldsymbol{\mathcal{V}}^h \mid \boldsymbol{v} = \boldsymbol{v}_f \text{ according to (5.41) and (5.42)}\}. \tag{5.61}$$

Rewriting Equation (5.57) in operator form yields,

$$\begin{bmatrix} \rho_f \boldsymbol{M} + \mu_f \delta_t \boldsymbol{K}(\boldsymbol{v}_f^{n+1}) & \delta_t \boldsymbol{B}^T \\ \delta_t \boldsymbol{B} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_f^{n+1} \\ p_f^{n+1} \end{bmatrix} = \begin{bmatrix} \rho_f \boldsymbol{M} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_f^{n} \\ p_f^{n} \end{bmatrix}, \tag{5.62}$$

where $\boldsymbol{M}$ is the mass matrix, and $\boldsymbol{K}$ and $\boldsymbol{B}$ refer to the discretized weak form Laplacian and divergence operators $\nabla^2()$ and $\nabla \cdot ()$, respectively. Rewriting (5.62) in $\boldsymbol{\Phi}$-form,

$$\begin{bmatrix} \boldsymbol{v}_f^{n+1} \\ p_f^{n+1} \end{bmatrix} = \underbrace{\begin{bmatrix} \rho_f \boldsymbol{M} + \mu_f \delta_t \boldsymbol{K}(\boldsymbol{v}_f^{n+1}) & \delta_t \boldsymbol{B}^T \\ \delta_t \boldsymbol{B} & \boldsymbol{0} \end{bmatrix}^{-1} \begin{bmatrix} \rho_f \boldsymbol{M} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}}_{\boldsymbol{\Phi} \approx \boldsymbol{\Phi}^{n+1}} \begin{bmatrix} \boldsymbol{v}_f^{n} \\ p_f^{n} \end{bmatrix}, \tag{5.63}$$

highlights the same sparsity pattern as in Equation (5.53), such that the convergence analysis can be restricted to the velocity block, see Section 5.4.4.

5.4.8 Multilevel MGRIT for the Navier-Stokes flow problem

The convergence framework presented in Chapter 4 was developed under the assumption of linear PDEs. In practice, an overwhelming number of applications, however, require the solution of nonlinear PDEs. It is thus important to investigate, how and if the estimates for the linear case are applicable to the nonlinear case. Here, two approaches are considered:

1. Convergence estimate based on $\boldsymbol{\Phi}$ from the Stokes equation, see Equation (5.52).

2. Convergence estimate based on a linearized $\widetilde{\boldsymbol{\Phi}}$ from a Newton iteration of the Navier-Stokes equation, see Equation (5.63).

Here, multilevel MGRIT V- and F-cycles with F- and FCF-relaxation are considered for $n_\ell \in \{2, 3, 4, 5, 6\}$ levels with temporal coarsening factor $m_\ell = 2$ for $\ell = 0, \ldots, n_\ell - 2$.

5.4.8.1 Convergence estimate based on the Stokes flow problem

The first approach to estimate convergence of MGRIT for the nonlinear case is motivated from the observation that the nonperiodic case provided reasonable estimates for time-periodic MGRIT. Thus, convergence estimates for the Stokes equation are compared with observed worst-case convergence factors of MGRIT for the parallel-in-time integration of the Navier-Stokes equation.

Figure 5.44 and Figure 5.45 highlight that the observed worst-case convergence factor for MGRIT for the Navier-Stokes equation is larger than for the Stokes equation. This makes sense, in that the presence of the nonlinear advective term changes the character of the PDE toward a regime that has traditionally experienced more challenges for parallel-in-time integration: the field of hyperbolic PDEs. It is further noted, that while the trends can still be predicted reasonably well, there is a visible mismatch between the predicted and observed magnitude of the convergence factor, e.g., see Figure 5.44.

Thus, the estimates for the Stokes equation do not directly translate to the nonlinear case, e.g., because the advective term is ignored. In the following section, a different approach is taken that takes the advective term into account to provide better estimates of the observed worst-case convergence of multilevel MGRIT.

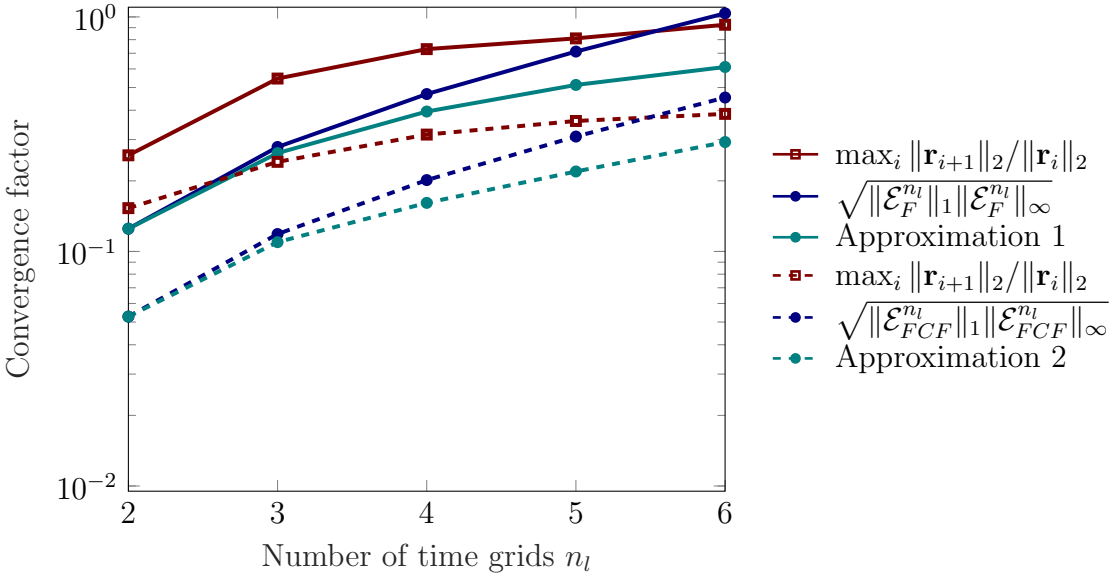


Figure 5.44: *Multilevel convergence of MGRIT V-cycles with F- and FCF-relaxation for the Navier-Stokes problem: Observed worst-case convergence factor compared with a priori estimates based on Stokes flow.*

5.4.8.2 Convergence estimate based on a linearization of the Navier-Stokes flow problem

In Section 5.4.8.1, it was shown that convergence estimates for the linear PDE case do not necessarily translate to the nonlinear PDE case directly. Here, an alternative approach is proposed. Instead of ignoring the role of the advective term in the convergence framework, it is taken into account by computing a linearized version of the time stepping operator $\mathbf{\Phi}_\ell$ on each level ℓ during a Newton iteration of the spatial solve. Since the magnitude

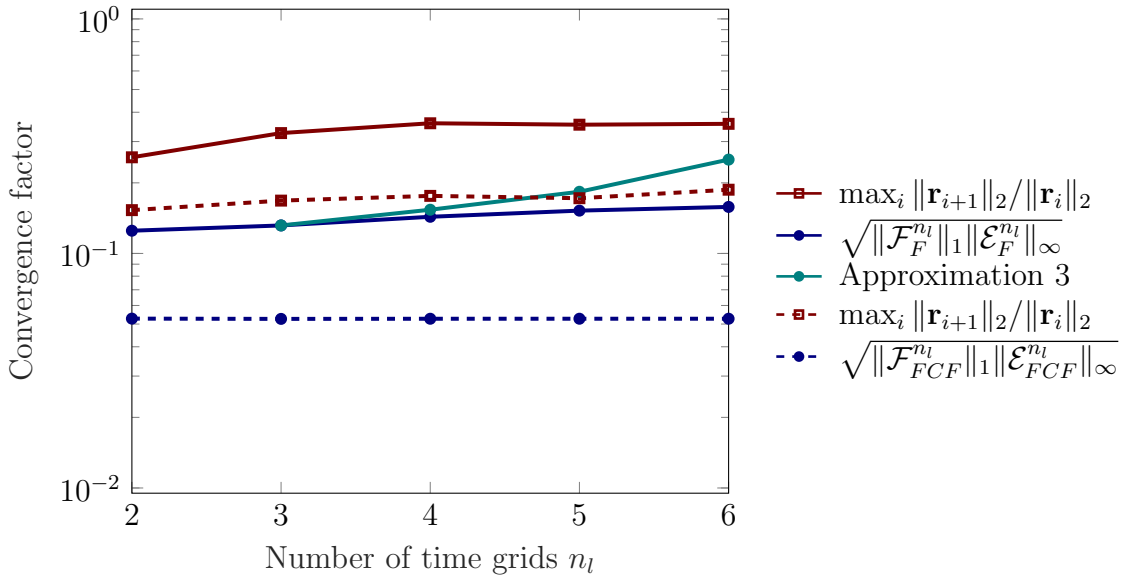


Figure 5.45: *Multilevel convergence of MGRIT F-cycles with F- and FCF-relaxation for the Navier-Stokes problem: Observed worst-case convergence factor compared with a priori estimates based on Stokes flow.*

of the advective term constantly changes throughout the temporal domain, the flow field $\boldsymbol{v}_f$ at $t = 0.384$ s is selected to compute the linearized $\boldsymbol{\Phi}_\ell$ operator on each level ℓ. This time point corresponds to a snapshot of the space-time solution (row 2 in Figure 5.27 and Figure 5.28), where advection is expected to play a significant role.

The convergence analysis proceeds as before, however, the assumption of simultaneous diagonalization is somewhat modified. For example, the linearized time stepping operator $\boldsymbol{\Phi}_0$ on level 0 is diagonalized by,

$$\boldsymbol{U}^{-1}\boldsymbol{\Phi}_0\boldsymbol{U} = \mathrm{diag}(\lambda_{0,1}, \ldots, \lambda_{0,N_x}), \tag{5.64}$$

however, for all other levels $\ell = 1, \ldots, n_\ell - 1$, let,

$$[\lambda_{\ell,1}, \ldots, \lambda_{\ell,N_x}]^T := \mathrm{diag}(\boldsymbol{U}^{-1}\boldsymbol{\Phi}_\ell\boldsymbol{U})\,. \tag{5.65}$$

When computing $\boldsymbol{U}^{-1}\boldsymbol{\Phi}_\ell\boldsymbol{U}$ numerically, there are slight nonzero offdiagonal entries. These nonzero entries could be present due to limited precision in numerical arithmetic and / or bad conditioning. It is, however, difficult to exactly explain this observation without extensive further research, e.g., by deriving the respective Fourier symbols. Deriving the Fourier symbols is straightforward to do for simple linear PDEs and a fixed spatiotemporal discretization scheme (e.g., see [24]). But it is not clear if it is indeed possible for the (nonlinear) Navier-Stokes equation. Despite the potential violation of the simultaneously diagonalizable assumption, we proceed with the eigenvalues $\lambda_{\ell,k}$ (for $\ell \in \{0, \ldots, n_\ell - 1\}$, $k \in \{1, \ldots, N_x\}$) obtained from Equation (5.64) and Equation (5.65) for two reasons:

- The simultaneously diagonalizable assumption is potentially not violated.
- If it is violated, the approach could still hint at a possible pathway for a future generalization of the convergence framework for nonlinear PDEs.

Figure 5.46 and Figure 5.47 illustrate the benefit of the proposed approach. While not mathematically rigorous, the obtained estimates of the worst-case convergence factor are strikingly close to observed worst-case convergence of MGRIT V- and F-cycles for the parallel-in-time integration of the Navier-Stokes equation. The estimates predict the correct trend and the correct magnitude with surprising sharpness. Irrespective of the use of V- or F-cycles and F- or FCF-relaxation: The observed worst-case convergence can be predicted with good quality.

Thus, the observed slightly nonzero offdiagonal entries of $\boldsymbol{U}^{-1}\boldsymbol{\Phi}_\ell\boldsymbol{U}$ in Equation (5.65) could indeed stem from the limited precision of numerical arithmetic and / or bad conditioning. As a matter of course, this observation needs to be confirmed in further studies, however, the proposed approach could possibly be the basis for future generalizations of the theory in Chapter 4 to the nonlinear PDE case.

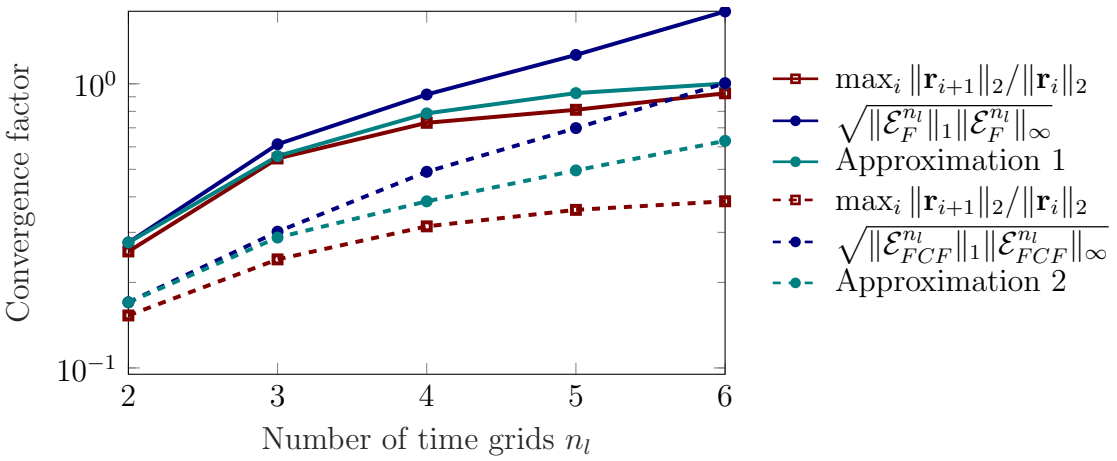


Figure 5.46: *Multilevel convergence of MGRIT V-cycles with F- and FCF-relaxation for the Navier-Stokes problem at $t = 0.384$ s: Observed worst-case convergence factor compared with a priori estimates based on a linearization of the $\boldsymbol{\Phi}$ operator.*

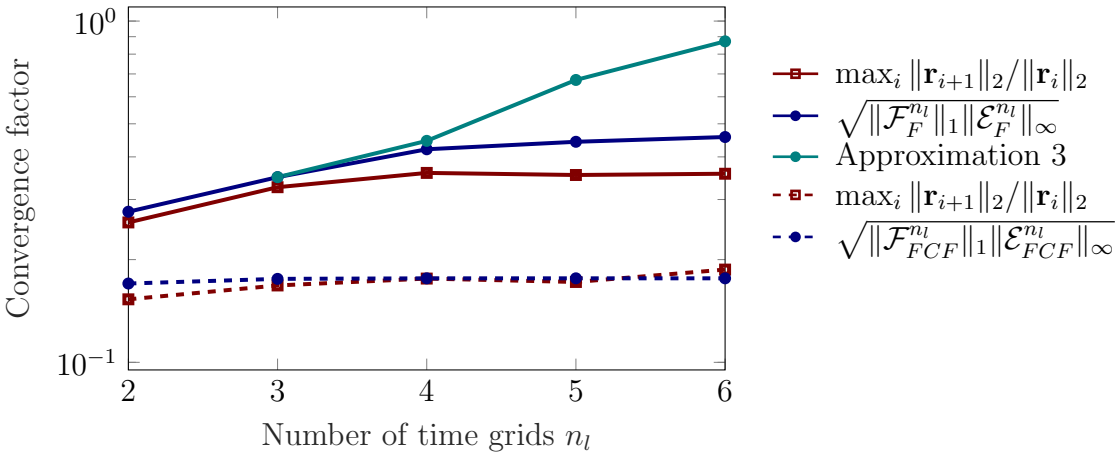


Figure 5.47: *Multilevel convergence of MGRIT F-cycles with F- and FCF-relaxation for the Navier-Stokes problem at $t = 0.384$ s: Observed worst-case convergence factor compared with a priori estimates based on a linearization of the $\boldsymbol{\Phi}$ operator.*

5.4.9 Conclusion

In this section, flow through a simplified stenosed valve geometry was investigated both from a theoretical perspective and in numerical experiments. Overall, theoretical estimates predicted worst case convergence in numerical experiments well, both in the linear PDE case and for a proposed extension to the nonlinear PDE case.

Significant speedups were achieved by exploiting the time-periodic property of the flow problem by means of the new time-periodic MGRIT algorithm proposed in Section 3.6.

5.5 Analytic FSI solutions: Transient fluid / transient linear solid in 2D[10]

In this section, MGRIT is applied to the transient linear two-dimensional FSI test case from Chapter 2. The domains are given in Figure 2.1 and parameters are selected similar to Section 2.4.2.1, see Table 5.2. Further, four space-time discretizations are considered and referred to as *refinement levels*. Each space-time refinement level has an MGRIT tolerance associated with it, see Table 5.3.

In the following, the **Φ**-form of the space-time discrete incompressible Stokes equation is derived. Then, the multilevel convergence framework from Chapter 4 is applied and the theoretical estimates are compared with observed convergence factors. Furthermore, the new time-periodic MGRIT algorithm (see Section 3.6), that has proven beneficial in Section 5.4, is assessed, where significant speedups are achieved.

Fluid density	Fluid viscosity	Solid density	Solid stiffness	Cycle length
$\rho_f = 1.000$	$\mu_f = 0.010$	$\rho_s = 1.000$	$\mu_s = 0.100$	$T = 1.024$

Table 5.2: *Parameters for the 2D transient fluid / transient linear solid case.*

Refinement level	Time step size	# Time steps	Spatial step size	# DOFs	MGRIT tol
1	0.128	80	0.200	378	$1.398 \cdot 10^{-7}$
2	0.032	320	0.100	1288	$5.590 \cdot 10^{-7}$
3	0.008	1280	0.050	4728	$2.236 \cdot 10^{-6}$
4	0.002	5120	0.025	18088	$8.944 \cdot 10^{-6}$

Table 5.3: *Space-time refinement levels used for 2D transient fluid / transient linear solid case with $\delta_t/\delta_x^2 = 3.2 = const$ and scaled MGRIT tolerance $10^{-8}/(\sqrt{\delta_t}\delta_x)$.*

5.5.1 Φ-form of coupled PDEs

In this section, first-order backward Euler time-discretization schemes are considered for the fluid and solid PDEs along with inf-sup stable quadratic-linear Taylor-Hood finite elements, similar to Chapter 2, Section 5.3 and Section 5.4.

The backward Euler time-discretization of the incompressible Stokes equation is given as,

$$\rho_f \boldsymbol{v}_f^{n+1} - \delta_t \mu_f \nabla \cdot \left[\nabla \boldsymbol{v}_f^{n+1} + \nabla^T \boldsymbol{v}_f^{n+1}\right] + \delta_t \nabla \cdot (p_f^{n+1} \boldsymbol{I}) = \rho_f \boldsymbol{v}_f^n, \tag{5.66}$$

$$-\delta_t \nabla \cdot \boldsymbol{v}_f^{n+1} = 0. \tag{5.67}$$

On the other hand, the backward Euler time discretization of the linear elasticity equa-

[10] Results in this section have previously appeared [62]

tion is given as Scheme II from Section 5.3,

$$\boldsymbol{u}_s^{n+1} - \delta_t \boldsymbol{v}_s^{n+1} = \boldsymbol{u}_s^n, \tag{5.68}$$

$$\rho_s \boldsymbol{v}_s^{n+1} - \delta_t^2 \mu_s \nabla \cdot \left[\nabla \boldsymbol{v}_s^{n+1} + \nabla^T \boldsymbol{v}_s^{n+1}\right] \ldots$$
$$\ldots + \delta_t \nabla \cdot (p_s^{n+1} \boldsymbol{I}) = \rho_s \boldsymbol{v}_s^n + \mu_s \delta_t \nabla \cdot \left[\nabla \boldsymbol{u}_s^n + \nabla^T \boldsymbol{u}_s^n\right], \tag{5.69}$$

$$-\delta_t^2 \nabla \cdot \boldsymbol{v}_s^{n+1} = 0, \tag{5.70}$$

where the symmetric Cauchy stress tensor[11] is considered to govern the stress response of a linear-elastic incompressible, isotropic and homogeneous solid material.

The dynamic and kinematic interface constraints,

$$\boldsymbol{t}_f^{n+1} + \boldsymbol{t}_s^{n+1} = \boldsymbol{0} \tag{5.71}$$

$$\boldsymbol{v}_f^{n+1} - \boldsymbol{v}_s^{n+1} = \boldsymbol{0} \tag{5.72}$$

are enforced at the coupling surface, with equal but opposite tractions,

$$\boldsymbol{t}_f = \left[\nabla \boldsymbol{v}_f + \nabla^T \boldsymbol{v}_f\right] \cdot \boldsymbol{n}_f, \tag{5.73}$$

$$\boldsymbol{t}_s = \left[\nabla \boldsymbol{u}_s + \nabla^T \boldsymbol{u}_s\right] \cdot \boldsymbol{n}_s, \tag{5.74}$$

where $\boldsymbol{n}_f$ and $\boldsymbol{n}_s$ are the outer normal directions on the fluid and solid coupling boundaries. The constraints are weakly enforced by introducing a Lagrange multiplier variable, $\boldsymbol{\lambda} = \boldsymbol{t}_f = -\boldsymbol{t}_s$.

Here, inf-sup stable Taylor-Hood finite elements are employed to discretize the coupled PDEs in space. Quadratic interpolation is used for the fluid velocity, the solid velocity and displacement, and the Lagrange multiplier; linear interpolation is used for the fluid and solid pressure. For conciseness and to avoid redundancy we skip over the weak form of the coupled FSI problem, which is given in general form in Section 2.3.1. In matrix form, we can write,

$$\begin{aligned}
\left(\rho_f M_f + \mu_f \delta_t K_f\right) \boldsymbol{v}_f^{n+1} + \delta_t B_f^T \boldsymbol{p}_f^{n+1} + C_f^T \boldsymbol{\lambda}^{n+1} &= \rho_f M_f \boldsymbol{v}_f^n, \\
\delta_t B_f \boldsymbol{v}_f^{n+1} &= 0, \\
\boldsymbol{u}_s^{n+1} - \delta_t \boldsymbol{v}_s^{n+1} &= \boldsymbol{u}_s^n, \\
\left(\rho_s M_s + \mu_s \delta_t^2 K_s\right) \boldsymbol{v}_s^{n+1} + \delta_t B_s^T \boldsymbol{p}_s^{n+1} - C_s^T \lambda_i^{n+1} &= \rho_s M_s \boldsymbol{v}_s^n - \mu_s \delta_t K_s \boldsymbol{u}_s^n, \\
\delta_t B_s \boldsymbol{v}_s^{n+1} &= 0, \\
C_f \boldsymbol{v}_f^{n+1} - C_s \boldsymbol{v}_s^{n+1} &= 0,
\end{aligned}$$

where M_f and M_s are the fluid and solid mass matrix, and K_f and K_s and B_f and B_s refer to the discretized weak form Laplacian and divergence operators $\nabla^2()$ and $\nabla \cdot ()$,

[11] Note, that the asymmetric Cauchy stress tensor was considered in Section 5.3 and [66].

respectively. In matrix form, we can write,

$$\begin{bmatrix} \rho_f M_f + \mu_f \Delta_t K_f & \Delta_t B_f^T & & & C_f^T \\ \Delta_t B_f & & & & \\ & I & -\Delta_t I & & \\ & & \rho_s M_s + \mu_s \Delta_t^2 K_s & \Delta_t B_s^T & -C_s^T \\ & & \Delta_t B_s & & \\ C_f & & -C_s & & \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_f^{n+1} \\ \boldsymbol{p}_f^{n+1} \\ \boldsymbol{u}_s^{n+1} \\ \boldsymbol{v}_s^{n+1} \\ \boldsymbol{p}_s^{n+1} \\ \boldsymbol{\lambda}^{n+1} \end{bmatrix}$$
$$= \begin{bmatrix} \rho_f M_f & & & & & \\ & & & & & \\ & & I & & & \\ & & -\mu_s \Delta_t K_s & \rho_s M_s & & \\ & & & & & \\ & & & & & \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_f^n \\ \boldsymbol{p}_f^n \\ \boldsymbol{u}_s^n \\ \boldsymbol{v}_s^n \\ \boldsymbol{p}_s^n \\ \boldsymbol{\lambda}^n \end{bmatrix}. \tag{5.75}$$

With $[\boldsymbol{\mathcal{D}}]^{-1}$ and $\boldsymbol{\mathcal{C}}$ referring to the matrices on the left-hand-side and right-hand-side of the equation, we can compute the evolution matrix as,

$$\boldsymbol{\Phi} = \boldsymbol{\mathcal{D}}\boldsymbol{\mathcal{C}}. \tag{5.76}$$

5.5.2 Simultaneously diagonalized time-stepping operators

Similar to Section 5.3 and [66], $\boldsymbol{\Phi}$ will have zero columns corresponding to the constraint variables, that is, $\boldsymbol{p}_f$ and $\boldsymbol{p}_s$ and $\boldsymbol{\lambda}$. Thus, the convergence analysis can be done on block rows and columns corresponding to $\boldsymbol{v}_f$, $\boldsymbol{u}_s$ and $\boldsymbol{v}_s$.

Considering the operator given in Equation (5.76) in an MGRIT setting, i.e. for various time step sizes, then the time-stepping operators $\{\boldsymbol{\Phi}_\ell\}$ are simultaneously diagonalizable, similar to the fluid and solid subproblems in Section 5.3 and Section 5.4:

$$\boldsymbol{U}^{-1}\boldsymbol{\Phi}_\ell\boldsymbol{U} = \mathrm{diag}(\lambda_{\ell,1}, \ldots, \lambda_{\ell,N_x}), \quad \text{for } \ell = 0. \ldots, n_\ell - 1. \tag{5.77}$$

Figure 5.48 shows the diagonal values $\lambda_{\ell,k}$ for space-time refinement levels $1-4$ (see Table 5.3) and $n_\ell = 7$ time grid levels with temporal coarsening factor 2 between all levels. It is quite clear that the values $\lambda_{\ell,k}$ can be distributed fairly differently in the complex plane, depending on the employed temporal and spatial step sizes.

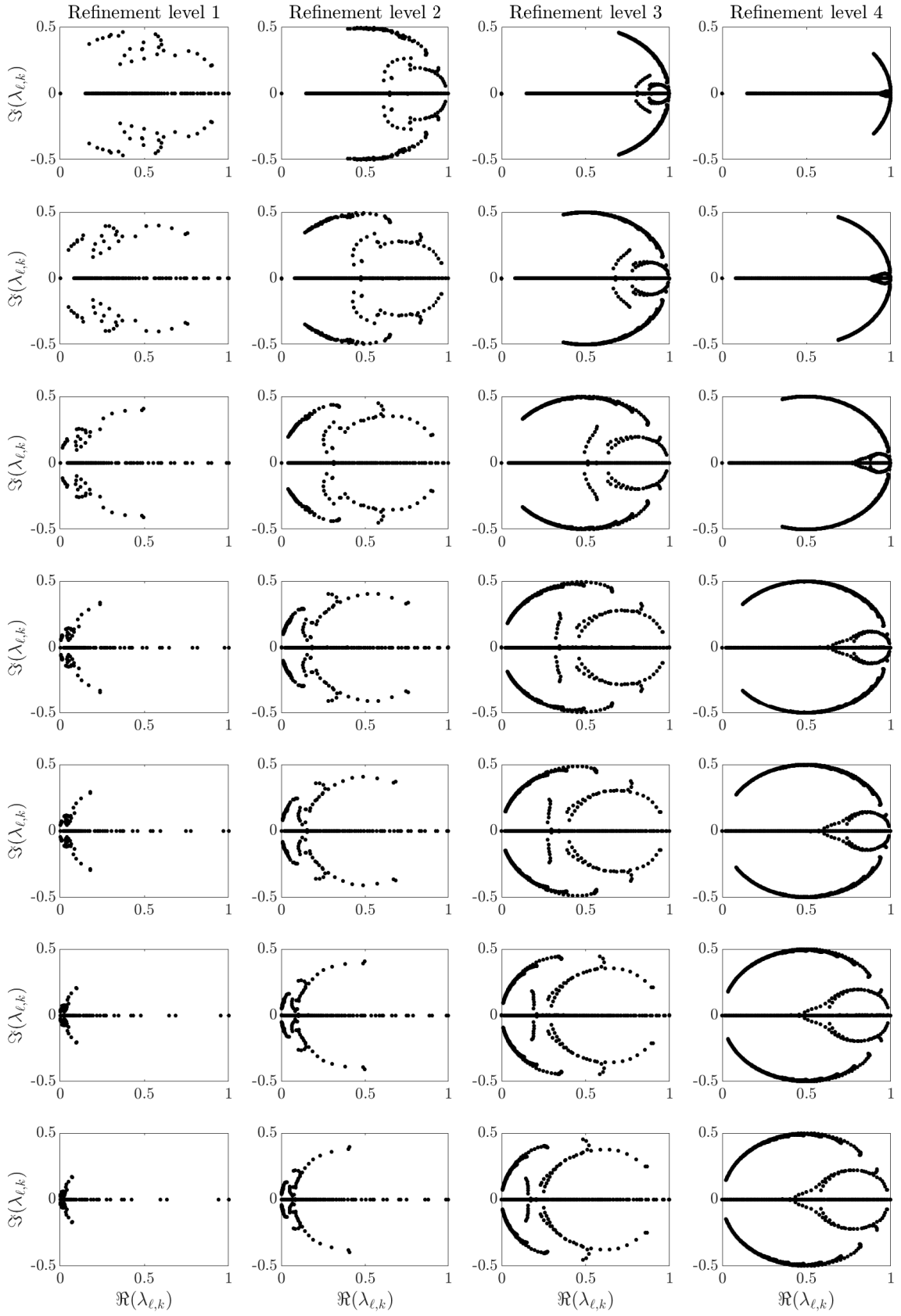


Figure 5.48: *Diagonal values of* $\boldsymbol{U}^{-1}\boldsymbol{\Phi}_\ell\boldsymbol{U}$ *in the complex plane depending on the space-time refinement level (see Table 5.3) and the temporal coarsening factor:* $\{1, 2, 4, \ldots, 64\}$ *from top-to-bottom.*

5.5.3 MGRIT convergence

In literature and the previous sections, it was observed that MGRIT convergence (with some exceptions) deteriorates:

- with more aggressive temporal coarsening,
- and when refining in space-time.

On the other hand, MGRIT V-cycle convergence was found to improve:

- when using a stronger relaxation scheme,
- when switching from V-cycles to F-cycles.

Here, these observations are reviewed for the coupled linear FSI problem; first in a two-level setting and later in the multilevel setting.

5.5.3.1 Two-level MGRIT with rFCF-relaxation

Figure 5.49 - Figure 5.52 show observed worst-case and mean convergence of two-level MGRIT for space-time refinement levels $1-4$, depending on the number of relaxation steps r and the coarsening factor m_0; and in comparison with the estimated upper bound in Theorem 1.

First, we remind the reader that each MGRIT iteration propagates the exact solution over $(r+1)m_0$ time steps, and thus, MGRIT is guarantueed to converge in $N_0/((r+1)m_0)$ iterations. For space-time refinement level 1, this means that MGRIT converges in "less than" one iteration for $r = 4$ and $m_0 = 20$; thus, this data point is excluded in Figure 5.49.

In most cases, the estimated worst-case convergence is bounding the observed worst-case convergence as illustrated in Figure 5.51. For the larger space-time refinement levels, however, observed and estimated convergence factors are a little underwhelming since this setting prohibits the use of larger temporal coarsening factors or requires more relaxation steps to achieve good convergence factors.

It is also noted, that the number of required MGRIT iterations to satisfy the convergence tolerance increases with larger space-time meshes; a property that is not ideal and usually contradicts with the requirements of a scalable solver (e.g., constant iterations / cost under refinement).

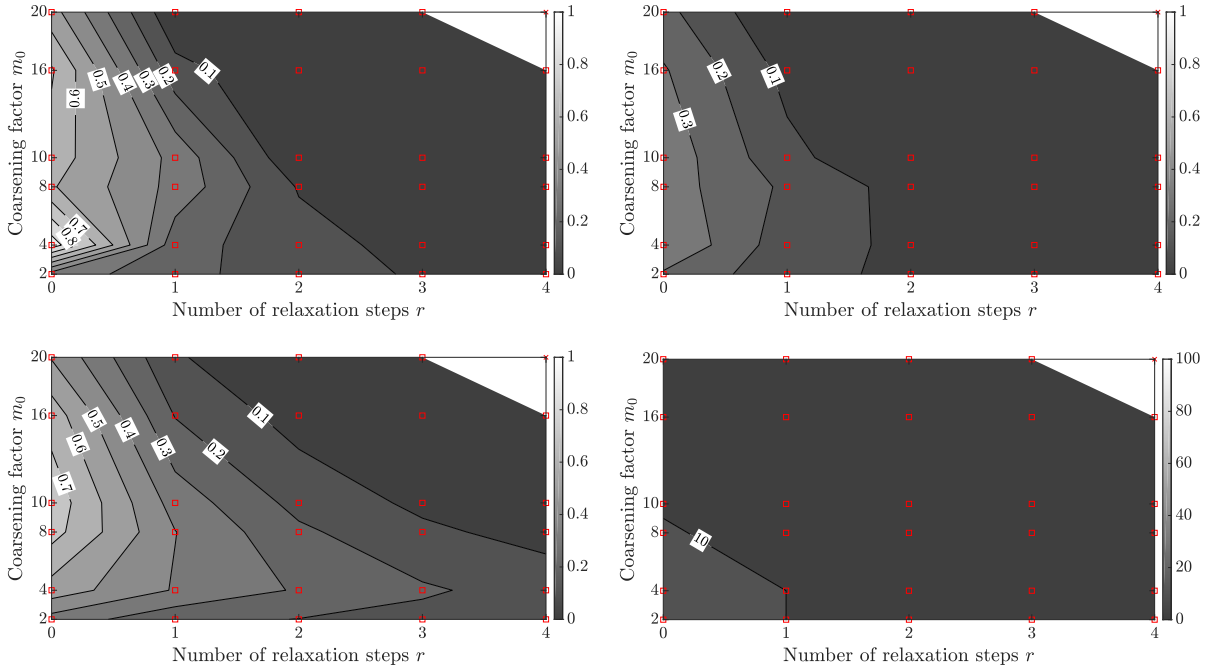


Figure 5.49: *Top: Measured worst-case and mean convergence factor; Bottom: Estimated worst-case and number of iterations. Space-time refinement 1. Note: Red × means convergence in 1 iteration.*

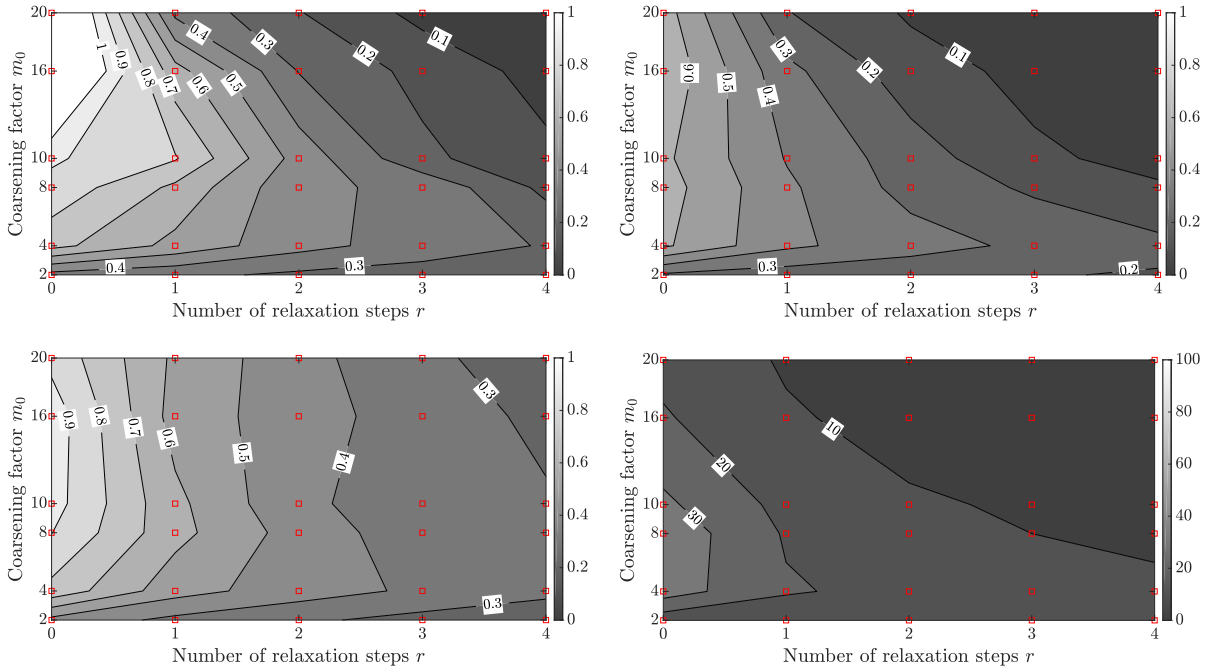


Figure 5.50: *Top: Measured worst-case and mean convergence factor; Bottom: Estimated worst-case and number of iterations. Space-time refinement 2.*

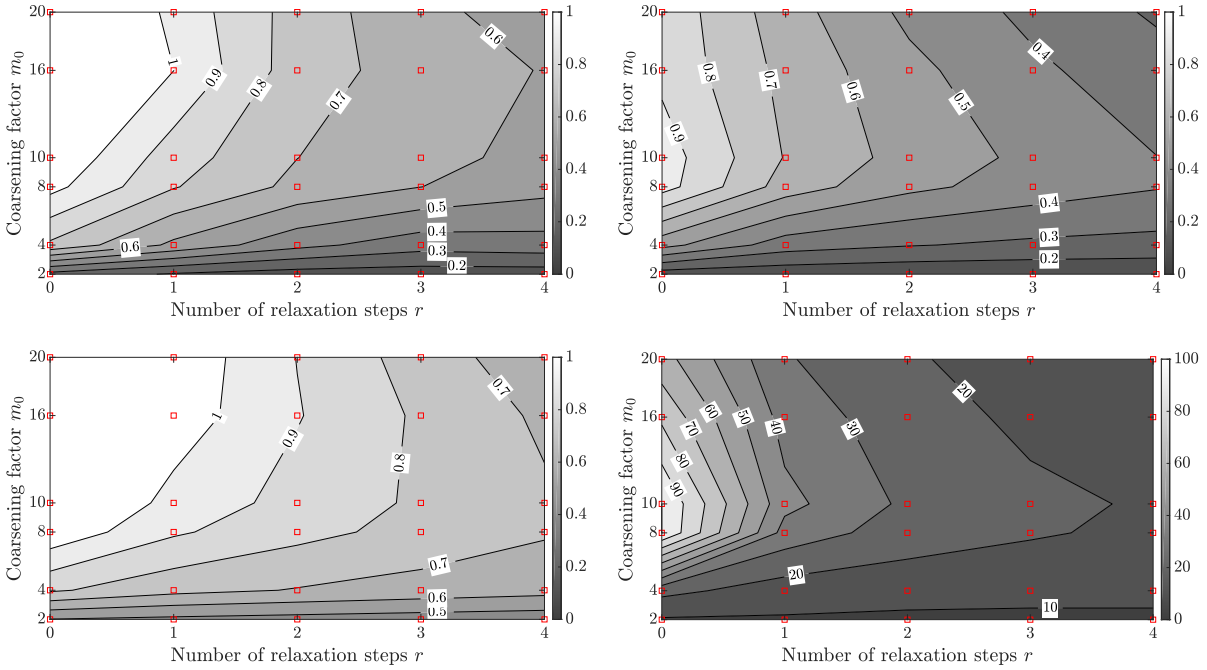


Figure 5.51: *Top: Measured worst-case and mean convergence factor; Bottom: Estimated worst-case and number of iterations. Space-time refinement 3.*

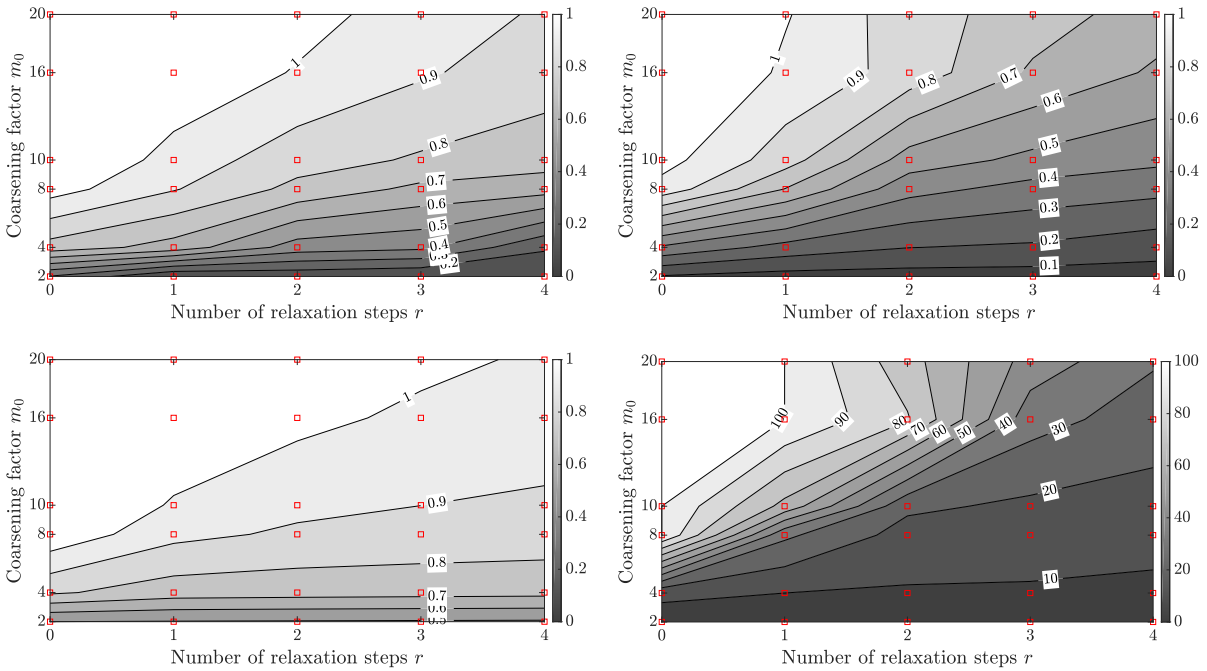


Figure 5.52: *Top: Measured worst-case and mean convergence factor; Bottom: Estimated worst-case and number of iterations. Space-time refinement 4.*

5.5.3.2 Multilevel MGRIT with V- and F-cycles

Figure 5.53 and Figure 5.54 show observed worst-case and mean convergence factors of multilevel MGRIT with V- and F-cycles, depending on the number of relaxation steps r and the employed space-time refinement level.

For MGRIT V-cycles, the inequality bound overestimates observed convergence factors significantly for four and five grid levels (see Figure 5.53), whereas magnitudes and trends are captured for two and three levels. Here, Approximation 1 and Approximation 2 are more valuable in the sense that the approximate convergence factors capture the observed worst-case much better. It is also noted that for space-time refinement levels 1 and 2, the observed convergence factors are constant or only slightly increasing but for refinement levels 3 and 4, the observed worst-case convergence factors are steadily increasing. This is not ideal. For example, for five levels and F- or FCF-relaxation, the worst-case convergence factor increases with a finer space-time mesh.

MGRIT F-cycles on the other hand (see Figure 5.54) show almost constant convergence factors with only a slight increase for refinement levels 3 and 4 and F-relaxation and refinement level 4 and FCF-relaxation. It is further noted that for refinement levels 1 and 2 and FCF-relaxation, the inequality bound gives quite sharp estimates of the observed worst-case. A general trend is further, that the mean and maximum observed convergence factors are close in the case of F-cycles but show a bigger gap for V-cycles.

Furthermore, we note the cases where the inequality bound could not be evaluated due to memory constraints (here, a machine with 128 GB memory was used), and thus, Approximation 1, 2 and 3 (with negligible memory consumption) are the only tools at hand:

- V-cycles:
  - FCF-relaxation: Refinement level 4 and $n_\ell > 3$.
- F-cycles:
  - F-relaxation: Refinement level 4 and $n_\ell > 2$.
  - FCF-relaxation: Refinement level 4 and $n_\ell > 2$.

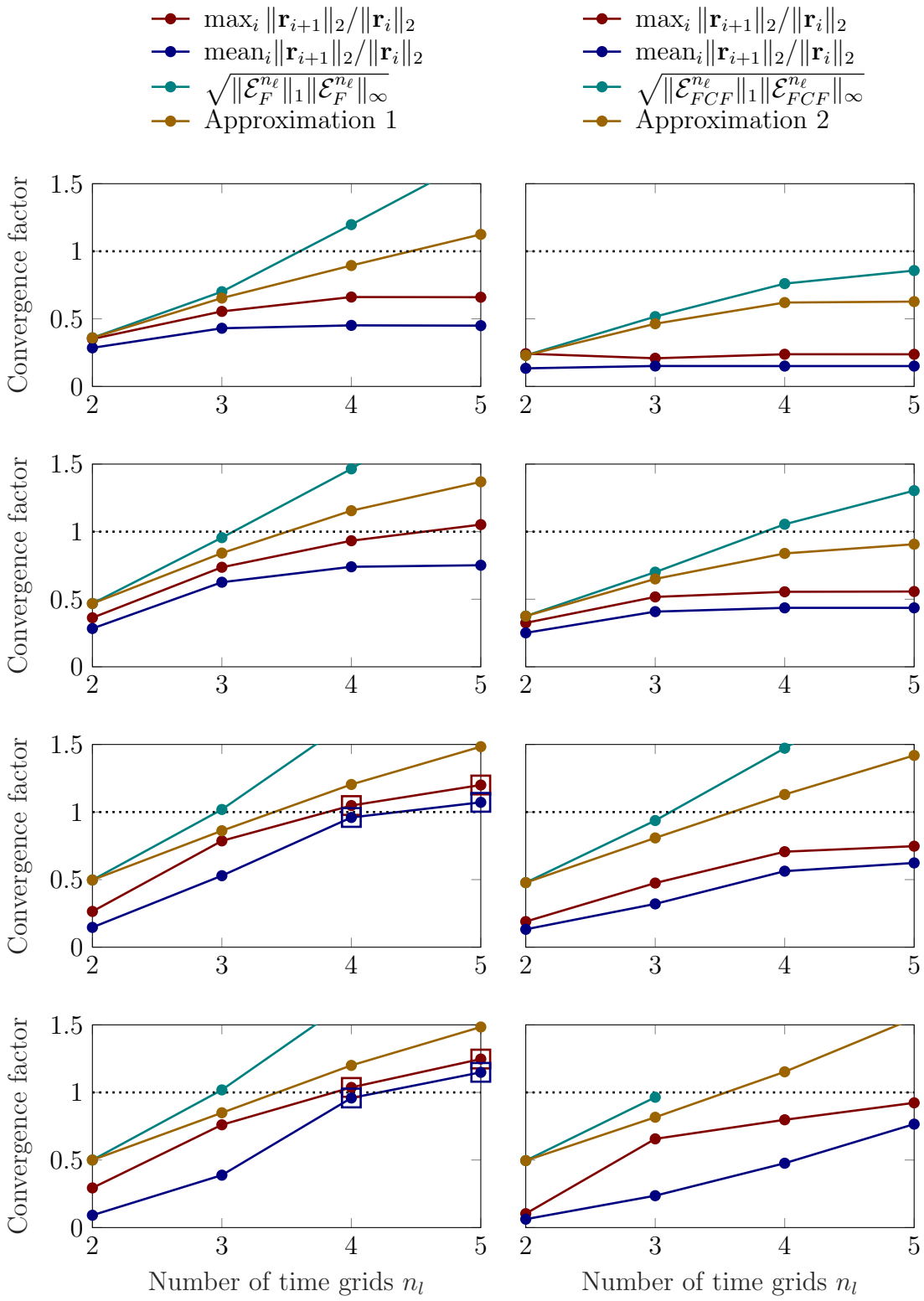


Figure 5.53: *Multilevel MGRIT V-cycle convergence with F-relaxation (left) and FCF-relaxation (right). Top-to-bottom: Space-time refinement level 1 - 4. Squares denote cases when MGRIT did not converge in* 100 *iterations. Missing data points are due to out-of-memory errors.*

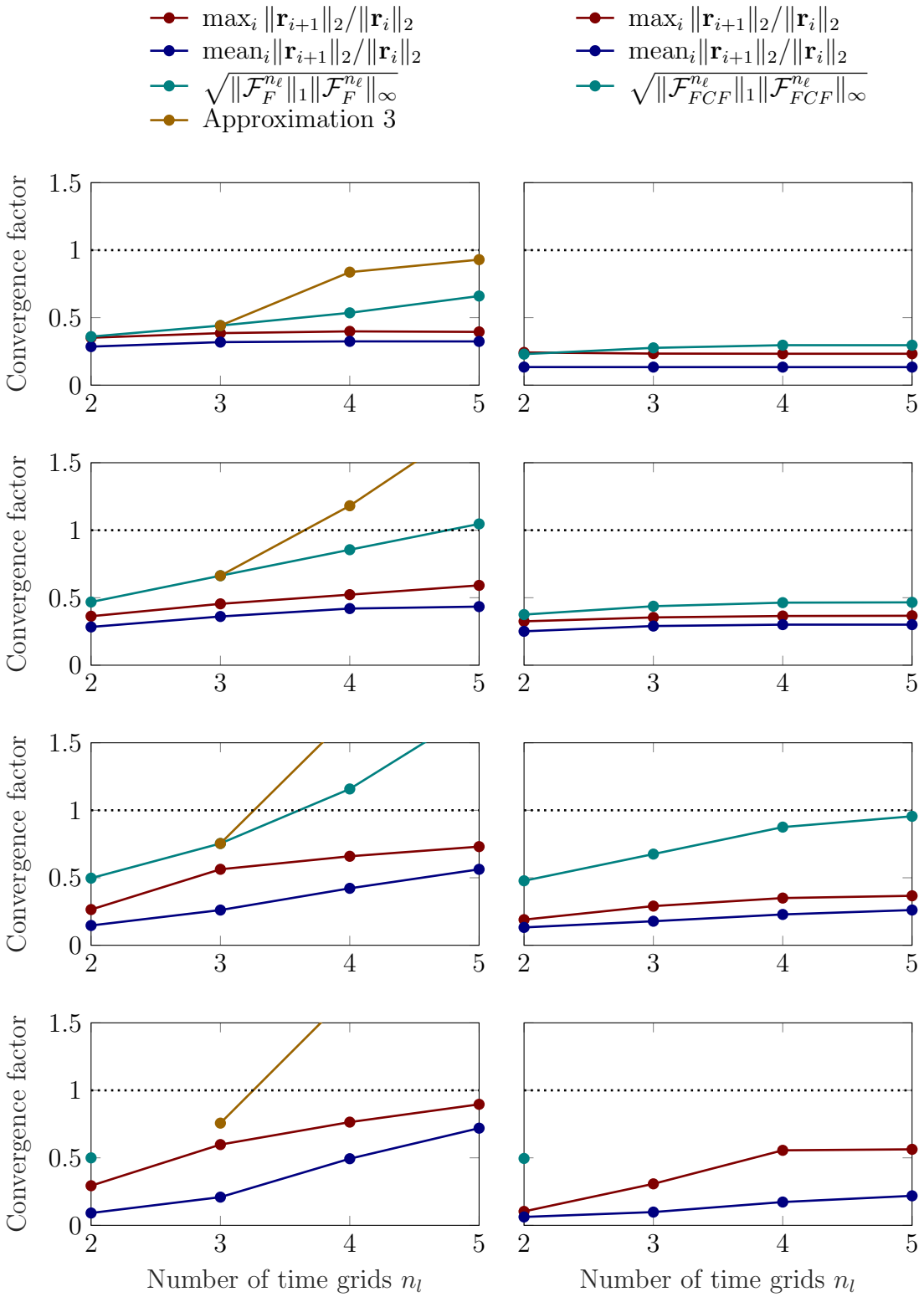


Figure 5.54: *Multilevel MGRIT F-cycle convergence with F-relaxation (left) and FCF-relaxation (right). Top-to-bottom: Space-time refinement level 1 - 4. Missing data points are due to out-of-memory errors.*

5.5.3.3 Convergence factor for each spatial mode $\lambda_{\ell,k}$

Now, let's look closer at the distribution of the spatial modes $\lambda_{\ell,k}$ in the complex plane and the correlation to the convergence factor for each spatial mode. For example, consider two-level MGRIT with F- and FCF-relaxation. As Figure 5.55 and Figure 5.56 illustrate, the slow converging eigenmodes are not near 0 or 1 (i.e. near the stability boundary) on the real axis (which are sometimes suspected to be the culprit) but rather eigenmodes with nonzero imaginary part. It is also quite clear that FCF-relaxation effectively scales down the convergence factors with the most visible effect on those eigenmodes that are farthest away from the real axis.

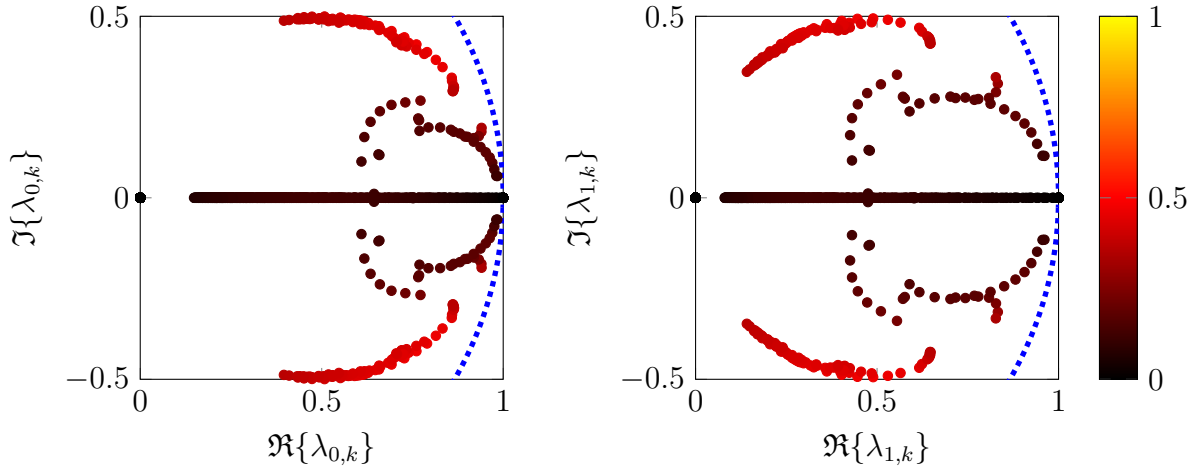


Figure 5.55: *Two-level MGRIT with F-relaxation for space-time refinement level 2: Convergence factor for each spatial mode in the complex plane as estimated by the inequality bound, see Theorem 1. The dotted blue line indicates the stability boundary $|x| = 1$ for $x \in \mathbb{C}$.*

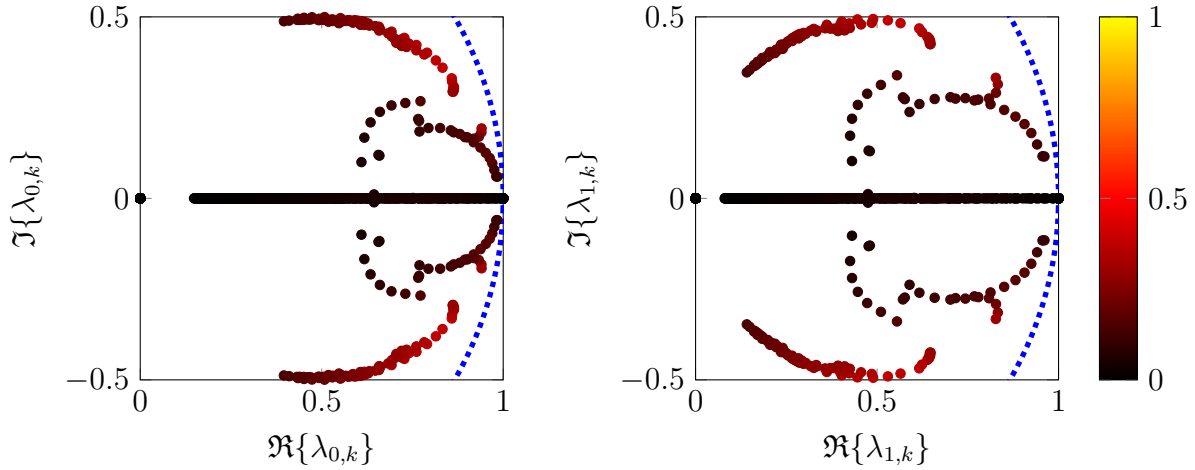


Figure 5.56: *Two-level MGRIT with FCF-relaxation for space-time refinement level 2: Convergence factor for each spatial mode in the complex plane as estimated by the inequality bound, see Theorem 1. The dotted blue line indicates the stability boundary $|x| = 1$ for $x \in \mathbb{C}$.*

Now moving on to the multilevel case with four-level V-cycles with F-relaxation (see Figure 5.57), the difference is quite remarkable. Eigenmodes $\lambda_{0,k}$ with nonzero imaginary part that are closer to the stability boundary (and their coarse-grid equivalents) are generally converging more slowly. Eigenmodes along the real axis show slower convergence as well, in contrast to the two-level algorithm (see Figure 5.55), however, eigenmodes near 0 and 1 are still converging fast. Although, it is clear that an algorithm acting on so many slow eigenmodes can be expected to perform poorly, which is reflected in the observed worst-case convergence factor approaching 1 in Figure 5.53 (second row, left).

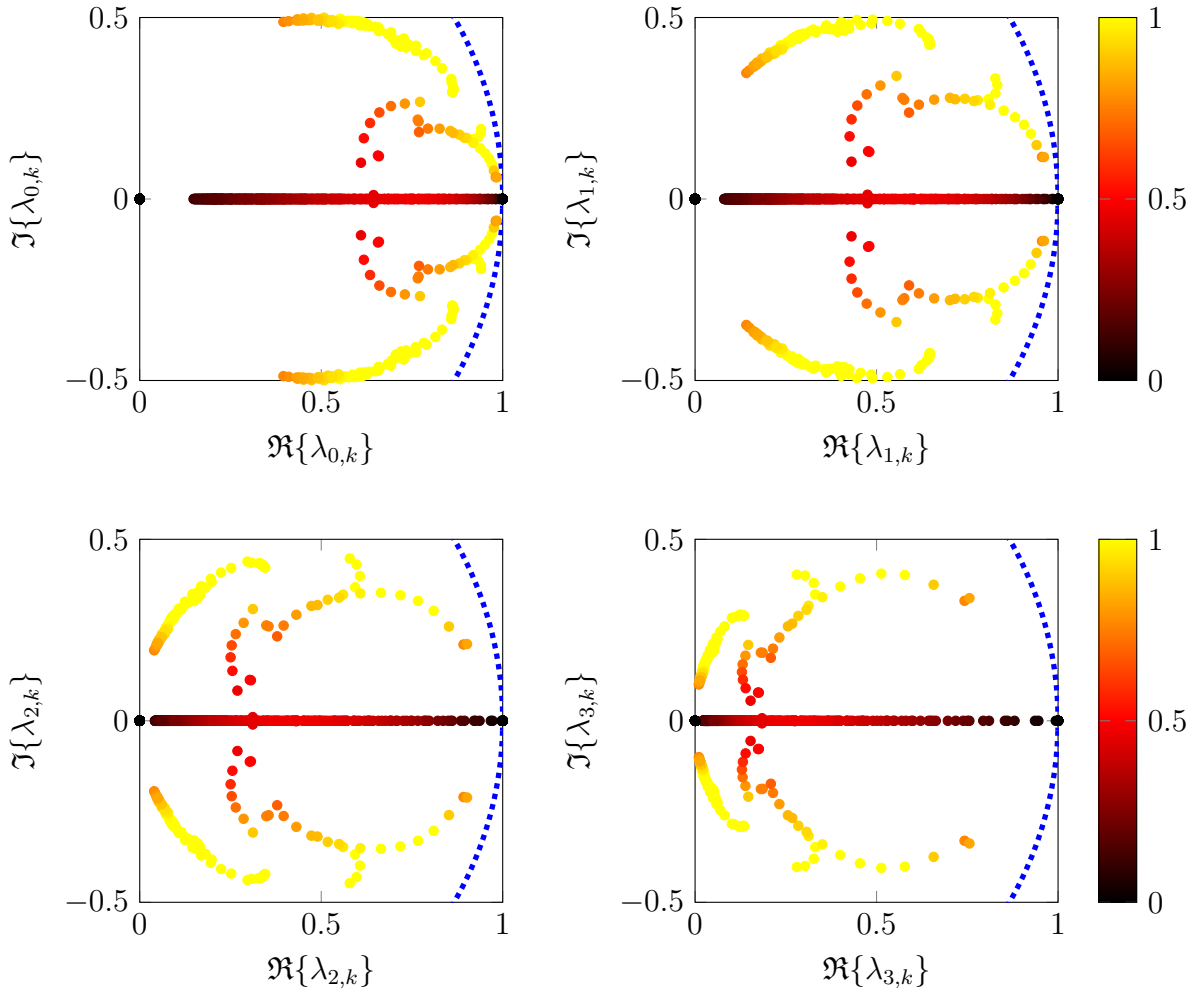


Figure 5.57: *Four-level MGRIT V-cycles with F-relaxation for space-time refinement level 2: Convergence factor for each spatial mode in the complex plane as estimated by the inequality bound, see Theorem 1. The dotted blue line indicates the stability boundary $|x| = 1$ for $x \in \mathbb{C}$.*

Figure 5.58 highlights how switching from V-cycles to F-cycles (keeping the number of grid levels $n_\ell = 4$ fixed) can significantly improve convergence. For example, all purely real eigenmodes are converging fast. While some eigenmodes still converge more slowly with convergence factors of approximately 0.86, they are significantly smaller in numbers. Thus, it is less probable that slow eigenmodes are actually present in the discrete error yielding faster observed convergence, which is true for the considered numerical exper-

iments. For example, the observed worst-case convergence factor for four-level F-cycles with F-relaxation is approximately 0.42, see Figure 5.54.

While FCF-relaxation can have a similar beneficial effect on the convergence of each eigenmodes (see Figure 5.59), it is not as pronounced. Eigenmodes on the real axis are still converging slower for a four-level V-cycle with FCF-relaxation than for a four-level F-cycle with F-relaxation. Further, the effect is different depending on the location of the eigenmode in the complex plane, e.g., eigenmodes $\lambda_{0,k}$ near $0.5+0.5i$ (and their coarse-grid equivalents) are actually converging slower for F-cycles with F-relaxation.

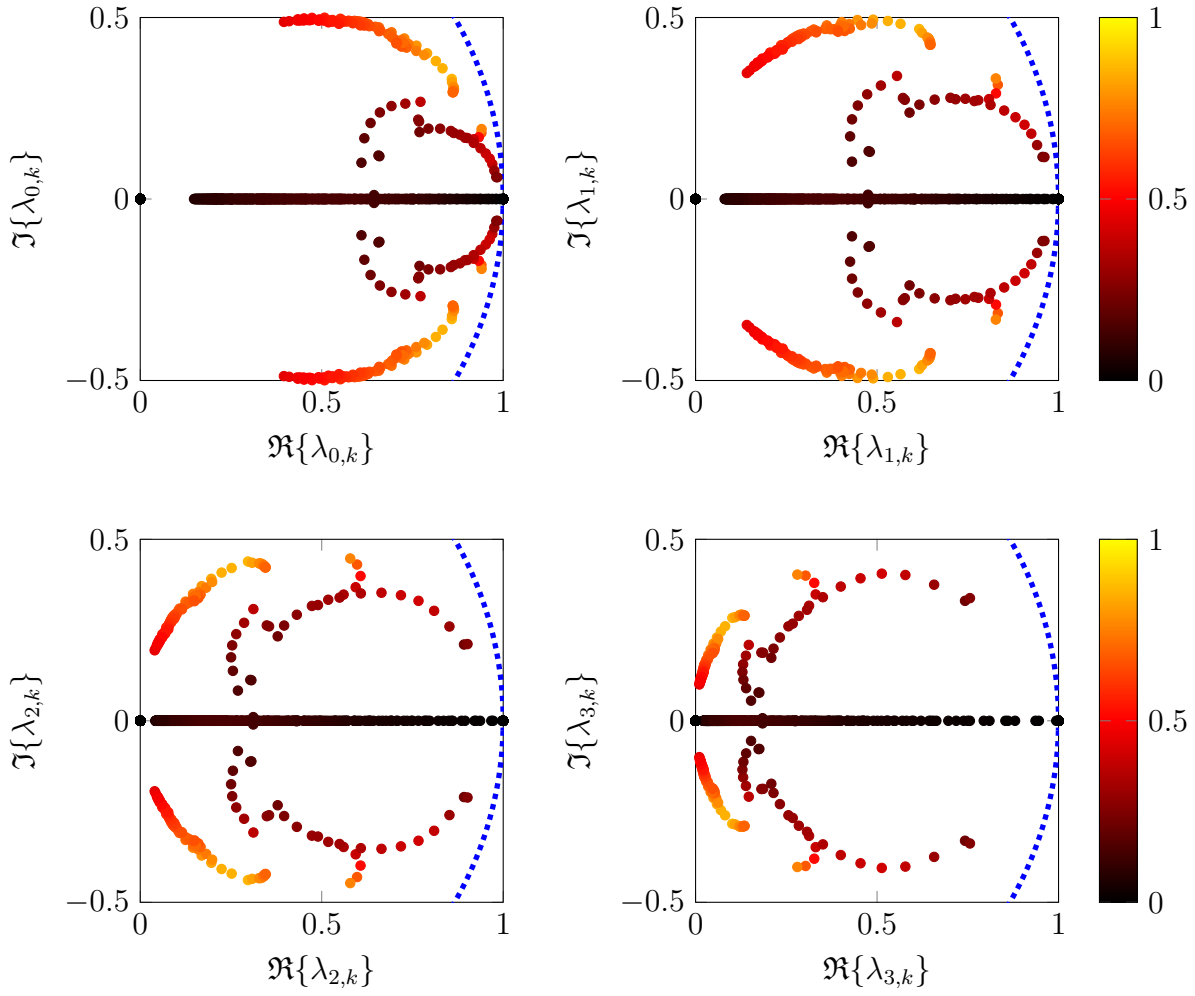


Figure 5.58: *Four-level MGRIT F-cycles with F-relaxation for space-time refinement level 2: Convergence factor for each spatial mode in the complex plane as estimated by the inequality bound, see Theorem 1. The dotted blue line indicates the stability boundary $|x| = 1$ for $x \in \mathbb{C}$.*

The largest improvement, compared to V-cycles with F-relaxation, can be achieved by switching to F-cycles *and* FCF-relaxation. In this case, only few eigenmodes are estimated to converge at a rate of approximately 0.46, whereas the observed worst-case convergence factor is approximately 0.36. This highlights that if the objective is to minimize the convergence factor while maximizing the number of grid levels (and thus, the potential for parallelism) for this particular application, it is required to choose a fairly expensive

algorithm, i.e. F-cycles with FCF-relaxation. Thus, the potential for speedup may be limited and it is worthwhile to explore how temporal periodicity can be exploited in the following.

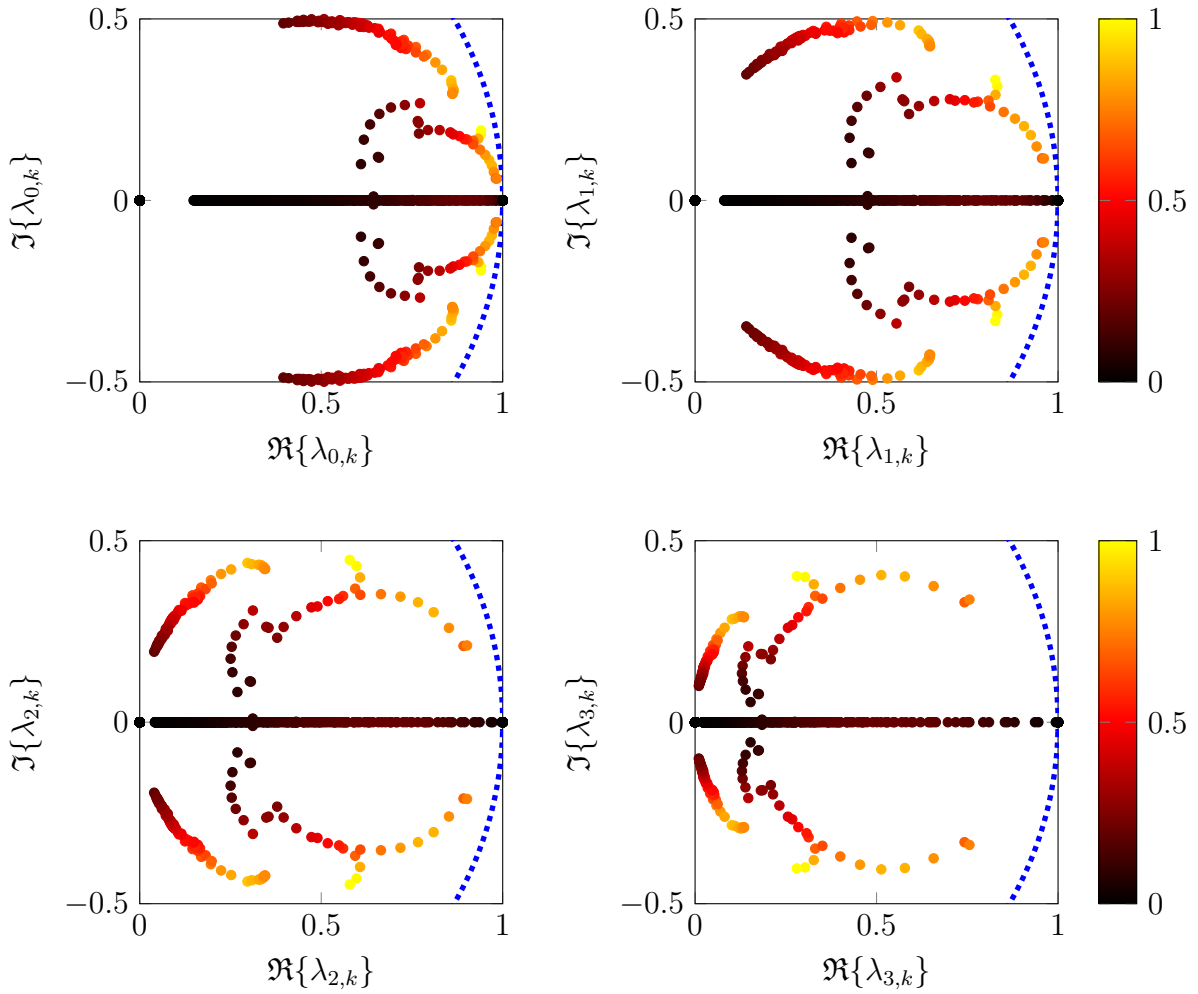


Figure 5.59: *Four-level MGRIT V-cycles with FCF-relaxation for space-time refinement level 2: Convergence factor for each spatial mode in the complex plane as estimated by the inequality bound, see Theorem 1. The dotted blue line indicates the stability boundary $|x| = 1$ for $x \in \mathbb{C}$.*

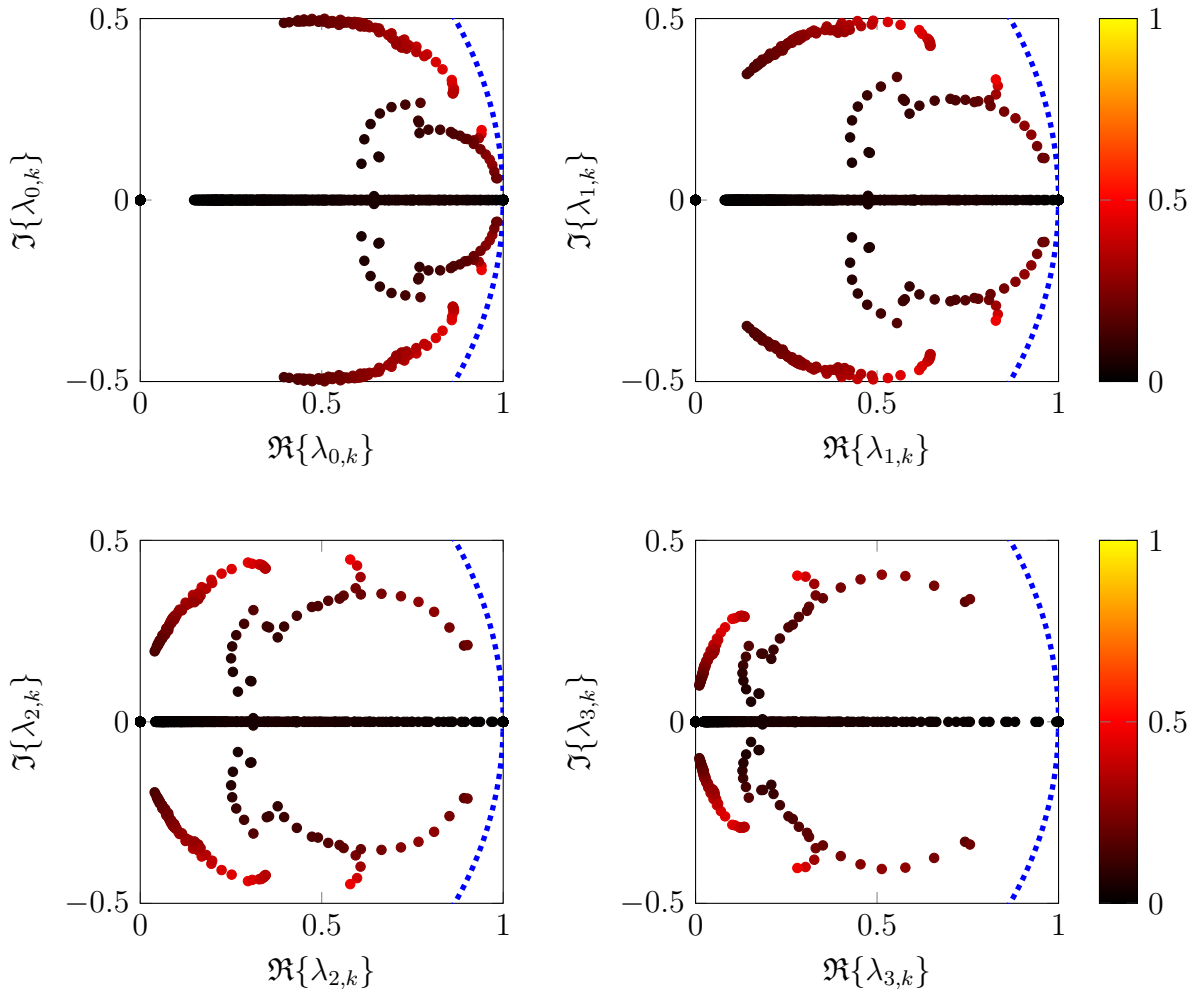


Figure 5.60: *Four-level MGRIT F-cycles with FCF-relaxation for space-time refinement level 2: Convergence factor for each spatial mode in the complex plane as estimated by the inequality bound, see Theorem 1. The dotted blue line indicates the stability boundary $|x| = 1$ for $x \in \mathbb{C}$.*

5.5.4 Time-periodic MGRIT

There are three main arguments for exploring how the time-periodic MGRIT algorithm can be applied to the considered FSI problem:

1. The convergence factor for the nonperiodic MGRIT algorithm is underwhelming; and in most cases, the convergence factor is smaller for smaller fine grid sizes.

2. Exploiting the time-periodic property yielded a large benefit for the flow problem in Section 5.4.

3. Here, there is an analytic solution that can help to study what an appropriate convergence criterion is for updating the initial condition when using the time-periodic MGRIT algorithm.

Thus, we consider the application of the time-periodic MGRIT algorithm (with skip-first-down option) from Section 5.4.6 with FCF-relaxation and measure how the error can

be reduced. The goal is to achieve a comparable order of error as sequential time-stepping over 10 cycles. For example, the spatial error at each time point during cycles $1, 2, \ldots, 10$ for the fluid velocity, solid displacement and solid velocity is shown in Figure 5.61, Figure 5.63 and Figure 5.65. While the cycle-to-cycle error reduction is initially quite fast, e.g., from cycle 1 to 5, it then decreases and almost becomes negligible from cycle 9 to 10. Thus, after 10 cycles, the system can be considered to have reached a periodic steady-state, i.e. cycle-to-cycle fluctuations become negligible.

After 10 cycles, sequential time-stepping yields a spatial error of order $O(10^{-6})$ in the fluid and solid velocity variables. Now, let's consider a two-level MGRIT algorithm with FCF-relaxation and temporal coarsening factor $m_0 = 8$ for space-time refinement level 4, see Table 5.3, to study how quickly the time-periodic MGRIT algorithm can yield a comparable error.

Figure 5.62, Figure 5.64 and Figure 5.66 illustrate the respective spatial errors over $t \in [0, T]$ for each time-periodic MGRIT iteration. Comparing the fluid velocity error from sequential time-stepping and time-periodic MGRIT (i.e. Figure 5.61 and Figure 5.62), it becomes clear that using the skip-first-down option for MGRIT yields a smaller error after iteration 0 than after cycle 1 with sequential time-stepping. This makes sense, since iteration 0 effectively generates a cheaper initial space-time guess using a larger time step size $\delta_1 = m_0 \delta_0$.

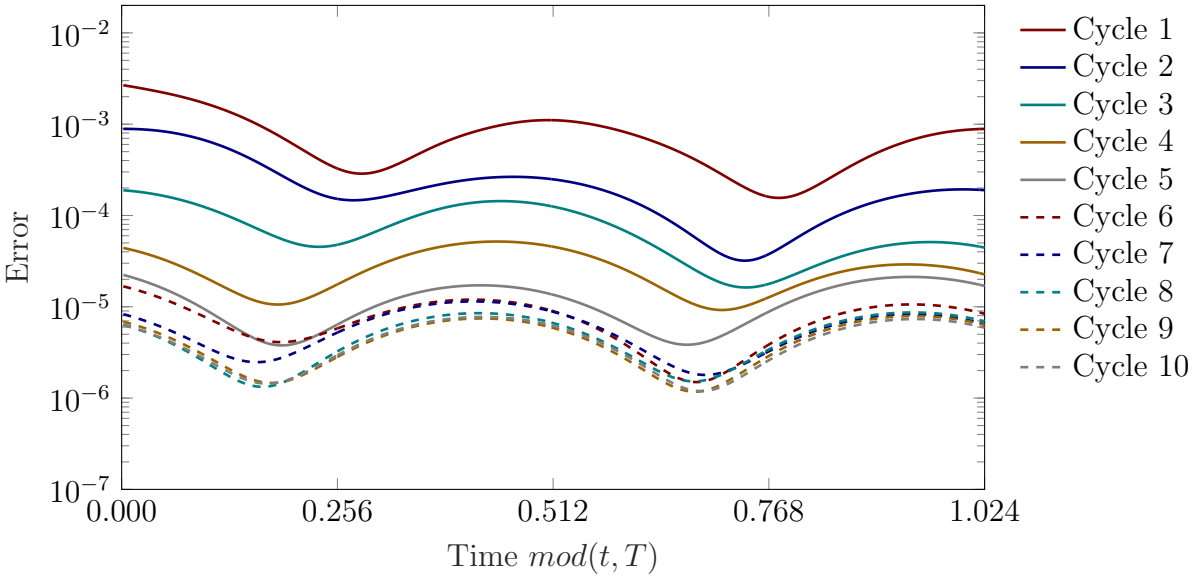


Figure 5.61: *Fluid velocity error reduction for sequential time-stepping over* 10 *cycles.*

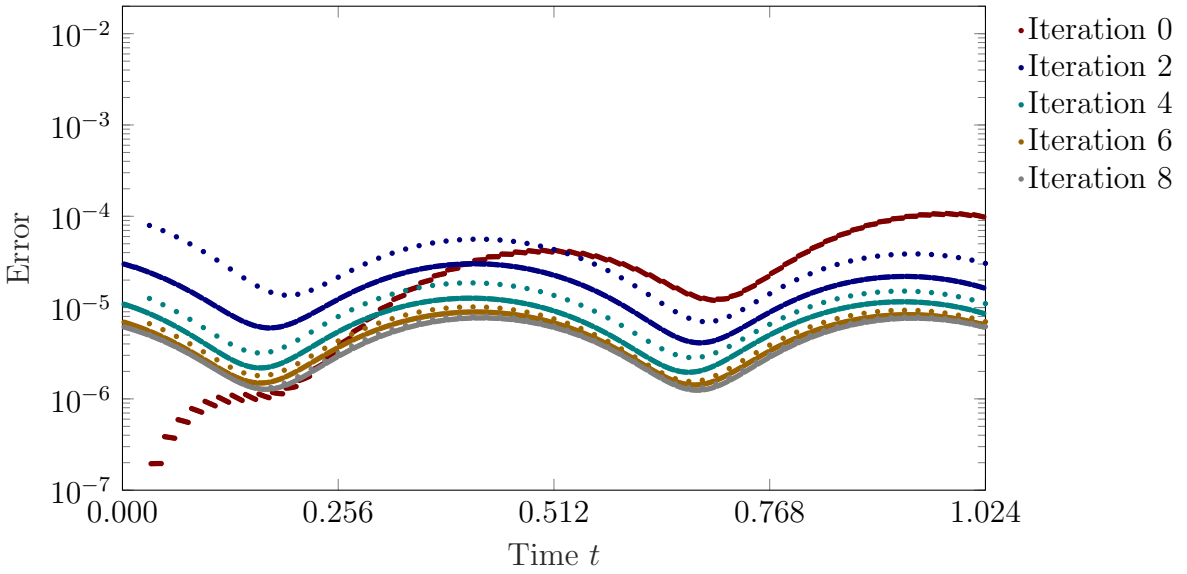


Figure 5.62: *Fluid velocity error reduction of two-level MGRIT with FCF-relaxation and temporal coarsening factor* $m_0 = 8$. *It takes* 8 *iterations to achieve a comparable error as sequential time-stepping over* 10 *cycles, see Figure 5.61.*

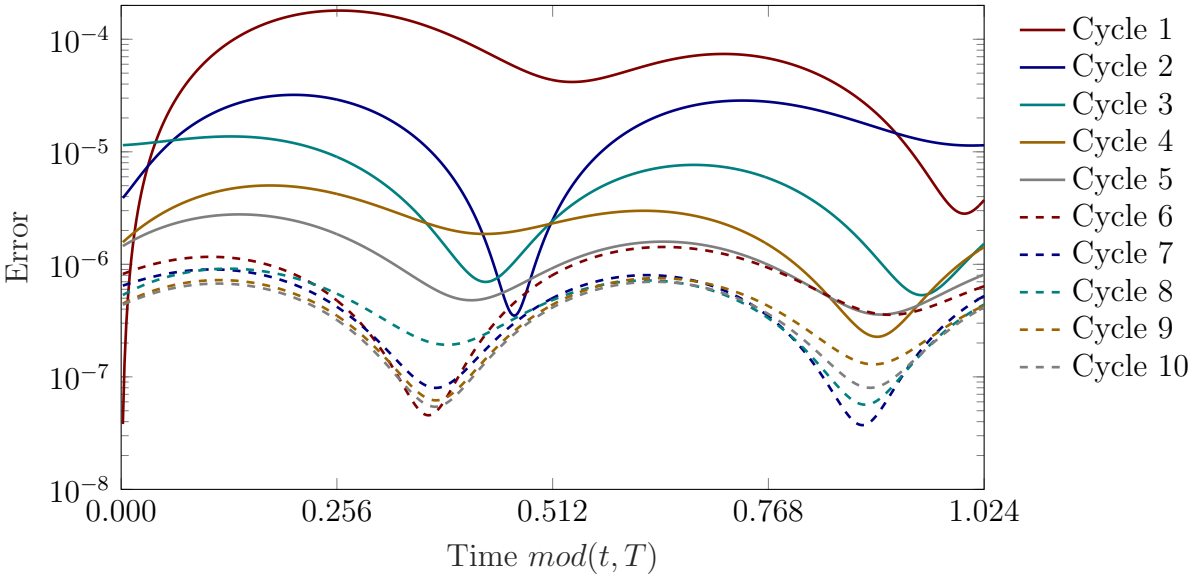


Figure 5.63: *Displacement error reduction for sequential time-stepping over* 10 *cycles.*

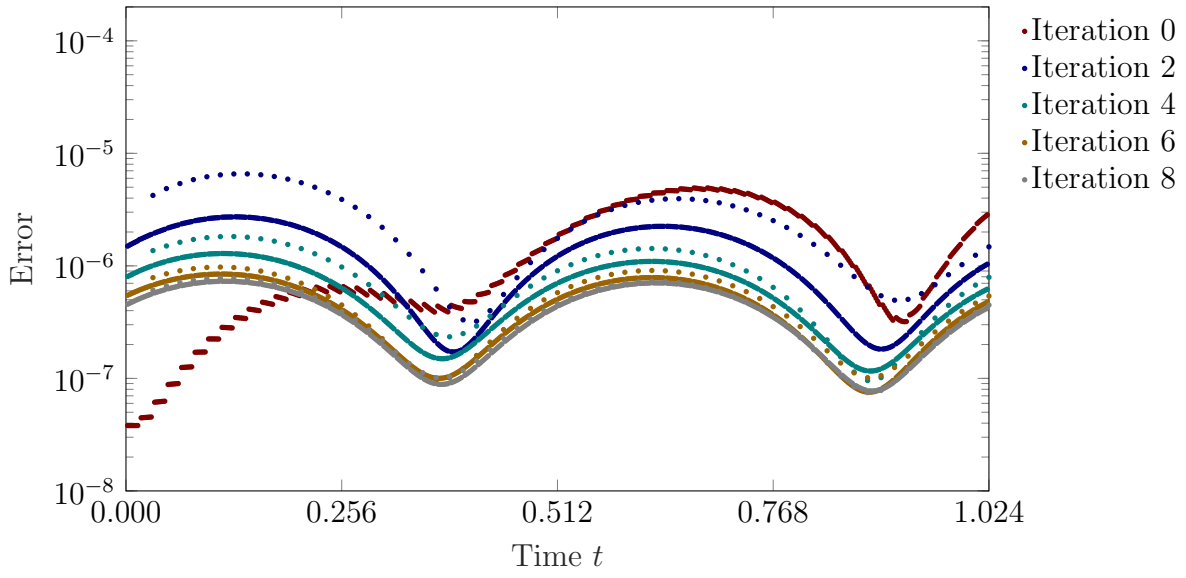


Figure 5.64: *Displacement error reduction of two-level MGRIT with FCF-relaxation and temporal coarsening factor* $m_0 = 8$*. It takes* 8 *iterations to achieve a comparable error as sequential time-stepping over* 10 *cycles, see Figure 5.63.*

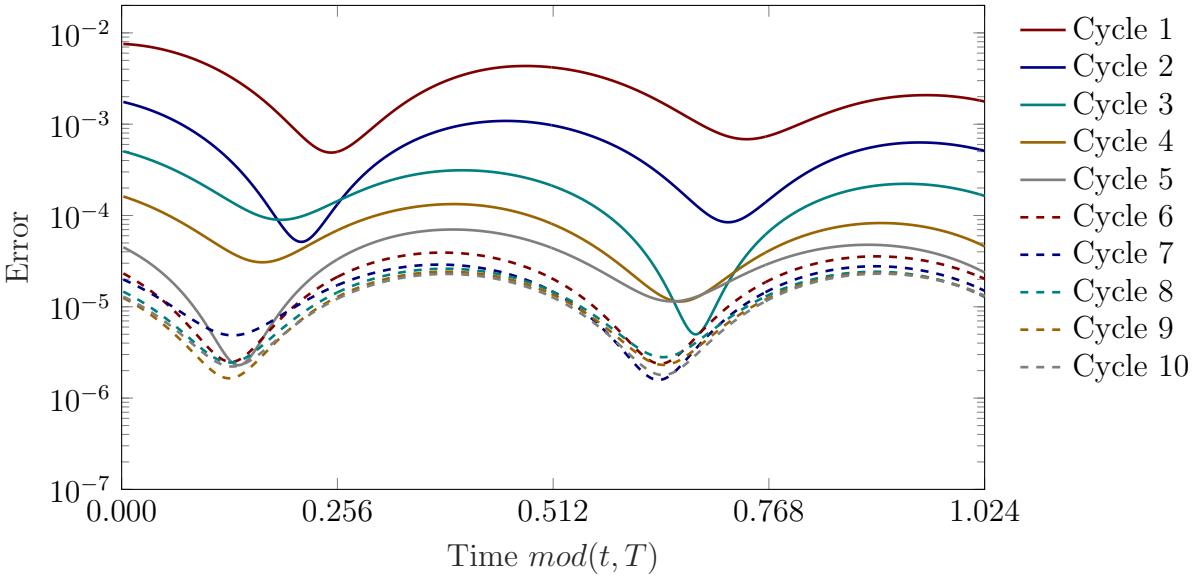


Figure 5.65: *Solid velocity error reduction for sequential time-stepping over* 10 *cycles.*

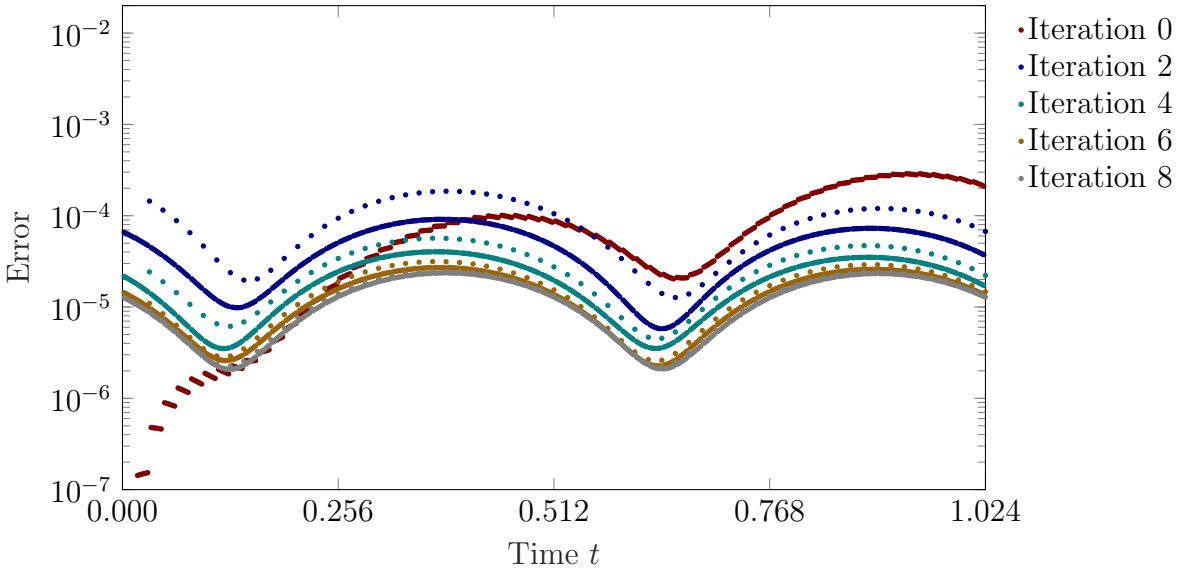


Figure 5.66: *Solid velocity error reduction of two-level MGRIT with FCF-relaxation and temporal coarsening factor* $m_0 = 8$. *It takes* 8 *iterations to achieve a comparable error as sequential time-stepping over* 10 *cycles, see Figure 5.65.*

Another typical property of MGRIT is that errors and residuals are generally larger towards the end of the temporal domain and can be reduced much quicker at the beginning of the temporal domain. This shows after the initial coarse-grid solve in iteration 0, see Figure 5.62. The error then quickly *equilibrates* and has a similar order throughout the temporal domain; with oscillations reflecting oscillations in the magnitudes of the solution itself. The error can further be quickly reduced within just 8 time-periodic MGRIT iterations. The errors for iteration 6 and iteration 8 are of the same order as for the sequential time-stepping solution during cycles 9 and 10.

Similar observations can be made for the error in the solid displacement (see Figure 5.64

and Figure 5.64) and solid velocity (see Figure 5.66 and Figure 5.66) with the only difference that the error in the solid displacement is of smaller order due to the relation (5.68) between the solid displacement and velocity, and due to the one-to-one coupling (5.72) between the fluid and solid velocity.

Thus, the time-periodic MGRIT algorithm (*recall*: two-level, FCF-relaxation, coarsening $m_0 = 8$) only needs 8 iterations to converge to the same order of error as sequential time-stepping over 10 cycles. This is despite the fact that the MGRIT residual is only reduced by roughly two orders of magnitude (see Figure 5.67), which again highlights the fact that the initial error after iteration 0 is already much smaller than with sequential time-stepping after cycle 1.

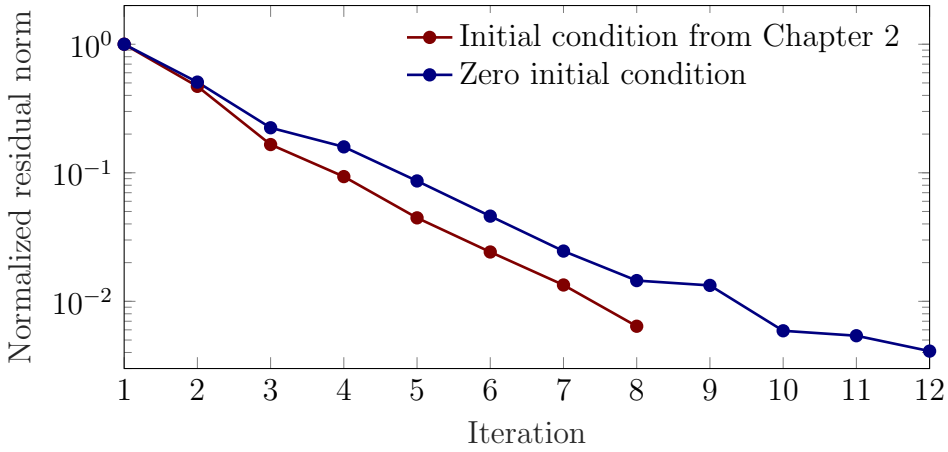


Figure 5.67: *Residual reduction of two-level MGRIT with FCF-relaxation and temporal coarsening factor $m_0 = 8$ depending on choice of initial condition.*

We can then compare the wall clock times (using ORCA, see Section A) for sequential time-stepping over $1, 3, 5, 10$ cycles and for the time-periodic MGRIT algorithm over 8 iterations (see Figure 5.68 and Table 5.4) with no data export (i.e. only measuring elapsed times for computation and communication). The wall clock time of the sequential time-stepping algorithm varies between 45 seconds for 1 cycle and 454 seconds for 10 cycles. On the other hand, the time-periodic MGRIT algorithm takes 174 seconds when no parallelism is employed and only 20 seconds when using 32 processors in the temporal domain (i.e. no spatial parallelism), yielding a maximum speedup of 22.57x, see Table 5.4.

Figure 5.69 further highlights the achieved space-time error for each simulated cycle of sequential time-stepping or each iteration of time-periodic MGRIT. Time-periodic MGRIT can achieve the same order of accuracy at a lower cost, which results in the best observed speedup of 22.57x using 32 processors.

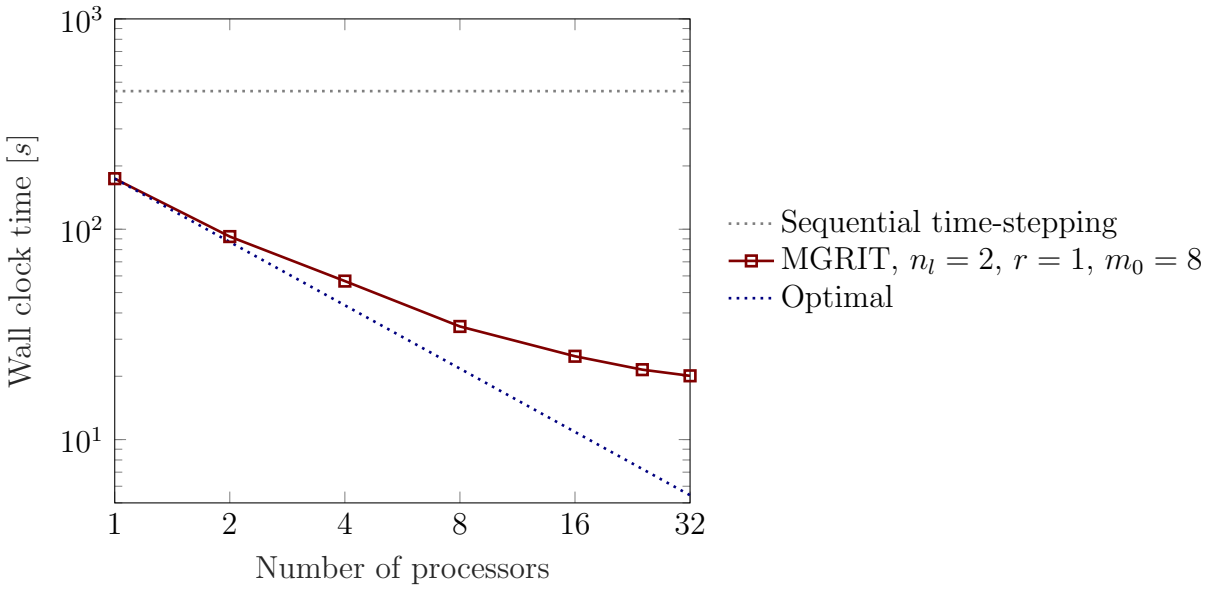


Figure 5.68: *Wall clock time of sequential time-stepping and time-periodic MGRIT to achieve a comparable space-time error.*

Algorithm	Number of cycles	Number of iterations	Number of processors	Wall clock time	Speedup vs. 10 cycles
Time-stepping	10	-	1	453.6 s	-
MGRIT	1	8	1	174.0 s	2.61x
MGRIT	1	8	2	92.3 s	4.91x
MGRIT	1	8	4	56.7 s	8.00x
MGRIT	1	8	8	34.5 s	13.15x
MGRIT	1	8	16	24.9 s	18.22x
MGRIT	1	8	24	21.5 s	21.10x
MGRIT	1	8	32	20.1 s	22.57x

Table 5.4: *Wall clock time and speedups for two-level MGRIT with FCF-relaxation and temporal coarsening factor $m_0 = 8$.*

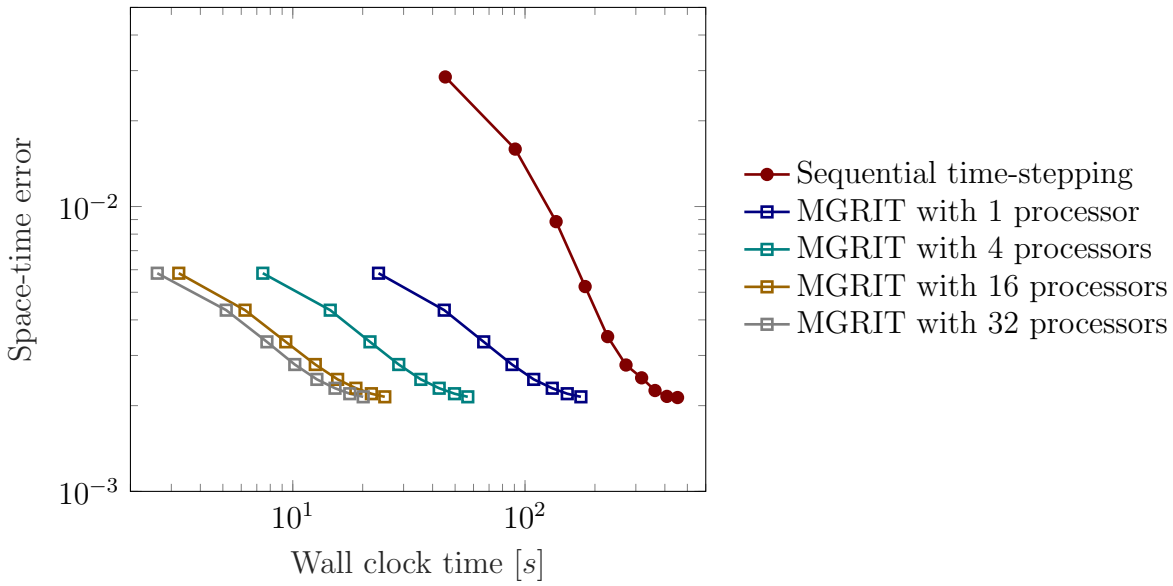


Figure 5.69: *Space-time error (velocity) compared to wall clock time of sequential time stepping and time-periodic MGRIT. Better accuracy can be achieved by simulating additional cycles or performing more MGRIT iterations.*

The question that arises is why the time-periodic MGRIT algorithm without parallelism can outperform sequential time-stepping over 5 cycles. There are multiple explanations that all work together:

1. **Size of temporal domain:**

   Parallel-in-time integration through time-periodic MGRIT is applied to a temporal domain that is much smaller, compared to simulating a given number of cycles for sequential time-stepping.

2. **Updating the initial condition:**

   The time-periodic MGRIT algorithm only requires an approximate update of the initial condition, and thus, already works with an improved initial condition after the very first iteration. This can be seen in Figure 5.69, where the first iteration of MGRIT can achieve a better accuracy than simulating three cycles with sequential time-stepping.

3. **Multigrid reduction:**

   MGRIT is an iterative solver that achieves a speedup over sequential time-stepping, which can be seen as a direct solver. The sequential component (i.e. the coarse-grid) is reduced by the temporal coarsening factor m_0.

4. **Jump-starting MGRIT:**

   Using the skip-first-down option for MGRIT yields a cheap initial guess over the entire space-time domain. In particular, it yields an already improved initial condition, see Bullet 2.

Previously, the initial condition (for sequential time stepping and the first iteration of time-periodic MGRIT) was selected as the analytic solution at $t = 0$ (compare Chapter 2). In typical applications, however, there is usually no good initial condition available, such that oftentimes the initial condition is selected as *zero*. Thus, now a zero initial condition is selected and studied how such a choice affects the solution process. Figure 5.70 illustrates the wall clock time depending on the required accuracy. It is noted, that now 12 time-periodic MGRIT iterations are required to achieve the same space-time error as sequential time-stepping over 10 cycles because the initial error is larger. Thus, the observed speedup is smaller with 1.77x using 1 processor, 5.56x using 4 processors and 14.98x using 32 processors. Considering the MGRIT residual for each iteration (see Figure 5.67), one can see that the residual norm is still reduced by roughly two orders of magnitude, however, the reduction is a bit slower compared to using the analytic solution as initial condition. This is not necessarily a problem because a slow-down in residual reduction could simply be caused by a redistribution or equilibration of the error over the temporal domain that still reduces the global space-time error.

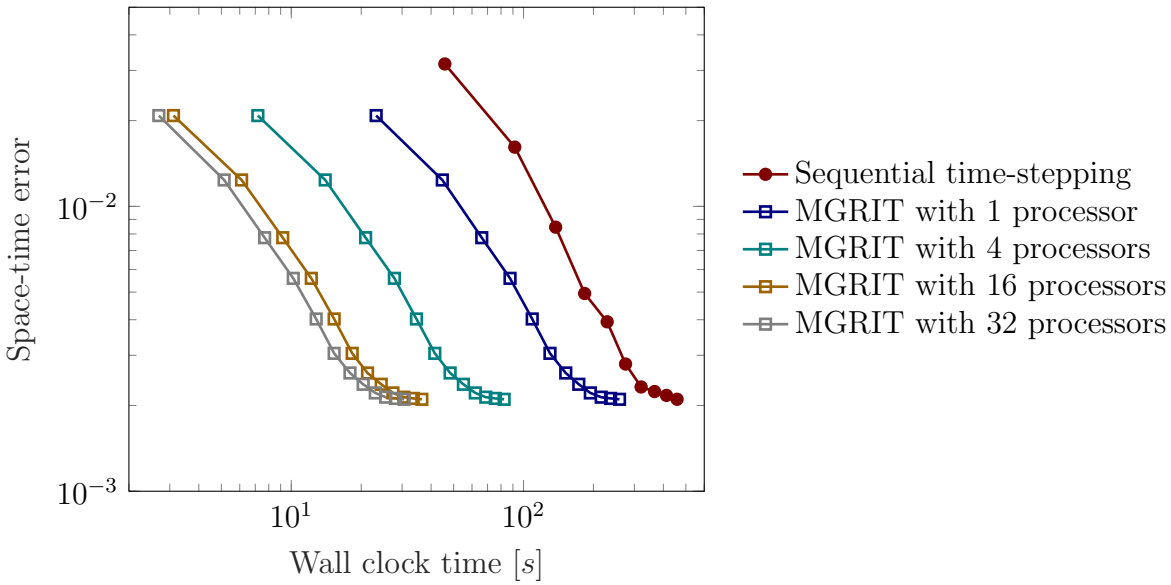


Figure 5.70: *Space-time error (velocity) compared to wall clock time of sequential time stepping and time-periodic MGRIT when using a zero initial condition. There is still a significant benefit of using time-periodic MGRIT, e.g., the maximum observed speedup is* $14.98x$ *using* 32 *processors. But using a better initial condition can accelerate time-periodic MGRIT relative to sequential time-stepping, see Figure 5.69.*

5.6 Analytic FSI solutions: Transient fluid / transient nonlinear solid in 3D[12]

In this section, complexity is added to assess the new time-periodic MGRIT algorithm in a more challenging setting. More specifically, the time-periodic MGRIT algorithm is applied to the three-dimensional FSI (transient fluid, transient nonlinear solid) test case from Chapter 2 (see Section 2.2.1.2). The domains are given in Figure 2.2 and parameters are selected similar to Section 2.4.2.2, see Table 5.5.

Fluid density	Fluid viscosity	Solid density	Solid stiffness	Cycle length
$\rho_f = 2.100$	$\mu_f = 0.030$	$\rho_s = 1.000$	$\mu_s = 0.100$	$T = 1.024$

Table 5.5: *Parameters for the 3D transient fluid / transient nonlinear solid case.*

The *coarse* and *medium* refinement levels from Table 2.2 are considered for the spatial mesh (i.e. step size $\delta_x, \delta_y, \delta_z \in \{0.314, 0.157\}$, number of DOFs $N_x \in \{32793, 245009\}$) and the temporal domain is discretized using 512 and 2048 time steps (i.e. step size $\delta_t \in \{0.002, 0.0005\}$) with constant $\delta_t/\delta_x^2 \approx 0.02$.[13]

Here, the initial condition is considered to be the analytic solution for all variables except for the fluid velocity, which is initially zero everywhere. This is to achieve a good initial condition and initial guess with a pertubation to simulate a practical case where some (experimental) data are available but convergence to a periodic steady-state is still required.

While the linear PDE convergence framework from Chapter 4 has shown promising pathways for extensions to the nonlinear PDE case (see Section 5.4.8.2), no theoretical convergence analysis is performed here. Instead, the knowledge from the linear PDE convergence analysis from previous sections is applied in combination with the time-periodic MGRIT algorithm to investigate if similarly large speedups can be obtained for the nonlinear FSI test case. Thus, a two-level time-periodic MGRIT algorithm with FCF-relaxation and temporal coarsening factor $m_0 = 8$ is employed.

First, consider the convergence of the fluid velocity solution to its time-periodic steady-state for the *coarse* mesh. Figure 5.71 illustrates how the error in the fluid velocity is reduced over 7 consecutive cycles using sequential time-stepping. On the other hand, Figure 5.72 illustrates error reduction over the cycle length when using time-periodic MGRIT. Initially, the error in the MGRIT approximation is less equally distributed over the cycle length. The error over the cycle length is also larger than during the first cycle of the sequential time-stepping solution, see Figure 5.74. It seems likely that this is the nonlinearity in the PDEs manifesting itself since a change in time step size (i.e. fine grid to coarse grid time step size) can be expected to have a bigger impact on solution accuracy compared to the linear PDE setting.

The error, however, can be reduced equally rapidly and a comparable order of error can be achieved after 6 time-periodic MGRIT iterations. Thus, compared to the linear FSI case (see Section 5.5) error reduction takes a similar number of time-periodic MGRIT iterations. There is further a slight difference in how the residual and error are reduced.

[12] Results in this section have previously appeared [62]

[13] Reported runtimes for ORCA (*coarse* mesh) and TOM (*medium* mesh), see Appendix A.

For example, the MGRIT residual increases in iteration 4 (see Figure 5.73), and similarly, the error plateaus, before both the residual and the error are further reduced. This is in contrast to the linear FSI case, where both the residual and error were reduced monotonously (e.g., see Figure 5.67 and Figure 5.70).

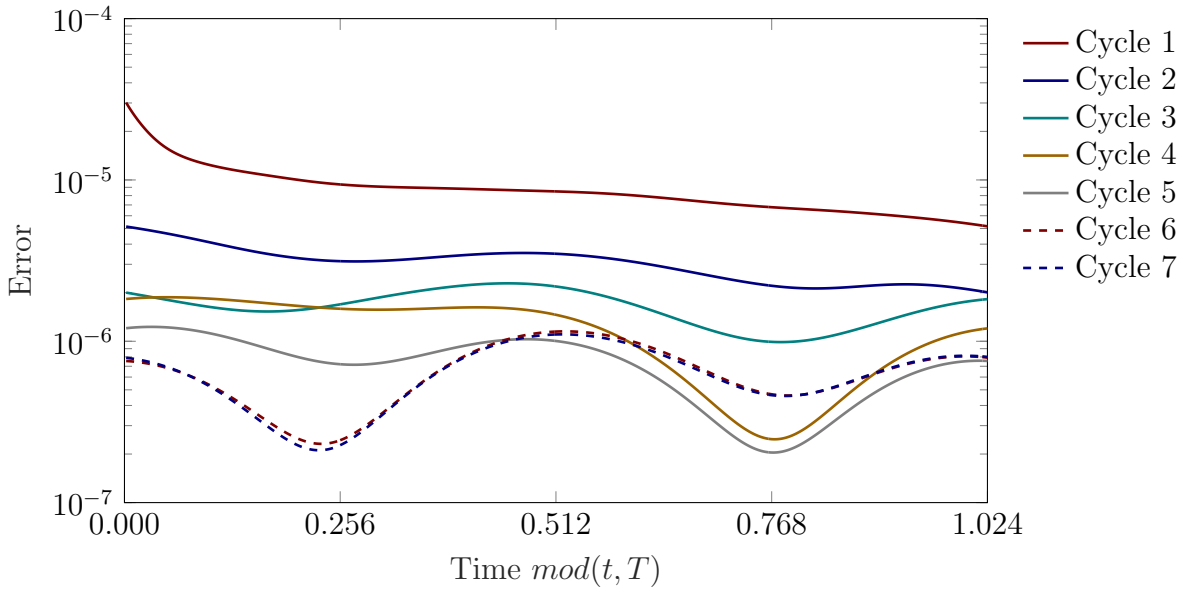


Figure 5.71: *Coarse mesh: Fluid velocity error reduction for sequential time-stepping over* 7 *cycles.*

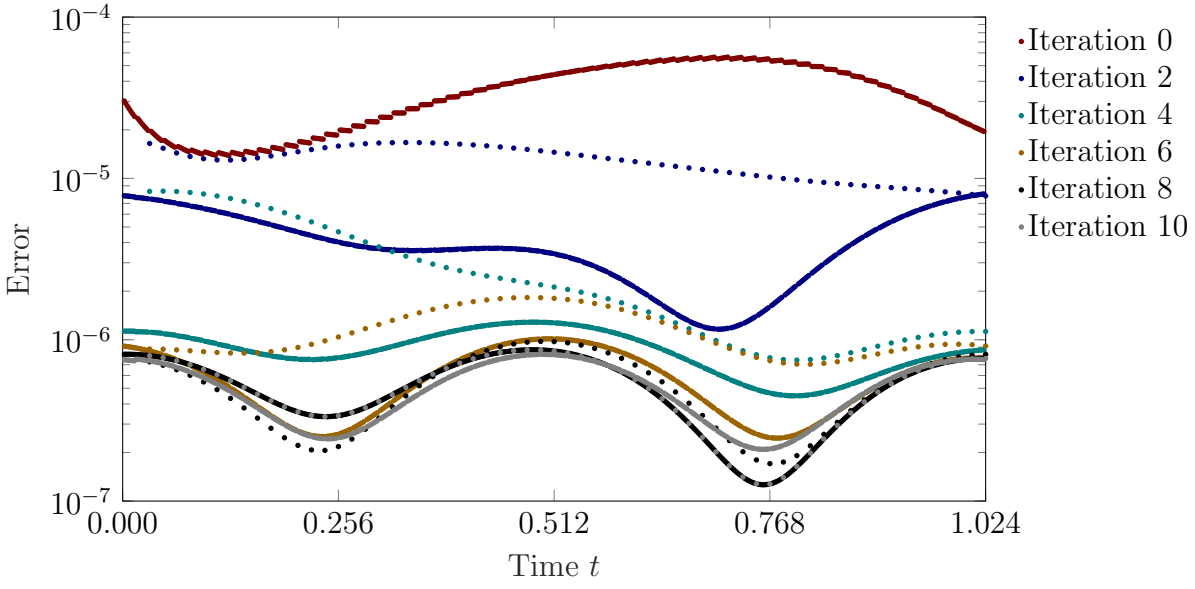


Figure 5.72: *Coarse mesh: Fluid velocity error reduction of two-level MGRIT with FCF-relaxation and temporal coarsening factor* $m_0 = 8$*. It takes* 6 *iterations to achieve a comparable error as sequential time-stepping over* 7 *cycles, see Figure 5.71.*

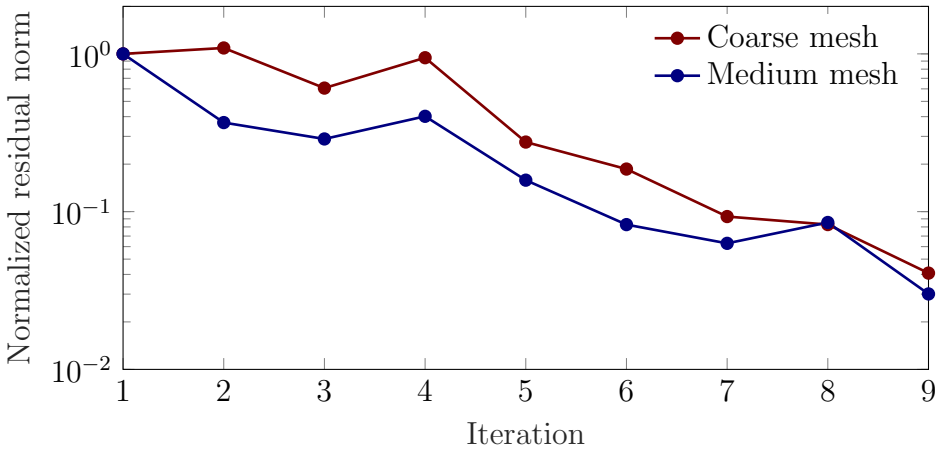


Figure 5.73: *Residual reduction of two-level MGRIT with FCF-relaxation and temporal coarsening factor* $m_0 = 8$.

Comparing wall clock times (see Figure 5.74) of sequential time-stepping and MGRIT for the *coarse* mesh, it can be seen that achieved speedups are not as large as for the linear FSI case. However, significant speedups over sequential time-stepping are achieved when using 16 processors to parallelize in the temporal domain. Depending on the required accuracy (see Figure 5.74 and Table 5.6), the speedup ranges between 2.96x and 6.20x using 16 processors. The largest achieved speedup is 6.64x using 32 processors. Thus, the benefit of adding more parallelism beyond 16 processors is negligible, while in the linear FSI case, a larger additional speedup can be achieved. This is likely due to the nonlinear solver taking more iterations toward the end of the temporal domain, and thus, causing less optimal load balancing. To achieve an even larger speedup at a higher number of processors, a different decomposition of the temporal domain could be considered that takes the cost of each time step (with respect to its position along the temporal domain) into account.

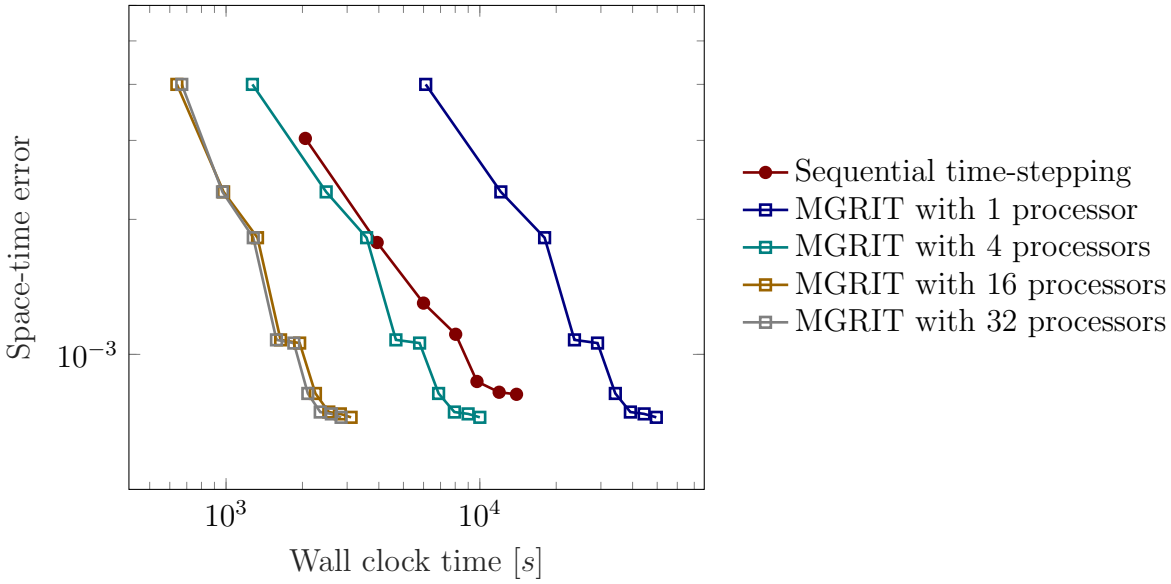


Figure 5.74: *Coarse mesh: Space-time error (velocity) compared to wall clock time of sequential time stepping and time-periodic MGRIT when using a zero initial condition for the fluid velocity and the analytic solution for all other variables. The largest achieved speedup is* $6.20x$ *using* 16 *processors and* $6.64x$ *using* 32 *processors. See Table 5.6.*

Sequential time-stepping		MGRIT		Speedup			
#cycles	Error	#iterations	Error	1 proc	4 procs	16 procs	32 procs
2	$1.78 \cdot 10^{-3}$	3	$1.82 \cdot 10^{-3}$	0.22x	1.10x	2.96x	3.08x
4	$1.11 \cdot 10^{-3}$	4	$1.08 \cdot 10^{-3}$	0.34x	1.72x	4.90x	5.08x
7	$8.15 \cdot 10^{-4}$	6	$8.18 \cdot 10^{-4}$	0.41x	2.03x	6.20x	6.64x
		9	$7.24 \cdot 10^{-4}$	0.28x	1.40x	4.49x	4.91x

Table 5.6: *Coarse mesh: Speedup depending on required accuracy and number of processors for two-level MGRIT with FCF-relaxation and temporal coarsening factor* $m_0 = 8$. *For a graphical presentation, see Figure 5.74.*

Now considering space-time refinement, it is possible to consider larger numbers of processors. Figure 5.75 illustrates accuracy (for the velocity approximation) vs. associated cost for sequential time-stepping, and time-periodic MGRIT with 16, 32 and 64 processors in time. Again, it is possible to rapidly reduce the space-time error with time-periodic MGRIT, giving the new algorithm a significant edge over sequential time-stepping in terms of wall clock time. Considering a comparable space-time error after 6 cycles (sequential time-stepping) and 6 iterations (time-periodic MGRIT), observed speedups are range from 6.40x using 16 processors to 9.05x using 64 processors, see Table 5.7. It is further noted, that a small *rebound* effect is observed between cycle 5 and 6 for sequential time-stepping, which similarly occurs for the time-periodic MGRIT algorithm between iteration 4 and 5. After iteration 5, however, the time-periodic MGRIT algorithm can further reduce the space-time error, while the wall clock time is still smaller than for sequential time-stepping.

As a matter of course, larger speedups might be observed when switching to a true

multilevel MGRIT algorithm and are expected when combining time-parallelism with space-parallelism, see Section 5.4.6.

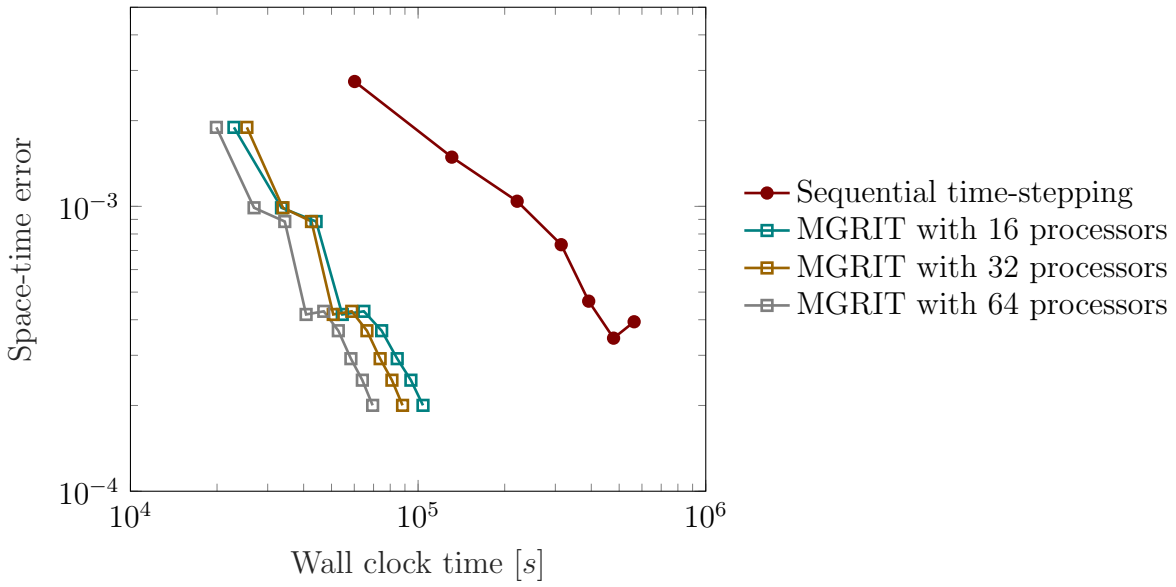


Figure 5.75: *Medium mesh: Space-time error (velocity) compared to wall clock time of sequential time stepping and time-periodic MGRIT when using a zero initial condition for the fluid velocity and the analytic solution for all other variables. The largest achieved speedup is 7.73x using 32 processors and 9.60x using 64 processors. See Table 5.7.*

Sequential time-stepping		MGRIT		Speedup		
#cycles	Error	#iterations	Error	16 procs	32 procs	64 procs
3	$1.04 \cdot 10^{-3}$	2	$9.98 \cdot 10^{-4}$	6.60x	6.52x	8.22x
5	$4.65 \cdot 10^{-4}$	4	$4.17 \cdot 10^{-4}$	7.19x	7.73x	9.60x
6	$3.93 \cdot 10^{-4}$	6	$3.66 \cdot 10^{-4}$	6.40x	7.20x	9.05x
		9	$2.00 \cdot 10^{-4}$	4.60x	5.41x	6.88x

Table 5.7: *Medium mesh: Speedup depending on required accuracy and number of processors for two-level MGRIT with FCF-relaxation and temporal coarsening factor $m_0 = 8$.*

6 Application: Flow through left atrium and ventricle in a Cardiac Resynchronization Therapy patient[1]

This chapter focuses on applying the time-periodic MGRIT algorithm introduced in Section 3.6 to a cardiac application: dynamic flow through the left atrium (LA) and left ventricle (LV).

In general, the cardiac cycle can be split into two phases: diastole and systole, see Figure 6.1. During ventricular diastole, the ventricles relax and fill with blood from the atria. During ventricular systole, the ventricles contract and eject blood to the lungs (right ventricle) and the rest of the body (left ventricle).

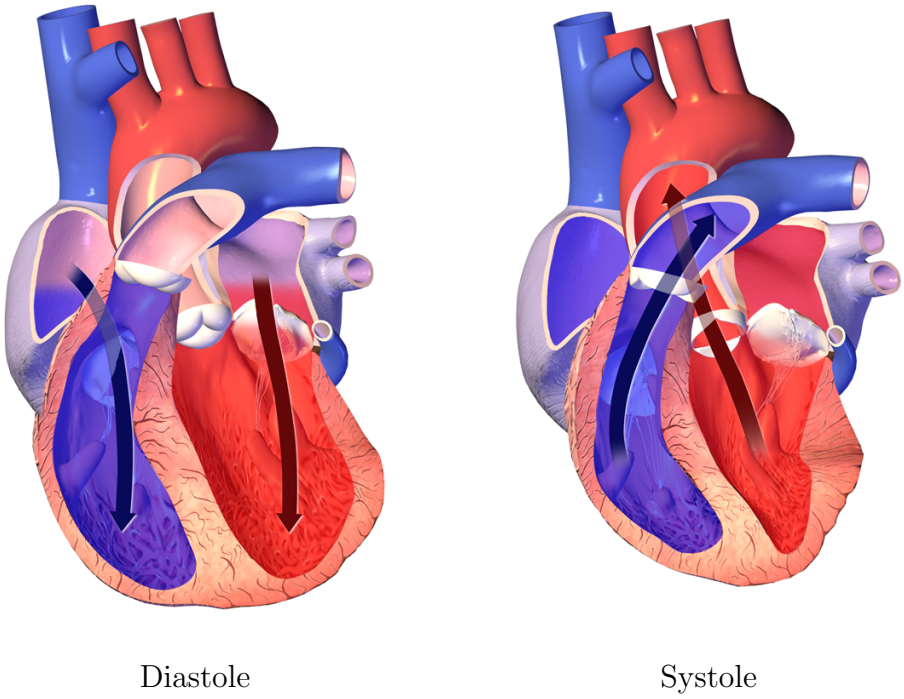


Figure 6.1: *The two phases of the cardiac cycle: diastole and systole. During ventricular diastole, the ventricles relax and fill with blood from the atria. During ventricular systole, the ventricles contract and eject blood to the lungs (right ventricle) and the rest of the body (left ventricle).*[2]

[1] Results in this section were submitted as [62, 101]

[2] Image by BruceBlaus, licensed under CC BY-SA 4.0 [16]. Image adapted from `https://commons.wikimedia.org/wiki/File:Systolevs_Diastole.png`.

The human heart rate is fairly constant unless there are changes in physical activity (e.g., standing up, faster gait, etc.) or in the environment (e.g., temperature change, recognizing a risk, etc.). From experience, for example, the heart rate of a subject lying in an MRI scanner may vary about ± 5 beats per minute from the resting heart rate. Thus, under stable conditions one can assume that the heart rate does not vary significantly. In fact, assuming a constant heart rate is the basis for nearly all clinical imaging techniques, including CT, MRI, and others. Such a situation is ideal to employ the time-periodic MGRIT algorithm as a solver to obtain a periodic steady-state solution more efficiently.

The geometry used in this study (see Figure 6.2), as well as the time-dependent deformation of the LA and LV were extracted from computed tomography (CT) data and provided by Adelaide de Vecchi at King's College London, UK. The CT data set stems from a Cardiac Resynchronization Therapy (CRT) patient that, as we will see in Section 6.5, suffers from an abnormal wall motion and a reduced ejection fraction. Due to the exposure of the patient to ionizing radiation during a CT scan, the acquisition time was kept as short as possible. Here, this means that no (additional) flow data (e.g., using Doppler echocardiography [109]) could be acquired. It is thus unclear, how cardiac function and, in particular, how blood flow through the LA and LV are affected. This is why numerical models come into play.

In the following, the preprocessing pipeline to prepare the raw data for the use in a numerical model is discussed in Section 6.1. The numerical model is presented in Section 6.2. Section 6.3 discusses how a periodic steady-state is detected. Solver settings are detailed in Section 6.4. Numerical results and runtime results are presented in Section 6.5, demonstrating faster performance of time-periodic MGRIT over sequential time-stepping. Section 6.6 gives an outlook on how the numerical model could be improved and how even better runtimes (absolute runtimes, as well as relative to sequential time-stepping) could be achieved by tuning parameters in the future.

6.1 Preprocessing pipeline

In this section, a detailed pipeline is described to prepare the raw image data for use in a numerical model. The received data contained:

- Volume mesh in CHeart X/T/B format for LA and LV
- Deformation of LA and LV: 800 snapshots with 1 ms increments
- Parameters, see Table 6.1:

  Density of blood: $\rho = 1.025\ g/cm^3$

  Viscosity of blood: $\mu = 0.004\ Pa \cdot s$

  Duration of diastole: $0.370\ s$

The following steps were performed to preprocess the data set:

1. Python 3.5.2:
   - Convert CHeart X/T/B format to Paraview VTU format [19]
2. Paraview 5.4.1:

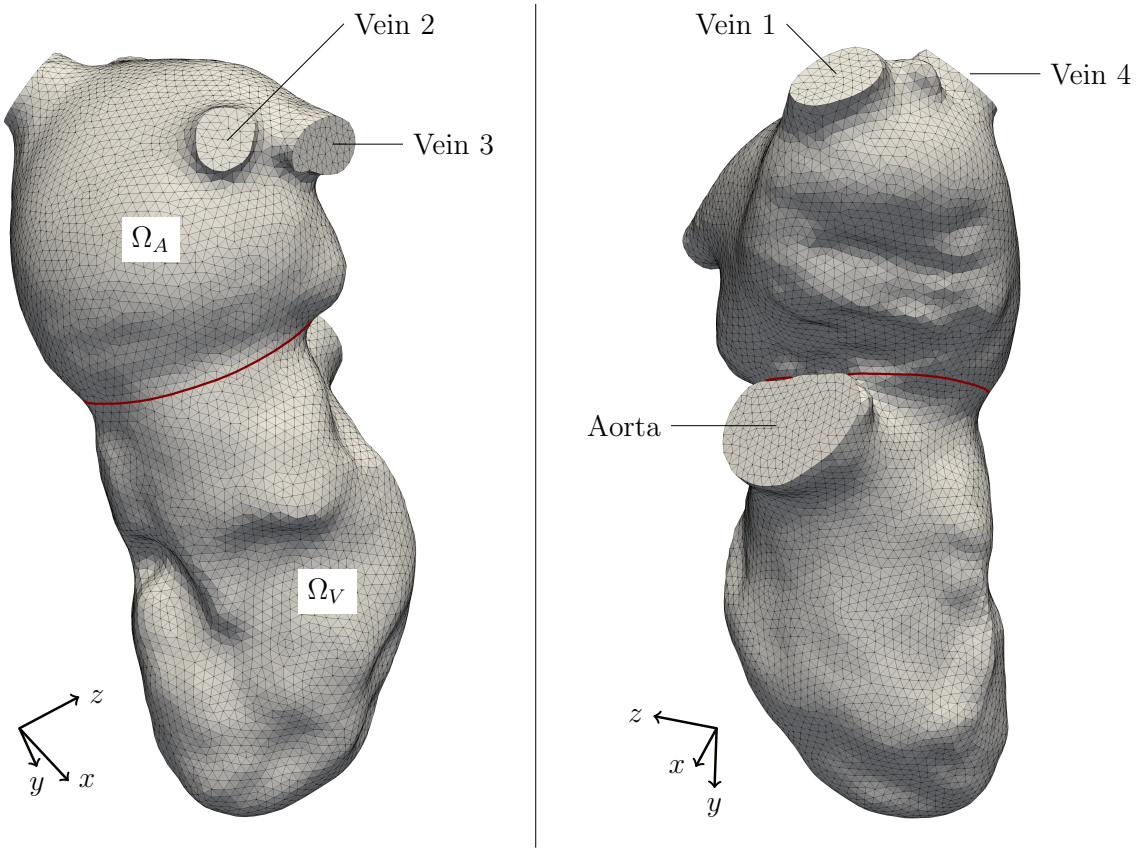


Figure 6.2: *Left atrium and left ventricle: Initial mesh and different inflow / outflow boundaries. Red line indicates coupling surface between atrium (top) and ventricle (bottom).*

- Import VTU mesh
- Clip pulmonary veins to achieve plane boundary, see Figure 6.2
- Identify coupling boundary between LA and LV
- Extract surface including coupling surface
- Export facet file in STL format

3. Simmetrix SimModeler 6.0:
   - Import STL file
   - Create triangular surface mesh
   - Create unique labels for veins, aorta, coupling surface and endocardial wall
   - Create tetrahedral volume mesh
   - Export mesh in UGRID format
4. Python 3.5.2:
   - Convert UGRID format to CHeart X/T/B format [149]
   - Convert X/T/B format to Paraview VTU format

5. Paraview 5.4.1:
   - Import VTU mesh
   - Inspect mesh
   - Extract nodes on coupling surface and export to TXT file
6. Python 3.5.2:
   - Import X/T/B mesh and TXT file
   - Find mesh elements that contain nodes in TXT file
   - Duplicate nodes on coupling surface and renumber nodes on ventricle side to split mesh into two parts (LA and LV)
   - Export X/T/B mesh for LA and LV to separate files
   - Export surface mesh on coupling boundary for Lagrange multiplier (LM) definition, see Section 6.2
   - Convert X/T/B format to Paraview VTU format
7. Paraview 5.4.1:
   - Import VTU mesh
   - Inspect mesh for LA and LV
8. CHeart (Git commit SHA 85e19268):
   - Compute mapping between nodes on LA coupling boundary and LM domain
   - Compute mapping between nodes on LV coupling boundary and LM domain
9. Python 3.5.2:
   - Map boundary motion data to LA and LV meshes for all 800 time steps
   - LA: Compute distance of interior nodes to wall to define diffusion coefficient for ALE mesh motion problem
   - Convert mapped boundary domain motion to VTU format
10. Paraview 5.4.1:
    - Import VTU boundary data
    - Inspect boundary domain motion for LA and LV
11. CHeart (Git commit SHA 85e19268):
    - Run ALE diffusion problem for LA domain to get interior domain motion for steps $1 - 799$, while maintaining mesh quality
    - Run ALE mesh dependent stiffness problem for LV domain to get interior domain motion for steps $1 - 799$, while maintaining mesh quality
    - Compute mesh velocity at step 800, such that space coincides with initial undeformed configuration at step 0; set mesh velocity at step 0 to mesh velocity at 800

- Convert mesh motion data to VTU format

12. Paraview 5.4.1:
    - Import VTU mesh motion data
    - Inspect domain motion for LA and LV
    - Doublecheck mesh quality
13. Setup model in CHeart (Git commit SHA 85e19268), see Section 6.2

Parameter	Symbol	Value
Density	ρ	$1.025\ g/cm^3$
Viscosity	μ	$0.004\ Pa \cdot s$
Cycle length	T	$0.800\ s$
Duration of diastole		$0.370\ s$
Duration of systole		$0.430\ s$
Bounding box (extents)	l_x	$13.680\ cm$
	l_y	$12.044\ cm$
	l_z	$9.111\ cm$
Time step size	δ_0	$0.001\ s$
Spatial step size (LA)	δ_x	$0.163 - 0.290\ cm$
Spatial step size (LV)	δ_x	$0.194 - 0.287\ cm$
Number of time steps	N_0	800
Number of elements (LA)		66484
Number of elements (LV)		69716
Number of DOFs (LA+LV)	N_x	54822
Stabilization parameters [71]	v_f^{max}	$100\ cm/s$
	ζ_1	$\rho\delta_x^{max}/v_f^{max}$
	ζ_2	$\rho\delta_x^{max}v_f^{max}$
	ζ_3	δ_x^{max}/v_f^{max}
Outflow stabilization parameter [106]		$0.2 \cdot \rho$

Table 6.1: *Material and other parameters for left atrium / left ventricle flow simulation. Here, δ_x^{max} refers to the maximum per-element edge length.*

6.2 Model problem

The mitral valve (MV) geometry and deformation were not available from the data set provided, and thus, the MV opening at $t = 0\ s$ is approximated as an ellipse with minor and major axes of $l_{MV}^{min} \approx 2.66\ cm$ and $l_{MV}^{max} \approx 4.64\ cm$. The MV is open for $t \in (0, 0.37]\ s$ (diastole) and closed for $t \in (0.37, 0.8]\ s$ (systole).

The domains and domain boundaries (see Figure 6.2) are denoted as follows:

Ω_A	:	LA domain
Γ_{V1}	:	Vein 1 boundary
Γ_{V2}	:	Vein 2 boundary
Γ_{V3}	:	Vein 3 boundary
Γ_{V4}	:	Vein 4 boundary
Γ_{Vn}	:	$\Gamma_{V1} \cup \Gamma_{V2} \cup \Gamma_{V3} \cup \Gamma_{V4}$
Γ_{CA}	:	Common surface with LV
Γ_{CAo}	:	Part of coupling surface of LA domain that is open during diastole
Γ_{WA}	:	Endocardial wall of LA domain
Ω_V	:	LV domain
Γ_{AV}	:	Aortic valve boundary
Γ_{CV}	:	Common surface with LA
Γ_{CVo}	:	Part of coupling surface of LV domain that is open during diastole
Γ_{WV}	:	Endocardial wall of LV domain

Furthermore, the coupling surface $\Omega_A \cap \Omega_V$ is denoted as:

Ω_{LM}	:	Lagrange multiplier (LM) domain
Γ_{LM}	:	LM boundary

6.2.1 Strong form equations

Blood is modeled as an incompressible Newtonian fluid with density $\rho = 1.025\ g/cm^3$ and viscosity $\mu = 0.004\ Pa \cdot s$, see Table 6.1. Flow in the atrium and ventricle is governed by the nonconservative ALE Navier-Stokes equations for a Newtonian fluid. The strong

form equations, coupling, boundary and initial conditions are given as,

$$\rho\partial_t \boldsymbol{v}_A + \rho\left(\boldsymbol{v}_A - \boldsymbol{w}_A\right)\cdot\nabla_{\boldsymbol{x}}\boldsymbol{v}_A - \nabla_{\boldsymbol{x}}\cdot\boldsymbol{\sigma}_A = \boldsymbol{0} \quad \text{in } \Omega_A, \tag{6.1}$$
$$\nabla_{\boldsymbol{x}}\cdot\boldsymbol{v}_A = 0 \quad \text{in } \Omega_A, \tag{6.2}$$
$$\boldsymbol{v}_A = \boldsymbol{w}_A \quad \text{on } \Gamma_{WA}, \tag{6.3}$$
$$\boldsymbol{\sigma}_A\cdot\boldsymbol{n}_A = \boldsymbol{t}_A \quad \text{on } \Gamma_{Vn}, \tag{6.4}$$
$$\boldsymbol{v}_A(\cdot,0) = \boldsymbol{w}_A \quad \text{on } \Omega_A(0), \tag{6.5}$$

$$\rho\partial_t \boldsymbol{v}_V + \rho\left(\boldsymbol{v}_V - \boldsymbol{w}_V\right)\cdot\nabla_{\boldsymbol{x}}\boldsymbol{v}_V - \nabla_{\boldsymbol{x}}\cdot\boldsymbol{\sigma}_V = \boldsymbol{0} \quad \text{in } \Omega_V, \tag{6.6}$$
$$\nabla_{\boldsymbol{x}}\cdot\boldsymbol{v}_V = 0 \quad \text{in } \Omega_V, \tag{6.7}$$
$$\boldsymbol{v}_V = \boldsymbol{w}_V \quad \text{on } \Gamma_{WV}, \tag{6.8}$$
$$\boldsymbol{v}_V = \boldsymbol{w}_V \quad \text{on } \Gamma_{AV} \text{ for } t \in (0, 0.37], \tag{6.9}$$
$$\boldsymbol{\sigma}_V\cdot\boldsymbol{n}_V = \boldsymbol{t}_V \quad \text{on } \Gamma_{AV} \text{ for } t \in (0.37, 0.8], \tag{6.10}$$
$$\boldsymbol{v}_V = \boldsymbol{w}_V \quad \text{on } \Gamma_{CV}\setminus\Gamma_{CVo} \text{ for } t \in (0, 0.37], \tag{6.11}$$
$$\boldsymbol{v}_V(\cdot,0) = \boldsymbol{w}_V \quad \text{on } \Omega_V(0), \tag{6.12}$$

$$\boldsymbol{v}_A - \alpha_1\boldsymbol{w}_A - \alpha_2\boldsymbol{v}_V = \boldsymbol{0} \quad \text{on } \Omega_{LM}, \tag{6.13}$$
$$\alpha_1\left(\boldsymbol{v}_V - \boldsymbol{w}_V\right) + \alpha_2\left(\boldsymbol{\sigma}_V\cdot\boldsymbol{n}_V + \boldsymbol{\sigma}_A\cdot\boldsymbol{n}_A\right) = \boldsymbol{0} \quad \text{on } \Omega_{LM}, \tag{6.14}$$

with velocity $\boldsymbol{v}_A$ (atrium) and $\boldsymbol{v}_V$ (ventricle), domain velocity $\boldsymbol{w}_A$ (atrium) and $\boldsymbol{w}_V$ (ventricle) obtained from CT data (see Section 6.1), outward boundary normal $\boldsymbol{n}_A$ (atrium) and $\boldsymbol{n}_V$ (ventricle), and Cauchy stress tensor,

$$\boldsymbol{\sigma}_A = \mu\left[\nabla_{\boldsymbol{x}}\boldsymbol{v}_A + \nabla_{\boldsymbol{x}}^T\boldsymbol{v}_A\right] - p_A\boldsymbol{I}, \tag{6.15}$$
$$\boldsymbol{\sigma}_V = \mu\left[\nabla_{\boldsymbol{x}}\boldsymbol{v}_V + \nabla_{\boldsymbol{x}}^T\boldsymbol{v}_V\right] - p_V\boldsymbol{I}, \tag{6.16}$$

with pressure p_A (atrium) and p_V (ventricle). Further, the scalar functions in Equation (6.13) and Equation (6.14) are defined as,

$$\alpha_1 = \begin{cases} 0 & \text{for } t \in (0, 0.37], \\ 1 & \text{for } t \in (0.37, 0.8], \end{cases} \tag{6.17}$$

$$\alpha_2 = \begin{cases} 1 & \text{for } t \in (0, 0.37], \\ 0 & \text{for } t \in (0.37, 0.8], \end{cases} \tag{6.18}$$

and enable switching the coupling constraints between diastole and systole:

- Diastole: Continuity of velocity and traction across coupling boundary.
- Systole: No-slip condition on coupling boundary.

Further, outflow stabilization is employed at the veins (see Equation 6.4) to deal with potential reflow on Γ_{Vn}, and at the aortic valve boundary during systole (see Equation (6.10)),

$$\boldsymbol{t}_A = \frac{\rho\beta}{2} \frac{(\boldsymbol{v}_A \cdot \boldsymbol{n}_A)^2}{(\boldsymbol{v}_A \cdot \boldsymbol{n}_A)^2 + 0.01} \min\{\boldsymbol{v}_A \cdot \boldsymbol{n}_A, 0\} \boldsymbol{v}_A, \tag{6.19}$$

$$\boldsymbol{t}_V = \frac{\rho\beta}{2} \frac{(\boldsymbol{v}_V \cdot \boldsymbol{n}_V)^2}{(\boldsymbol{v}_V \cdot \boldsymbol{n}_V)^2 + 0.01} \min\{\boldsymbol{v}_V \cdot \boldsymbol{n}_V, 0\} \boldsymbol{v}_V. \tag{6.20}$$

Note, that Equation (6.19) and Equation (6.20) provide a modified version of the outflow stabilization proposed in [8] with a scaling of $\beta = 0.2$, as suggested in [106].

6.2.2 Finite element formulation using Lagrange multipliers

"The normal mitral valve peak diastolic velocity is less than 1.3 m/s" [109]. Thus, the peak Reynolds number at the MV opening can be estimated as:

$$Re = \frac{\rho v_f^{max} l_{MV}^{min}}{\mu} \approx \frac{1.025\ g/cm^3 \cdot 130.0\ cm/s \cdot 2.66\ cm}{0.04\ g/(cm \cdot s)} \approx 8861. \tag{6.21}$$

Due to the expected peak Reynolds number of 8861, the flow is modeled by a stabilized general Galerkin scheme (instead of using, e.g., an inf-sup stable Taylor-Hood finite element discretization scheme) for the incompressible Navier-Stokes equations; namely the cG(1)cG(1) scheme as given in the study of Hoffman et al. [71]. The scheme was implemented in CHeart [18, 94], and validated in a previous work [67].

On the coupling domain, Lagrange mulitplier variables are defined as $\boldsymbol{\lambda}_A = \boldsymbol{\sigma}_A \cdot \boldsymbol{n}_A$ and $\boldsymbol{\lambda}_V = -\boldsymbol{\sigma}_V \cdot \boldsymbol{n}_V$ to enforce the coupling constraints, see Section 6.2.1, such that during dyastole: $\boldsymbol{\lambda}_A - \boldsymbol{\lambda}_V = \boldsymbol{0}$.

6.2.2.1 Spatiotemporal discretization

The temporal domain is discretized using 800 equidistant time steps $0\ s = t_0 < t_1 < \ldots < t_{N_0-1} = 0.8\ s$ with constant time step size $\delta_0 = t_{n+1} - t_n = 0.001\ s$ for $n = 0, \ldots, N_0 - 2$. Similar to Scheme I in Section 5.3.3, the time discretization had originally been implemented in CHeart using an average between the current and previous velocity. In preliminary tests, this resulted in unphysical oscillations at inflow and outflow boundaries (similar to those observed for the linear elasticity problem in Section 5.3) when MGRIT was used for parallel-in-time integration. Thus, a pure backward Euler time discretization scheme (similar to Scheme II in Section 5.3.6) was implemented that results in a stable PinT algorithm.

The atrial and ventricular domains, Ω_A and Ω_V, are discretized using 66484 and 69716 tetrahedral elements. The coupling domain Ω_{LM} is discretized using 945 triangular elements that conform with the tetrahedral elements at the coupling boundary of the atrium and ventricle. In the following, we use Ω_A^0 and Ω_V^0 to refer to the initial meshes discretizing atrium and ventricle; further, Ω_A^n and Ω_V^n refer to the respective current meshes. In particular, due to the construction of the domain deformation (see Section 6.1): $\Omega_A^0 = \Omega_A^{N_0-1}$ and $\Omega_V^0 = \Omega_V^{N_0-1}$.

6.2.2.2 Weak formulation

Finite element discretizations were constructed using $\mathbb{P}^1-\mathbb{P}^1$ elements for fluid velocity and pressure and $\mathbb{P}^1$ elements for the Lagrange multipliers on the coupling domain, resulting in 55842 degrees-of-freedom (DOFs). The discrete solution at each time step n can then be written as follows:

Find $\boldsymbol{s}^n := (\boldsymbol{v}_A^n, \boldsymbol{v}_V^n, \boldsymbol{\lambda}_A^n, \boldsymbol{\lambda}_V^n, p_A^n, p_V^n) \in \boldsymbol{\mathcal{S}}_D^h := \boldsymbol{\mathcal{V}}_D^h \times \boldsymbol{\mathcal{U}}_D^h \times \boldsymbol{\mathcal{M}}_0^h \times \boldsymbol{\mathcal{N}}_0^h \times \mathcal{W}_A^h \times \mathcal{W}_V^h$, such that for every $d := (\boldsymbol{y}_A, \boldsymbol{y}_V, \boldsymbol{\varphi}_A, \boldsymbol{\varphi}_V, q_A, q_V) \in \boldsymbol{\mathcal{S}}_0^h := \boldsymbol{\mathcal{V}}_0^h \times \boldsymbol{\mathcal{U}}_0^h \times \boldsymbol{\mathcal{M}}_0^h \times \boldsymbol{\mathcal{N}}_0^h \times \mathcal{W}_A^h \times \mathcal{W}_V^h$:

$$
\begin{aligned}
&R\left(\boldsymbol{s}^n, \boldsymbol{s}^{n-1}, \boldsymbol{w}_A^n, \boldsymbol{w}_V^n; \boldsymbol{d}\right) \\
&\quad := \int_{\Omega_A^n} \left[\frac{\boldsymbol{v}_A^n - \boldsymbol{v}_A^n}{\delta_0} + (\boldsymbol{v}_A^n - \boldsymbol{w}_A^n) \cdot \nabla_{\boldsymbol{x}} \boldsymbol{v}_A^n\right] \cdot \boldsymbol{y}_A \, d\boldsymbol{x} \\
&\qquad + \int_{\Omega_A^n} \boldsymbol{\sigma}_A^n : \nabla_{\boldsymbol{x}} \boldsymbol{y}_A + \boldsymbol{\varphi}_A \nabla_{\boldsymbol{x}} \cdot \boldsymbol{v}_A^n \, d\boldsymbol{x} \\
&\qquad - \int_{\Gamma_{Vn}^n} \boldsymbol{t}_A^n \cdot \boldsymbol{y}_A \, d\boldsymbol{x} \\
&\qquad + \int_{\Omega_V^n} \left[\frac{\boldsymbol{v}_V^n - \boldsymbol{v}_V^n}{\delta_0} + (\boldsymbol{v}_V^n - \boldsymbol{w}_V^n) \cdot \nabla_{\boldsymbol{x}} \boldsymbol{v}_V^n\right] \cdot \boldsymbol{y}_V \, d\boldsymbol{x} \\
&\qquad + \int_{\Omega_V^n} \boldsymbol{\sigma}_V^n : \nabla_{\boldsymbol{x}} \boldsymbol{y}_V + \boldsymbol{\varphi}_V \nabla_{\boldsymbol{x}} \cdot \boldsymbol{v}_A^n \, d\boldsymbol{x} \\
&\qquad - \alpha_1 \int_{\Gamma_{AV}^n} \boldsymbol{t}_V^n \cdot \boldsymbol{y}_V \, d\boldsymbol{x} \\
&\qquad + \int_{\Omega_{LM}^0} \boldsymbol{\varphi}_A \cdot (\boldsymbol{v}_A^n - \alpha_1 \boldsymbol{w}_A^n - \alpha_2 \boldsymbol{v}_V^n) \, d\boldsymbol{x} \\
&\qquad + \int_{\Omega_{LM}^0} \alpha_1 \boldsymbol{\varphi}_V (\boldsymbol{v}_A^n - \boldsymbol{v}_V^n) \, d\boldsymbol{x} \\
&\qquad + \int_{\Omega_{LM}^0} \alpha_2 (\boldsymbol{\lambda}_A \cdot \boldsymbol{y}_A - \boldsymbol{\lambda}_V \cdot \boldsymbol{y}_V) \, d\boldsymbol{x} \\
&\qquad + \zeta_1 \int_{\Omega_A^n} [(\boldsymbol{v}_A^n - \boldsymbol{w}_A^n) \cdot \nabla_{\boldsymbol{x}} \boldsymbol{v}_A^n] \cdot [\boldsymbol{v}_A^n \cdot \nabla_{\boldsymbol{x}} \boldsymbol{y}_A + \nabla_{\boldsymbol{x}} q_A] \, d\boldsymbol{x} \\
&\qquad + \zeta_2 \int_{\Omega_A^n} (\nabla_{\boldsymbol{x}} \cdot \boldsymbol{v}_A) \cdot (\nabla_{\boldsymbol{x}} \cdot \boldsymbol{y}_A) \, d\boldsymbol{x} \\
&\qquad + \zeta_3 \int_{\Omega_A^n} (\nabla_{\boldsymbol{x}} p_A^n) \cdot (\boldsymbol{v}_A^n \cdot \nabla_{\boldsymbol{x}} \boldsymbol{y}_A + \nabla_{\boldsymbol{x}} q_A) \, d\boldsymbol{x} \\
&\qquad + \zeta_1 \int_{\Omega_V^n} [(\boldsymbol{v}_V^n - \boldsymbol{w}_V^n) \cdot \nabla_{\boldsymbol{x}} \boldsymbol{v}_V^n] \cdot [\boldsymbol{v}_V^n \cdot \nabla_{\boldsymbol{x}} \boldsymbol{y}_V + \nabla_{\boldsymbol{x}} q_V] \, d\boldsymbol{x} \\
&\qquad + \zeta_2 \int_{\Omega_V^n} (\nabla_{\boldsymbol{x}} \cdot \boldsymbol{v}_V) \cdot (\nabla_{\boldsymbol{x}} \cdot \boldsymbol{y}_V) \, d\boldsymbol{x} \\
&\qquad + \zeta_3 \int_{\Omega_V^n} (\nabla_{\boldsymbol{x}} p_V^n) \cdot (\boldsymbol{v}_V^n \cdot \nabla_{\boldsymbol{x}} \boldsymbol{y}_V + \nabla_{\boldsymbol{x}} q_V) \, d\boldsymbol{x}
\end{aligned}
\tag{6.22}
$$

where the stabilization terms with parameters ζ_1, ζ_2 and ζ_3 (see Table 6.1) are added according to the cG(1)cG(1) scheme [71].

Note, that the domain velocity $\boldsymbol{w}_A^n$ and $\boldsymbol{w}_V^n$ is provided from the CT data, see Section 6.1.

The definitions of the function spaces are:

$$S^1\left(\Omega_i^0\right) = \{f : \Omega_i^0 \to \mathbb{R} \mid f \in \mathcal{C}^0(\bar{\Omega}_i^0),\ f|_{\tau_e} \in \mathbb{P}^1(\tau_e),\ \forall\ \tau_e \in \mathcal{T}_i^h\}, \tag{6.23}$$

which represent the first-order piecewise continuous polynomial spaces defined on Ω_i^0. Consequently, we can define:

$$\begin{aligned}
\boldsymbol{\mathcal{V}}^h &= \left[S^1(\Omega_A^0)\right]^3, & \boldsymbol{\mathcal{U}}^h &= \left[S^1(\Omega_V^0)\right]^3,\\
\boldsymbol{\mathcal{M}}^h &= \left[S^1(\Gamma_{CA})\right]^3, & \boldsymbol{\mathcal{N}}^h &= \left[S^1(\Gamma_{VA})\right]^3,\\
\mathcal{W}_A^h &= S^1(\Omega_A^0), & \mathcal{W}_V^h &= S^1(\Omega_V^0).
\end{aligned}$$

Further restrictions are applied on the respective spaces for the atrium and ventricle in order to incorporate the Dirichlet and homogeneous boundary conditions:

$$\boldsymbol{\mathcal{V}}_D^h = \{\boldsymbol{v} \in \boldsymbol{\mathcal{V}}^h \mid \boldsymbol{v} = \boldsymbol{w}_A \text{ on } \Gamma_{WA}^h\}, \tag{6.24}$$

$$\boldsymbol{\mathcal{V}}_0^h = \{\boldsymbol{v} \in \boldsymbol{\mathcal{V}}^h \mid \boldsymbol{v} = \boldsymbol{0} \text{ on } \Gamma_{WA}^h\}, \tag{6.25}$$

$$\begin{aligned}
\boldsymbol{\mathcal{U}}_D^h = \{\boldsymbol{v} \in \boldsymbol{\mathcal{U}}^h \mid\ & \boldsymbol{v} = \boldsymbol{w}_V \text{ on } \Gamma_{WV}^h,\\
& \boldsymbol{v} = \boldsymbol{w}_V \text{ on } \Gamma_{AV}^h \text{ for } t \in (0, 0.37],\\
& \boldsymbol{v} = \boldsymbol{w}_V \text{ on } \Gamma_{CV}^h \setminus \Gamma_{CVo}^h \text{ for } t \in (0.37, 0.8]\},
\end{aligned} \tag{6.26}$$

$$\begin{aligned}
\boldsymbol{\mathcal{U}}_0^h = \{\boldsymbol{v} \in \boldsymbol{\mathcal{U}}^h \mid\ & \boldsymbol{v} = \boldsymbol{0} \text{ on } \Gamma_{WV}^h,\\
& \boldsymbol{v} = \boldsymbol{0} \text{ on } \Gamma_{AV}^h \text{ for } t \in (0, 0.37],\\
& \boldsymbol{v} = \boldsymbol{0} \text{ on } \Gamma_{CV}^h \setminus \Gamma_{CVo}^h \text{ for } t \in (0.37, 0.8]\},
\end{aligned} \tag{6.27}$$

and similarly for the coupling domains,

$$\boldsymbol{\mathcal{M}}_0^h = \{\boldsymbol{\lambda} \in \gamma_{\Gamma_{CA}} \boldsymbol{\mathcal{V}}_0^h\}, \tag{6.28}$$

$$\boldsymbol{\mathcal{N}}_0^h = \{\boldsymbol{\lambda} \in \gamma_{\Gamma_{CV}} \boldsymbol{\mathcal{U}}_0^h\}, \tag{6.29}$$

where $\gamma_{\Gamma_{CA}}$ and $\gamma_{\Gamma_{CV}}$ are the trace operators on Γ_{CA} and Γ_{CV}, respectively.

6.3 Obtaining and detecting a periodic steady-state

Similar to problems described in previous chapters (e.g., see Section 3.6, Section 5.4, etc.), the system has to be driven to a periodic steady-state because no data is available to prescribe a meaningful initial flow condition. Thus, zero initial flow is assumed and the system is driven to a time-periodic steady-state.

To detect such a periodic steady-state, the rate of inflow / outflow at the pulmonary veins[3] is measured over the entire cycle length at 10 ms increments.

[3]Note, that the flow rates at the MV and AV surfaces can be captured with machine precision since the fluxes reflect the prescribed volume change in the ventricle during diastole (flow between LA and LV) and systole (flow across AV surface).

6.4 Solver

CHeart (Git commit SHA 85e19268) [18, 94] and the matrix solver MUMPS (version 5.0.0) [2] were used to solve the considered problem using sequential time-stepping. A Shamanskii-Newton-Raphson (SNR) method [129] was employed to solve the nonlinear system (absolute tolerance 10^{-8}) and to enable reuse of the Jacobian matrix and its inverse, which reduces the computational cost of the algorithm. To obtain a periodic steady-state, 10 cycles are simulated.

The XBraid (version 2.3.0) [158] library was employed to enable parallel-in-time integration. XBraid wrappers and extensions were written to use CHeart in conjuction with the time-periodic MGRIT algorithm described in Section 3.6. When using MGRIT, separate Jacobian matrix objects were stored for the Jacobian on each time-grid level; that is, the appropriate Jacobian matrix object for a given grid level (i.e. time step size) was selected to enable using the more efficient SNR method, similar to sequential time-stepping. The *skip-first-down* option was used to generate a cheap initial space-time guess and XBraid was forced to perform 10 time-periodic MGRIT iterations to obtain a periodic steady-state. Further XBraid settings are given in Table 6.2.

In both cases, data are exported at 10 ms increments.

Parameter	Value
Number of time grids	$n_l = 2$
Temporal coarsening factor	$m_0 = 8$
Number of FC-relaxation steps	$r = 1$
Skip-first-down option	Enabled
Forced number of iterations	10

Table 6.2: *XBraid settings for the LA / LV flow problem.*

6.5 Results

6.5.1 Fluid volume

First, consider the volume of the LV over time, see Figure 6.3. For example, the end-diastolic volume (EDV) [73] at $t = 0.37$ s is approximately,

$$EDV \approx 192.50 \ cm^3, \tag{6.30}$$

and the end-systolic volume (ESV) at $t = 0.8$ s is approximately,

$$ESV \approx 152.46 \ cm^3.$$

Thus, the stroke volume (SV) is,

$$SV = EDV - ESV \approx 40.04 \ cm^3,$$

and the left ventricular ejection fraction (LVEF) is,

$$LVEF = 100 \ SV/EDV \approx 20.80 \ \%.$$

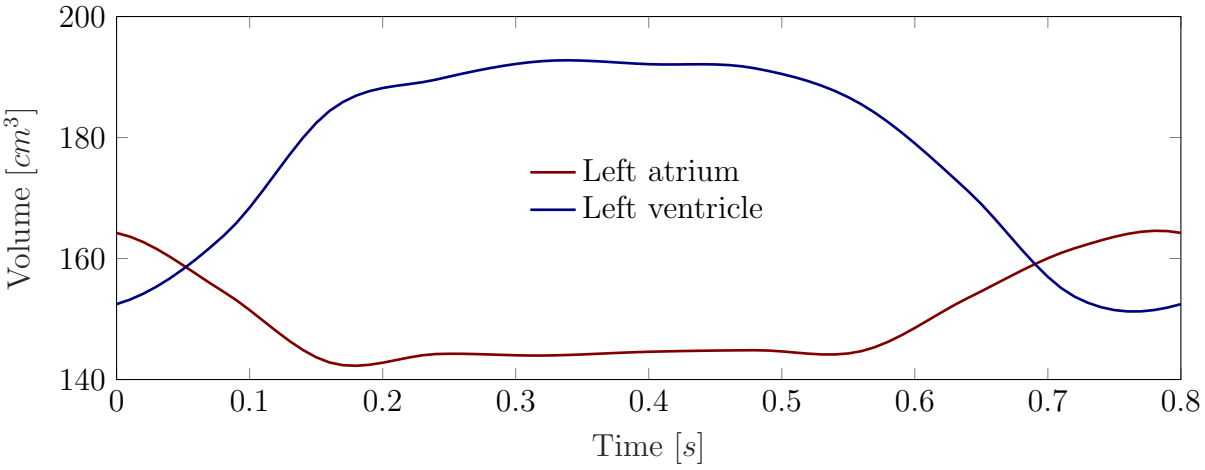


Figure 6.3: *Fluid volume in left atrium and left ventricle over one cardiac cycle.*

Thus, according to the LVEF categories defined in the *European Society of Cardiology Guidelines 2016* [115, Section 3.2.1], the LVEF is significantly reduced:

$\geq 50\%$: normal
$40 - 49\%$: midrange
$< 40\%$: reduced

Further, the SV is reduced and the EDV and ESV are high compared to the normal ranges [98, Table 1] in healthy individuals:

Quantity	Subjects	CAFU method [98]	QGS method [98]
ESV	Women	28 ml	25 ml
	Men	46 ml	47 ml
EDV	Women	99 ml	75 ml
	Men	132 ml	112 ml
SV	Women	71 ml	50 ml
	Men	86 ml	65 ml
LVEF	Women	72 ml	68 ml
	Men	65 ml	59 ml

6.5.2 Periodic steady-state: Flow rate at pulmonary veins

Figure 6.4 illustrates the flow rate at all pulmonary veins over the duration of one cycle for all even numbered cycles, as obtained with sequential time-stepping. Figure 6.4 highlights that it is necessary to run multiple cycles to minimize cycle-to-cycle variations. Furthermore, the flow rate at vein 1 converges to a periodic steady-state quicker than the flow rate at vein 4, however, a periodic steady-state is obtained at each vein.

On the other hand, Figure 6.5 highlights a similar convergence behavior when the time-periodic MGRIT algorithm is employed to obtain a periodic steady-state solution. In contrast to the sequential time-stepping solution (see Figure 6.4), however, the initial error in the flow rates can be reduced slightly more quickly. It seems reasonable to assume that this is achieved by providing a better initial guess of the initial condition by means of XBraid's *skip-first-down* option (see Section 6.4) and by updating the initial condition multiple times during each time-periodic MGRIT iteration (see Section 3.6).

Selecting the space-time solution for cycle 10 or iteration 10 as the periodic steady-state solution, the error for each cycle and each iteration can be quantified. Figure 6.6 highlights that a periodic steady-state can be achieved by running sequential time-stepping for 7 cycles or time-periodic MGRIT for 6 iterations (note, that this does not include the first coarse-grid solve according to XBraid's *skip-first-down* option). Thus, in terms of convergence to a periodic steady-state, one time-periodic MGRIT iteration corresponds to running one cycle using sequential time-stepping. This is ideal in the sense that each MGRIT iteration can update the initial condition at relatively cheaper cost (compared to running one cycle using sequential time-stepping), whereas the underlying multigrid algorithm achieves convergence in the interior of the temporal domain; thus, the significant potential benefit in reducing the wall clock time of the algorithm, see Section 6.3.

6.5.3 Flow in left atrium and ventricle

Figure 6.7 and Figure 6.8 illustrate the flow in the left atrium and ventricle over cycle 10. During diastole, flow through the pulmonary veins causes a slight spiraling of flow in the left atrium and a filling of the left ventricle through the mitral valve. Peak velocities occurs at the pulmonary vein boundaries and the mitral valve section, see Figure 6.7. The peak flow through the MV was estimated as 46.18 cm/s, such that the corresponding Reynolds number is given as,

$$Re = \frac{\rho v_f^{max} l_c}{\mu} = \frac{1.025 g/cm^3 \cdot 46.18 cm/s \cdot 2.66 cm}{0.04 g/(cm \cdot s)} \approx 3147.74. \tag{6.31}$$

Further, the strong jet-like flow through the mitral valve causes a circular flow structure to develop around the jet with reflow regions between the jet and the endocardial wall (see Figure 6.8, top-left), which moves downward after the MV closes (see Figure 6.8, top-right and row 2).

During systole, the vortices near the MV decelerate and blood is ejected through the aortic valve boundary, e.g., see $t = 0.6\ s$ and $t = 0.7\ s$ in Figure 6.8.

6.5.4 Runtimes & Speedup

Here, runtimes and errors for sequential time-stepping and time-periodic MGRIT are compared. Timing results were obtained on TheoSim (see Appendix A) with 8 or 16 processors per node. Errors are those reported in Figure 6.6b and Figure 6.6c.

As can be seen in Figure 6.9, spatial parallelism can reduce the wall clock time for sequential time-stepping. For example, the speedup of switching from 8 processors to 16 processors in space is 1.58x, see Table 6.3. Spatial parallelism, however, saturates with

Algorithm	p_x	p_t	$p_x p_t$	Wall clock time $[s]$	Speedup	Efficiency
Sequential time-stepping	8	–	8	15814		
	16	–	16	10017	1.58x	0.79
	32	–	32	12199	1.30x	0.33
Time-periodic MGRIT	16	5	80	4940	3.20x	0.32
	8	10	80	4310	3.67x	0.37
	8	20	160	3470	4.56x	0.23

Table 6.3: *Runtimes and respective speedups for sequential time-stepping (7 cycles) and time-periodic MGRIT (6 iterations) with p_x processors in space and p_t processors in time.*

a subsequent increase of the *time-to-solution* for 32 processors compared to 16 processors in space.

For any required space-time error, time-periodic MGRIT is faster than sequential time-stepping. The results in Figure 6.9 – Figure 6.12 further highlight that it is better to employ more processors for parallelizing in the temporal domain intead of the spatial domain. For example, when using a total number of 80 processors, the wall clock time for time-periodic MGRIT can be reduced when more processors are assigned to the temporal component than the spatial component.

6.6 Future work

In the future, spatial mesh refinement should be investigated as the considered mesh was fairly coarse for the employed cG(1)cG(1) scheme. It should be assessed, how spatial mesh refinement changes the flow rates at the pulmonary veins. For example, in this study mostly outflow occured at vein 1, which is not necessarily physical. This could, however, be related to employing outflow stabilization (see Section 6.2.1), which can affect the solution and has to be studied.

Furthermore, it needs to be assessed whether the time step size of 1 ms is sufficiently small. If not, this could increase the potential speedup when using time-periodic MGRIT by employing a larger temporal coarsening factor or more processors to parallelize in the time component. Additionally, switching to a true multilevel scheme (i.e. more then two time grid levels) could prove beneficial and reduce the overall runtime of the algorithm. Switching from diastole to systole (at $t = 0.37$) at a fine-grid C-point might further reduce the wall clock time.

Including a model for the mitral valve to capture the dynamics and deformation of the valve and its effect on the surrounding flow could be a potential improvement of the overall model of the system; for example, by employing a PUFEM approach as proposed in [5].

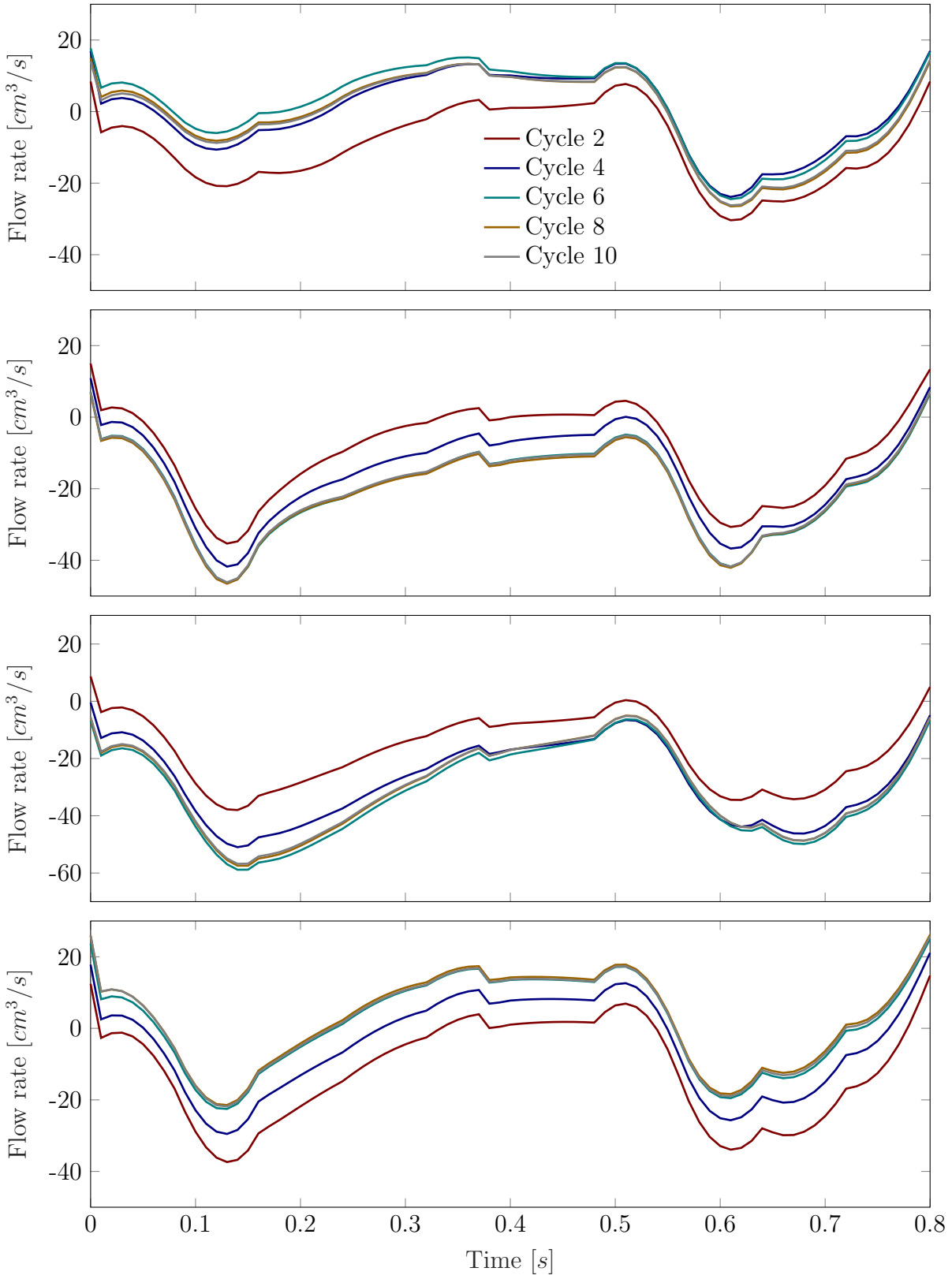


Figure 6.4: *Flow rates at pulmonary veins* $1-4$ *(top to bottom): A periodic steady-state can be achieved by running multiple cycles using sequential time stepping. Note: A negative flow rate corresponds to inflow.*

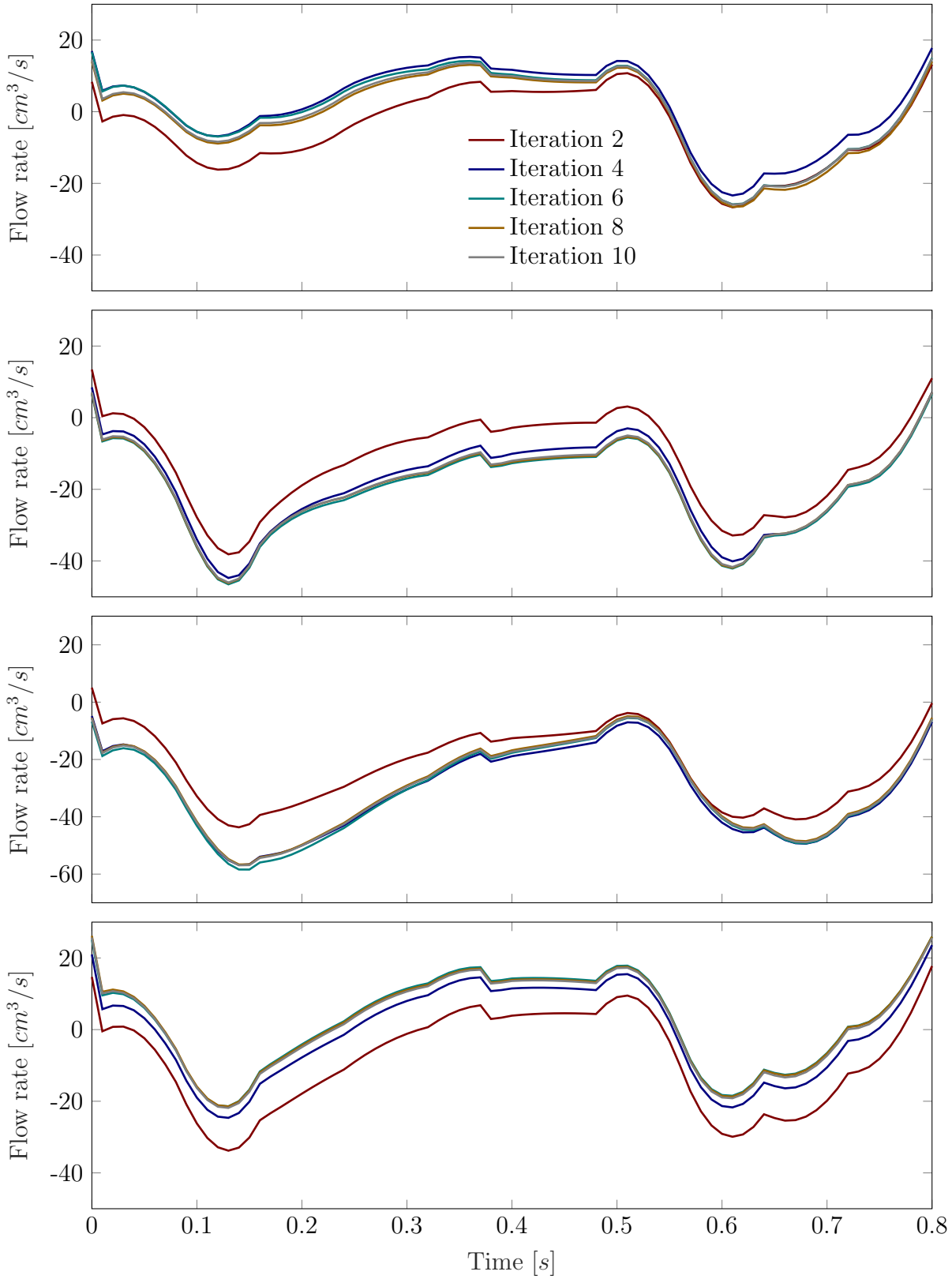
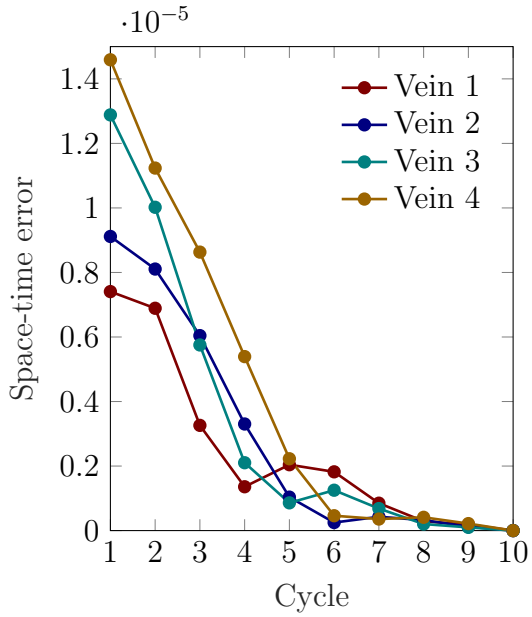
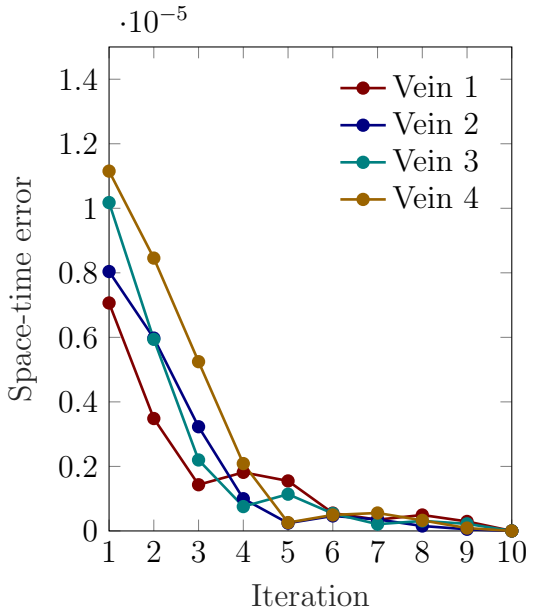


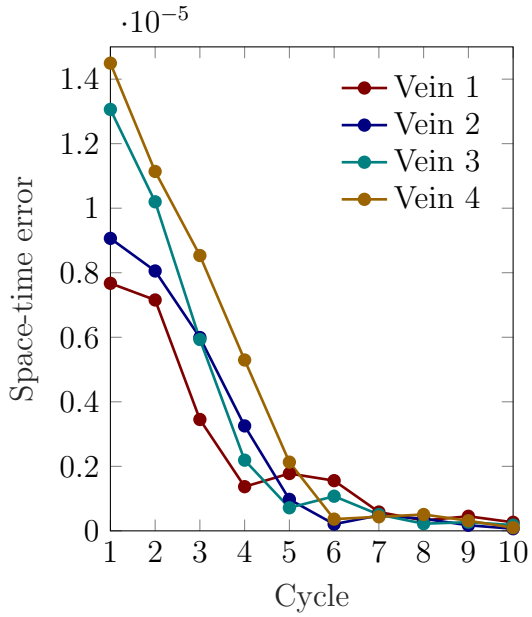
Figure 6.5: *Flow rates at pulmonary veins $1-4$ (top to bottom): A periodic steady-state can be achieved by running multiple iterations using time-periodic MGRIT. Note: A negative flow rate corresponds to inflow.*



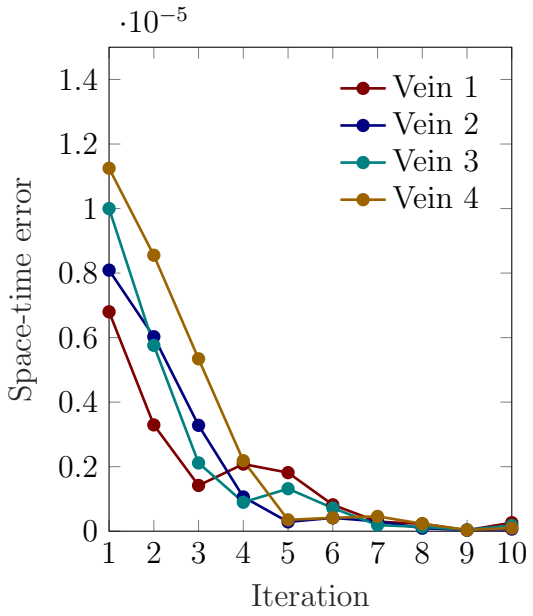
(a) *Per-cycle error compared to cycle* 10.



(b) *Per-iteration error compared to iteration* 10.



(c) *Per-cycle error compared to iteration* 10.



(d) *Per-iteration error compared to cycle* 10.

Figure 6.6: *Convergence of flow rates at veins* $1-4$ *boundaries with reference data selected as cycle* 10 *from sequential time-stepping or iteration* 10 *from time-periodic MGRIT. Space-time error using norm defined in Equation* (2.129).

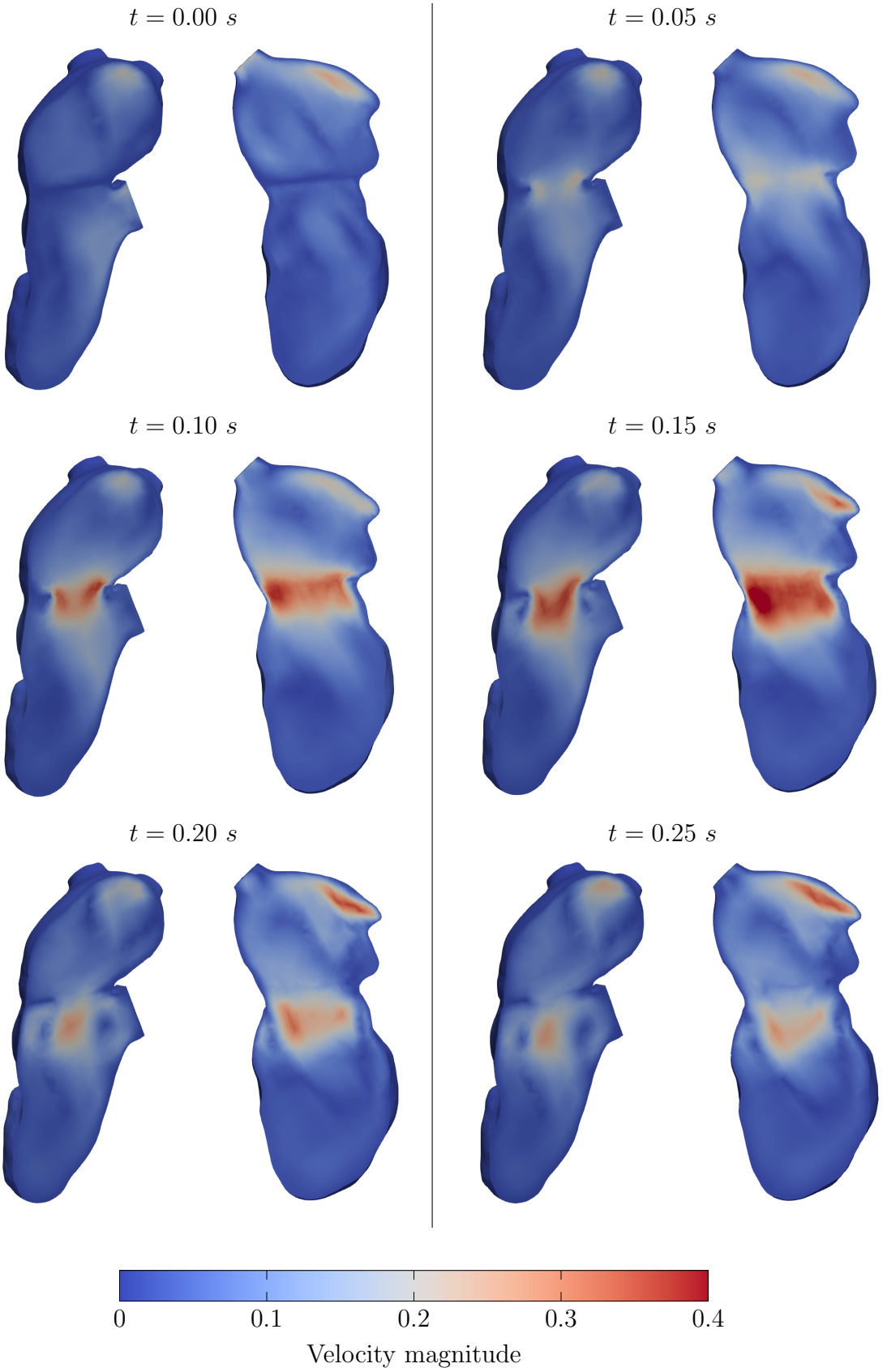


Figure 6.7: *Flow at midsections in the left atrium and ventricle during diastole for two different perspectives, similar to those in Figure 6.2.*

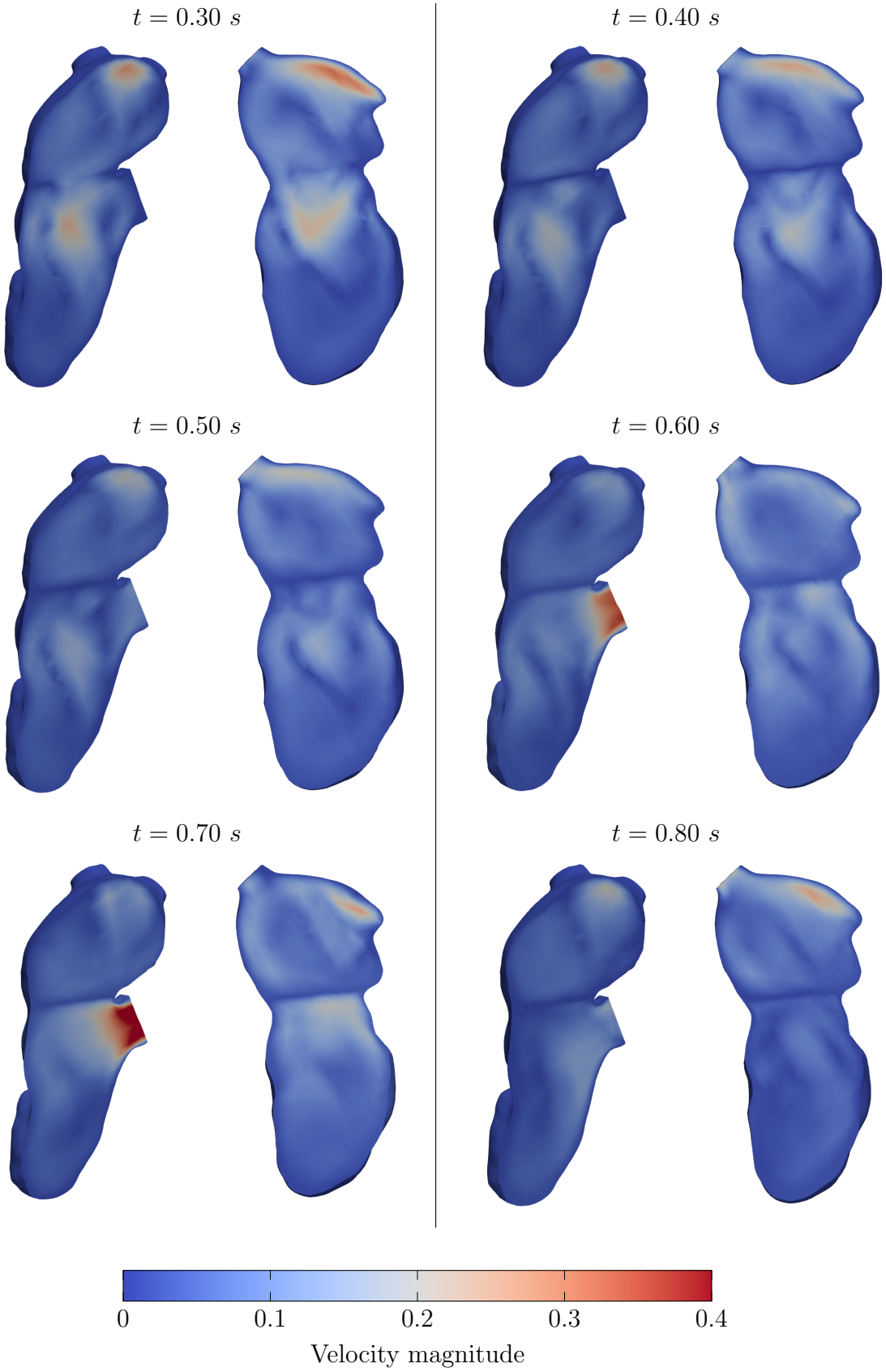


Figure 6.8: *Flow at midsections in the left atrium and ventricle at the end of diastole and during systole for two different perspectives, similar to those in Figure 6.2.*

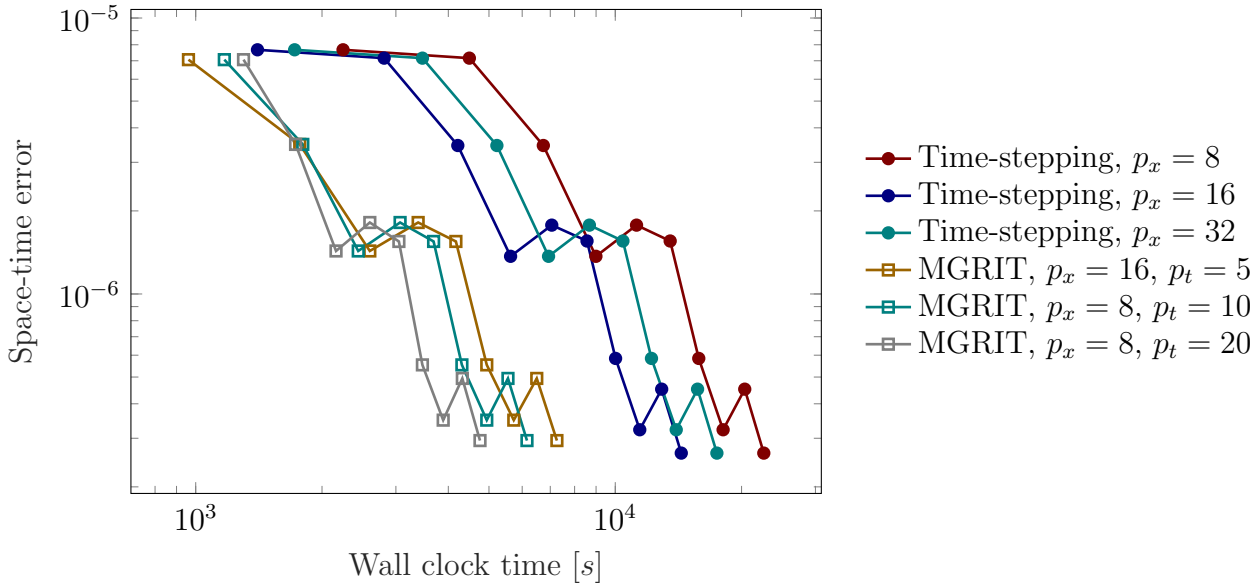


Figure 6.9: *Vein 1: Space-time error (flow rate) compared to wall clock time for sequential time stepping and time-periodic MGRIT. MGRIT is consistenly faster than sequential time-stepping and using a larger number of processors in time (denoted as p_t) instead of space (denoted as p_x) reduces the wall clock time of time-periodic MGRIT.*

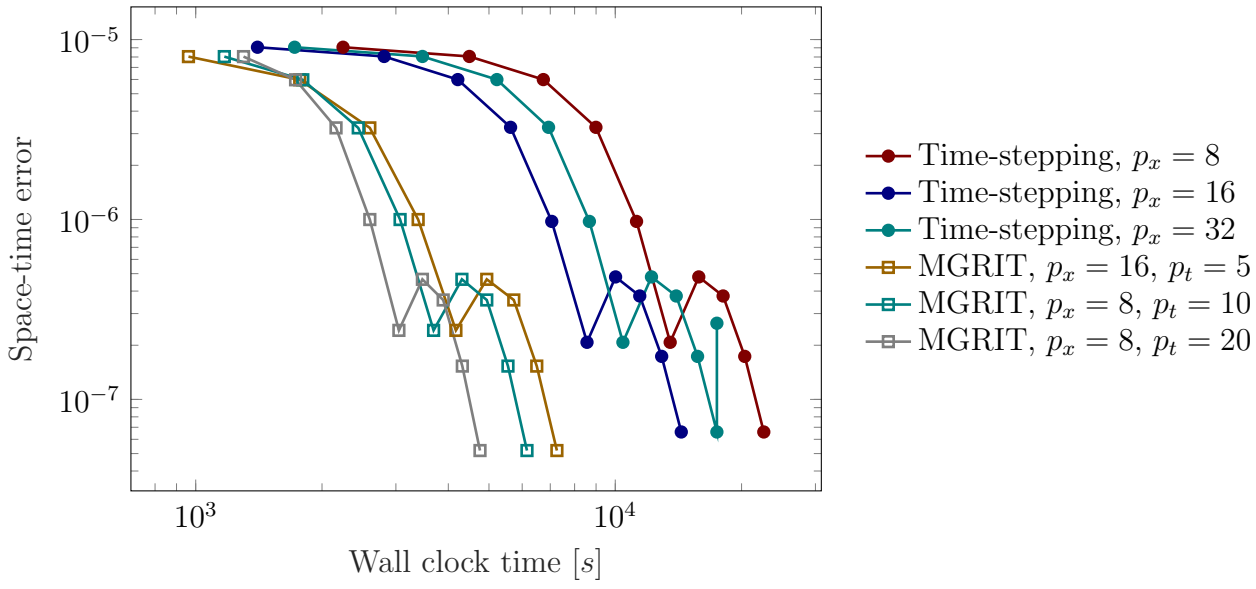


Figure 6.10: *Vein 2: Space-time error (flow rate) compared to wall clock time for sequential time stepping and time-periodic MGRIT. MGRIT is consistenly faster than sequential time-stepping and using a larger number of processors in time (denoted as p_t) instead of space (denoted as p_x) reduces the wall clock time of time-periodic MGRIT.*

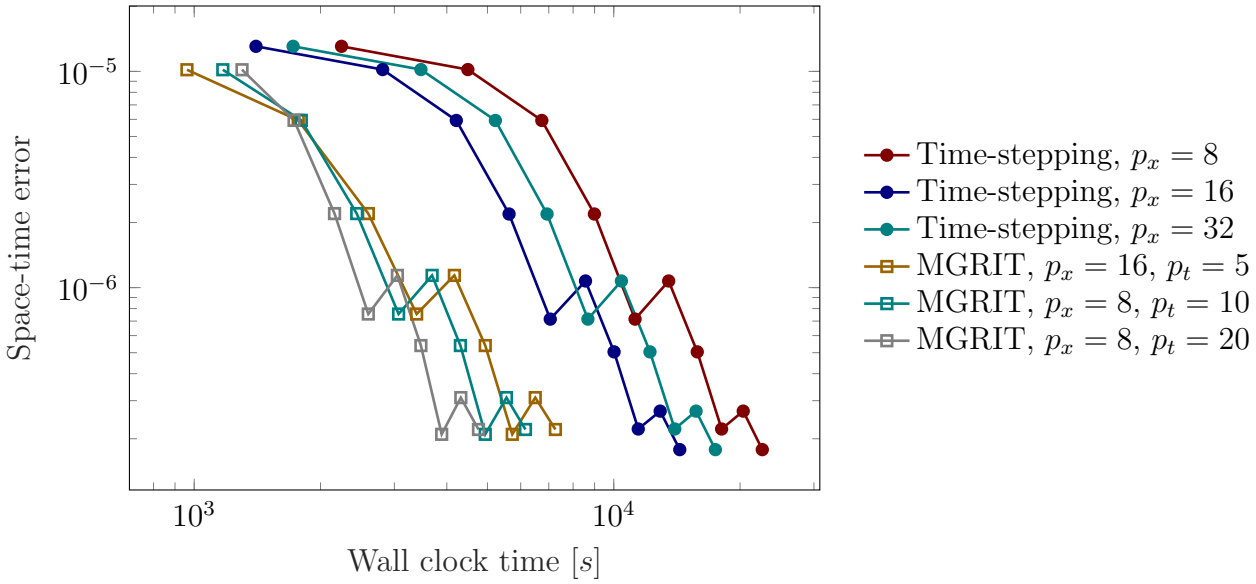


Figure 6.11: *Vein 3: Space-time error (flow rate) compared to wall clock time for sequential time stepping and time-periodic MGRIT. MGRIT is consistenly faster than sequential time-stepping and using a larger number of processors in time (denoted as p_t) instead of space (denoted as p_x) reduces the wall clock time of time-periodic MGRIT.*

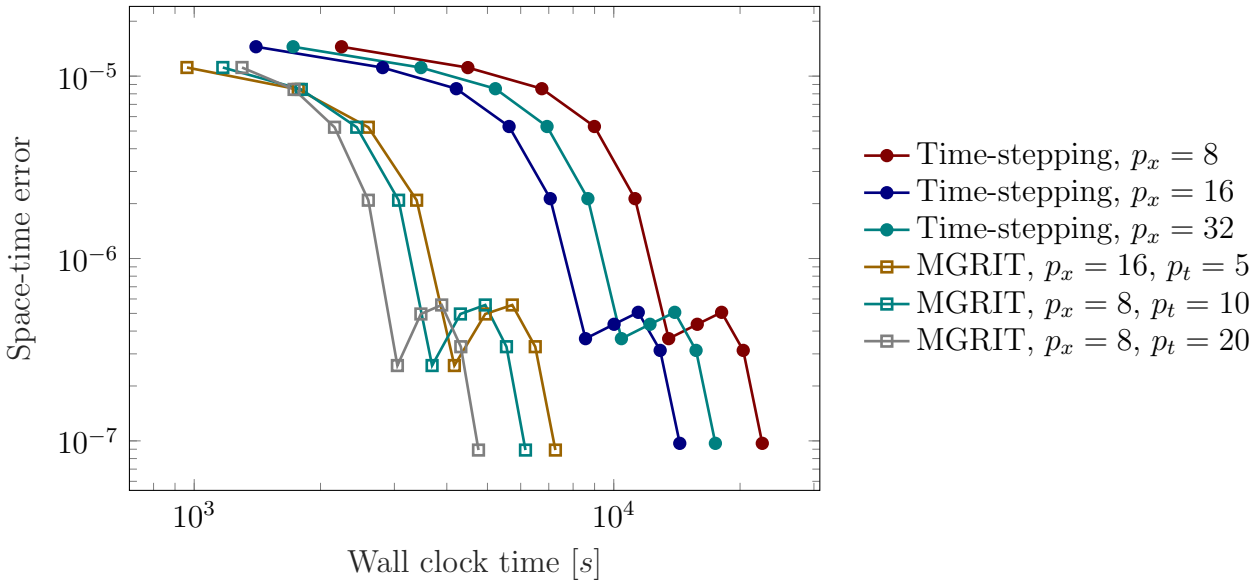


Figure 6.12: *Vein 4: Space-time error (flow rate) compared to wall clock time for sequential time stepping and time-periodic MGRIT. MGRIT is consistenly faster than sequential time-stepping and using a larger number of processors in time (denoted as p_t) instead of space (denoted as p_x) reduces the wall clock time of time-periodic MGRIT.*

7 Summary & Outlook

7.1 Summary

In this Ph.D. Thesis, a multilevel convergence framework for multigrid-reduction-in-time (MGRIT) was derived for various parameter settings and applied to a number of test problems relevant to fluid-structure interaction (FSI) modeling: the anisotropic diffusion equation, the wave equation, the linear elasticity equation, the Stokes and Navier-Stokes equations, and for a novel class of analytic solutions for coupled linear and nonlinear FSI.

The derived *a priori* upper bounds range from sharp but computationally expensive numerical upper bounds to cheap analytic formulae bounding the worst-case convergence of MGRIT. Furthermore, approximate convergence factors were proposed that capture MGRIT convergence with reasonable quality for cases, when analytic formulae are not available and the computational cost of numerical bounds is prohibitive. The convergence framework was derived under the assumption of solving linear partial differential equations (PDEs), however, promising extensions to the nonlinear domain were explored that can help guide future generalizations of the theory.

The benefits of the convergence framework were demonstrated, for example:

- It is capable of predicting and bounding worst-case convergence of MGRIT *a priori*.
- It can help to identify desirable properties of time integration schemes. For example, L-stable time integration schemes are more amendable to parallel-in-time integration than A-stable schemes.
- It can capture observations in numerical experiments that were reported in literature, for example:
  - Slow temporal coarsening in a multilevel hierarchy can lead to faster convergence than aggressive coarsening in a two-level hierarchy.
  - Stronger relaxation improves convergence of MGRIT for parabolic problems but not necessarily for purely hyperbolic problems.
  - F-cycles yield better performance than V-cycles.

Furthermore, an MGRIT variant was proposed that exploits the time-periodicity, which is present in many biomedical engineering applications, e.g., cyclic blood flow in the human heart. The time-periodic MGRIT algorithm was assessed for simple flow problems, linear and nonlinear coupled FSI models, as well as nonlinear flow in a patient-specific model of the left atrium and left ventricle. For the range of considered test problems, the time-periodic MGRIT algorithm proved capable of consistently reducing the *time-to-solution* of an existing simulation model with significant observed speedups. It further highlighted that one time-periodic MGRIT iteration corresponds to solving one cycle using sequential

time-stepping, which helped identifying a better convergence criterion. For most cases in literature, the tolerance imposed on the MGRIT residual was quite tight, which may often lead to a (potentially unnecessarily) large number of MGRIT iterations. By employing some knowledge about how the algorithm converges, one can decrease the required number of iterations and reduce the runtime of the algorithm.

7.2 Outlook

The multilevel convergence framework proposed in this work provides the basis for an in-depth understanding of existing MGRIT variants and developing new variants of the MGRIT algorithm, for example:

- **Design of optimal time integration scheme:** For example, select a time integration scheme on the fine grid and optimize over the Butcher coefficients for the coarse-grid time integration scheme. Approach similar to [37, 90].
- **Weighted relaxation:** Weight relaxation steps to improve convergence similar to other iterative schemes, e.g., the weighted Jacobi method.
- **Coarsening in integration order:** Like in spatial multigrid, complement or replace temporal coarsening with coarsening in integration order. Similar to coarsening in collocation order, see [135].
- **Dissipative spatial operators:** Use artificial dissipation to improve convergence, see [126].
- **Periodic coarse time grids:** Following the success of the time-periodic MGRIT algorithm proposed in Section 3.6, one could make all coarse time grids periodic in addition to the fine grid. Approach similiar to [40, 41].
- **Optimal speedup:** Combine the convergence framework with performance modeling [38] to find the *best* scheme with the shortest runtime and the largest speedup *a priori*.
- **Applications:** Apply MGRIT and its variants to further application areas, such as skeletal muscle models, porous media problems, etc.

Using the new class of analytic solutions for linear and nonlinear FSI from Chapter 2 helps to explore these new variants and develop robust solvers by exactly measuring and tracking the space-time error across the whole temporal domain, e.g., see Section 2.4.2.2.

A Parallel machines

Name	LEAD
Operating System	Scientific Linux 6.4 (Carbon)
Compute Nodes	1x AMD Opteron 6373 G34
	2.3 GHz, 4x 16 cores, Abu Dhabi
	7x AMD Opteron 6328 G34
	3.2 GHz, 4x 8 cores, Abu Dhabi
Memory per Node	256 GB
Disks per Node	4 x 128 GB Samsung SSD
Network	2 x 10 GBit Intel 82599ES Ethernet Dual SFP+
	1 x Melanox MCX353A-QCBT,
	Infiniband single-port QSFP, IB 40 GB/s
	2 x 1 GBit
Total Number of Processors	288

Table A.1: *Hardware and software specifications for LEAD cluster.*

Name	TheoSim
Operating System	Scientific Linux 6.10 (Carbon)
Compute Nodes	52x Intel(R) Xeon(R) E5-2650 v2
	2.8 GHz, 2x 10 cores, Ivy Bridge
	Available for project: 18 nodes
Memory per Node	256 GB
Disks per Node	4 x 500 GB Samsung SSD
Network	1 x QDR-Infiniband Interconnect (40 GBit)
	1 x 1 GBit Ethernet
Total Number of Processors	1040
	Available for project: 360

Table A.2: *Hardware and software specifications for TheoSim cluster.*

Name	ORCA
Operating System	Ubuntu 16.04
Compute Nodes	1x AMD Ryzen Threadripper 2990WX 3.0 GHz, 32 cores
Memory per Node	128 GB
Disks per Node	3 x 1 TB Samsung NVME SSD
Total Number of Processors	32

Table A.3: *Hardware and software specifications for ORCA cluster.*

Name	TOM
Operating System	Suse SLES 11 SP1
Compute Nodes	76x Intel(R) Xeon(R) E7-8837 2.66GHz, 8 cores, Westmere EX
Memory per Node	64 GB
Network	NUMAlink 5 Interconnect
Total Number of Processors	608

Table A.4: *Hardware and software specifications for TOM cluster.*

B Butcher tableaux of SDIRK schemes

$$\begin{array}{c|c} 1 & 1 \\ \hline & 1 \end{array} \qquad \begin{array}{c|cc} 1-\gamma & 1-\gamma & 0 \\ \gamma & 2\gamma-1 & 1-\gamma \\ \hline & 1/2 & 1/2 \end{array} \qquad \begin{array}{c|ccc} q & q & 0 & 0 \\ s & s-q & q & 0 \\ 1 & r & 1-q-r & q \\ \hline & r & 1-q-r & q \end{array}$$

Table B.1: *Butcher tableau for L-stable SDIRK scheme of orders 1 - 3 with* $\gamma = 1/\sqrt{2}$, $q = 0.4358665215\ldots$, $r = 1.2084966491\ldots$ *and* $s = 0.7179332607\ldots$*; See [25].*

$$\begin{array}{c|ccccc} 1/4 & 1/4 & 0 & 0 & 0 & 0 \\ 3/4 & 1/2 & 1/4 & 0 & 0 & 0 \\ 11/20 & 17/50 & -1/25 & 1/4 & 0 & 0 \\ 1/2 & 371/1360 & -137/2720 & 15/544 & 1/4 & 0 \\ 1 & 25/24 & -49/48 & 125/16 & -85/12 & 1/4 \\ \hline & 25/24 & -49/48 & 125/16 & -85/12 & 1/4 \end{array}$$

Table B.2: *Butcher tableau for L-stable SDIRK scheme of orders 4; See [27], Appendix C.*

$$\begin{array}{c|cc} 1/4 & 1/4 & 0 \\ 3/4 & 1/2 & 1/4 \\ \hline & 1/2 & 1/2 \end{array} \qquad \begin{array}{c|cc} \gamma & \gamma & 0 \\ 1-\gamma & 1-2\gamma & \gamma \\ \hline & 1/2 & 1/2 \end{array} \qquad \begin{array}{c|ccc} q & q & 0 & 0 \\ 1/2 & 1/2-q & q & 0 \\ 1-q & 2q & 1-4q & q \\ \hline & r & 1-2r & r \end{array}$$

Table B.3: *Butcher tableau for A-stable SDIRK scheme of orders 2 - 4 with* $\gamma = (3+\sqrt{3})/6$, $q = \cos(\pi/18)/\sqrt{3} + 1/2$ *and* $r = 1/(6(2q-1)^2)$*; See [12].*

Bibliography

[1] J. Aguado-Sierra, A. Krishnamurthy, C. Villongco, J. Chuang, E. Howard, M. J. Gonzales, J. Omens, D. E. Krummen, S. Narayan, R. C. P. Kerckhoffs, and A. D. McCulloch, *Patient-specific modeling of dyssynchronous heart failure: a case study*, Progress in Biophysics and Molecular Biology, 107 (2011), pp. 147–155.

[2] P. R. Amestoy, I. S. Duff, and J.-Y. L'excellent, *Multifrontal parallel distributed symmetric and unsymmetric solvers*, Computer Methods in Applied Mechanics and Engineering, 184 (2000), pp. 501–520.

[3] K. E. Atkinson, *An Introduction to Numerical Analysis*, John Wiley & Sons, 1989.

[4] Y. Bai, D. Sun, J. Lin, D. Kennedy, and F. Williams, *Numerical aerodynamic simulations of a NACA airfoil using CFD with block-iterative coupling and turbulence modelling*, International Journal of Computational Fluid Dynamics, 26 (2012), pp. 119–132.

[5] M. Balmus, A. Massing, J. Hoffman, R. Razavi, and D. Nordsletten, *A partition of unity approach to fluid mechanics and fluid-structure interaction*, arXiv preprint arXiv:1902.06168, (under review).

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

[7] O. A. Bauchau and J. I. Craig, *Euler-Bernoulli beam theory*, in Structural Analysis, Springer, 2009, pp. 173–221.

[8] Y. Bazilevs, J. R. Gohean, T. J. R. Hughes, R. D. Moser, and Y. Zhang, *Patient-specific isogeometric fluid-structure interaction analysis of thoracic aortic blood flow due to implantation of the Jarvik 2000 left ventricular assist device*, Computer Methods in Applied Mechanics and Engineering, 198 (2009), pp. 3534–3550.

[9] Y. Bazilevs, M.-C. Hsu, J. Kiendl, R. Wüchner, and K.-U. Bletzinger, *3D simulation of wind turbine rotors at full scale. Part II: Fluid-structure interaction modeling with composite blades*, International Journal for Numerical Methods in Fluids, 65 (2011), pp. 236–253.

[10] C. Bertram and J. Tscherry, *The onset of flow-rate limitation and flow-induced oscillations in collapsible tubes*, Journal of Fluids and Structures, 22 (2006), pp. 1029–1045.

[11] P. E. Black, *Big-O notation*, in Dictionary of Algorithms and Data Structures [online], 2019.

[12] L. Bonaventura and A. Della Rocca, *Monotonicity, positivity and strong stability of the TR-BDF2 method and of its SSP extensions*, arXiv preprint arXiv:1510.04303, (2015).

[13] C. Bradley, A. Bowery, R. Britten, V. Budelmann, O. Camara, R. Christie, A. Cookson, A. F. Frangi, T. B. Gamage, T. Heidlauf, et al., *OpenCMISS: a multi-physics & multi-scale computational infrastructure for the VPH/Physiome project*, Progress in Biophysics and Molecular Biology, 107 (2011), pp. 32–47.

[14] C. P. Bradley, N. Emamy, T. Ertl, D. Göddeke, A. Hessenthaler, T. Klotz, A. Krämer, M. Krone, B. Maier, M. Mehl, T. Rau, and O. Röhrle, *Enabling Detailed, Biophysics-Based Skeletal Muscle Models on HPC Systems*, Frontiers in Physiology - Computational Physiology and Medicine, 9 (2018).

[15] M. Burghardt, *Parallele Netzgenerierung für ebene und räumliche Problemstellungen aus dem Bauwesen*, PhD thesis, Institut für Numerische Methoden und Informatik im Bauwesen, Technische Universität Darmstadt, 2001.

[16] *Creative Commons Attribution-ShareAlike 4.0 International (CC BY-SA 4.0).*

[17] R. Chabiniok, V. Y. Wang, M. Hadjicharalambous, L. Asner, J. Lee, M. Sermesant, E. Kuhl, A. A. Young, P. Moireau, M. P. Nash, et al., *Multiphysics and multiscale modelling, data–model fusion and integration of organ physiology in the clinic: ventricular cardiac mechanics*, Interface focus, 6 (2016), p. 20150083.

[18] *CHeart: A scientific software application for simulating the physics of the human heart.* `http://cheart.co.uk`.

[19] *Converter script: CHeart data to Paraview VTU.* `https://bitbucket.org/hessenthaler/cheart2vtu.git`.

[20] G. Chesshire and W. D. Henshaw, *Composite overlapping meshes for the solution of partial differential equations*, Journal of Computational Physics, 90 (1990), pp. 1–64.

[21] A. J. Christlieb, C. B. Macdonald, and B. W. Ong, *Parallel high-order integrators*, SIAM Journal on Scientific Computing, 32 (2010), pp. 818–835.

[22] J. Cortial and C. Farhat, *A time-parallel implicit method for accelerating the solution of non-linear structural dynamics problems*, International Journal for Numerical Methods in Engineering, 77 (2009), pp. 451–470.

[23] G. De Nayer, A. Kalmbach, M. Breuer, S. Sicklinger, and R. Wüchner, *Flow past a cylinder with a flexible splitter plate: A complementary experimental-numerical investigation and a new FSI test case (FSI-Pfs-1a)*, Computers & Fluids, 99 (2014), pp. 18–43.

[24] H. De Sterck, S. Friedhoff, A. J. M. Howse, and S. P. MacLachlan, *Convergence analysis for parallel-in-time solution of hyperbolic systems*, arXiv preprint arXiv:1903.08928, (2019).

[25] V. A. Dobrev, T. V. Kolev, N. A. Petersson, and J. B. Schroder, *Two-level convergence theory for multigrid reduction in time (MGRIT)*, SIAM Journal on Scientific Computing, 39 (2017), pp. S501–S527.

[26] J. Donea, S. Giuliani, and J.-P. Halleux, *An arbitrary Lagrangian-Eulerian finite element method for transient dynamic fluid-structure interactions*, Computer Methods in Applied Mechanics and Engineering, 33 (1982), pp. 689–723.

[27] M. Duarte, R. Dobbins, and M. Smooke, *High order implicit time integration schemes on multiresolution adaptive grids for stiff PDEs*, arXiv preprint arXiv:1604.00355, (2016).

[28] M. Emmett and M. L. Minion, *Toward an Efficient Parallel in Time Method for Partial Differential Equations*, Communications in Applied Mathematics and Computational Science, 7 (2012), pp. 105–132.

[29] C. R. Ethier and D. A. Steinman, *Exact fully 3D Navier-Stokes solutions for benchmarking*, International Journal for Numerical Methods in Fluids, 19 (1994), pp. 369–375.

[30] R. D. Falgout, S. Friedhoff, T. V. Kolev, S. P. MacLachlan, and J. B. Schroder, *Parallel time integration with multigrid*, SIAM Journal on Scientific Computing, 36 (2014), pp. C635–C661.

[31] R. D. Falgout, A. Katz, T. V. Kolev, J. B. Schroder, A. Wissink, and U. M. Yang, *Parallel time integration with multigrid reduction for a compressible fluid dynamics application*, tech. rep., Lawrence Livermore National Lab.(LLNL), Livermore, CA (United States), 2015.

[32] R. D. Falgout, T. A. Manteuffel, B. O'Neill, and J. B. Schroder, *Multigrid reduction in time for nonlinear parabolic problems: A case study*, SIAM Journal on Scientific Computing, 39 (2017), pp. S298–S322.

[33] C. Farhat and M. Chandesris, *Time-decomposed parallel time-integrators: theory and feasibility studies for fluid, structure, and fluid-structure applications*, International Journal for Numerical Methods in Engineering, 58 (2003), pp. 1397–1434.

[34] C. Farhat, J. Cortial, C. Dastillung, and H. Bavestrello, *Time-parallel implicit integrators for the near-real-time prediction of linear structural dynamic responses*, International Journal for Numerical Methods in Engineering, 67 (2006), pp. 697–724.

[35] S. Friedhoff, R. D. Falgout, T. V. Kolev, S. MacLachlan, and J. B. Schroder, *A multigrid-in-time algorithm for solving evolution equations in parallel*, tech. rep., Lawrence Livermore National Lab.(LLNL), Livermore, CA (United States), 2012.

[36] S. Friedhoff and S. P. MacLachlan, *A generalized predictive analysis tool for multigrid methods*, Numerical Linear Algebra with Applications, 22 (2015), pp. 618–647.

[37] S. Friedhoff and B. S. Southworth, *On "Optimal" h-Independent Convergence of Parareal and MGRIT using Runge-Kutta Time Integration*, arXiv preprint arXiv:1906.06672, (2019).

[38] H. Gahvari, V. A. Dobrev, R. D. Falgout, T. V. Kolev, J. B. Schroder, M. Schulz, and U. M. Yang, *A performance model for allocating the parallelism in a multigrid-in-time solver*, in International Workshop on Performance Modeling, Benchmarking and Simulation of High Performance Computer Systems (PMBS), IEEE, 2016, pp. 22–31.

[39] M. J. Gander, *50 years of Time Parallel Time Integration*, in Multiple Shooting and Time Domain Decomposition, Springer, 2015.

[40] M. J. Gander, Y.-L. Jiang, B. Song, and H. Zhang, *Analysis of two parareal algorithms for time-periodic problems*, SIAM Journal on Scientific Computing, 35 (2013), pp. A2393–A2415.

[41] M. J. Gander, I. Kulchytska-Ruchka, and S. Schöps, *A new parareal algorithm for time-periodic problems with discontinuous inputs*, arXiv preprint arXiv:1810.12372, (2018).

[42] M. J. Gander, F. Kwok, and H. Zhang, *Multigrid interpretations of the parareal algorithm leading to an overlapping variant and MGRIT*, Computing and Visualization in Science, 19 (2018), pp. 59–74.

[43] O. Ghattas and X. Li, *A variational finite element method for stationary nonlinear fluid-solid interaction*, Journal of Computational Physics, 121 (1995), pp. 347–356.

[44] A. J. Gil, A. A. Carreño, J. Bonet, and O. Hassan, *The immersed structural potential method for haemodynamic applications*, Journal of Computational Physics, 229 (2010), pp. 8613–8641.

[45] A. J. Gil, A. A. Carreno, J. Bonet, and O. Hassan, *An enhanced immersed structural potential method for fluid-structure interaction*, Journal of Computational Physics, 250 (2013), pp. 178–205.

[46] R. Glowinski, T.-W. Pan, T. I. Hesla, and D. D. Joseph, *A distributed Lagrange multiplier/fictitious domain method for particulate flows*, International Journal of Multiphase Flow, 25 (1999), pp. 755–794.

[47] R. Glowinski, T.-W. Pan, and J. Periaux, *A fictitious domain method for Dirichlet problem and applications*, Computer Methods in Applied Mechanics and Engineering, 111 (1994), pp. 283–303.

[48] J. P. Gomes and H. Lienhart, *Experimental study on a fluid-structure interaction reference test case*, in Fluid-structure interaction I – modelling, simulation, optimization, vol. 53, Lecture Notes in Computational Science and Engineering, Springer, 2006, pp. 356–370.

[49] ——, *Experimental benchmark: Self-excited fluid-structure interaction test cases*, in Fluid-structure interaction II – modelling, simulation, optimization, vol. 73, Lecture Notes in Computational Science and Engineering, Springer, 2010, pp. 383–411.

[50] H. Gray, *Anatomy of the Human Body*, vol. 20, 1918.

[51] S. Günther, N. R. Gauger, and J. B. Schroder, *A non-intrusive parallel-in-time adjoint solver with the XBraid library*, Computing and Visualization in Science, 19 (2018), pp. 85–95.

[52] ——, *A non-intrusive parallel-in-time approach for simultaneous optimization with unsteady PDEs*, Optimization Methods and Software, (2018), pp. 1–16.

[53] W. Hackbusch and U. Trottenberg, *Multigrid methods: proceedings of the conference held at Köln-Porz, November 23-27, 1981*, vol. 960, Springer, 2006.

[54] E. Hairer, S. P. Nørsett, and G. Wanner, *Solving Ordinary Differential Equations I: Nonstiff Problems*, 1993.

[55] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*, Springer Series in Computational Mathematics, 14 (1996).

[56] F. P. Hamon, M. Schreiber, and M. L. Minion, *Multi-level spectral deferred corrections scheme for the shallow water equations on the rotating sphere*, Journal of Computational Physics, 376 (2019), pp. 435–454.

[57] M. Heil, A. L. Hazel, and J. Boyle, *Solvers for large-displacement fluid-structure interaction problems: segregated versus monolithic approaches*, Computational Mechanics, 43 (2008), pp. 91–101.

[58] A. Hessenthaler, *Direct and iterative solvers in Python*. `https://bitbucket.org/hessenthaler/solvers-python`.

[59] A. Hessenthaler, M. Balmus, O. Röhrle, and D. Nordsletten, *A Class of Analytic Solutions for Verification and Convergence Analysis of Linear and Nonlinear Fluid-Structure Interaction Algorithms*, Computer Methods in Applied Mechanics and Engineering, 362 (2020), p. 112841.

[60] A. Hessenthaler, R. D. Falgout, J. B. Schroder, D. Nordsletten, and O. Röhrle, *Analytic F-cycle bounds for multigrid-reduction-in-time*, (in preparation).

[61] ——, *Time-periodic multigrid-reduction-in-time for a stenosed valve problem*, (in preparation).

[62] ——, *Time-periodic steady-state solution of fluid-structure interaction and cardiac flow problems through multigrid-reduction-in-time*, ((in preparation).

[63] A. Hessenthaler, S. Friedhoff, O. Röhrle, and D. A. Nordsletten, *3D Fluid-Structure Interaction Experiment and Benchmark Results*, PAMM, 16 (2016), pp. 451–452.

[64] A. Hessenthaler, N. Gaddum, O. Holub, R. Sinkus, O. Röhrle, and D. Nordsletten, *Experiment for validation of fluid-structure interaction models and algorithms*, International Journal for Numerical Methods in Biomedical Engineering, 33 (2017).

[65] A. Hessenthaler, T. Klotz, D. Nordsletten, and O. Röhrle, *Solving the multidomain equations with multigrid-reduction-in-time*, (in preparation).

[66] A. Hessenthaler, D. Nordsletten, O. Röhrle, J. B. Schroder, and R. D. Falgout, *Convergence of the multigrid reduction in time algorithm for the linear elasticity equations*, Numerical Linear Algebra with Applications, 25 (2018), p. e2155. e2155 nla.2155.

[67] A. Hessenthaler, O. Röhrle, and D. Nordsletten, *Validation of a non-conforming monolithic fluid-structure interaction method using phase-contrast MRI*, International Journal for Numerical Methods in Biomedical Engineering, 33 (2017).

[68] A. Hessenthaler, B. S. Southworth, D. Nordsletten, O. Rhrle, R. D. Falgout, and J. B. Schroder, *Multilevel convergence analysis of multigrid-reduction-in-time*, SIAM Journal on Scientific Computing, 42 (2020), pp. A771–A796.

[69] N. J. Higham, *Estimating the matrix p-norm*, Numerische Mathematik, 62 (1992), pp. 539–555.

[70] C. W. Hirt, A. A. Amsden, and J. L. Cook, *An arbitrary Lagrangian-Eulerian computing method for all flow speeds*, Journal of Computational Physics, 14 (1974), pp. 227–253.

[71] J. Hoffman, J. Jansson, and R. V. De Abreu, *Adaptive modeling of turbulent flow with residual based turbulent kinetic energy dissipation*, Computer Methods in Applied Mechanics and Engineering, 200 (2011), pp. 2758–2767.

[72] J. Hoffman, J. Jansson, and M. Stöckli, *Unified continuum modeling of fluid-structure interaction*, Mathematical Models and Methods in Applied Sciences, 21 (2011), pp. 491–513.

[73] J. Holt, *Estimation of the residual volume of the ventricle of the dog's heart by two indicator dilution technics*, Circulation Research, 4 (1956), pp. 187–195.

[74] G. Horton and S. Vandewalle, *A Space-Time Multigrid Method for Parabolic Partial Differential Equations*, SIAM Journal on Scientific Computing, 16 (1995), pp. 848–864.

[75] G. Houzeaux and R. Codina, *A Chimera method based on a Dirichlet/Neumann (Robin) coupling for the Navier-Stokes equations*, Computer Methods in Applied Mechanics and Engineering, 192 (2003), pp. 3343–3377.

[76] A. Howse, H. De Sterck, S. MacLachlan, R. Falgout, and J. Schroder, *Multigrid reduction in time with adaptive spatial coarsening for the linear advection equation*, tech. rep., Lawrence Livermore National Laboratory (LLNL), Livermore, CA, 2017.

[77] T. J. R. Hughes, W. K. Liu, and T. K. Zimmermann, *Lagrangian-Eulerian finite element formulation for incompressible viscous flows*, Computer Methods in Applied Mechanics and Engineering, 29 (1981), pp. 329–349.

[78] S. R. Idelsohn, J. Marti, A. Souto-Iglesias, and E. Onate, *Interaction between an elastic structure and free-surface flows: experimental versus numerical comparisons using the pfem*, Computational Mechanics, 43 (2008), pp. 125–132.

[79] C. G. J. Jacobi, *Über ein leichtes Verfahren, die in der Theorie der Säkularstörungen vorkommenden Gleichungen numerisch aufzulösen*, Journal für die Reine und Angewandte Mathematik, 30 (1846), pp. 51–95.

[80] J. Jansson, N. C. Degirmenci, and J. Hoffman, *Adaptive unified continuum FEM modeling of a 3D FSI benchmark problem*, International Journal for Numerical Methods in Biomedical Engineering, 33 (2017), p. e2851.

[81] C. R. Johnson, *Positive definite matrices*, The American Mathematical Monthly, 77 (1970), pp. 259–264.

[82] I. E. Kaporin, *Explicitly preconditioned conjugate gradient method for the solution of unsymmetric linear systems*, International Journal of Computer Mathematics, 44 (1992), pp. 169–187.

[83] ——, *New convergence results and preconditioning strategies for the conjugate gradient method*, Numerical Linear Algebra with Applications, 1 (1994), pp. 179–210.

[84] ——, *Talk: Convergence Rate Estimates for the Conjugate Gradient Method*. The Rome-Moscow School of Matrix Methods (Moscow Part: September 10-15, 2012), 2012.

[85] G. Karypis and V. Kumar, *ParMETIS – parallel graph partitioning and fill-reducing matrix ordering*, 2013.

[86] D. E. Keyes, L. C. McInnes, C. Woodward, W. Gropp, E. Myra, M. Pernice, J. Bell, J. Brown, A. Clo, J. Connors, E. Constantinescu, D. Estep, K. Evans, C. Farhat, A. Hakim, G. Hammond, G. Hansen, J. Hill, T. Isaac, X. Jiao, K. Jordan, D. Kaushik, E. Kaxiras, A. Koniges,

K. Lee, A. Lott, Q. Lu, J. Magerlein, R. Maxwell, M. McCourt, M. Mehl, R. Pawlowski, A. P. Randles, D. Reynolds, B. Rivière, U. Rüde, T. Scheibe, J. Shadid, B. Sheehan, M. Shepard, A. Siegel, B. Smith, X. Tang, C. Wilson, and B. Wohlmuth, *Multiphysics simulations: Challenges and opportunities*, The International Journal of High Performance Computing Applications, 27 (2013), pp. 4–83.

[87] J. Kim and P. Moin, *Application of a fractional-step method to incompressible Navier-Stokes equations*, Journal of Computational Physics, 59 (1985), pp. 308–323.

[88] D. E. Knuth, *Big omicron and big omega and big theta*, ACM Sigact News, 8 (1976), pp. 18–24.

[89] J. F. B. M. Kraaijevanger, *Contractivity of Runge-Kutta methods*, BIT Numerical Mathematics, 31 (1991), pp. 482–528.

[90] O. A. Krzysik, H. De Sterck, S. P. MacLachlan, and S. Friedhoff, *On selecting coarse-grid operators for Parareal and MGRIT applied to linear advection*, arXiv preprint arXiv:1902.07757, (2019).

[91] M. Landajuela, M. Vidrascu, D. Chapelle, and M. A. Fernández, *Coupling schemes for the FSI forward prediction challenge: comparative study and validation*, International Journal for Numerical Methods in Biomedical Engineering, 33 (2017).

[92] P. D. Lax and R. D. Richtmyer, *Survey of the Stability of Linear Finite Difference Equations*, Communications on Pure and Applied Mathematics, 9 (1956), pp. 267–293.

[93] M. Lecouvez, R. D. Falgout, C. S. Woodward, and P. Top, *A parallel multigrid reduction in time method for power systems*, in Power and Energy Society General Meeting (PESGM), 2016, IEEE, 2016, pp. 1–5.

[94] J. Lee, A. Cookson, I. Roy, E. Kerfoot, L. Asner, G. Vigueras, T. Sochi, S. Deparis, C. Michler, N. P. Smith, et al., *Multiphysics computational modeling in CHeart*, SIAM Journal on Scientific Computing, 38 (2016), pp. C150–C178.

[95] R. J. LeVeque, *Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems*, vol. 98, SIAM, 2007.

[96] A. E. Li, I. Kamel, F. Rando, M. Anderson, B. Kumbasar, J. A. Lima, and D. A. Bluemke, *Using MRI to assess aortic wall thickness in the multiethnic study of atherosclerosis: distribution by race, sex, and age*, American Journal of Roentgenology, 182 (2004), pp. 593–597.

[97] J.-L. Lions, Y. Maday, and G. Turinici, *A "parareal" in time discretization of PDE's*, Comptes Rendus de l'Académie des Sciences - Series I - Mathematics, 332 (2001), pp. 661–668.

[98] M. Lomsky, L. Johansson, P. Gjertsson, J. Björk, and L. Edenbrandt, *Normal limits for left ventricular ejection fraction and volumes determined by gated single photon emission computed tomography - a comparison between two quantification methods*, Clinical Physiology and Functional Imaging, 28 (2008), pp. 169–173.

[99] C. Lubich and A. Ostermann, *Multi-grid dynamic iteration for parabolic equations*, BIT Numerical Mathematics, 27 (1987), pp. 216–234.

[100] T. A. Manteuffel, L. N. Olson, J. B. Schroder, and B. S. Southworth, *A Root-node Based Algebraic Multigrid Method*, SIAM Journal on Scientific Computing, (accepted) (2017).

[101] D. Marlevi, M. Balmus, A. Hessenthaler, F. Viola, D. Fovargue, A. de Vecchi, P. Lamata, N. S. Burris, F. D. Pagani, E. Edelman, T. Ebbers, and D. A. Nordsletten, *Non-invasive estimation of relative pressure for intracardiac flows using virtual work-energy*, Medical Image Analysis, (submitted).

[102] M. McCormick, D. A. Nordsletten, D. Kay, and N. Smith, *Simulating left ventricular fluid-solid mechanics through the cardiac cycle under LVAD support*, Journal of Computational Physics, 244 (2013), pp. 80–96.

[103] J. Mediavilla Varas, M. Philippens, S. Meijer, A. Van Den Berg, P. Sibma, J. Van Bree, and D. De Vries, *Physics of IED blast shock tube simulations for mTBI research*, Frontiers in Neurology, 2 (2011), p. 58.

[104] R. B. Medvitz, V. Reddy, S. Deutsch, K. B. Manning, and E. G. Paterson, *Validation of a CFD methodology for positive displacement LVAD analysis using PIV data*, Journal of Biomechanical Engineering, 131 (2009), p. 111009.

[105] R. Mittal and G. Iaccarino, *Immersed boundary methods*, Annual Review of Fluid Mechanics, 37 (2005), pp. 239–261.

[106] M. E. Moghadam, Y. Bazilevs, T.-Y. Hsia, I. E. Vignon-Clementel, A. L. Marsden, et al., *A comparison of outlet boundary treatments for prevention of backflow divergence with relevance to blood flow simulations*, Computational Mechanics, 48 (2011), pp. 277–291.

[107] D. P. Mok, *Partitionierte Lösungsansätze in der Strukturdynamik und der Fluid-Struktur-Interaktion*, PhD thesis, Institut für Baustatik, Universität Stuttgart, 2001.

[108] J. Nievergelt, *Parallel methods for integrating ordinary differential equations*, Commun. ACM, 7 (1964), pp. 731–733.

[109] R. A. Nishimura, F. A. Miller Jr, M. J. Callahan, R. C. Benassi, J. B. Seward, and A. J. Tajik, *Doppler echocardiography: theory, instrumentation, technique, and application*, in Mayo Clinic Proceedings, vol. 60, Elsevier, 1985, pp. 321–343.

[110] D. Nordsletten, D. Kay, and N. Smith, *A non-conforming monolithic finite element method for problems of coupled mechanics*, Journal of Computational Physics, 229 (2010), pp. 7571–7593.

[111] B. O'Neill, *Multigrid Reduction in Time for Nonlinear Parabolic Problems*, 2017.

[112] *OpenCMISS: A mathematical modeling environment.* `http://opencmiss.org`.

[113] C. S. Peskin, *Flow patterns around heart valves: a digital computer method for solving the equations of motion*, IEEE Transactions on Biomedical Engineering, (1973), pp. 316–317.

[114] ——, *The immersed boundary method*, Acta Numerica, 11 (2002), pp. 479–517.

[115] P. Ponikowski, A. A. Voors, S. D. Anker, H. Bueno, J. G. F. Cleland, A. J. S. Coats, V. Falk, J. R. Gonzlez-Juanatey, V.-P. Harjola, E. A. Jankowska, M. Jessup, C. Linde, P. Nihoyannopoulos, J. T. Parissis, B. Pieske, J. P. Riley, G. M. C. Rosano, L. M. Ruilope, F. Ruschitzka, F. H. Rutten, P. van der Meer, and E. S. D. Group, *2016 ESC guidelines for the diagnosis and treatment of acute and chronic heart failure*, European Heart Journal, 37 (2016), pp. 2129–2200.

[116] P. Reymond, P. Crosetto, S. Deparis, A. Quarteroni, and N. Stergiopulos, *Physiological simulation of blood flow in the aorta: comparison of hemodynamic indices as predicted by 3-D FSI, 3-D rigid wall and 1-D models*, Medical Engineering & Physics, 35 (2013), pp. 784–791.

[117] P. J. Roache, *Code verification by the method of manufactured solutions*, Journal of Fluids Engineering, 124 (2002), pp. 4–10.

[118] K. Rupp, *Microprocessor Trend Data.* `https://github.com/karlrupp/microprocessor-trend-data`.

[119] Y. Saad, *Iterative methods for sparse linear systems*, vol. 82, SIAM, 2003.

[120] K. Salari and P. Knupp, *Code verification by the method of manufactured solutions*, tech. rep., Sandia National Laboratories, Albuquerque, NM 87185 (US); Sandia National Laboratories, Livermore, CA 94550 (US), 2000.

[121] C. Sanderson and R. Curtin, *Armadillo: a template-based C++ library for linear algebra*, Journal of Open Source Software, (2016).

[122] ——, *A User-Friendly Hybrid Sparse Matrix Class in C++*, arXiv preprint arXiv:1805.03380, (2018).

[123] R. Schreiber and H. B. Keller, *Driven cavity flows by efficient numerical techniques*, Journal of Computational Physics, 49 (1983), pp. 310–333.

[124] R. Schreiber and H. B. Keller, *Spurious solutions in driven cavity calculations*, Journal of Computational Physics, 49 (1983), pp. 165–172.

[125] J. B. Schroder, *Parallelizing Over Artificial Neural Network Training Runs with Multigrid*, arXiv preprint arXiv:1708.02276, (2017).

[126] ——, *On the Use of Artificial Dissipation for Hyperbolic Problems and Multigrid Reduction In Time (MGRIT)*, Technical Report, LLNL-TR-681918, (2018).

[127] J. B. Schroder, R. D. Falgout, C. S. Woodward, P. Top, and M. Lecouvez, *Parallel-in-Time Solution of Power Systems with Scheduled Events*, 2018 Power and Energy Society General Meeting (PESGM), IEEE, (2018).

[128] J. L. Semple, N. Woolridge, and C. J. Lumsden, *In vitro, in vivo, in silico: computational systems in tissue engineering and regenerative medicine*, Tissue Engineering, 11 (2005), pp. 341–356.

[129] V. Shamanskii, *A modification of Newton's method*, Ukrainian Mathematical Journal, 19 (1967), pp. 118–122.

[130] J. Simo and N. Tarnow, *The discrete energy-momentum method. Conserving algorithms for nonlinear elastodynamics*, Zeitschrift für angewandte Mathematik und Physik ZAMP, 43 (1992), pp. 757–792.

[131] J. C. Simo and K. K. Wong, *Unconditionally stable algorithms for rigid body dynamics that exactly preserve energy and momentum*, International Journal for Numerical Methods in Engineering, 31 (1991), pp. 19–52.

[132] N. Smith, A. de Vecchi, M. McCormick, D. Nordsletten, O. Camara, A. F. Frangi, H. Delingette, M. Sermesant, J. Relan, N. Ayache, M. W. Krueger, W. H. W. Schulze, R. Hose, I. Valverde, P. Beerbaum, C. Staicu, M. Siebes, J. Spaan, P. Hunter, J. Weese, H. Lehmann, D. Chapelle, and R. Rezavi, *euHeart: personalized and integrated cardiac care using patient-specific cardiovascular modelling*, Interface focus, 1 (2011), pp. 349–364.

[133] B. S. Southworth, *Necessary conditions and tight two-level convergence bounds for Parareal and multigrid reduction in time*, SIAM J. on Matrix Analysis and Applications, (2019).

[134] B. S. Southworth, W. Mitchell, and A. Hessenthaler, *Tight two-level convergence of Linear Parareal and MGRIT: Extensions and Implications in Practice*, (in preparation).

[135] R. Speck, D. Ruprecht, M. Emmett, M. Minion, M. Bolten, and R. Krause, *A multi-level spectral deferred correction method*, BIT Numerical Mathematics, 55 (2015), pp. 843–867.

[136] R. Speck, D. Ruprecht, R. Krause, M. Emmett, M. L. Minion, M. Winkel, and P. Gibbon, *A massively space-time parallel N-body solver*, in Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis, SC '12, Los Alamitos, CA, USA, 2012, IEEE Computer Society Press, pp. 92:1–92:11.

[137] J. L. Steger and J. A. Benek, *On the use of composite grid schemes in computational aerodynamics*, Computer Methods in Applied Mechanics and Engineering, 64 (1987), pp. 301–320.

[138] J. L. Steger, F. C. Dougherty, and J. A. Benek, *A chimera grid scheme*, in Advances in Grid Generation, vol. ASME FED-5, 1983, pp. 59–69.

[139] S. Steinberg and P. J. Roache, *Symbolic manipulation and computational fluid dynamics*, Journal of Computational Physics, 57 (1985), pp. 251–284.

[140] V. Strassen, *Gaussian Elimination is not Optimal*, Numerische Mathematik, 13 (1969), pp. 354–356.

[141] K. Takizawa, Y. Bazilevs, and T. E. Tezduyar, *Space-time and ALE-VMS techniques for patient-specific cardiovascular fluid-structure interaction modeling*, Archives of Computational Methods in Engineering, 19 (2012), pp. 171–225.

[142] K. Takizawa and T. E. Tezduyar, *Multiscale space-time fluid-structure interaction techniques*, Computational Mechanics, 48 (2011), pp. 247–267.

[143] C. A. Taylor and C. Figueroa, *Patient-specific modeling of cardiovascular mechanics*, Annual Review of Biomedical Engineering, 11 (2009), pp. 109–134.

[144] T. E. Tezduyar, *Stabilized finite element formulations for incompressible flow computations*, in Advances in Applied Mechanics, vol. 28, Elsevier, 1991, pp. 1–44.

[145] T. E. Tezduyar, M. Behr, S. Mittal, and J. Liou, *A new strategy for finite element computations involving moving boundaries and interfaces – the deforming-spatial-domain/space-time procedure: II. Computation of free-surface flows, two-liquid flows, and flows with drifting cylinders*, Computer Methods in Applied Mechanics and Engineering, 94 (1992), pp. 353–371.

[146] U. Trottenberg, C. W. Oosterlee, and A. Schüller, *Multigrid*, Academic Press, 2001.

[147] S. Turek and J. Hron, *Proposal for numerical benchmarking of fluid-structure interaction between an elastic object and laminar incompressible flow*, in Fluid-structure interaction, Springer, 2006, pp. 371–385.

[148] S. Turek, J. Hron, M. Razzaq, H. Wobker, and M. Schäfer, *Numerical benchmarking of fluid-structure interaction: A comparison of different discretization and solution approaches*, in Fluid Structure Interaction II, Springer, 2011, pp. 413–424.

[149] *Mesh converter script: UGRID to CHeart.* `https://bitbucket.org/hessenthaler/ugrid2xtb.git`.

[150] D. Van Heesch, *Doxygen: Source code documentation generator tool*, (2008).

[151] G. van Rossum, *Python tutorial*, Tech. Rep. CS-R9526, Centrum voor Wiskunde en Informatica (CWI), Amsterdam, May 1995.

[152] S. Vandewalle and E. Van de Velde, *Space-time concurrent multigrid waveform relaxation*, Annals of Numerical Mathematics, 1 (1994), pp. 347–360.

[153] W. A. Wall, *Fluid-Struktur-Interaktion mit stabilisierten Finiten Elementen*, PhD thesis, Institut für Baustatik, Universität Stuttgart, 1999.

[154] W. A. Wall, P. Gamnitzer, and A. Gerstenberger, *Fluid-structure interaction approaches on fixed grids based on two different domain decomposition ideas*, International Journal of Computational Fluid Dynamics, 22 (2008), pp. 411–427.

[155] J. R. Womersley, *Method for the calculation of velocity, rate of flow and viscous drag in arteries when the pressure gradient is known*, The Journal of Physiology, 127 (1955), pp. 553–563.

[156] J. R. Womersley, *Oscillatory Flow in Arteries: the Constrained Elastic Tube as a Model of Arterial Flow and Pulse Transmission*, Physics in Medicine and Biology, 2 (1957), pp. 178–187.

[157] P. Wriggers, *Nonlinear Finite Element Methods*, Springer Science & Business Media, 2008.

[158] *XBraid: Parallel multigrid in time*. `http://llnl.gov/casc/xbraid`.

[159] X. Yue, S. Shu, X. Xu, W. Bu, and K. Pan, *Parallel-in-Time with Fully Finite Element Multigrid for 2-D Space-fractional Diffusion Equations*, arXiv preprint arXiv:1805.06688, (2018).

[160] O. C. Zienkiewicz, R. L. Taylor, P. Nithiarasu, and J. Z. Zhu, *The Finite Element Method*, vol. 3, McGraw-Hill London, 1977.