



LUND UNIVERSITY

Partitioned methods for time-dependent thermal fluid-structure interaction

Monge, Azahar

2018

Document Version:

Publisher's PDF, also known as Version of record

[Link to publication](#)

Citation for published version (APA):

Monge, A. (2018). *Partitioned methods for time-dependent thermal fluid-structure interaction*. [Doctoral Thesis (compilation), Centre for Mathematical Sciences]. Lund University, Faculty of Science, Centre for Mathematical Sciences.

Total number of authors:

1

General rights

Unless other specific re-use rights are stated the following general rights apply:

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Read more about Creative commons licenses: <https://creativecommons.org/licenses/>

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

LUND UNIVERSITY

PO Box 117
221 00 Lund
+46 46-222 00 00

Partitioned methods for time-dependent thermal fluid-structure interaction

AZAHAR MONGE

Lund University
Faculty of Science
Centre for Mathematical Sciences
Numerical Analysis



Partitioned methods for time-dependent thermal fluid-structure interaction

Azahar Monge



LUND
UNIVERSITY

ACADEMIC THESIS

which, by due permission of the Faculty of Science at Lund University, will be publicly defended on Monday 10th of September, 2018, at 12:15 in lecture hall MH:Hörmander, Centre for Mathematical Sciences, Sölvegatan 18A, Lund, for the degree of Doctor of Philosophy in Numerical Analysis.

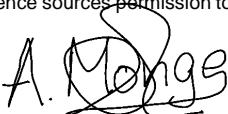
Faculty opponent

Prof. Dr. Oliver Rheinbach, TU Bergakademie Freiberg, Germany.

Organization Numerical Analysis Centre for Mathematical Sciences Faculty of Science LUND UNIVERSITY	Document name DOCTORAL DISSERTATION IN MATHEMATICAL SCIENCES	
Author(s) Azahar Monge	Date of issue August 2018	
	Sponsoring organization	
	Supervisors Philipp Birken, Claus Führer, Gustaf Söderlind	
Title and subtitle Partitioned methods for time-dependent thermal fluid-structure interaction		
<p>Abstract</p> <p>The efficient simulation of thermal interaction between fluids and structures is crucial in the design of many industrial products, e.g. turbine blades or rocket nozzles. The main goal of this work is to present a high order time adaptive multirate parallel partitioned coupled method for the efficient numerical solution of two parabolic problems with strong jumps in the material coefficients that can be further extended to thermal fluid-structure interaction simulation.</p> <p>Our starting point was to analyze the convergence rate of the Dirichlet-Neumann iteration, which is one of the basic methods for simulating fluid-structure problems, for the fully discretized unsteady transmission problem. Specifically, we consider the coupling of two linear heat equations on two identical non overlapping domains with jumps in the material coefficients across these as a model for thermal fluid-structure interaction. We provide an exact formula for the spectral radius of the iteration matrix in 1D. We then show numerically that the 1D result estimates the convergence rates of 2D examples and even of nonlinear thermal fluid-structure interaction test cases with unstructured grids.</p> <p>However, an important challenge when coupling two different time-dependent problems is to increase parallelization in time. We suggest a multirate Neumann-Neumann waveform relaxation algorithm to solve two heterogeneous coupled heat equations as an alternative to the Dirichlet-Neumann method. In order to fix the mismatch produced by the multirate feature at the space-time interface a linear interpolation is constructed.</p> <p>Furthermore, we perform a one-dimensional convergence analysis for the nonmultirate fully discretized heat equations to find the optimal relaxation parameter in terms of the material coefficients, the step size and the mesh resolution. This gives a very efficient method which needs only two iterations. Numerical results confirm the analysis and show that the 1D nonmultirate optimal relaxation parameter is a very good estimator for the multirate 1D case and even for multirate and nonmultirate 2D examples.</p> <p>Finally, we also include in this work a time adaptive version of the multirate Neumann-Neumann waveform relaxation method mentioned above. Building a variable step size multirate scheme allows each of the subsolvers to freely construct its own time grid independently of each other. Therefore, the overall coupled method is more efficient than the previous multirate version.</p>		
Key words thermal FSI, conjugate heat transfer, domain decomposition, time adaptivity, coupled problems, multirate		
Classification system and/or index terms (if any)		
Supplementary bibliographical information	Language English	
ISSN and key title 1404-0034	ISBN 978-91-7753-775-5	
Recipient's notes	Number of pages xvi + 175	Price
	Security classification	

I, the undersigned, being the copyright owner of the abstract of the above-mentioned dissertation, hereby grant to all reference sources permission to publish and disseminate the abstract of the above-mentioned dissertation.

Signature



Date 7 August 2018

PARTITIONED METHODS FOR
TIME-DEPENDENT THERMAL
FLUID-STRUCTURE INTERACTION

AZAHAR MONGE



LUND UNIVERSITY

Faculty of Science
Centre for Mathematical Sciences
Numerical Analysis

Numerical Analysis
Centre for Mathematical Sciences
Lund University
Box 118
SE-221 00 Lund
Sweden
<http://www.maths.lth.se/>

Doctoral Theses in Mathematical Sciences 2018:5
ISSN 1404-0034

ISBN 978-91-7753-775-5 (print)
ISBN 978-91-7753-776-2 (electronic)
LUNFNA-1007-2018

© Azahar Monge, 2018

Printed in Sweden by Media-Tryck, Lund 2018

*We live on an island
surrounded by a sea of ignorance.
As our island of knowledge grows,
so does the shore of our ignorance.*

Archibald Wheeler

Abstract

The efficient simulation of thermal interaction between fluids and structures is crucial in the design of many industrial products, e.g. turbine blades or rocket nozzles. The main goal of this work is to present a high order time adaptive multirate parallel partitioned coupled method for the efficient numerical solution of two parabolic problems with strong jumps in the material coefficients that can be further extended to thermal fluid-structure interaction simulation.

Our starting point was to analyze the convergence rate of the Dirichlet-Neumann iteration, which is one of the basic methods for simulating fluid-structure problems, for the fully discretized unsteady transmission problem. Specifically, we consider the coupling of two linear heat equations on two identical non overlapping domains with jumps in the material coefficients across these as a model for thermal fluid-structure interaction. We provide an exact formula for the spectral radius of the iteration matrix in 1D. We then show numerically that the 1D result estimates the convergence rates of 2D examples and even of nonlinear thermal fluid-structure interaction test cases with unstructured grids.

However, an important challenge when coupling two different time-dependent problems is to increase parallelization in time. We suggest a multirate Neumann-Neumann waveform relaxation algorithm to solve two heterogeneous coupled heat equations as an alternative to the Dirichlet-Neumann method. In order to fix the mismatch produced by the multirate feature at the space-time interface a linear interpolation is constructed.

Furthermore, we perform a one-dimensional convergence analysis for the nonmultirate fully discretized heat equations to find the optimal relaxation parameter in terms of the material coefficients, the step size and the mesh resolution. This gives a very efficient method which needs only two iterations. Numerical results confirm the analysis and show that the 1D nonmultirate optimal relaxation parameter is a very good estimator for the multirate 1D case and even for multirate and nonmultirate 2D examples.

Finally, we also include in this work a time adaptive version of the multirate Neumann-Neumann waveform relaxation method mentioned above. Building a variable step size multirate scheme allows each of the subsolvers to freely construct its own time grid independently of each other. Therefore, the overall coupled method is more efficient than the previous multirate version.

Popular Science Summary

The invention of computers revolutionized the way of doing science and in particular the field of mathematics. Computers were faster than any human mind in doing calculations and they did not make mistakes opening a wide range of possibilities that earlier where in practice uncomputable. Scientists observe the behavior of nature and find mathematical equations that model the different phenomena. In other words, they translate the multiplicity of phenomena observed in nature into mathematical language. Sometimes one is able to find a solution of an equation in the classical way, by pen and paper. This is called an analytical solution. However, most of the times this is not possible because the equations are too complicated and either the analytical solution is yet unknown or it has been proved to not exist. In those cases, it is still possible to find a numerical solution which is a discrete approximation to the unknown continuous analytical solution. Numerical analysis is the discipline that builds and analyzes new numerical methods to approximate solutions to all kinds of equations; from climate models to rocket engines.

The work in this thesis is motivated by the simulation of thermal fluid-structure interaction (FSI). The thermal interaction between fluids and structures, also called conjugate heat transfer, occurs when a deformable or moving structure transfers or receives heat from a surrounding or internal fluid flow. Examples of this are cooling of gas-turbine blades, thermal anti-icing systems of airplanes, supersonic reentry of vehicles from space or gas quenching, which is an industrial heat treatment of metal workpieces. These problems are usually too complex to solve them analytically, and therefore, numerical simulations of the conjugate heat transfer are essential.

There are three different aspects that one needs to take into account for the simulation of thermal FSI. Firstly, we need a fluid solver that models the behavior of the gas in the quenching process of metal workpieces or in the liquid chemicals of the anti-icing systems of airplanes. Secondly, a structure solver is needed to model the temperature distribution over the metal workpiece or the airplane. Thirdly, the temperature interaction in the places where the fluid and the structure meet needs to be taken into consideration as well. There are basically two approaches for the numerical simulation of thermal FSI. On one hand, one can set a numerical method that includes the fluid model, the structure model and the corresponding interaction building a new holistic model for each specific

application. This is called monolithic approach. As an alternative, one can reuse existing models for the simulation of the fluid and the structure and set a coupled numerical method to handle the interaction between fields in an iterative manner. This is known as partitioned approach and even though the advantages with respect to the monolithic are clear because *only* the coupling needs to be taken into account, it depends on an iterative procedure that does not guarantee in general to achieve a solution.

My contribution is focused on providing efficient partitioned numerical methods for the simulation of thermal FSI. The efficiency of a partitioned method is measured through the speed of the iterative solver to achieve an accurate numerical solution. Three scenarios are possible; the iteration does not converge to any solution, the method approaches to the solution but very slowly, meaning that needs many iterates to find it or the method is very fast and achieves the solution in few iterates. The first scenario is uninteresting and the speed of the iteration to determine if the method is fast and efficient or slow and inefficient is measured through its rate of convergence.

In this thesis, we have measured the rate of convergence of the Dirichlet-Neumann iteration which is one of the classical coupled partitioned methods for FSI simulation. We are interested in time-dependent problems. This means that we find the numerical solution over a certain time grid corresponding to a time interval. Then, for each of the values of the time grid, one performs the Dirichlet-Neumann iteration to coordinate the solution of the fluid and the structure models. The rate of convergence of the Dirichlet-Neumann method is highly dependent on the materials that one couples. In particular, the rate will be very small and consequently the coupled method will be very fast when there exist strong jumps in the material properties. This means that when for instance one couples air with steel where their densities and heat conductivities are strongly different, one gets a very fast method. Conversely, the rate will be larger and consequently the numerical method will be very slow or even divergent when the properties of the coupled materials are very similar to each other. In conclusion, the Dirichlet-Neumann iteration is a very good choice when coupling fields with strongly different material properties. This is exactly the situation we have in the air cooling of metal workpieces.

In spite of the efficient behavior of the Dirichlet-Neumann iteration in the thermal FSI framework, it has a main disadvantage. The subsolvers for the fluid and the structure wait for each other, and therefore, they perform the iterative procedure sequentially. In order to increase time parallelization we use the Neumann-Neumann waveform relaxation (NNWR) algorithm as an alternative to the Dirichlet-Neumann method. Using the NNWR algorithm we were able to construct a new method that allows at each iteration to find the solution of the two subproblems in parallel over the whole time interval before performing the coupling across the interfaces. In addition, this method also allows each of the subsolvers to freely construct their own time grid independently of each other. Therefore, our proposal increases parallelism and it is more efficient than the classical Dirichlet-Neumann method in most cases.

List of Papers

This thesis is based on the following papers, which will be referred to in the text by their Roman numerals. The papers are appended at the end of the work and I describe my contribution to each of them on the next page.

- I. P. Birken and A. Monge
Numerical Methods for Unsteady Thermal Fluid Structure Interaction,
in S. Frei, B. Holm, T. Richter, T. Wick and H. Yang (eds), Fluid-Structure
Interaction. Modeling, Adaptive Discretisations and Solvers, Contributions in
Mathematical and Computational Sciences, Springer, Berlin, 2017, pp. 129–168.
- II. A. Monge and P. Birken
**On the convergence rate of the Dirichlet-Neumann iteration for unsteady
thermal fluid-structure interaction.**
Computational Mechanics, First Online 2017.
DOI: 10.1007/s00466-017-1511-3.
- III. A. Monge and P. Birken
**A multirate Neumann-Neumann waveform relaxation method for
heterogeneous coupled heat equations.**
Preprint, arXiv:1805.04336, submitted to SISC, 2018.

Author's contribution

Since all the appended papers were co-authored I here describe my contribution to each.

- I. I wrote most of the introduction, performed the convergence analysis, the numerical results and was the main author in sections 4.3 and 4.4. Finally, to calculate the entries of the inverse matrices using their eigendecomposition was my idea. This is the core point of all the convergence analysis presented in this thesis.
- II. I performed the convergence analysis and implemented the numerical experiments (except the implementations regarding the nonlinear thermal fluid-structure interaction test cases). I was also the main author of the paper.
- III. I constructed the algorithms, performed the convergence analysis and implemented all the numerical results included in the paper. I was also the main author of the paper.

All the unpublished work to be presented in this thesis is my own contribution.

Acknowledgments

This PhD thesis represents the culmination of all my academic records and specially the work of the last five years. However, it would never have happened without the right circumstances and the right people surrounding me. Let me take the space here to acknowledge them.

Firstly, I would like to thank my supervisor Philipp Birken for sharing his best ideas and letting me play with them. Thank you for your support, guidance, motivation and infinite patience during all this time. Your passion for science and your optimistic view on modern research has been very inspiring for me. I am sure I will carry those values to wherever I will go.

Next, I want to thank my co-supervisors Claus Führer and Gustaf Söderlind for always being supportive and for all the valuable feedback and fruitful discussions. It has been a pleasure to share space time coordinates with you.

I address a big thank you to the Numerical Analysis group for the friendly environment. My gratitude to Eskil Hansen for reading my work in details, for his valuable criticisms and for all the interesting discussions afterwards. A special thanks to Carmen Arévalo for being the most inspiring reference in teaching I have ever met. It has been an honor to be your student and your teaching assistant. I also want to thank my fellow PhD students, present and past: Tony, Dara, Christian, Erik, Fatemeh, Peter, Lea and Julio for sharing a portion of this experience together. Thank you for being supportive in the ups and downs in the life as a PhD student and for understanding that better than anyone else. In particular I thank my academic twin sister Fatemeh Mohammadi for sharing this journey with me from the beginning and becoming a great friend on the way.

Let me also thank my family; my mother Estrella for being a constant source of inspiration and for transmitting us the passion for knowledge and education, and my siblings Aritz, Tanit and Coral for being as proud of me as I am of them and for being my first and best work team. A special thanks to my grandfather Sánchez for being the first person who showed me the beauty of mathematics through his logical card games and his endless square roots on the newspapers.

A warm thanks to all my friends for their support. To the ones growing up on the same little Mediterranean stone as me; Alba, Darío, María Marí, Núria and Tere and to

the international family I have been building over my adventure abroad; Abtihal, Alberto, Álvaro, Diego, Maria Navarro, Najmeh and Ylva.

Last but not least, I thank my love Pepe for his contagious positive attitude towards everything and for the thousands of kilometers traveled from the world to Lund over the last five years.

Azahar Monge
Lund, September, 2018

Funding

Part of the research presented in this thesis was made possible by funding from the strategic collaborative research programme in e-science eSENCE under the project *3M: Multiphysics, multicore, multirate solution of coupled dynamics problems*.

Contents

1	Introduction	1
1.1	State of the art	2
1.2	Thesis overview	3
1.3	Organization of the thesis	5
2	Thermal fluid-structure interaction	7
2.1	Motivation	7
2.2	Thermal FSI methodology	11
2.2.1	Governing equations	11
2.2.2	Partitioned coupled method	14
2.3	Thermal FSI test cases	15
2.3.1	Flow over a plate	15
2.3.2	Cooling of a flanged shaft	16
3	The classical partitioned approach	19
3.1	Model problem	20
3.2	Finite element discretization	22
3.3	Time discretization	24
3.4	Dirichlet-Neumann iteration	25
3.5	Semidiscrete analysis	26
4	Convergence analysis for FE discretizations	29
4.1	Discrete analysis in 1D	29
4.2	Discrete analysis in 2D	34
4.3	Numerical results	38
5	Convergence analysis for mixed discretizations	43
5.1	Two alternative FVM formulations	43
5.1.1	Numerical order comparison	47
5.2	Discrete analyses in 1D and 2D	47

6 Partitioned multirate approach 53

- 6.1 The Neumann-Neumann Waveform Relaxation algorithm 53
- 6.2 Space-time interface interpolation 54
- 6.3 Optimal relaxation parameter 56
- 6.4 Multirate SDIRK2-NNWR algorithm 57

7 Partitioned time adaptive approach 63

- 7.1 Time adaptive method 63
- 7.2 Time adaptive relaxation parameter 66
- 7.3 Numerical results 69

8 Conclusions and Outlook 77

- 8.1 Conclusions 77
- 8.2 Further work 78

Bibliography 81

Papers I–III

Chapter 1

Introduction

The space industry has grown strongly over the last years. The three major sectors of the space industry are satellite manufacturing, support ground equipment manufacturing and the launch industry. The "2017 Space Report" [31] estimates that in 2016 total global space activity was \$329 billion. The companies SpaceX and PLD Space founded in 2002 and 2011, respectively, believe that a fully and rapidly reusable rocket is the pivotal breakthrough needed to substantially reduce the cost of space access. On the other hand, nowadays communities are increasingly concerned about the impact of these industries on the environment. For example, the European commission published a report, named "European Aeronautics: a vision for 2020" [2], where as a main goal for this industry a 50 % reduction of the noise and the CO₂ emissions is specified. All these facts show the need for rapid development in this area.

Numerical analysis plays an important role in the development of space launch vehicles. The invention of super computers and advanced numerical methods allows numerical modelling of different aspects of an engine, helping in the design process. Modeling of the cooling/heating of the thrust chamber nozzle is a typical example. Thus, for a reasonable description a coupled simulation needs to be performed.

Due to new developments in computer technology during the last years the simulation of a coupled interaction between fluids and structures is now feasible. Nevertheless, investigating only the interaction between fluid forces and structural deformations in a rocket engine is not sufficient because the temperature plays a crucial role in the engine. The temperature achieved in the turbines has increased over the years due to progress in the building materials and therefore, advanced cooling methods are needed. Thus, developing an efficient model for the heat transfer is essential for the numerical simulation of these problems.

1.1 State of the art

The efficient numerical simulation of fluid-structure interaction (FSI) models is one of the current challenges in scientific computing as stated in [15]:

“The issue of coupling models of different events at different scales and governed by different physical laws is largely wide open and represents an enormously challenging area for future research.”

One finds examples in diverse fields like aeronautics [24], turbomachinery [98], biomechanics [73, 89], or even in the food industry [64]. There exist two main methods to simulate FSI models: the monolithic and the partitioned approaches.

In the monolithic method (or direct coupling), a new code is tailored for the coupled equations. Different methods exist for the boundary formulations, for instance, moving reference frames [53] or coordinate transformations [63].

On the other hand, the partitioned approach allows to reuse existing software for each sub-problem. The coupling is done by a master program which calls interface functions of the other codes [18, 20, 27]. If the data transfer between the subsolvers is done only once per time step, we are using a loosely coupled scheme [29]. However, for stability reasons, often a strongly coupled scheme needs to be used [51]. In this case the data exchange at every time step is repeated until a convergence criterion is satisfied.

The partitioned coupling methods offer the attractive advantage of reusing existing solvers for the fluid and the structure when simulating FSI problems [93, 14]. One of the basic methods for managing the coupling is the Dirichlet-Neumann iteration. More specifically, the Dirichlet-Neumann iteration is a basic method in both domain decomposition and fluid-structure interaction. To satisfy the coupling conditions at the interface, the subsolvers are iterated by providing Dirichlet- and Neumann data for the other solver in a sequential manner.

In the domain decomposition context, the iteration has two main problems, namely slow convergence and the need for an implementation using a red-black colouring. The slow convergence can be improved using a relaxation procedure. In fluid-structure interaction, there are only two domains, coupled along an interface, making the application straight forward. However, the convergence rate is not great for the coupling between a compressible fluid and a structure [23], which is why a lot of effort goes into convergence acceleration. On the other hand, the Dirichlet-Neumann iteration was reported to be a very fast solver for thermal fluid-structure interaction [9] although a convergence analysis has not been performed.

Thermal FSI applications typically consist of two domains with jumps in the material coefficients across a connecting interface. For this reason, the transmission problem is considered a basic building block in thermal fluid-structure interaction. For the case of coupled heat equations, a one dimensional stability analysis was presented by Giles [37].

There, an explicit time integration method was chosen with respect to the interface unknowns. On the other hand, Henshaw and Chand provided in [44] a method to analyze stability and convergence speed of the Dirichlet-Neumann iteration in 2D based on applying the continuous Fourier transform to the semi-discretized equations. Their result depends on ratios of thermal conductivities and diffusivities of the materials. This is similar to the situation in [18, 3] where the performance of the coupling for incompressible fluids is affected by the added mass effect.

In spite of the efficient behavior of the Dirichlet-Neumann iteration in the thermal FSI framework, it has two main disadvantages. Firstly, the subsolvers wait for each other, and therefore, they perform sequentially. Secondly, in the time-dependent case the Dirichlet-Neumann iteration is used at each time step and consequently, both fields are solved with a common time resolution. Using instead a time adaptive multirate scheme that allows for different time resolutions on each subdomain would be more efficient.

Some work has already been done in that direction for the simulation of FSI problems. A time adaptive partitioned approach built over the Dirichlet-Neumann iteration for thermal FSI was presented in [11]. In [57], two new iterative partitioned coupling methods that allow for the simultaneous execution of flow and structure solvers were introduced. Furthermore, coupled schemes with two different time integration methods or using two different time steps sometimes leads to a degradation of the overall time integration method to first order convergence affecting stability [7, 13, 38]. A more general solution when coupling time integration methods with arbitrary order using the coupling library preCICE is given by [71].

A new method that at each iteration solves the two subproblems over the whole time window before exchanging information across the interfaces for the coupling of two parabolic problems was introduced in [50, 34]. There, the Neumann-Neumann waveform relaxation (NNWR) method which is a waveform relaxation (WR) method based on the classical Neumann-Neumann iteration is described. The WR methods were originally introduced in [52] for ordinary differential equation (ODE) systems, and they were used for the first time to solve time-dependent PDEs in [35, 36]. They allow the use of different spatial and time discretizations for each subdomain which is specially useful in problems with strong jumps in the material coefficients [33] or the coupling of different models for the subdomains [32]. In [65], a pipeline implementation of the NNWR method together with its parallel efficiency is analyzed for the coupling of homogeneous materials. However, parallelization in time for the coupling of heterogeneous materials was not yet considered.

1.2 Thesis overview

The aim of this work is to present a high order time adaptive multirate partitioned coupled method for the efficient simulation of the solution of two coupled parabolic problems with strong jumps in the material coefficients that can be further extended to thermal

FSI simulation. This work is not simple and faces two main challenges. One hand, the efficiency of a partitioned approach always depends on the speed of its coupled iteration across the interfaces. For that reason, we analyze the convergence rates of different coupled methods explaining under which circumstances are those a great option for thermal FSI simulation. On the other hand, our time adaptive multirate method that couples two variable time integration schemes on the two subdomains increases efficiency of the overall coupled simulation with respect to current approaches.

The research collected in this thesis consists of three stages. Firstly, we performed a convergence analysis of the classical Dirichlet-Neumann partitioned coupled method motivated by the fast behavior observed in [9] for the simulation of thermal FSI. We concluded that strong jumps in the material coefficients of the coupled PDEs imply fast convergence. This is often the situation in thermal FSI where the density and thermal conductivity for the fluid are typically way smaller than for the structure. Conversely, the coupling iteration will be slow when the material coefficients are continuous over all the subdomains. Secondly, we proposed and analyzed a multirate partitioned approach based on the NNWR algorithm [50, 34] for two coupled parabolic problems with heterogeneous material coefficients. Additionally, we determined the optimal relaxation parameter Θ_{opt} in terms of the material coefficients and the temporal and spatial resolutions Δt and Δx . The method using Θ_{opt} converges extremely fast, typically within two iterations. Finally, time adaptivity is added to the multirate approach resulting in a partitioned coupled scheme that allows at each iteration to find the local solutions of the subproblems over a certain time window using different time step controllers. In this setting, one does not need to exchange information across the interface after each time step. The numerical results show the advantages of the time adaptive method over the previous multirate approach.

All the analyses contained in this thesis are performed after the full discretization of the coupled problem meaning both space and time dimensions. With regards to the space, we consider two different cases. We either use a finite element method (FEM) to discretize both subdomains or use a finite volume method (FVM) on one domain and a FEM on the other. With regards to time integration methods, we either use implicit Euler or a second order singly diagonally implicit Runge-Kutta method (SDIRK2). Once we have a fully discrete coupled problem, the efficiency of the resulting iterative method is measured through its convergence rate. In principle, the convergence rate of the method is analyzed in any standard book on domain decomposition methods, e.g. [69, 91]. There, the iteration matrix is derived in terms of the stiffness and mass matrices of finite element discretizations and the convergence rate is the spectral radius of that. However, this does not provide a quantitative answer, since the spectral radius is unknown. Computing the spectral radius is in general a non trivial task. In our context, the material properties are discontinuous across the interface and as a consequence, computing the spectral radius of the iteration matrix is even more difficult.

The core idea that it is used is the following: we computed the spectral radius of the

corresponding iteration matrix exactly in terms of the eigendecomposition of the resulting matrices for the model problem. In the 2D case, we provided an approximation of the convergence rate. The asymptotics of the convergence rates when approaching the continuous case in either time or space are also computed for the different choices of spatial discretizations. These results are consistent with numerical experiments. They describe the convergence rates of the thermal interaction between two different materials using the Dirichlet-Neumann iteration or the NNWR algorithm. Furthermore, we showed numerically that the theoretical results describing the convergence behavior of the two coupled linear heat equations are a very good estimate for the convergence rates of two nonlinear thermal FSI test cases.

1.3 Organization of the thesis

In chapter 2, the general framework of the thesis is presented. Basic concepts about thermal fluid-structure interaction are explained as well as its simulation. In chapter 3, we define the time-dependent transmission problem as a simplified model for thermal fluid-structure interaction in terms of the partial differential equations, boundary conditions and interface conditions. We also give a description of the discretizations and the Dirichlet-Neumann model is explained. Our convergence analysis for the discrete case of the model problem using Dirichlet-Neumann interface conditions is presented in chapter 4. In chapter 5, the analysis is repeated for mixed discretizations. Here we discretize one domain with finite volumes and the other with finite elements as usually done in the FSI models. In chapter 6, we present our multirate approach for the coupling of two heterogeneous materials based on the NNWR algorithm. This approach is further extended to its time adaptive version in chapter 7. Conclusions and future work can be found in the final chapter.

Chapter 2

Thermal fluid-structure interaction

Fluid structure interactions occurs when a deformable or moving structure interacts with a surrounding or internal fluid flow. Fluid structure interaction covers a wide range of subjects as aero-elasticity [28, 47], hydro-elasticity [75, 12], flow-induced vibration [74], thermal deformation [54, 100], the analysis of aneurysms in large arteries [90, 4, 5] and artificial heart valves [17, 22, 19] among others.

Our specific field of interest is thermal interaction between fluids and structures, the so-called conjugate heat transfer. Examples for thermal fluid-structure interaction are cooling of gas-turbine blades, thermal anti-icing systems of airplanes [16], supersonic reentry of vehicles from space [58, 45], gas quenching, which is an industrial heat treatment of metal workpieces [43, 88] (see figure 2.1) or the cooling of rocket nozzles [48, 49]. These problems are too complex to solve them analytically, and therefore, numerical simulations of conjugate heat transfer are essential in many applications.

2.1 Motivation

The research that will be presented in this thesis was inspired by the problem of cooling of rocket thrust chambers. In particular, we consider the Ariane 5 rockets which are manufactured under the authority of the European Space Agency (ESA). The Ariane 5 is used to deliver payloads into the geostationary transfer orbit (GTO) or the low Earth orbit (LEO), see figure 2.2a. The cooling of rocket thrust chambers we consider takes place in the first stage rocket engines for the Ariane 5 called Vulcain 2, see figure 2.2b.

Figure 2.2c shows a combustion chamber with an opening, the nozzle, through which gas can escape. The pressure distribution within the chamber varies a little, but near the nozzle it decreases. At the bottom of the combustion chamber the force due to gas

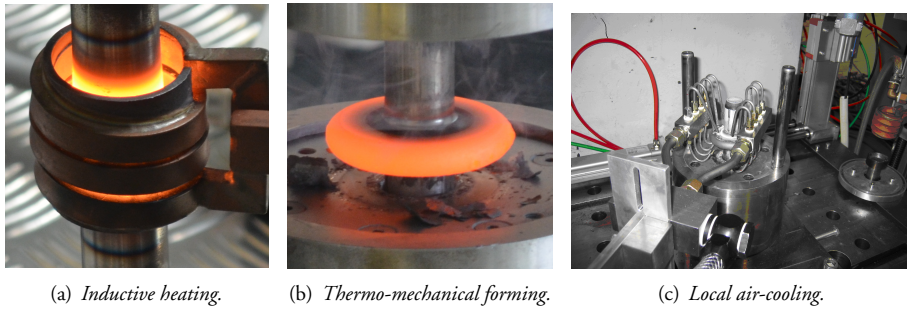


Figure 2.1: Gas quenching. Left and center picture taken from [39]. Right picture taken from [86].

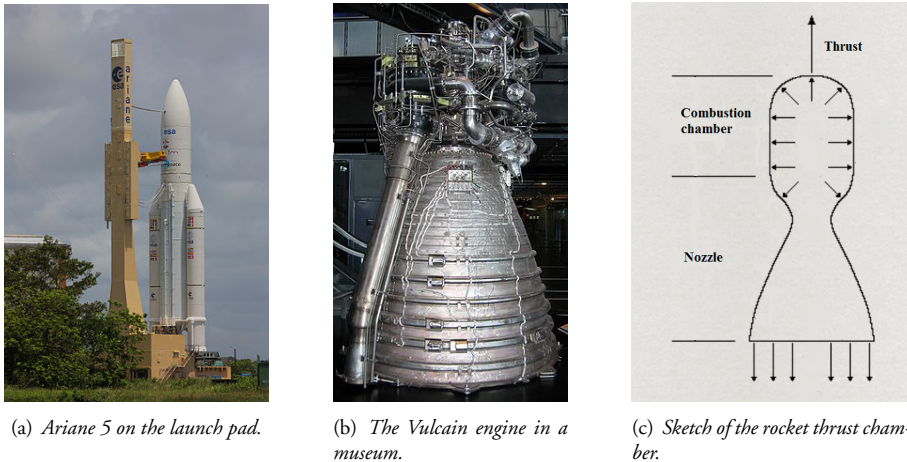


Figure 2.2: Cooling of rocket thrust chambers. Left picture: DLR German Aerospace Center, CC BY 2.0. Center picture: Pline, CC BY-SA 3.0

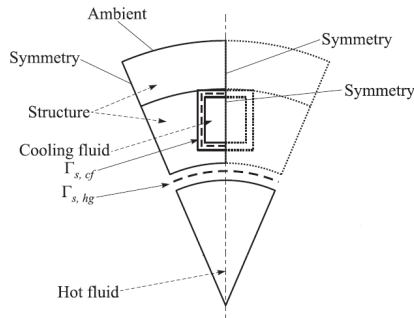


Figure 2.3: Sketch of the coupling surfaces. Figure taken from [48].

pressure difference is not compensated from the outside. The resulting force (the thrust) is opposite to the direction of the hot gas and it pushes the chamber upwards.

The cooling process of the rocket thrust chamber takes place during the combustion. Then, the hot gas produced during the combustion process is flowing inside the nozzle. The nozzle is delimited by a structure that can be damaged due to the high temperature of the gas flowing inside. In order to avoid this, the nozzle has a cooling system. A cooling fluid flows through small channels contained inside the structure. This results in a system with two related thermal interactions between fluids and structures. On one hand, the thermal interaction between the hot gas coming out from the combustion chamber and the structure recovering the nozzle. On the other hand, the thermal interaction between the cooling fluid and the structure. A sketch of the coupling surfaces can be consulted in figure 2.3 where $\Gamma_{s,cf}$ corresponds to the interface between the structure and the cooling fluid and $\Gamma_{s,hg}$ to the interface between the structure and the hot gas.

Currently, there exist several research goals related to the simulation of the cooling process of the rocket thrust chamber. First of all, engineers are concerned about the design of the cooling system in the nozzle. In other words, if the small channels are well designed to cool down the nozzle preventing damage in the structure. On the other hand, the first stage rocket engine is only used during the launch of the rocket. Therefore, recovering and reusing the first stage will reduce the cost of space access and the environment impact.

Related to this, the company SpaceX is developing a set of new technologies for an orbital launch system that may be reused many times in a manner similar to the reusability of aircraft. The first controlled vertical splashdown of an orbital rocket stage on the ocean surface was achieved in April 2014. The next two flights in January and April 2015 attempted to land the returning first stage on a floating platform. Both of them were guided accurately to the target, but they did not succeed in landing vertically on the floating platform and were destroyed [84]. The first vertical landing was achieved on December 21, 2015, when the first stage of Falcon 9 Flight 20 successfully landed on



(a) *Unsuccessful vertical landing attempt.* (b) *First stage landing vertically on solid ground in December 2015.* (c) *First stage landed on autonomous drone ship in April 2016.*

Figure 2.4: SpaceX reusable launch system development program. SpaceX Photos, CC0 1.0.

solid ground [79, 81]. On April 8, 2016, Flight 23 achieved the first soft landing on a drone ship in the Atlantic Ocean [80, 82], see figure 2.4. Finally, SpaceX achieved the world's first reflight of an orbital class rocket launching a Falcon 9 on March 30, 2017. The first stage successfully landed on a drone ship in the Atlantic ocean for a second time [83, 85].

With similar objectives, the Spanish company PLD Space founded in 2011 is developing reusable cost-effective launch vehicles. Since 2014 the company is operating a liquid fuel engine test stand located at Teruel airport where they have successfully accomplished their first tests [68]. Currently, the company is developing two rockets: ARION 1 [66] and ARION 2 [67]. On one hand, ARION 1 is designed to perform research or technology development in micro-gravity environment and/or in the upper atmosphere during sub-orbital flights. It can fly a payload of up to 200 kg and in its first mission scheduled for the middle of 2019 it will carry 100 kg of payload to an apogee of 150 km. A second test flight will follow at the end of the year and the commercial flight service is planned for 2020 targeting up to eight sub-orbital launches per year. In addition, ARION 1 is equipped with a recovery system, that enables PLD Space to recover and reuse the complete launch vehicle becoming the first recoverable rocket in Europe. On the other hand, ARION 2 provides orbital capabilities for smaller payloads (up to 150 kg), that need a flexible and dedicated launch vehicle and therefore can not fly with traditional vehicles. The first stage is designed to be reused several times. The first test flight of ARION 2 is planned for the end of 2020, with the commercial launch service



(a) Headquarters of PLD Space in Elche, Alicante.



(b) The reusable engine of ARION 1.

Figure 2.5: PLD Space reusable launch system development program. Benjamín Núñez González, CC BY-SA 4.0.

starting in 2021.

In view of these achievements, it is of main interest to simulate several cycles of the combustion-cooling process of the thrust chamber. This will tell how many times the engine can be used without damaging the structure. More details about this can be found in [48, 49].

2.2 Thermal FSI methodology

The basic setting we are in is that on a domain $\Omega_1 \subset \mathbb{R}^d$ where d corresponds to the spatial dimension, the physics is described by a fluid model, whereas on a domain $\Omega_2 \subset \mathbb{R}^d$, a different model describing the structure is used. The two domains are almost disjoint in that they are connected via an interface $\Gamma \subset \partial\Omega_1 \cup \partial\Omega_2$. At the interface Γ , coupling conditions are prescribed that model the interaction between fluid and structure. For the thermal coupling problem, these conditions are that temperature and the normal component of the heat flux are continuous across the interface. This thermal FSI methodology was used in papers I and II in the appendix.

2.2.1 Governing equations

The fluid is modeled using the time-dependent compressible Navier-Stokes equations, which are a second order system of conservation laws (mass, momentum, energy) modeling compressible flow. We consider the two dimensional case, written in conservative

variables density ρ , momentum $\mathbf{m} = \rho\mathbf{v}$ and energy per unit volume ρE as:

$$\begin{aligned} \partial_t \rho + \nabla \cdot \rho \mathbf{v} &= 0, \\ \partial_t \rho v_i + \sum_{j=1}^2 \partial_{x_j} (\rho v_i v_j + p \delta_{ij}) &= \frac{1}{Re} \sum_{j=1}^2 \partial_{x_j} S_{ij}, \quad i = 1, 2, \\ \partial_t \rho E + \nabla \cdot (\rho H \mathbf{v}) &= \frac{1}{Re} \sum_{j=1}^2 \partial_{x_j} \left(\sum_{i=1}^2 S_{ij} v_i + \frac{q_j}{Pr} \right). \end{aligned} \quad (2.1)$$

As the equations are dimensionless, the Reynolds number Re and the Prandtl number Pr appear. Here, enthalpy is given by $H = E + p/\rho$ with $p = (\gamma - 1)\rho(E - 1/2(v_1^2 + v_2^2))$ being the pressure and $\gamma = 1.4$ the adiabatic index for an ideal gas. Furthermore, $\mathbf{q}_f = (q_1, q_2)^T$ represents the heat flux and $\mathbf{S} = (S_{ij})_{i,j=1,2}$ the viscous shear stress tensor given by

$$S_{ij} = \mu(\partial_{x_j} v_i + \partial_{x_i} v_j) + \delta_{ij} \lambda (\partial_{x_1} v_1 + \partial_{x_2} v_2), \quad (2.2)$$

with μ being the viscosity and λ the thermal conductivity. The relation between μ and λ is defined by the Stoke's hypothesis to be $\lambda = -2/3\mu$. Moreover, the Sutherland law represents the correlation between temperature and viscosity:

$$\mu = T^{3/2} \left(\frac{1 + Su}{T + Su} \right), \quad (2.3)$$

with Su being the Sutherland constant and T the fluid temperature.

The nondimensional Navier-Stokes equations in (2.1) using the vector of conservative variables $\mathbf{u} = (\rho, \rho v_1, \rho v_2, \rho E)^T$, the convective fluxes

$$\mathbf{f}_1^c(\mathbf{u}) = \begin{pmatrix} \rho v_1 \\ \rho v_1 v_1 + p \\ \rho v_1 v_2 \\ \rho H v_1 \end{pmatrix}, \quad \mathbf{f}_2^c(\mathbf{u}) = \begin{pmatrix} \rho v_2 \\ \rho v_2 v_1 \\ \rho v_2 v_2 + p \\ \rho H v_2 \end{pmatrix}, \quad (2.4)$$

and the viscous fluxes

$$\mathbf{f}_1^v(\mathbf{u}) = \frac{1}{Re} \begin{pmatrix} 0 \\ S_{11} \\ S_{12} \\ \sum_{i=1}^2 S_{1i} v_i - \frac{q_1}{Pr} \end{pmatrix}, \quad \mathbf{f}_2^v(\mathbf{u}) = \frac{1}{Re} \begin{pmatrix} 0 \\ S_{21} \\ S_{22} \\ \sum_{i=1}^2 S_{2i} v_i - \frac{q_2}{Pr} \end{pmatrix}, \quad (2.5)$$

can be written as

$$\mathbf{u}_t + \partial_{x_1} \mathbf{f}_1^c(\mathbf{u}) + \partial_{x_2} \mathbf{f}_2^c(\mathbf{u}) = \partial_{x_1} \mathbf{f}_1^v(\mathbf{u}, \nabla \mathbf{u}) + \partial_{x_2} \mathbf{f}_2^v(\mathbf{u}, \nabla \mathbf{u}), \quad (2.6)$$

or in a more compact form:

$$\mathbf{u}_t + \nabla \cdot \mathbf{f}^c(\mathbf{u}) = \nabla \cdot \mathbf{f}^v(\mathbf{u}, \nabla \mathbf{u}). \quad (2.7)$$

The Reynolds Averaged Navier-Stokes equations (RANS) are a turbulence model that tries to resolve only the larger eddies and not smaller eddies and it is derived from the Navier-Stokes equations. The effect of small scale eddies is incorporated using additional terms in the original equations. In particular, the RANS equations are obtained by averaging the Navier-Stokes equations. Thus, every quantity is represented by a mean value plus a fluctuation:

$$\phi(x, t) = \bar{\phi}(x, t) + \phi'(x, t). \quad (2.8)$$

Furthermore, to avoid the computation of mean values of products, the Favre or density weighted average is introduced:

$$\tilde{\phi} = \frac{\overline{\rho\phi}}{\bar{\rho}}, \quad (2.9)$$

with a corresponding different fluctuation

$$\phi(x, t) = \tilde{\phi}(x, t) + \phi''(x, t). \quad (2.10)$$

We thus obtain the RANS equations:

$$\begin{aligned} \partial_t \bar{\rho} + \nabla \cdot \bar{\rho} \tilde{\mathbf{v}} &= 0, \\ \partial_t \bar{\rho} \tilde{\mathbf{v}} + \sum_{j=1}^2 \partial_{x_j} (\bar{\rho} \tilde{v}_i \tilde{v}_j + \bar{\rho} \tilde{\delta}_{ij}) &= \frac{1}{Re} \sum_{j=1}^2 \partial_{x_j} \left(\tilde{S}_{ij} + S_{ij}^R \right), \quad i = 1, 2, \\ \partial_t \bar{\rho} \tilde{E} + \nabla \cdot (\bar{\rho} \tilde{H} \tilde{\mathbf{v}}) &= \sum_{j=1}^2 \partial_{x_j} \left(\sum_{i=1}^2 \left(\overline{\left(\frac{1}{Re} S_{ij} - S_{ij}^R \right)} v_i + \widetilde{S_{ij} v_i''} \right) \right. \\ &\quad \left. - \bar{\rho} \tilde{v}_j'' - \bar{\rho} \widetilde{v_j'' k} + \frac{\bar{q}_j}{RePr} \right), \end{aligned} \quad (2.11)$$

where the Reynolds stresses are given by

$$S_{ij}^R = -\overline{\rho v_i'' v_j''}, \quad (2.12)$$

and the turbulent energy by

$$k = \frac{1}{2} \sum_{j=1}^2 \overline{v_j' v_j'}. \quad (2.13)$$

Additionally, we prescribe appropriate boundary conditions at the boundary of Ω_1 except for Γ , where we have the coupling conditions. In the Dirichlet-Neumann coupling, a temperature value is enforced at Γ .

Regarding the structure model, we will consider heat conduction only. Thus, we have the nonlinear heat equation for the structure temperature Θ

$$\rho(\mathbf{x})c_p(\Theta)\frac{d}{dt}\Theta(\mathbf{x}, t) = -\nabla \cdot \mathbf{q}(\mathbf{x}, t), \quad (2.14)$$

where

$$\mathbf{q}_s(\mathbf{x}, t) = -\lambda(\Theta)\nabla\Theta(\mathbf{x}, t)$$

denotes the heat flux vector. For alloys, the specific heat capacity c_p and heat conductivity λ are temperature-dependent and highly nonlinear.

2.2.2 Partitioned coupled method

As mentioned earlier in the introduction, we choose a partitioned implementation because this allows us to reuse existing software for each sub-problem. We use an inhouse code named Native for the structure computation and we got permission to use the DLR-TAU-Code [72] for solving the fluid. The partitioned approach is dealt with a C++ library named Component Template Library (CTL) [56]. With regards to the space discretization, we discretize the two fields separately following the partitioned approach. In particular, we use a finite element method (FEM) for the structure and a finite volume method (FVM) for the fluid. The coupling conditions have to be enforced via boundary conditions at the interface. In particular, one imposes that the temperature and the heat flux have to be continuous across the interface. This can be interpreted as prescribing the temperatures at the interface Θ_Γ provided by the structure solver to the fluid solver as Dirichlet boundary conditions at the interface. Similarly, the heat fluxes at the interface \mathbf{q}_Γ provided by the fluid solver are used as Neumann boundary conditions at the interface for the structure solver. This interaction between the fluid and the structure results in a fixed point iteration.

We study transient processes and therefore a numerical method needs to be chosen for the time discretization. In [55], the implicit midpoint rule is used in a monolithic

scheme to analyze energy conservation of an aeroelasticity problem. Already in [6, 21], it is suggested to use an explicit high order Runge-Kutta scheme for both subproblems with data exchange at each stage. However, the resulting scheme has limited time steps due to the explicit nature of the method. The order of coupling schemes on moving meshes is analyzed in [40], but only first order convergence is proved for p -th order schemes. Moreover, higher order implicit Runge-Kutta schemes on moving meshes are analyzed in [95] (in 1D) and in [96] (in 3D). There, so called explicit first stage, singly diagonally implicit Runge-Kutta schemes (ESDIRK) are employed and higher order in time is shown by numerical results. The master program of the FSI procedure can be extended to SDIRK methods very easily as explained in [8, ch.4]. Finally, time adaptivity can be added into this framework as explained in [10, 11].

Summarizing, simulation of thermal FSI involves two subsolvers (fluid and structure) and two embedded iterations. There exists an outer loop corresponding to the time integration. Then, at each time step, a subiteration enforces the temperature and the heat flux across the interface to be continuous, the so called Dirichlet-Neumann coupling. Namely, the boundary conditions for the two subsolvers are chosen such that we prescribe Dirichlet data for one solver and Neumann for the other. Figure 2.6 illustrates this.

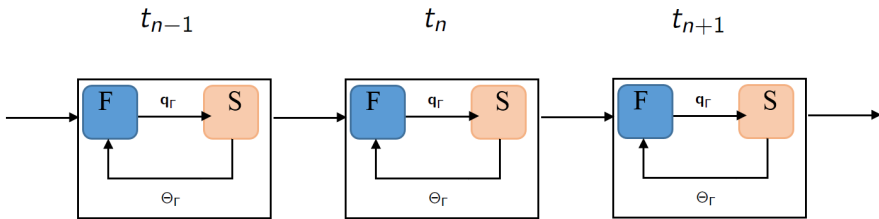


Figure 2.6: Illustration of the FSI solver.

2.3 Thermal FSI test cases

In this section we present two thermal FSI test cases that have been used in papers I and II to test our numerical methods. The two examples are cooling systems: the cooling of a flat plate and the cooling of a flanged shaft. For the first problem, structured grids are used and for the second, unstructured grids.

2.3.1 Flow over a plate

The first test case is the cooling of a flat steel plate resembling a simple work piece [11]. The work piece is initially at a much higher temperature than the fluid and then cooled

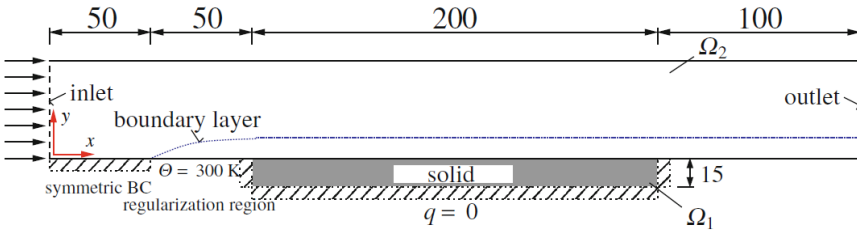


Figure 2.7: Sketch of the cooling of a flat plate.

by a constant laminar air stream, see figure 2.7.

The inlet is given on left, where air enters the domain with an initial velocity of $Ma_\infty = 0.8$ in horizontal direction and a temperature of 273K. Regarding the initial condition in the structure, a constant temperature of 900K at $t = 0$ is chosen throughout. For this test case, the obtained results were compared with the experiments of van Driest for the temperature boundary layer [94].

The grid, see figure 2.8, is chosen cartesian and equidistant in the structural part. In the fluid region the thinnest cells are on the boundary and then become coarser in y -direction. Furthermore, the points of the primary fluid grid and the nodes of the structural grid match on the interface Γ .

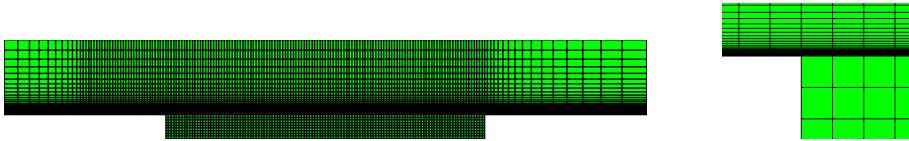


Figure 2.8: Full grid (left) and zoom into shaft region (right).

2.3.2 Cooling of a flanged shaft

The second test case is the cooling of a flanged steel shaft by cold high pressured air (this process is also known as gas quenching) [97]. Here, we have a hot flanged shaft that is cooled by cold high pressured air coming out of small tubes, see figure 2.9. We assume symmetry along the horizontal axis in order to consider one half of the flanged shaft and two tubes blowing air at it. We also assume that the air leaves the tube in straight and uniform way at a Mach number of 1.2. Moreover, we assume a freestream in x -direction of Mach 0.005. The Reynolds number is $Re = 2500$ and the Prandtl number $Pr = 0.72$.

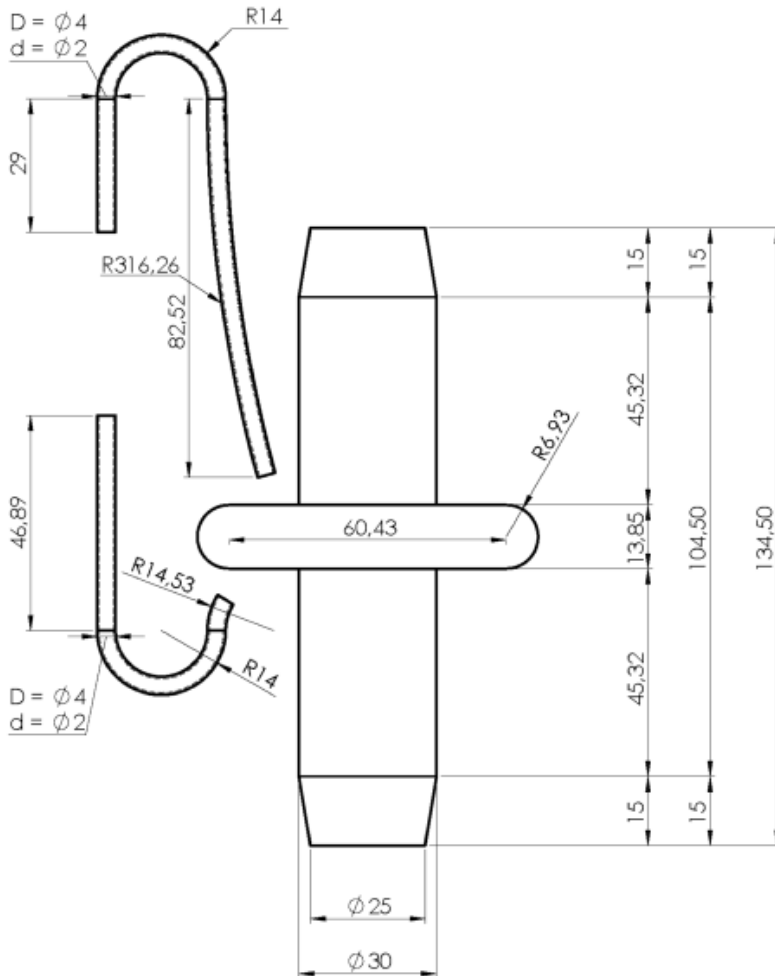


Figure 2.9: Sketch of the cooling of the flanged shaft.

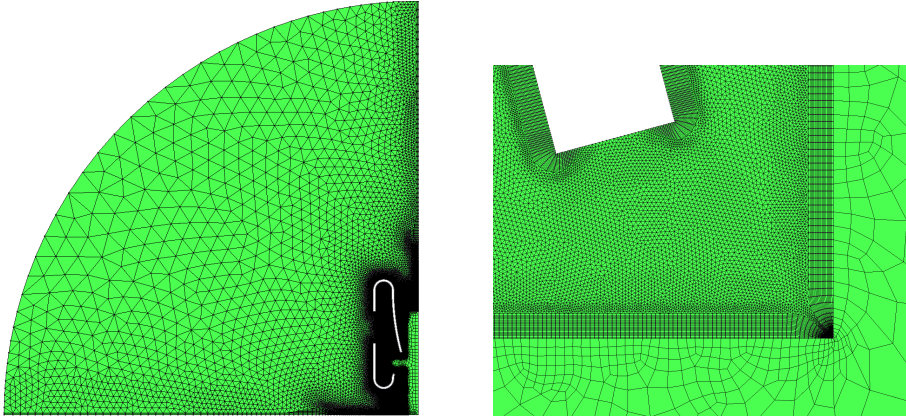


Figure 2.10: Full grid (left) and zoom into shaft region (right).

The grid, see figure 2.10, consists of an unstructured grid of quadrilaterals in the boundary layer and triangles in the rest of the domain for the fluid and quadrilateral elements in the structure. Regarding the initial conditions, we use the procedure explained in [9].

Chapter 3

The classical partitioned approach

In this chapter we are going to analyze the convergence behavior of the Dirichlet-Neumann iteration which is a classical domain decomposition method widely used for partitioned FSI simulation. This chapter and the next one were part of my Licentiate thesis which is a midterm degree in the Swedish high education system.

The subsolvers involving FSI simulations are usually computationally expensive. We are then interested in a coupling method that converges fast and helps the FSI procedure to gain efficiency. Therefore, a convergence analysis of the subiteration will help experts to choose parameters to run efficient FSI simulations. As explained before, the interactions between fluids and structures offer a wide range of possibilities and result in very different models depending on the context. All this influences the convergence analysis of the Dirichlet-Neumann coupling which will behave differently depending on diverse factors. Let's give some examples to illustrate this. For instance, the convergence rate of the Dirichlet-Neumann iteration is not great for the coupling between a compressible fluid and a structure [23], which is why a lot of effort goes into convergence acceleration. On the other hand, the Dirichlet-Neumann iteration was reported to be very fast solver for thermal fluid-structure interaction. More specifically, in [9] the subiteration is extremely efficient and achieves a very accurate solution with at most two iterates per time step. These numerical results motivated the work to be explained in this chapter. To analyze the convergence behavior of the Dirichlet-Neumann iteration will explain why this coupling method is very efficient in some FSI models and very inefficient in some others.

Some work has been done with respect to the convergence behavior of the subiteration in other FSI applications. In [3, 18] the design of the partitioned approach is improved when the coupling is affected by the added mass effect. In [92], it is shown for a model problem that on increasingly small time intervals, the added mass of a compress-

ible flow is proportional to the length of the time interval, whereas the added mass of an incompressible flow approaches a constant. The distinct proportionalities of the added mass to the time step for compressible and incompressible flows yield essentially different behavior of the subiteration method for fluid-structure problems. In a latter article [93], the convergence behavior of several partitioned iterative solution methods for FSI are examined. However, the thermal fluid-structure interaction model explained in the previous chapter is beyond current convergence theory.

3.1 Model problem

We are basically interested in the conjugate heat transfer between a compressible fluid and a solid. We typically think of the coupling between air and steel. It is then a sensible simplification to consider the linear heat equation with different constant coefficients corresponding to the two fields as a model for the thermal interaction between a fluid and a structure. In particular, we consider a domain $\Omega \subset \mathbb{R}^d$ which is cut into two subdomains $\Omega_1 \cup \Omega_2 = \Omega$. Each of the fields is restricted to one of the subdomains. Thus, we consider the following model:

$$\begin{aligned} \alpha(\mathbf{x}) \frac{\partial u(\mathbf{x}, t)}{\partial t} - \nabla \cdot (\lambda(\mathbf{x}) \nabla u(\mathbf{x}, t)) &= 0, \quad t \in [t_0, t_f], \quad \mathbf{x} \in \Omega \subset \mathbb{R}^d, \\ u(\mathbf{x}, t) &= 0, \quad t \in [t_0, t_f], \quad \mathbf{x} \in \partial\Omega, \\ u(\mathbf{x}, 0) &= u^0(\mathbf{x}), \quad \mathbf{x} \in \Omega, \end{aligned} \tag{3.1}$$

where the functions $\alpha(\mathbf{x})$ and $\lambda(\mathbf{x})$ are defined by

$$\alpha(\mathbf{x}) = \begin{cases} \alpha_1 & \text{if } \mathbf{x} \in \Omega_1 \\ \alpha_2 & \text{if } \mathbf{x} \in \Omega_2 \end{cases}, \quad \lambda(\mathbf{x}) = \begin{cases} \lambda_1 & \text{if } \mathbf{x} \in \Omega_1 \\ \lambda_2 & \text{if } \mathbf{x} \in \Omega_2 \end{cases}. \tag{3.2}$$

Here, the constants λ_1 and λ_2 describe the thermal conductivities of the materials on Ω_1 and Ω_2 respectively. D_1 and D_2 represent the thermal diffusivities of the materials and they are defined by

$$D_m = \frac{\lambda_m}{\alpha_m}, \quad \text{with } \alpha_m = \rho_m c_{p_m} \tag{3.3}$$

where ρ_m represents the density and c_{p_m} the heat capacity of the material placed in Ω_m , $m = 1, 2$. Note that (3.1) has a unique weak solution as proven by any book in PDEs, e.g. [26, chapt.7].

However, we want to rewrite (3.1) in a multi-domain formulation to be able to define the coupling iteration between the two subdomains and later analyze the convergence

behavior of that coupling. To that end, we consider the unsteady transmission problem with two coupling conditions at the interface $\Gamma = \Omega_1 \cap \Omega_2$. Those are continuity of the solutions and continuity of the heat fluxes across the interface:

$$\begin{aligned}
 \alpha_m \frac{\partial u_m(\mathbf{x}, t)}{\partial t} - \nabla \cdot (\lambda_m \nabla u_m(\mathbf{x}, t)) &= 0, \quad t \in [t_0, t_f], \quad \mathbf{x} \in \Omega_m \subset \mathbb{R}^d, \\
 u_m(\mathbf{x}, t) &= 0, \quad t \in [t_0, t_f], \quad \mathbf{x} \in \partial\Omega_m \setminus \Gamma, \\
 u_1(\mathbf{x}, t) &= u_2(\mathbf{x}, t), \quad \mathbf{x} \in \Gamma, \\
 \lambda_2 \frac{\partial u_2(\mathbf{x}, t)}{\partial \mathbf{n}_2} &= -\lambda_1 \frac{\partial u_1(\mathbf{x}, t)}{\partial \mathbf{n}_1}, \quad \mathbf{x} \in \Gamma, \\
 u_m(\mathbf{x}, 0) &= u_m^0(\mathbf{x}), \quad \mathbf{x} \in \Omega_m,
 \end{aligned} \tag{3.4}$$

where \mathbf{n}_m is the outward normal to Ω_m for $m = 1, 2$, and we consider $d = 1, 2$ (see figure 3.1).



Figure 3.1: Illustration of the domains for the model problem (3.4). On the left, domains when $d = 1$. On the right, domains when $d = 2$.

Problems (3.1) and (3.4) are equivalent in a variational sense as proved in any basic book on domain decomposition methods, e.g: [69, chap. 7].

In the one-dimensional case ($d = 1$), we discretize this problem with a constant mesh width of $\Delta x = 1/(N + 1)$ with N being the number of interior space discretization points in the intervals Ω_m , $m = 1, 2$. If instead we consider (3.4) with $d = 2$, we will use a constant mesh width with respect to both spatial components ($\Delta y := \Delta x$) resulting in N^2 interior space discretization points in both Ω_1 and Ω_2 . We use the implicit Euler method for the time discretization for both $d = 1, 2$. With regards to the spatial discretization, we consider two cases. On one hand, we use a finite element method (FEM) on both domains. On the other hand, we use a finite volume method (FVM) on Ω_1 and FEM on Ω_2 .

3.2 Finite element discretization

In this section we describe the FEM formulation of (3.4). Let $V_m := H_0^1(\Omega_m)$ be the closure of $C_0^\infty(\Omega)$ in the Sobolev space $H^1(\Omega)$. More specifically,

$$H_0^1(\Omega) := \{u \in H^1(\Omega) : \exists \{u_m\}_{m=1}^\infty \subset C_0^\infty(\Omega) \text{ s.t. } u_m \rightarrow u \text{ in } H^1(\Omega)\}, \quad (3.5)$$

where the Sobolev space $H^1(\Omega)$ is defined by

$$H^1(\Omega) := \{u \in L^2(\Omega) : D^b u \in L^2(\Omega) \text{ for } |b| \leq 1\}, \quad (3.6)$$

with

$$D^b u := \frac{\partial^{|b|}}{\partial x_1^{b_1} \dots \partial x_n^{b_n}}, \quad (3.7)$$

for $b \in \mathbb{N}_0^n$ and $b \neq 0$ with $|b| = \sum_{i=1}^n b_i$. If $b = 0$, then $D^0 u := u$.

A semi-discretization in space of the first two equations in (3.4) can be defined via a Galerkin approximation of the spaces V_m by finite dimensional subspaces $V_{m,h}$ for $m = 1, 2$. Then, the semi-discrete approximate problem reads as follows: Given $u_{m,h}^0 \in V_{m,h}$ being a suitable approximation of the initial data u_m^0 , find $u_{m,h} \in V_{m,h}$ such that for each $t \in [t_0, t_f]$,

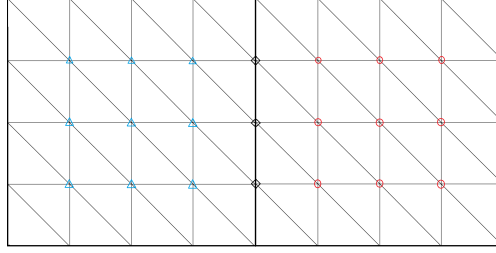
$$\begin{aligned} \alpha_m \int_{\Omega_m} \frac{d}{dt} u_{m,h} v_{m,h} d\mathbf{x} - \lambda_m \int_{\Omega_m} \Delta u_{m,h} v_{m,h} d\mathbf{x} &= 0 \quad \forall v_{m,h} \in V_{m,h}, \quad \mathbf{x} \in \Omega_m, \\ u_{1,h} &= u_{2,h} \quad \text{on } \Gamma, \end{aligned} \quad (3.8)$$

for $m = 1, 2$.

We suppose that the interface Γ does not cut any element (see figure 3.2). This implies that a global triangulation of Ω induces the two triangulations of Ω_1 and Ω_2 that are compatible on Γ (they share the same nodes on Γ).

It is now useful to consider a global finite dimensional subspace V_h in $H_0^1(\Omega)$. Let $\{\phi_j\}$ be a nodal basis of V_h and consequently $u_h(t) = \sum_j u_j(t) \phi_j$. Therefore, applying integration by parts to the first equation in (3.8) in order to remove the Laplacian operator, we can write the resulting discrete systems in the following compact form:

$$\mathbf{M}_1 \dot{\mathbf{u}}_I^{(1)} + \mathbf{M}_{I\Gamma}^{(1)} \dot{\mathbf{u}}_\Gamma + \mathbf{A}_1 \mathbf{u}_I^{(1)} + \mathbf{A}_{I\Gamma}^{(1)} \mathbf{u}_\Gamma = \mathbf{0}, \quad (3.9)$$

Figure 3.2: Splitting of Ω and finite element triangulation.

$$\mathbf{M}_2 \dot{\mathbf{u}}_I^{(2)} + \mathbf{M}_{I\Gamma}^{(2)} \dot{\mathbf{u}}_\Gamma + \mathbf{A}_2 \mathbf{u}_I^{(2)} + \mathbf{A}_{I\Gamma}^{(2)} \mathbf{u}_\Gamma = \mathbf{0}. \quad (3.10)$$

Here, the unknown coefficient functions $\mathbf{u}_I^{(1)}$ and $\mathbf{u}_I^{(2)}$ correspond to the interior nodes on Ω_1 and Ω_2 respectively and \mathbf{u}_Γ corresponds to the nodes at the interface Γ . \mathbf{A}_m and \mathbf{M}_m are the stiffness and the mass matrices for the interior nodes on Ω_m , $m = 1, 2$ and they are given by

$$(\mathbf{A}_m)_{ij} = \lambda_m \int_{\Omega_m} \nabla \phi_i \nabla \phi_j d\mathbf{x}, \quad \mathbf{x} \in \Omega_m, \quad (3.11)$$

$$(\mathbf{M}_m)_{ij} = \alpha_m \int_{\Omega_m} \phi_i \phi_j d\mathbf{x}, \quad \mathbf{x} \in \Omega_m. \quad (3.12)$$

The required data from the interface is inserted in the equations by the matrices $\mathbf{A}_{I\Gamma}^{(m)}$ and $\mathbf{M}_{I\Gamma}^{(m)}$, $m = 1, 2$ given by (3.11) and (3.12) as well, but with i running over the interior nodes of Ω_m and j over the nodes at Γ .

Finally, if ϕ_j is a nodal basis function for a node on Γ we observe that the normal derivatives in the fourth equation of (3.4) can be written as linear functionals using Green's formula [91, pp. 3]. Thus,

$$\begin{aligned} \lambda_m \int_{\Gamma} \frac{\partial u_m}{\partial \mathbf{n}_m} \phi_j dS &= \lambda_m \int_{\Omega_m} (\Delta u_m \phi_j + \nabla u_m \nabla \phi_j) d\mathbf{x} \\ &= \alpha_m \int_{\Omega_m} \frac{d}{dt} u_m \phi_j + \lambda_m \int_{\Omega_m} \nabla u_m \nabla \phi_j d\mathbf{x}, \quad m = 1, 2. \end{aligned} \quad (3.13)$$

Letting j run over the nodes on Γ we obtain the following compact expression equivalent to the fourth equation in (3.4):

$$\mathbf{M}_{\Gamma\Gamma}^{(2)}\dot{\mathbf{u}}_{\Gamma} + \mathbf{M}_{\Gamma I}^{(2)}\dot{\mathbf{u}}_I^{(2)} + \mathbf{A}_{\Gamma\Gamma}^{(2)}\mathbf{u}_{\Gamma} + \mathbf{A}_{\Gamma I}^{(2)}\mathbf{u}_I^{(2)} = -\mathbf{M}_{\Gamma\Gamma}^{(1)}\dot{\mathbf{u}}_{\Gamma} - \mathbf{M}_{\Gamma I}^{(1)}\dot{\mathbf{u}}_I^{(1)} - \mathbf{A}_{\Gamma\Gamma}^{(1)}\mathbf{u}_{\Gamma} - \mathbf{A}_{\Gamma I}^{(1)}\mathbf{u}_I^{(1)}. \quad (3.14)$$

Here, $\mathbf{A}_{\Gamma\Gamma}^{(m)}$ and $\mathbf{M}_{\Gamma\Gamma}^{(m)}$ are the stiffness and the mass matrix with respect to the nodes located at Γ and they are given by (3.11) and (3.12) with i, j running over the nodes at Γ . Finally, the required data from domains Ω_1 and Ω_2 is inserted in the equation (3.14) by the matrices $\mathbf{A}_{\Gamma I}^{(m)}$ and $\mathbf{M}_{\Gamma I}^{(m)}$, $m = 1, 2$ given by (3.11) and (3.12) with i running over the nodes at Γ and j over the interior nodes of Ω_m .

Thus, equation (3.14) completes the system (3.9)-(3.10). We now reformulate the coupled equations (3.9), (3.10) and (3.14) into an ODE for the vector of unknowns $\mathbf{u} = (\mathbf{u}_I^{(1)}, \mathbf{u}_I^{(2)}, \mathbf{u}_{\Gamma})^T$

$$\tilde{\mathbf{M}}\dot{\mathbf{u}} + \tilde{\mathbf{A}}\mathbf{u} = 0, \quad (3.15)$$

where

$$\tilde{\mathbf{M}} = \begin{pmatrix} \mathbf{M}_1 & \mathbf{0} & \mathbf{M}_{\Gamma\Gamma}^{(1)} \\ \mathbf{0} & \mathbf{M}_2 & \mathbf{M}_{\Gamma\Gamma}^{(2)} \\ \mathbf{M}_{\Gamma I}^{(1)} & \mathbf{M}_{\Gamma I}^{(2)} & \mathbf{M}_{\Gamma\Gamma}^{(1)} + \mathbf{M}_{\Gamma\Gamma}^{(2)} \end{pmatrix}, \quad \tilde{\mathbf{A}} = \begin{pmatrix} \mathbf{A}_1 & \mathbf{0} & \mathbf{A}_{\Gamma\Gamma}^{(1)} \\ \mathbf{0} & \mathbf{A}_2 & \mathbf{A}_{\Gamma\Gamma}^{(2)} \\ \mathbf{A}_{\Gamma I}^{(1)} & \mathbf{A}_{\Gamma I}^{(2)} & \mathbf{A}_{\Gamma\Gamma}^{(1)} + \mathbf{A}_{\Gamma\Gamma}^{(2)} \end{pmatrix}.$$

3.3 Time discretization

Applying the implicit Euler method with time step Δt to the system (3.15), we get for the vector of unknowns $\mathbf{u}^{n+1} = (\mathbf{u}_I^{(1),n+1}, \mathbf{u}_I^{(2),n+1}, \mathbf{u}_{\Gamma}^{n+1})^T$

$$\mathbf{A}\mathbf{u}^{n+1} = \tilde{\mathbf{M}}\mathbf{u}^n, \quad (3.16)$$

where

$$\mathbf{A} = \tilde{\mathbf{M}} + \Delta t\tilde{\mathbf{A}} = \begin{pmatrix} \mathbf{M}_1 + \Delta t\mathbf{A}_1 & \mathbf{0} & \mathbf{M}_{\Gamma\Gamma}^{(1)} + \Delta t\mathbf{A}_{\Gamma\Gamma}^{(1)} \\ \mathbf{0} & \mathbf{M}_2 + \Delta t\mathbf{A}_2 & \mathbf{M}_{\Gamma\Gamma}^{(2)} + \Delta t\mathbf{A}_{\Gamma\Gamma}^{(2)} \\ \mathbf{M}_{\Gamma I}^{(1)} + \Delta t\mathbf{A}_{\Gamma I}^{(1)} & \mathbf{M}_{\Gamma I}^{(2)} + \Delta t\mathbf{A}_{\Gamma I}^{(2)} & \mathbf{M}_{\Gamma\Gamma} + \Delta t\mathbf{A}_{\Gamma\Gamma} \end{pmatrix},$$

with $\mathbf{M}_{\Gamma\Gamma} = \mathbf{M}_{\Gamma\Gamma}^{(1)} + \mathbf{M}_{\Gamma\Gamma}^{(2)}$ and $\mathbf{A}_{\Gamma\Gamma} = \mathbf{A}_{\Gamma\Gamma}^{(1)} + \mathbf{A}_{\Gamma\Gamma}^{(2)}$.

3.4 Dirichlet-Neumann iteration

We face now the task of solving the discrete multi-domain problem (3.16) by an iterative procedure. In particular, we employ the standard Dirichlet-Neumann iteration that consists of two fractional steps corresponding to the two subdomains Ω_1 and Ω_2 . Given an initial guess \mathbf{u}_Γ^0 , we first solve a Dirichlet problem on Ω_1 with Dirichlet data \mathbf{u}_Γ^0 at the interface Γ , and then a Neumann problem on Ω_2 with a Neumann condition at Γ determined by the solution $\mathbf{u}_\Gamma^{(1)}$ on Ω_1 obtained in the previous step.

Thus, from (3.16) employing the Dirichlet-Neumann method one gets for the k -th iteration the two equation systems

$$(\mathbf{M}_1 + \Delta t \mathbf{A}_1) \mathbf{u}_I^{(1),n+1,k+1} = -(\mathbf{M}_{\Gamma}^{(1)} + \Delta t \mathbf{A}_{\Gamma}^{(1)}) \mathbf{u}_\Gamma^{n+1,k} + \mathbf{M}_1 \mathbf{u}_I^{(1),n} + \mathbf{M}_{\Gamma}^{(1)} \mathbf{u}_\Gamma^n, \quad (3.17)$$

$$\hat{\mathbf{A}} \hat{\mathbf{u}}^{k+1} = \hat{\mathbf{M}} \mathbf{u}^n - \mathbf{b}^k, \quad (3.18)$$

to be solved in succession. Here,

$$\hat{\mathbf{A}} = \begin{pmatrix} \mathbf{M}_2 + \Delta t \mathbf{A}_2 & \mathbf{M}_{\Gamma}^{(2)} + \Delta t \mathbf{A}_{\Gamma}^{(2)} \\ \mathbf{M}_{\Gamma}^{(2)} + \Delta t \mathbf{A}_{\Gamma}^{(2)} & \mathbf{M}_{\Gamma}^{(2)} + \Delta t \mathbf{A}_{\Gamma}^{(2)} \end{pmatrix}, \quad \hat{\mathbf{M}} = \begin{pmatrix} \mathbf{0} & \mathbf{M}_2 & \mathbf{M}_{\Gamma}^{(2)} \\ \mathbf{M}_{\Gamma}^{(1)} & \mathbf{M}_{\Gamma}^{(2)} & \mathbf{M}_{\Gamma} \end{pmatrix},$$

and

$$\mathbf{b}^k = \begin{pmatrix} \mathbf{0} \\ (\mathbf{M}_{\Gamma}^{(1)} + \Delta t \mathbf{A}_{\Gamma}^{(1)}) \mathbf{u}_I^{(1),n+1,k+1} + (\mathbf{M}_{\Gamma}^{(1)} + \Delta t \mathbf{A}_{\Gamma}^{(1)}) \mathbf{u}_\Gamma^{n+1,k} \end{pmatrix},$$

$$\hat{\mathbf{u}}^{k+1} = \begin{pmatrix} \mathbf{u}_I^{(2),n+1,k+1} \\ \mathbf{u}_\Gamma^{n+1,k+1} \end{pmatrix},$$

with some initial condition, here $\mathbf{u}_\Gamma^{n+1,0} = \mathbf{u}_\Gamma^n$. The iteration is terminated according to the standard criterion $\|\mathbf{u}_\Gamma^{k+1} - \mathbf{u}_\Gamma^k\| \leq \tau$ where τ is a user defined tolerance.

One way to analyze this method is to write it as a splitting method for (3.16) and try to estimate the spectral radius of that iteration. However, the results obtained in this way are much too inaccurate. For that reason, we now rewrite (3.17)-(3.18) as an iteration for \mathbf{u}_Γ^{n+1} to restrict the size of the space to the dimension of \mathbf{u}_Γ . To this end, we isolate the term $\mathbf{u}_I^{(1),n+1,k+1}$ from (3.17) and $\mathbf{u}_I^{(2),n+1,k+1}$ from the first equation in (3.18):

$$\mathbf{u}_I^{(1),n+1,k+1} = (\mathbf{M}_1 + \Delta t \mathbf{A}_1)^{-1} (-(\mathbf{M}_{I\Gamma}^{(1)} + \Delta t \mathbf{A}_{I\Gamma}^{(1)}) \mathbf{u}_\Gamma^{n+1,k} + \mathbf{M}_1 \mathbf{u}_I^{(1),n} + \mathbf{M}_{I\Gamma}^{(1)} \mathbf{u}_\Gamma^n), \quad (3.19)$$

$$\mathbf{u}_I^{(2),n+1,k+1} = (\mathbf{M}_2 + \Delta t \mathbf{A}_2)^{-1} (-(\mathbf{M}_{I\Gamma}^{(2)} + \Delta t \mathbf{A}_{I\Gamma}^{(2)}) \mathbf{u}_\Gamma^{n+1,k+1} + \mathbf{M}_2 \mathbf{u}_I^{(2),n} + \mathbf{M}_{I\Gamma}^{(2)} \mathbf{u}_\Gamma^n). \quad (3.20)$$

Inserting (3.19) and (3.20) into the second equation of (3.18) one obtains the iteration $\mathbf{u}_\Gamma^{n+1,k+1} = \Sigma \mathbf{u}_\Gamma^{n+1,k} + \psi^n$, with iteration matrix

$$\Sigma = -\mathbf{S}^{(2)-1} \mathbf{S}^{(1)}, \quad (3.21)$$

where

$$\mathbf{S}^{(m)} = (\mathbf{M}_{I\Gamma}^{(m)} + \Delta t \mathbf{A}_{I\Gamma}^{(m)}) - (\mathbf{M}_{I\Gamma}^{(m)} + \Delta t \mathbf{A}_{I\Gamma}^{(m)}) (\mathbf{M}_m + \Delta t \mathbf{A}_m)^{-1} (\mathbf{M}_{I\Gamma}^{(m)} + \Delta t \mathbf{A}_{I\Gamma}^{(m)}), \quad (3.22)$$

for $m = 1, 2$ and ψ^n contains terms that depend only on the solutions at the previous time step. Notice that Σ is a discrete version of the Steklov-Poincaré operator.

Then, the Dirichlet-Neumann iteration $\mathbf{u}_\Gamma^{n+1,k+1} = \Sigma \mathbf{u}_\Gamma^{n+1,k} + \psi^n$ is convergent only if the powers Σ^k go to zero. And those approach zero if and only if every eigenvalue of Σ satisfies that its absolute value is smaller than 1. The rate of convergence of the iterative method is then governed by the largest absolute eigenvalue, which is called the spectral radius of Σ . This result is proved in any basic book on numerical analysis, e.g. [87, chap. 5]. Thus, the Dirichlet-Neumann iteration is a linear iteration and the rate of convergence is described by the spectral radius of the iteration matrix Σ .

3.5 Semidiscrete analysis

Before we present our analysis for the fully discrete equations, we want to describe previous results which analyze the behavior of the Dirichlet-Neumann iteration for the transmission problem in the semidiscrete case. Henshaw and Chand applied in [44] the implicit Euler method for the time discretization on both equations in (3.4) but kept the space continuous. Then, they applied the Fourier transform in space (with dual variable k) in order to transform the second order derivatives into algebraic expressions. Once they have a coupled system of algebraic equations, they insert one into the other and obtain the Dirichlet-Neumann convergence rate β :

$$\beta = \left| -\frac{\lambda_1}{\lambda_2} \sqrt{\frac{1/(D_1\Delta t) + k^2}{1/(D_2\Delta t) + k^2}} \frac{\tanh\left(-\sqrt{1/(D_2\Delta t) + k^2}\right)}{\tanh\left(\sqrt{1/(D_1\Delta t) + k^2}\right)} \right|. \quad (3.23)$$

In the 1D case, the transverse Fourier mode k is zero. Then, for Δt small enough, we have $\tanh(-1/\sqrt{D_2\Delta t}) \approx -1$ and $\tanh(1/\sqrt{D_1\Delta t}) \approx 1$ and therefore:

$$\beta \approx \frac{\lambda_1}{\lambda_2} \sqrt{\frac{D_2}{D_1}}. \quad (3.24)$$

On the other hand, for Δt big enough, we have $\tanh(-1/\sqrt{D_2\Delta t}) \approx -1/\sqrt{D_2\Delta t}$ and $\tanh(1/\sqrt{D_1\Delta t}) \approx 1/\sqrt{D_1\Delta t}$ and therefore:

$$\beta \approx \frac{\lambda_1}{\lambda_2} \sqrt{\frac{D_2}{D_1}} \frac{\sqrt{D_1\Delta t}}{\sqrt{D_2\Delta t}} = \frac{\lambda_1}{\lambda_2}. \quad (3.25)$$

Chapter 4

Convergence analysis for FE discretizations

In this chapter we perform a convergence analysis of the coupling iteration for the unsteady transmission problem (3.4) presented in the previous chapter. In FSI models, the solid is typically discretized using finite elements. On the other hand, although the finite element method (FEM) is applicable to computational fluid dynamics, the finite volume method (FVM) is generally a better choice for the discretization of the fluid. This method guarantees the conservation of fluxes through a particular control volume. Moreover, FVM has also an advantage in memory usage and solution speed.

Nevertheless, this analysis with mixed discretizations will not be presented until the next chapter. Firstly, we introduce in this chapter a convergence analysis of the coupling iteration for the unsteady transmission problem using finite elements for the discretization of both subdomains. This simplification of the model let us compare the obtained results with previous research that is summarized in section 3.5.

4.1 Discrete analysis in 1D

We now study the iteration matrix Σ for two specific linear FEM discretizations in 1D and 2D. A previous work using finite differences can be found in [62, pp. 452-463] and a summary of the one-dimensional case using finite elements in [61, pp. 733-734]. Here, we will give an exact formula for the convergence rates in 1D and we will propose an approximation for the 2D case. The behavior of the rates when approaching both the continuous case in time and space is also given.

Firstly, we analyze the iteration matrix for the 1D case. Specifically, we use $\Omega_1 = [-1, 0]$, $\Omega_2 = [0, 1]$ and the standard piecewise-linear polynomials

$$\phi_k(x) := \begin{cases} \frac{x-x_{k-1}}{x_k-x_{k-1}}, & \text{if } x_{k-1} < x \leq x_k \\ \frac{x_{k+1}-x}{x_{k+1}-x_k}, & \text{if } x_k < x \leq x_{k+1} \\ 0, & \text{otherwise} \end{cases} \quad (4.1)$$

as test functions.

If we consider $\mathbf{e}_j = (0 \ \cdots \ 0 \ 1 \ 0 \ \cdots \ 0)^T \in \mathbb{R}^N$ where the only nonzero entry is located at the j -th position, the discretization matrices are given by

$$\mathbf{A}_m = \frac{\lambda_m}{\Delta x^2} \begin{pmatrix} 2 & -1 & & 0 \\ -1 & 2 & \ddots & \\ & \ddots & \ddots & -1 \\ 0 & & -1 & 2 \end{pmatrix}, \quad \mathbf{M}_m = \frac{\alpha_m}{6} \begin{pmatrix} 4 & 1 & & 0 \\ 1 & 4 & \ddots & \\ & \ddots & \ddots & 1 \\ 0 & & 1 & 4 \end{pmatrix},$$

$$\begin{aligned} \mathbf{A}_{\Gamma\Gamma}^{(1)} &= \frac{\lambda_1}{\Delta x^2}, \quad \mathbf{A}_{\Gamma\Gamma}^{(2)} = \frac{\lambda_2}{\Delta x^2}, \quad \mathbf{M}_{\Gamma\Gamma}^{(1)} = \frac{2\alpha_1}{6}, \quad \mathbf{M}_{\Gamma\Gamma}^{(2)} = \frac{2\alpha_2}{6}, \\ \mathbf{A}_{\Gamma I}^{(1)} &= -\frac{\lambda_1}{\Delta x^2} \mathbf{e}_N, \quad \mathbf{A}_{\Gamma I}^{(2)} = -\frac{\lambda_2}{\Delta x^2} \mathbf{e}_1, \quad \mathbf{A}_{I\Gamma}^{(1)} = -\frac{\lambda_1}{\Delta x^2} \mathbf{e}_N^T, \quad \mathbf{A}_{I\Gamma}^{(2)} = -\frac{\lambda_2}{\Delta x^2} \mathbf{e}_1^T, \\ \mathbf{M}_{I\Gamma}^{(1)} &= \frac{\alpha_1}{6} \mathbf{e}_N, \quad \mathbf{M}_{I\Gamma}^{(2)} = \frac{\alpha_2}{6} \mathbf{e}_1, \quad \mathbf{M}_{\Gamma I}^{(1)} = \frac{\alpha_1}{6} \mathbf{e}_N^T, \quad \mathbf{M}_{\Gamma I}^{(2)} = \frac{\alpha_2}{6} \mathbf{e}_1^T, \end{aligned}$$

where $\Delta x = 1/(N+1)$, $\mathbf{A}_m, \mathbf{M}_m \in \mathbb{R}^{N \times N}$, $\mathbf{A}_{\Gamma\Gamma}^{(m)}, \mathbf{M}_{\Gamma\Gamma}^{(m)} \in \mathbb{R}^{N \times 1}$ and $\mathbf{A}_{\Gamma I}^{(m)}, \mathbf{M}_{\Gamma I}^{(m)} \in \mathbb{R}^{1 \times N}$ for $m = 1, 2$.

Note that the iteration matrix Σ is just a real number in this case and thus its spectral radius is its modulus. The goal now is to compute $\mathbf{S}^{(1)}$ and $\mathbf{S}^{(2)}$. Inserting the corresponding matrices specified in (3.22) we have

$$\begin{aligned} \mathbf{S}^{(1)} &= \left(\frac{\alpha_1}{3} + \Delta t \frac{\lambda_1}{\Delta x^2} \right) - \left(\frac{\alpha_1}{6} - \Delta t \frac{\lambda_1}{\Delta x^2} \right)^2 \mathbf{e}_N^T (\mathbf{M}_1 + \Delta t \mathbf{A}_1)^{-1} \mathbf{e}_N \\ &= \left(\frac{\alpha_1}{3} + \Delta t \frac{\lambda_1}{\Delta x^2} \right) - \left(\frac{\alpha_1}{6} - \Delta t \frac{\lambda_1}{\Delta x^2} \right)^2 \alpha_{NN}^1, \end{aligned} \quad (4.2)$$

$$\begin{aligned} \mathbf{S}^{(2)} &= \left(\frac{\alpha_2}{3} + \Delta t \frac{\lambda_2}{\Delta x^2} \right) - \left(\frac{\alpha_2}{6} - \Delta t \frac{\lambda_2}{\Delta x^2} \right)^2 \mathbf{e}_1^T (\mathbf{M}_2 + \Delta t \mathbf{A}_2)^{-1} \mathbf{e}_1 \\ &= \left(\frac{\alpha_2}{3} + \Delta t \frac{\lambda_2}{\Delta x^2} \right) - \left(\frac{\alpha_2}{6} - \Delta t \frac{\lambda_2}{\Delta x^2} \right)^2 \alpha_{11}^2, \end{aligned} \quad (4.3)$$

where α_{ij}^m represent the entries of the matrices $(\mathbf{M}_m + \Delta t \mathbf{A}_m)^{-1}$ for $i, j = 1, \dots, N$, $m = 1, 2$. Observe that the matrices $\mathbf{M}_m + \Delta t \mathbf{A}_m$, $m = 1, 2$ are tridiagonal Toeplitz matrices but their inverses are full matrices. The computation of the exact inverses is based on a recursive formula which runs over the entries [30] and consequently, it is non trivial to compute α_{NN}^1 and α_{11}^2 this way.

Due to these difficulties, we propose to rewrite the matrices $(\mathbf{M}_m + \Delta t \mathbf{A}_m)^{-1}$, $m = 1, 2$ in terms of their eigendecomposition:

$$\begin{aligned} & (\mathbf{M}_m + \Delta t \mathbf{A}_m)^{-1} \\ &= \left[\text{tridiag} \left(\frac{\alpha_m \Delta x^2 - 6\lambda_m \Delta t}{6\Delta x^2}, \frac{2\alpha_m \Delta x^2 + 6\lambda_m \Delta t}{3\Delta x^2}, \frac{\alpha_m \Delta x^2 - 6\lambda_m \Delta t}{6\Delta x^2} \right) \right]^{-1} \\ &= \mathbf{V} \Lambda_m^{-1} \mathbf{V}, \end{aligned} \tag{4.4}$$

for $m = 1, 2$ where the matrix \mathbf{V} has the eigenvectors of any symmetric tridiagonal Toeplitz matrix as columns. The entries of \mathbf{V} are not dependent on the entries of $\mathbf{M}_m + \Delta t \mathbf{A}_m$ due to their symmetry. Moreover, the matrix Λ_m is a diagonal matrix having the eigenvalues of $\mathbf{M}_m + \Delta t \mathbf{A}_m$ as entries. These are known and given e.g. in [59, pp. 514-516]:

$$\begin{aligned} v_{ij} &= \frac{1}{\sqrt{\sum_{k=1}^N \sin^2 \left(\frac{k\pi}{N+1} \right)}} \sin \left(\frac{ij\pi}{N+1} \right), \\ \lambda_j^m &= \frac{1}{3\Delta x^2} \left(2\alpha_m \Delta x^2 + 6\lambda_m \Delta t + (\alpha_m \Delta x^2 - 6\lambda_m \Delta t) \cos \left(\frac{j\pi}{N+1} \right) \right), \end{aligned} \tag{4.5}$$

for $i, j = 1, \dots, N$ and $m = 1, 2$.

The entries α_{NN}^1 and α_{11}^2 of the matrices $(\mathbf{M}_1 + \Delta t \mathbf{A}_1)^{-1}$ and $(\mathbf{M}_2 + \Delta t \mathbf{A}_2)^{-1}$, respectively, are now computed through their eigendecomposition resulting in

$$\alpha_{NN}^1 = \frac{\sum_{i=1}^N \frac{1}{\lambda_i^1} \sin^2 \left(\frac{i\pi N}{N+1} \right)}{\sum_{i=1}^N \sin^2 \left(\frac{i\pi}{N+1} \right)} = \frac{s_1}{\sum_{i=1}^N \sin^2(i\pi \Delta x)}, \tag{4.6}$$

$$\alpha_{11}^2 = \frac{\sum_{i=1}^N \frac{1}{\lambda_i^2} \sin^2 \left(\frac{i\pi}{N+1} \right)}{\sum_{i=1}^N \sin^2 \left(\frac{i\pi}{N+1} \right)} = \frac{s_2}{\sum_{i=1}^N \sin^2(i\pi \Delta x)}, \tag{4.7}$$

with

$$s_m = \sum_{i=1}^N \frac{3\Delta x^2 \sin^2(i\pi\Delta x)}{2\alpha_m\Delta x^2 + 6\lambda_m\Delta t + (\alpha_m\Delta x^2 - 6\lambda_m\Delta t) \cos(i\pi\Delta x)}, \quad (4.8)$$

for $m = 1, 2$.

To simplify this, the finite sum $\sum_{i=1}^N \sin^2(i\pi\Delta x)$ can be computed. We first rewrite the sum of squared sinus terms into a sum of cosinus terms using the identity $\sin^2(x/2) = (1 - \cos(x))/2$. Then, the resulting sum can be converted into a geometric sum using Euler's formula:

$$\begin{aligned} \sum_{j=1}^N \sin^2(j\pi\Delta x) &= \frac{1 - \Delta x}{2\Delta x} - \frac{1}{2} \sum_{j=1}^N \cos(2j\pi\Delta x) \\ &= \frac{1 - \Delta x}{2\Delta x} - \frac{1}{2} \operatorname{Re} \left(\sum_{j=1}^N e^{2ij\pi\Delta x} \right) \\ &= \frac{1 - \Delta x}{2\Delta x} - \frac{1}{2} \operatorname{Re} \left(\frac{e^{2i\pi\Delta x} (1 - e^{2iN\pi\Delta x})}{1 - e^{2i\pi\Delta x}} \right) = \frac{1}{2\Delta x}. \end{aligned} \quad (4.9)$$

Now, inserting (4.9) into (4.6) and (4.7), and those into (4.2) and (4.3) we get for $\mathbf{S}^{(1)}$ and $\mathbf{S}^{(2)}$,

$$\mathbf{S}^{(1)} = \left(\frac{\alpha_1\Delta x^2 + 3\lambda_1\Delta t}{3\Delta x^2} \right) - \frac{(\alpha_1\Delta x^2 - 6\lambda_1\Delta t)^2 s_1}{18\Delta x^3}, \quad (4.10)$$

$$\mathbf{S}^{(2)} = \left(\frac{\alpha_2\Delta x^2 + 3\lambda_2\Delta t}{3\Delta x^2} \right) - \frac{(\alpha_2\Delta x^2 - 6\lambda_2\Delta t)^2 s_2}{18\Delta x^3}. \quad (4.11)$$

With this we obtain an explicit formula for the spectral radius of the iteration matrix Σ as a function of Δx and Δt :

$$\rho(\Sigma) = |\Sigma| = |\mathbf{S}^{(2)-1} \mathbf{S}^{(1)}| = \frac{6\Delta x(\alpha_1\Delta x^2 + 3\lambda_1\Delta t) - (\alpha_1\Delta x^2 - 6\lambda_1\Delta t)^2 s_1}{6\Delta x(\alpha_2\Delta x^2 + 3\lambda_2\Delta t) - (\alpha_2\Delta x^2 - 6\lambda_2\Delta t)^2 s_2}. \quad (4.12)$$

We could not find a way of simplifying the finite sum (4.8) because Δx depends on N (i.e., $\Delta x = 1/(N + 1)$). However, (4.12) is a computable formula that gives exactly

the convergence rates of the Dirichlet-Neumann iteration for given Δx , Δt , α_m and λ_m , $m = 1, 2$.

We are now interested in the asymptotics of (4.12). In particular, we want to know the behavior of (4.12) when Δt or Δx tend to 0, so that we can relate this to the results of the semidiscrete analysis in section 3.5. However, the denominator of (4.8) becomes zero when Δx tends to 0. To solve this problem, we reformulate (4.12) in terms of $c := \Delta t / \Delta x^2$. Figure 4.1 illustrates the relation between computing $c \rightarrow 0$ or $\Delta t \rightarrow 0$ and $c \rightarrow \infty$ or $\Delta x \rightarrow 0$.

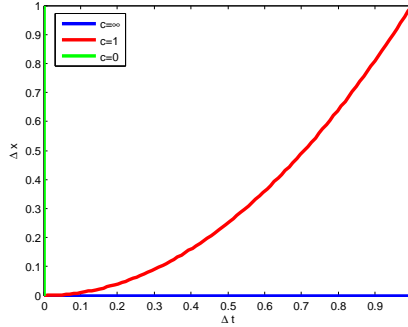


Figure 4.1: Relation between computing $c \rightarrow 0$ or $\Delta t \rightarrow 0$ and $c \rightarrow \infty$ or $\Delta x \rightarrow 0$.

From figure 4.1 one can observe that for a fixed Δx , if we choose a Δt that enforces the condition $\Delta t \ll \Delta x^2$ (i.e. $c \rightarrow 0$), we approach the green line. Similarly, for a fixed Δt , if we choose a Δx that enforces the condition $\Delta t \gg \Delta x^2$ (i.e. $c \rightarrow \infty$), we approach the blue line.

Thus, we multiply (4.12) by Δx^2 both in the numerator and denominator and get

$$|\Sigma| = \frac{6\Delta x(\alpha_1 + 3\lambda_1 c) - (\alpha_1 - 6\lambda_1 c)^2 s'_1}{6\Delta x(\alpha_2 + 3\lambda_2 c) - (\alpha_2 - 6\lambda_2 c)^2 s'_2} \quad (4.13)$$

where

$$s'_m = \sum_{i=1}^N \frac{3 \sin^2(i\pi\Delta x)}{2\alpha_m + 6\lambda_m c + (\alpha_m - 6\lambda_m c) \cos(i\pi\Delta x)} \quad (4.14)$$

for $m = 1, 2$.

Finally, computing the limits of (4.13) when $c \rightarrow 0$ and $c \rightarrow \infty$, we get

$$\begin{aligned}
\lim_{c \rightarrow 0} |\Sigma| &= \lim_{c \rightarrow 0} \frac{6\Delta x \alpha_1 - \alpha_1^2 \sum_{i=1}^N \frac{3 \sin^2(i\pi \Delta x)}{\alpha_1(2+\cos(i\pi \Delta x))}}{6\Delta x \alpha_2 - \alpha_2^2 \sum_{i=1}^N \frac{3 \sin^2(i\pi \Delta x)}{\alpha_2(2+\cos(i\pi \Delta x))}} \\
&= \lim_{c \rightarrow 0} \frac{\alpha_1 \left(6\Delta x - \sum_{i=1}^N \frac{3 \sin^2(i\pi \Delta x)}{2+\cos(i\pi \Delta x)} \right)}{\alpha_2 \left(6\Delta x - \sum_{i=1}^N \frac{3 \sin^2(i\pi \Delta x)}{2+\cos(i\pi \Delta x)} \right)} = \frac{\alpha_1}{\alpha_2} =: \gamma,
\end{aligned} \tag{4.15}$$

$$\begin{aligned}
\lim_{c \rightarrow \infty} |\Sigma| &= \lim_{c \rightarrow \infty} \frac{6\Delta x 3\lambda_1 c - (6\lambda_1 c)^2 \sum_{i=1}^N \frac{3 \sin^2(i\pi \Delta x)}{6\lambda_1 c(1-\cos(i\pi \Delta x))}}{6\Delta x 3\lambda_2 c - (6\lambda_2 c)^2 \sum_{i=1}^N \frac{3 \sin^2(i\pi \Delta x)}{6\lambda_2 c(1-\cos(i\pi \Delta x))}} \\
&= \lim_{c \rightarrow \infty} \frac{\lambda_1 \left(6\Delta x - 6 \sum_{i=1}^N \frac{3 \sin^2(i\pi \Delta x)}{1-\cos(i\pi \Delta x)} \right)}{\lambda_2 \left(6\Delta x - 6 \sum_{i=1}^N \frac{3 \sin^2(i\pi \Delta x)}{1-\cos(i\pi \Delta x)} \right)} = \frac{\lambda_1}{\lambda_2} =: \delta.
\end{aligned} \tag{4.16}$$

From the results obtained in (4.15) and (4.16) we can observe that strong jumps in the physical properties of the materials placed in Ω_1 and Ω_2 will imply fast convergence. This is the case when modelling thermal fluid-structure interaction, where often a fluid with low thermal conductivity and density is coupled with a structure having higher thermal conductivity and density.

On the other hand, in the domain decomposition context, the coupling iteration will be slow because the coefficients α_m and λ_m , $m = 1, 2$ are typically continuous across subdomains.

Finally, note that the space continuous analysis presented in section 3.5 coincides when $c \rightarrow \infty$. In that case, the semidiscrete result (3.25) agrees with the result obtained in (4.16). However, the semidiscrete analysis was performed in 2D and the discrete analysis in 1D.

4.2 Discrete analysis in 2D

In this section we extend the analysis presented in the previous section to the 2D case. The subdomains are here $\Omega_1 = [-1, 0] \times [0, 1]$ and $\Omega_2 = [0, 1] \times [0, 1]$. An equidistant grid is chosen i.e, $\Delta x = \Delta y = 1/(N+1)$. For the FEM discretization, we use triangular elements distributed as sketched in figure 4.2 and the following pyramidal test functions

$$\phi_k(x, y) = \begin{cases} \frac{x+y}{\Delta x} - 1, & \text{if } \mathbf{x} = (x, y) \in \text{Region 1,} \\ \frac{y}{\Delta x}, & \text{if } \mathbf{x} \in \text{Region 2,} \\ \frac{\Delta x - x}{\Delta x}, & \text{if } \mathbf{x} \in \text{Region 3,} \\ 1 - \frac{x+y}{\Delta x}, & \text{if } \mathbf{x} \in \text{Region 4,} \\ \frac{\Delta x - y}{\Delta x}, & \text{if } \mathbf{x} \in \text{Region 5,} \\ \frac{x}{\Delta x}, & \text{if } \mathbf{x} \in \text{Region 6,} \\ 0, & \text{otherwise.} \end{cases} \quad (4.17)$$

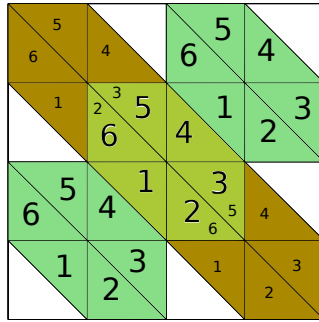


Figure 4.2: Sketch of the regions for the pyramidal test functions defined in (4.17).

The discretization matrices are given in this case by

$$\mathbf{A}_m = \frac{\lambda_m}{\Delta x^2} \begin{pmatrix} \mathbf{B} & -\mathbf{I} & \mathbf{0} \\ -\mathbf{I} & \mathbf{B} & \ddots \\ & \ddots & \ddots & -\mathbf{I} \\ \mathbf{0} & & -\mathbf{I} & \mathbf{B} \end{pmatrix} \quad \text{where } \mathbf{B} = \begin{pmatrix} 4 & -1 & & 0 \\ -1 & 4 & \ddots & \\ & \ddots & \ddots & -1 \\ 0 & & -1 & 4 \end{pmatrix}$$

and $\mathbf{I} \in \mathbb{R}^{N \times N}$ is an identity matrix. Note that each block of the matrices $\mathbf{A}_m \in \mathbb{R}^{N^2 \times N^2}$ has size $N \times N$.

$$\mathbf{M}_m = \alpha_m \begin{pmatrix} \mathbf{N} & \mathbf{N}_2 & \mathbf{0} \\ \mathbf{N}_1 & \mathbf{N} & \ddots \\ \mathbf{0} & & \mathbf{N}_1 & \mathbf{N} \end{pmatrix}, \quad \mathbf{N} = \begin{pmatrix} 5/6 & -1/12 & & 0 \\ -1/12 & 5/6 & \ddots & \\ & \ddots & \ddots & -1/12 \\ 0 & & -1/12 & 5/6 \end{pmatrix},$$

$$\mathbf{N}_1 = \begin{pmatrix} -1/12 & 0 & & 0 \\ 1/4 & -1/12 & \ddots & \\ & \ddots & \ddots & 0 \\ 0 & & 1/4 & -1/12 \end{pmatrix}, \quad \mathbf{N}_2 = \begin{pmatrix} -1/12 & 1/4 & & 0 \\ 0 & -1/12 & \ddots & \\ & \ddots & \ddots & 1/4 \\ 0 & & 0 & -1/12 \end{pmatrix}.$$

Each block of the matrices $\mathbf{M}_m \in \mathbb{R}^{N^2 \times N^2}$ has size $N \times N$ as well. As before, we consider here $\mathbf{E}_j = (\mathbf{0} \ \cdots \ \mathbf{0} \ \mathbf{I} \ \mathbf{0} \ \cdots \ \mathbf{0})^T \in \mathbb{R}^{N^2 \times N}$ where the only nonzero block is the j -th block of size $N \times N$. Thus,

$$\mathbf{A}_{\Gamma I}^{(1)} = -\frac{\lambda_1}{\Delta x^2} \mathbf{E}_N^T, \quad \mathbf{A}_{\Gamma I}^{(2)} = -\frac{\lambda_2}{\Delta x^2} \mathbf{E}_1^T, \quad \mathbf{A}_{I\Gamma}^{(1)} = -\frac{\lambda_1}{\Delta x^2} \mathbf{E}_N, \quad \mathbf{A}_{I\Gamma}^{(2)} = -\frac{\lambda_2}{\Delta x^2} \mathbf{E}_1,$$

$$\mathbf{M}_{\Gamma I}^{(1)} = \alpha_1 \mathbf{E}_N^T \mathbf{N}_1, \quad \mathbf{M}_{\Gamma I}^{(2)} = \alpha_2 \mathbf{E}_1^T \mathbf{N}_2, \quad \mathbf{M}_{I\Gamma}^{(1)} = \alpha_1 \mathbf{E}_N \mathbf{N}_1, \quad \mathbf{M}_{I\Gamma}^{(2)} = \alpha_2 \mathbf{E}_1 \mathbf{N}_2,$$

$$\mathbf{M}_{\Gamma\Gamma}^{(m)} = \alpha_m \begin{pmatrix} 5/12 & -1/24 & & 0 \\ -1/24 & 5/12 & \ddots & \\ & \ddots & \ddots & -1/24 \\ 0 & & -1/24 & 5/12 \end{pmatrix},$$

$$\mathbf{A}_{\Gamma\Gamma}^{(m)} = \frac{\lambda_m}{\Delta x^2} \begin{pmatrix} 2 & -1/2 & & 0 \\ -1/2 & 2 & \ddots & \\ & \ddots & \ddots & -1/2 \\ 0 & & -1/2 & 2 \end{pmatrix},$$

where $\mathbf{M}_{\Gamma\Gamma}^{(m)}, \mathbf{A}_{\Gamma\Gamma}^{(m)} \in \mathbb{R}^{N \times N}$ for $m = 1, 2$.

In the two-dimensional case, the iteration matrix Σ is a matrix of size $N \times N$. As in the 1D case, one computes $\mathbf{S}^{(1)}$ and $\mathbf{S}^{(2)}$ by inserting the corresponding matrices specified above in (3.22) obtaining

$$\begin{aligned}
\mathbf{S}^{(1)} &= \mathbf{M}_{\Gamma\Gamma}^{(1)} + \Delta t \mathbf{A}_{\Gamma\Gamma}^{(1)} - (\mathbf{M}_{\Gamma I}^{(1)} + \Delta t \mathbf{A}_{\Gamma I}^{(1)}) (\mathbf{M}_1 + \Delta t \mathbf{A}_1)^{-1} (\mathbf{M}_{\Gamma I}^{(1)} + \Delta t \mathbf{A}_{\Gamma I}^{(1)}) \\
&= \left(\alpha_1 \text{tridiag} \left(-\frac{1}{24}, \frac{5}{12}, -\frac{1}{24} \right) + \frac{\lambda_1 \Delta t}{\Delta x^2} \text{tridiag} \left(-\frac{1}{2}, 2, -\frac{1}{2} \right) \right) \\
&\quad - \left(\alpha_1 \mathbf{N}_1 - \frac{\lambda_1 \Delta t}{\Delta x^2} \mathbf{I} \right) \mathbf{E}_N^T (\mathbf{M}_1 + \Delta t \mathbf{A}_1)^{-1} \mathbf{E}_N \left(\alpha_1 \mathbf{N}_1 - \frac{\lambda_1 \Delta t}{\Delta x^2} \mathbf{I} \right)
\end{aligned} \tag{4.18}$$

$$\begin{aligned}
\mathbf{S}^{(2)} &= \mathbf{M}_{\Gamma\Gamma}^{(2)} + \Delta t \mathbf{A}_{\Gamma\Gamma}^{(2)} - (\mathbf{M}_{\Gamma I}^{(2)} + \Delta t \mathbf{A}_{\Gamma I}^{(2)}) (\mathbf{M}_2 + \Delta t \mathbf{A}_2)^{-1} (\mathbf{M}_{\Gamma I}^{(2)} + \Delta t \mathbf{A}_{\Gamma I}^{(2)}) \\
&= \left(\alpha_2 \text{tridiag} \left(-\frac{1}{24}, \frac{5}{12}, -\frac{1}{24} \right) + \frac{\lambda_2 \Delta t}{\Delta x^2} \text{tridiag} \left(-\frac{1}{2}, 2, -\frac{1}{2} \right) \right) \\
&\quad - \left(\alpha_2 \mathbf{N}_2 - \frac{\lambda_2 \Delta t}{\Delta x^2} \mathbf{I} \right) \mathbf{E}_1^T (\mathbf{M}_2 + \Delta t \mathbf{A}_2)^{-1} \mathbf{E}_1 \left(\alpha_2 \mathbf{N}_2 - \frac{\lambda_2 \Delta t}{\Delta x^2} \mathbf{I} \right)
\end{aligned} \tag{4.19}$$

In this case the iteration matrix Σ is not easy to compute for several reasons. First of all, the matrices $\mathbf{M}_1 + \Delta t \mathbf{A}_1$ and $\mathbf{M}_2 + \Delta t \mathbf{A}_2$ are sparse block tridiagonal matrices, and consequently, their inverses are not straight forward to compute. A block-by-block algorithm for inverting a block tridiagonal matrix is explained in [70]. However, the algorithm is based on the iterative application of the Schur complement [99], and it results in a sequence of block matrices and inverses of block matrices that we did not find possible to compute exactly. Moreover, the diagonal blocks of $\mathbf{M}_1 + \Delta t \mathbf{A}_1$ and $\mathbf{M}_2 + \Delta t \mathbf{A}_2$ are tridiagonal but their inverses are full matrices [30].

Due to these difficulties, we propose here to approximate Σ . One can observe that $\mathbf{M}_1 + \Delta t \mathbf{A}_1$ and $\mathbf{M}_2 + \Delta t \mathbf{A}_2$ are strictly diagonally dominant matrices, and therefore, we propose to approximate them by their block diagonal. The same reasoning is used to approximate the diagonal block matrices of $\mathbf{M}_1 + \Delta t \mathbf{A}_1$ and $\mathbf{M}_2 + \Delta t \mathbf{A}_2$ by their diagonal. Thus,

$$\begin{aligned}
\mathbf{S}^{(1)} &\approx \left(\frac{5\alpha_1}{12} + \frac{2\lambda_1 \Delta t}{\Delta x^2} \right) \mathbf{I} - \left(\frac{\alpha_1}{12} + \frac{\lambda_1 \Delta t}{\Delta x^2} \right)^2 \left(\frac{5\alpha_1}{6} + \frac{4\lambda_1 \Delta t}{\Delta x^2} \right)^{-1} \mathbf{I} \\
&= \left(\frac{2(5\alpha_1 \Delta x^2 + 24\lambda_1 \Delta t)^2 - (\alpha_1 \Delta x^2 + 12\lambda_1 \Delta t)^2}{24\Delta x^2(5\alpha_1 \Delta x^2 + 24\lambda_1 \Delta t)} \right) \mathbf{I},
\end{aligned} \tag{4.20}$$

$$\begin{aligned}
\mathbf{S}^{(2)} &\approx \left(\frac{5\alpha_2}{12} + \frac{2\lambda_2 \Delta t}{\Delta x^2} \right) \mathbf{I} - \left(\frac{\alpha_2}{12} + \frac{\lambda_2 \Delta t}{\Delta x^2} \right)^2 \left(\frac{5\alpha_2}{6} + \frac{4\lambda_2 \Delta t}{\Delta x^2} \right)^{-1} \mathbf{I} \\
&= \left(\frac{2(5\alpha_2 \Delta x^2 + 24\lambda_2 \Delta t)^2 - (\alpha_2 \Delta x^2 + 12\lambda_2 \Delta t)^2}{24\Delta x^2(5\alpha_2 \Delta x^2 + 24\lambda_2 \Delta t)} \right) \mathbf{I}.
\end{aligned} \tag{4.21}$$

Thus, we obtain an estimate of the spectral radius of the iteration matrix Σ :

$$\rho(\Sigma) \approx \frac{(5\alpha_2\Delta x^2 + 24\lambda_2\Delta t)(2(5\alpha_1\Delta x^2 + 24\lambda_1\Delta t)^2 - (\alpha_1\Delta x^2 + 12\lambda_1\Delta t)^2)}{(5\alpha_1\Delta x^2 + 24\lambda_1\Delta t)(2(5\alpha_2\Delta x^2 + 24\lambda_2\Delta t)^2 - (\alpha_2\Delta x^2 + 12\lambda_2\Delta t)^2)} =: \sigma. \quad (4.22)$$

Furthermore, computing the limits of (4.22) when $\Delta t \rightarrow 0$ and $\Delta x \rightarrow 0$ we get

$$\lim_{\Delta t \rightarrow 0} \rho(\Sigma) \approx \lim_{\Delta t \rightarrow 0} \frac{5\alpha_2\Delta x^2(2(5\alpha_1\Delta x^2)^2 - (\alpha_1\Delta x^2)^2)}{5\alpha_1\Delta x^2(2(5\alpha_2\Delta x^2)^2 - (\alpha_2\Delta x^2)^2)} = \frac{\alpha_1}{\alpha_2} =: \gamma, \quad (4.23)$$

$$\lim_{\Delta x \rightarrow 0} \rho(\Sigma) \approx \lim_{\Delta x \rightarrow 0} \frac{24\lambda_2\Delta t(2 \cdot 24^2\lambda_1^2\Delta t^2 - 12^2\lambda_1^2\Delta t^2)}{24\lambda_1\Delta t(2 \cdot 24^2\lambda_2^2\Delta t^2 - 12^2\lambda_2^2\Delta t^2)} = \frac{\lambda_1}{\lambda_2} =: \delta. \quad (4.24)$$

Thus, the asymptotic behavior of the convergence rates for the 2D case is consistent with the 1D case.

As already observed in the 1D case, when $c \rightarrow \infty$ (3.25) matches with the asymptotic computed in (4.24).

4.3 Numerical results

In this section we present numerical experiments that show how the theoretical formula $|\Sigma|$ in (4.12) predicts the convergence rates in the 1D case and the validity of the approximation of $\rho(\Sigma)$ in (4.22) as an estimator for the rates in the 2D case of the coupled problem formulated above. We also show that the theoretical asymptotics deduced in (4.15), (4.16), (4.23) and (4.24) match with the numerical results.

We first compare the semidiscrete estimator β in (3.23) with the discrete formula $|\Sigma|$ in 1D in (4.12) and experimental convergence rates. The latter are obtained from implementing the Dirichlet-Neumann method (3.17)-(3.18). The results are then compared to a reference solution u_{ref} over the whole domain Ω , obtained by choosing a tolerance of $1e - 10$ as a termination criterion. More specifically, the experimental convergence rates C_R are computed using the formula

$$C_R = \frac{\|u_3 - u_{ref}\|_2}{\|u_2 - u_{ref}\|_2}, \quad (4.25)$$

where u_2 and u_3 are the second and third iterates of the Dirichlet-Neumann iteration.

Figure 4.3 shows a comparison between β and $|\Sigma|$ for $\Delta x = 1/20$ and $\Delta x = 1/500$ and varying Δt . As can be seen, the experimental convergence rate matches exactly with

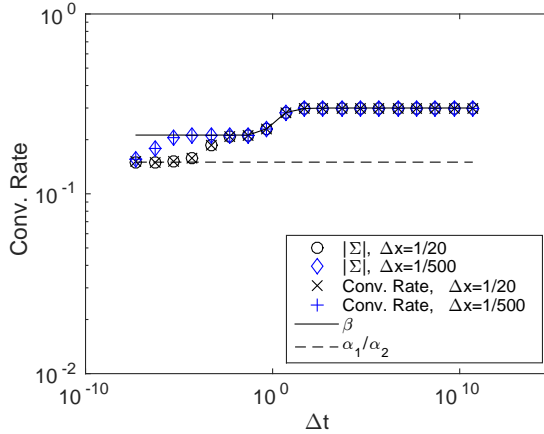


Figure 4.3: Semidiscrete estimator β , exact rate $|\Sigma|$ and numerical rates over Δt in 1D. $D_1 = 1$, $D_2 = 0.5$, $\lambda_1 = 0.3$, $\lambda_2 = 1$, $\Delta x = 1/20$ or $\Delta x = 1/500$ and $\Delta t = 5e-8, 5e-7, \dots, 5e10$.

the formula (4.12). Observe that β is almost constant, except for a short dynamic transition between $(\lambda_1/\lambda_2)\sqrt{D_2/D_1}$ and λ_1/λ_2 . We can conclude that the formula for the convergence rates in 1D presented in this chapter match the semidiscrete one proposed in [44] when $\Delta t/\Delta x^2 \gg 1$. In the, less relevant case, $\Delta t/\Delta x^2 \ll 1$ our formula also predicts the rates accurately, while the semidiscrete estimator deviates according to (3.24).

The difference to the semidiscrete analysis in [44] stems from different limits taking place. The semidiscrete analysis implicitly assumes that first a limit Δx to zero has taken place for Δt fixed. Thus, a limit first Δt , then Δx to zero is not addressed by it. This can be seen in figure 4.3 in the following way: for fixed Δx , letting Δt become very small causes the convergence rate to move into the direction predicted by the fully discrete analysis. However, then keeping this very small Δt fixed and decreasing Δx moves that rate back in the vicinity of β .

We now want to illustrate how $|\Sigma|$ in (4.12) and σ in (4.22) predict the convergence rates and tend to the limits computed previously in (4.15)-(4.16) and (4.23)-(4.24) respectively. As test cases we consider here the thermal interaction between air at $273K$ with steel at $900K$, water at $283K$ with steel at $900K$ and air at $273K$ with water at $283K$. Physical properties of the materials and resulting asymptotics for these three cases are shown in table 4.1 and 4.2 respectively.

Figures 4.4, 4.5 and 4.6 show the convergence rates of the three cases specified in table 4.2 for 1D and 2D. The circles correspond to our estimates ($|\Sigma|$ in 1D and σ in 2D), the crosses to the experimental convergence rates, the continuous line to the spatial limit δ and the dashed line to the temporal limit γ . In 4.4, 4.5 and 4.6 we have

Table 4.1: Physical properties of the materials. λ is the thermal conductivity, ρ the density, c_p the specific heat capacity and $\alpha = \rho c_p$.

Material	λ (W/mK)	ρ (kg/m ³)	c_p (J/kgK)	α (J/K m ³)
Air	0.0243	1.293	1005	1299.5
Water	0.58	999.7	4192.1	4.1908e6
Steel	48.9	7836	443	3471348

Table 4.2: The limits of the convergence rates when $\Delta t \rightarrow 0$ (γ) and $\Delta x \rightarrow 0$ (δ).

Case	γ	δ
Air-Steel	3.7434e-4	4.9693e-4
Water-Steel	1.2073	0.0119
Air-Water	3.1008e-4	0.0419

chosen a fixed $\Delta x = 1/20$ and we have varied $\Delta t = 5e - 4, 5e - 3, \dots, 5e10$ obtaining $c = 5e - 4/2.5e - 3, 5e - 3/2.5e - 3, \dots, 5e10/2.5e - 3$.

Note that both the 1D and 2D cases have a similar behavior. However, the convergence rates are computed exactly in the 1D case and estimated in the 2D case. We observe from the 2D plots that the approximation predicts the convergence rates quite well because the difference with respect to the experimental rates is really small. Furthermore, one can observe that the convergence rates tend to δ when $c \rightarrow \infty$ and to γ when $c \rightarrow 0$ both in 1D and 2D as computed previously in (4.15)-(4.16) and (4.23)-(4.24).

From figures 4.4 and 4.5 we can observe that the convergence rates are really fast (factor of $\sim 1e - 4$) when there exist strong jumps in the coefficient of the materials. For instance, when performing the thermal coupling between air and steel the Dirichlet-Neumann iteration only needs two iterations to achieve a tolerance of $1e - 10$.

Finally, the thermal coupling between water and steel illustrated in 4.6 shows us the importance of choosing carefully the spatial resolution (Δx) and the time step (Δt) when one of the asymptotics predicts divergence and the other convergence. Here, for the water-steel coupling we have $\gamma > 1$ and $\delta < 1$ (check table 4.2). Thus, if we choose a pair $\Delta t, \Delta x$ such that $\Delta t/\Delta x^2 \sim 0$, the rate will be $\gamma = 1.2073$ and the numerical method will be divergent. However, if we choose a pair $\Delta t, \Delta x$ such that $\Delta t/\Delta x^2 \sim \infty$, the rate will be $\delta = 0.0119$ and the numerical method will converge fast.

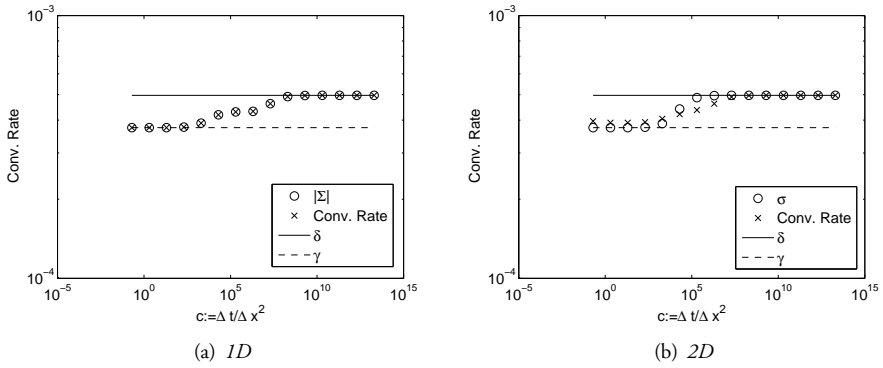


Figure 4.4: Air-steel thermal interaction convergence rates with respect to $c := \Delta t / \Delta x^2$. Here $c = 5e - 4 / 2.5e - 3, 5e - 3 / 2.5e - 3, \dots, 5e10 / 2.5e - 3$.

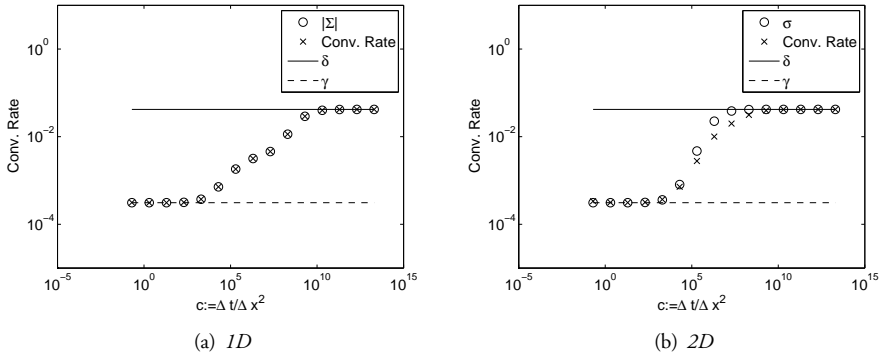


Figure 4.5: Air-water thermal interaction convergence rates with respect to $c := \Delta t / \Delta x^2$. Here $c = 5e - 4 / 2.5e - 3, 5e - 3 / 2.5e - 3, \dots, 5e10 / 2.5e - 3$.

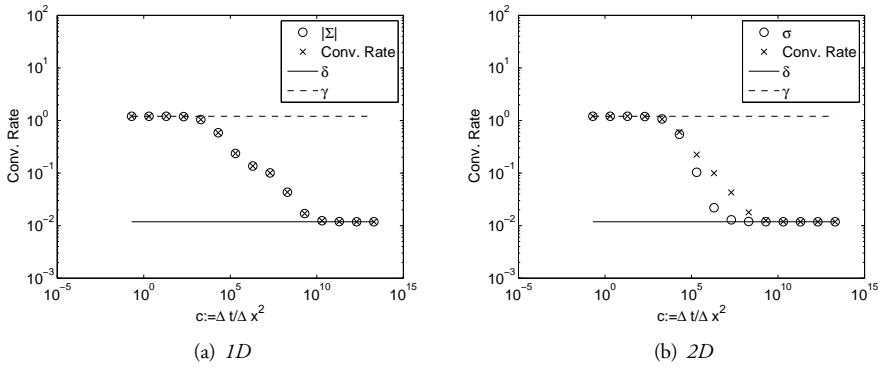


Figure 4.6: Water-steel thermal interaction convergence rates with respect to $c := \Delta t / \Delta x^2$. Here $c = 5e - 4 / 2.5e - 3, 5e - 3 / 2.5e - 3, \dots, 5e10 / 2.5e - 3$.

Chapter 5

Convergence analysis for mixed discretizations

As mentioned before, the finite volume method (FVM) is generally a better choice than the finite element method (FEM) for the discretization of fluids. Consequently, the model problem (3.4) will become a more realistic model for thermal fluid-structure interaction if we use finite volumes on one subdomain and finite elements on the other.

For this reason, we present in this chapter a modified 1D analysis of the model problem in (3.4). The novelty is that here we consider two different space discretizations on the subdomains. An FVM is used on Ω_1 and an FEM is used on Ω_2 . In particular, we propose two alternative FVM discretizations for the approximation of the normal derivatives. A first order and a second order approximation are presented and compared. Finally, a summary of the extension to 2D and the generalization to non equidistant grids included in papers I and II respectively is presented in the last section of this chapter. A previous work in mixed discretizations where we used finite differences on one domain and finite elements on the other can be found in [60, pp. 1530-1544].

5.1 Two alternative FVM formulations

We will focus on the FVM formulation on Ω_1 of problem (3.4) because the FEM formulation on Ω_2 was already presented in chapter 3. For this, we consider a primal grid, i.e, we discretize Ω_1 into $N + 1$ equal sized grid cells of size $\Delta x = 1/(N + 1)$, and define $x_i = i\Delta x$, so that x_i is the center of the cell i , see figure 5.1. The edges of cell i are then $x_{i-1/2}$ and $x_{i+1/2}$ and they form the corresponding dual grid.

Integrating the first equation for $m = 1$ in (3.4) over cell i we get

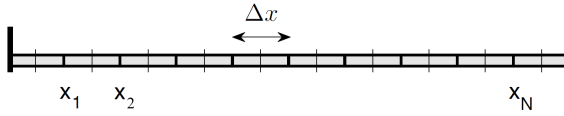


Figure 5.1: Grid cells over Ω_1 for the finite volume discretization.

$$\begin{aligned} \alpha_1 \frac{d}{dt} \int_{x_{i-1/2}}^{x_{i+1/2}} u_1(x, t) dx &= \lambda_1 \int_{x_{i-1/2}}^{x_{i+1/2}} \Delta u_1(x, t) dx \\ &= \lambda_1 (u_{1,x}(x_{i+1/2}, t) - u_{1,x}(x_{i-1/2}, t)). \end{aligned} \tag{5.1}$$

Thus, introducing the cell average

$$u_{1,i}(t) = \frac{1}{\Delta x} \int_{x_{i-1/2}}^{x_{i+1/2}} u_1(x, t) dx,$$

into (5.1) with the flux

$$F_i(t) = -\lambda_1 u_{1,i_x}(t),$$

we get the formula

$$\alpha_1 \Delta x \frac{du_{1,i}(t)}{dt} = -(F_{i+1/2}(t) - F_{i-1/2}(t)).$$

Finally, dividing the previous expression by Δx and approximating the flux by

$$F_{i-1/2}(t) = -\frac{\lambda_1}{\Delta x} (u_{1,i}(t) - u_{1,i-1}(t)) + O(\Delta x^2),$$

we get the second order update formula

$$\begin{aligned} \alpha_1 \frac{du_{1,1}(t)}{dt} - \frac{\lambda_1}{\Delta x^2} (u_{1,2}(t) - 2u_{1,1}(t)) &= 0, \quad i = 1, \\ \alpha_1 \frac{du_{1,i}(t)}{dt} - \frac{\lambda_1}{\Delta x^2} (u_{1,i+1}(t) - 2u_{1,i}(t) + u_{1,i-1}(t)) &= 0, \quad i = 2..N - 1, \\ \alpha_1 \frac{du_{1,N}(t)}{dt} - \frac{\lambda_1}{\Delta x^2} (u_{\Gamma} - 2u_{1,N}(t) + u_{1,N-1}(t)) &= 0, \quad i = N. \end{aligned} \tag{5.2}$$

Now, letting $\mathbf{u}_I^{(1)} = (u_{1,1}(t), u_{1,2}(t), \dots, u_{1,N}(t))^T$ correspond to the unknowns on Ω_1 and \mathbf{u}_Γ to the unknown at the interface Γ , we can rewrite (5.2) in a compact form:

$$\alpha_1 \dot{\mathbf{u}}_I^{(1)} - \mathbf{A}_1 \mathbf{u}_I^{(1)} - \mathbf{A}_{I\Gamma}^{(1)} \mathbf{u}_\Gamma = \mathbf{0}. \quad (5.3)$$

where \mathbf{A}_1 corresponds to the discretization of the Laplacian operator on Ω_1 and the prescribed Dirichlet boundary condition at the interface is inserted in the equation by the matrix $\mathbf{A}_{I\Gamma}^{(1)}$. These matrices are given by

$$\mathbf{A}_1 = \frac{\lambda_1}{\Delta x^2} \begin{pmatrix} -2 & 1 & & 0 \\ 1 & -2 & \ddots & \\ & \ddots & \ddots & 1 \\ 0 & & 1 & -2 \end{pmatrix}, \quad \mathbf{A}_{I\Gamma}^{(1)} = \frac{\lambda_1}{\Delta x^2} \mathbf{e}_N.$$

On the other hand, the discretization of the problem on Ω_2 using finite elements was already presented in chapter 3 resulting in equation (3.10). However, the system (5.3)-(3.10) is not enough to describe (3.4). Extending the approach (5.3)-(3.10) for the unsteady transmission problem, we will look for an approximation of the normal derivatives at Γ .

We have decided to use two possibilities to approximate the normal derivative with respect to u_1 for comparison reasons. Those are the first order forward finite differences and the second order one-sided finite differences.

The first choice will be

$$-\lambda_1 \frac{\partial u_1}{\partial \mathbf{n}_1} \approx \frac{\lambda_1}{\Delta x} (u_{1,N}(t) - u_\Gamma), \quad (5.4)$$

and the second one will be

$$-\lambda_1 \frac{\partial u_1}{\partial \mathbf{n}_1} \approx \frac{\lambda_1}{2\Delta x} (4u_{1,N}(t) - u_{1,N-1}(t) - 3u_\Gamma). \quad (5.5)$$

These two possibilities lead to the following compact expressions:

$$-\mu_{FVM}^{1st} = \mathbf{A}_{\Gamma I}^{(1),1st} \mathbf{u}_I^{(1)} - \mathbf{A}_{\Gamma\Gamma}^{(1),1st} \mathbf{u}_\Gamma. \quad (5.6)$$

$$-\mu_{FVM}^{2nd} = \mathbf{A}_{\Gamma I}^{(1),2nd} \mathbf{u}_I^{(1)} - \mathbf{A}_{\Gamma\Gamma}^{(1),2nd} \mathbf{u}_\Gamma. \quad (5.7)$$

Here, the matrices $\mathbf{A}_{\Gamma I}^{(1),1st}$, $\mathbf{A}_{\Gamma I}^{(1),2nd}$, $\mathbf{A}_{\Gamma\Gamma}^{(1),1st}$ and $\mathbf{A}_{\Gamma\Gamma}^{(1),2nd}$ are given by

$$\mathbf{A}_{\Gamma I}^{(1),1st} = \frac{\lambda_1}{\Delta x^2} \mathbf{e}_N^T, \quad \mathbf{A}_{\Gamma I}^{(1),2nd} = \frac{\lambda_1}{2\Delta x^2} (4\mathbf{e}_N^T - \mathbf{e}_{N-1}^T),$$

$$\mathbf{A}_{\Gamma\Gamma}^{(1),1st} = \frac{\lambda_1}{\Delta x^2}, \quad \mathbf{A}_{\Gamma\Gamma}^{(1),2nd} = \frac{3\lambda_1}{2\Delta x^2}.$$

On the other hand, the normal derivative with respect to u_2 is approximated as explained in (3.13) resulting in the following expression:

$$\mu_{FEM} = \mathbf{M}_{\Gamma\Gamma}^{(2)} \dot{\mathbf{u}}_{\Gamma} + \mathbf{M}_{\Gamma I}^{(2)} \dot{\mathbf{u}}_I^{(2)} + \mathbf{A}_{\Gamma\Gamma}^{(2)} \mathbf{u}_{\Gamma} + \mathbf{A}_{\Gamma I}^{(2)} \mathbf{u}_I^{(2)}. \quad (5.8)$$

Consequently, the equation

$$\mu_{FEM} = -\mu_{FVM}^{1st}, \quad \text{or} \quad \mu_{FEM} = -\mu_{FVM}^{2nd}, \quad (5.9)$$

completes the system (5.3)-(3.10). We now reformulate the coupled equations (5.3), (3.10) and (5.9) into an ODE for the vector of unknowns $\mathbf{u} = \left(\mathbf{u}_I^{(1)}, \mathbf{u}_I^{(2)}, \mathbf{u}_{\Gamma} \right)^T$

$$\hat{\mathbf{M}} \dot{\mathbf{u}} + \hat{\mathbf{A}} \mathbf{u} = 0, \quad (5.10)$$

where

$$\hat{\mathbf{M}} = \begin{pmatrix} \alpha_1 \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_2 & \mathbf{M}_{\Gamma I}^{(2)} \\ \mathbf{0} & \mathbf{M}_{\Gamma I}^{(2)} & \mathbf{M}_{\Gamma\Gamma}^{(2)} \end{pmatrix}, \quad \hat{\mathbf{A}} = \begin{pmatrix} -\mathbf{A}_1 & \mathbf{0} & -\mathbf{A}_{\Gamma\Gamma}^{(1)} \\ \mathbf{0} & \mathbf{A}_2 & \mathbf{A}_{\Gamma\Gamma}^{(2)} \\ -\mathbf{A}_{\Gamma I}^{(1),1st/2nd} & \mathbf{A}_{\Gamma I}^{(2)} & \mathbf{A}_{\Gamma\Gamma}^{(1),1st/2nd} + \mathbf{A}_{\Gamma\Gamma}^{(2)} \end{pmatrix}.$$

Although the mixed discretizations FVM-FEM explained in this chapter and the FEM-FEM discretizations explained in chapter 3 are used to solve the same model problem (3.4), we already observe here some differences in their structure. The first observation is that matrices $\hat{\mathbf{A}}$ and $\tilde{\mathbf{A}}$ in (3.15) have the same construction but when using second order one-sided differences, $\mathbf{A}_{\Gamma I}^{(1)} \neq -\mathbf{A}_{\Gamma I}^{(1),2nd}$ and $\mathbf{A}_{\Gamma\Gamma}^{(1)} \neq \mathbf{A}_{\Gamma\Gamma}^{(1),2nd}$. On the other hand, the matrices $\hat{\mathbf{M}}$ and $\tilde{\mathbf{M}}$ in (3.15) do not have the same structure. The tridiagonal matrix \mathbf{M}_1 in $\tilde{\mathbf{M}}$ corresponds to the diagonal matrix $\alpha_1 \mathbf{I}$ in $\hat{\mathbf{M}}$. Moreover, the matrices $\mathbf{M}_{\Gamma I}^{(1)}$ and $\mathbf{M}_{\Gamma I}^{(1)}$ in $\tilde{\mathbf{M}}$ are not existing for the mixed discretizations case. Therefore, we can already intuit here that the convergence analysis of the FVM-FEM discretizations might give different results than the ones presented for FEM-FEM discretizations.

Finally, proceeding as before in chapter 3 we can derive the corresponding iteration matrices:

$$\Sigma_{1st} = -\mathbf{S}^{(2)-1} \mathbf{S}_{1st}^{(1)}, \quad \Sigma_{2nd} = -\mathbf{S}^{(2)-1} \mathbf{S}_{2nd}^{(1)}, \quad (5.11)$$

where

$$\mathbf{S}_{1st}^{(1)} = \Delta t \mathbf{A}_{\Gamma\Gamma}^{(1),1st} - \Delta t^2 \mathbf{A}_{\Gamma I}^{(1),1st} (\alpha_1 \mathbf{I} - \Delta t \mathbf{A}_1)^{-1} \mathbf{A}_{I\Gamma}^{(1)}, \quad (5.12)$$

$$\mathbf{S}_{2nd}^{(1)} = \Delta t \mathbf{A}_{\Gamma\Gamma}^{(1),2nd} - \Delta t^2 \mathbf{A}_{\Gamma I}^{(1),2nd} (\alpha_1 \mathbf{I} - \Delta t \mathbf{A}_1)^{-1} \mathbf{A}_{I\Gamma}^{(1)}, \quad (5.13)$$

and $\mathbf{S}^{(2)}$ is given by (3.22). Here, Σ_{1st} and $\mathbf{S}_{1st}^{(1)}$ correspond to first order forward differences for the normal derivative and Σ_{2nd} and $\mathbf{S}_{2nd}^{(1)}$ to second order one-sided differences.

Note that (5.12) and (5.13) are different than $\mathbf{S}^{(1)}$ in (3.22). In particular, the mass matrices in (3.22) do not appear in $\mathbf{S}_{1st}^{(1)}$ and $\mathbf{S}_{2nd}^{(1)}$.

5.1.1 Numerical order comparison

We now compare the accuracy obtained in the numerical solution of the coupled problem using first order forward differences and second order one-sided differences to approximate the normal derivative with respect to Ω_1 . The order plot in figure 5.2 was produced comparing the numerical solution obtained after the Dirichlet-Neumann iteration has achieved an accuracy below $1e - 8$ on Ω to a reference solution using the root mean square norm. One observes that using forward differences a first order solution is achieved and using one-sided differences we get a second order solution. Note that the order of the global solution is affected by the order of the scheme used at one point of the mesh. In addition, the error of the numerical solution is roughly two orders of magnitude smaller using one-sided differences than using forward differences. Therefore, we strongly recommend to use a second order scheme to approximate the normal derivative.

5.2 Discrete analyses in 1D and 2D

The analyses presented in the previous chapter for FE discretizations in 1D and 2D are the basis for the analyses and results contained in papers I and II for mixed discretizations (meaning the FVM-FEM framework).

In paper I, a 2D convergence analysis applying the methodology explained in chapter 4, but using FVM on $\Omega_1 = [-1, 0] \times [0, 1]$, FEM on $\Omega_2 = [0, 1] \times [0, 1]$ and a second order approximation for the normal derivative as presented in the previous section is

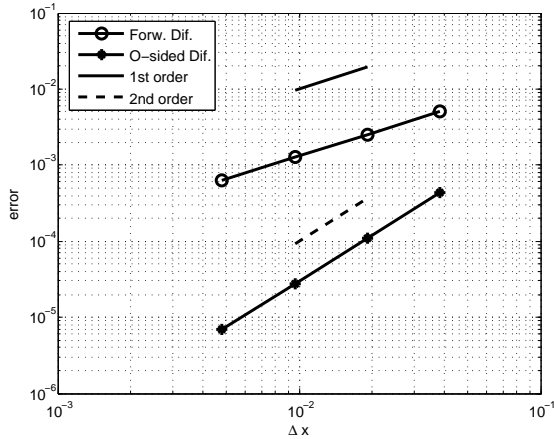


Figure 5.2: Order comparison between using forward differences or one-sided differences for the approximation of the normal derivative.

performed. These results assume that Ω_1 and Ω_2 are discretized with identical equidistant grids, i.e., $\Delta x := \Delta x_1 = \Delta x_2$.

In paper II, we generalize the analysis introduced in paper I to the case of non identical meshes on the subdomains Ω_1 and Ω_2 meaning $\Delta x_1 \neq \Delta x_2$. The scenario we consider in this work is illustrated in figure 5.3 (a), where $r := \Delta x_2 / \Delta x_1$ is a fixed aspect ratio for both 1D and 2D cases. Thus, the resolution on Ω_1 in direction tangential to the wall is the same as the resolution on Ω_2 . This means that the aspect ratio of the left subdomain cells in 2D corresponds to the ratio of grid spacings between the two subdomains in 1D. This is illustrated in figure 5.3 (b). This situation is interesting because the spatial grid of the fluid for the two nonlinear FSI test cases has this property. The fluid cells are highly non equidistant as can be seen in figures 2.8 and 2.10. Additionally, we assume that we have matching nodes at the interface as we already did in the FEM-FEM case.

Note that paper II is a follow-up of paper I because the analysis in paper I is a particular case of the analysis in paper II. More specifically, if the aspect ratio $r = 1$, then the grids on Ω_1 and Ω_2 are identical and equidistant recovering the scenario presented in paper I. For that reason, we are going to summarize here only the results presented in paper II.

There, performing a similar one-dimensional analysis as the one above in section 4.1 for this new setting of the spatial discretization, we got the following exact formula for the convergence rates:

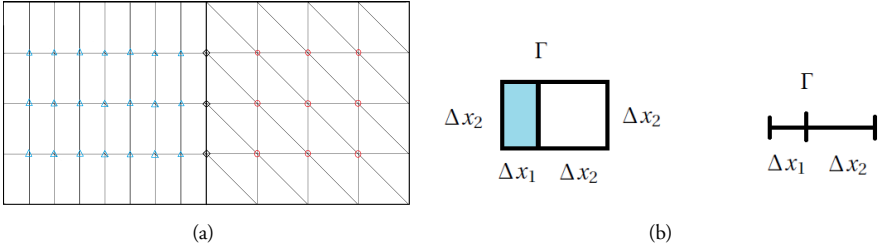


Figure 5.3: (a): Splitting of Ω into non equidistant subdomains. (b): Relation between the aspect ratio of the left subdomain cells in 2D and the ratio of grid spacings between both subdomains in 1D.

$$|\Sigma| = \frac{3\Delta x_2^2(3\lambda_1\Delta t - 2\lambda_1^2\Delta x_1\Delta t^2(4s_1 - s_0))}{\Delta x_1^2(2(\alpha_2\Delta x_2^2 + 3\lambda_2\Delta t) - \Delta x_2(\alpha_2\Delta x_2^2 - 6\lambda_2\Delta t)^2s_2)}, \quad (5.14)$$

where s_0 , s_1 and s_2 are given by

$$s_0 = \sum_{i=1}^{N_1} \frac{\sin(i\pi\Delta x_1) \sin(2i\pi\Delta x_1)}{\alpha_1\Delta x_1^2 + 2\lambda_1\Delta t(1 - \cos(i\pi\Delta x_1))}, \quad (5.15)$$

$$s_1 = \sum_{i=1}^{N_1} \frac{\sin^2(i\pi\Delta x_1)}{\alpha_1\Delta x_1^2 + 2\lambda_1\Delta t(1 - \cos(i\pi\Delta x_1))}, \quad (5.16)$$

$$s_2 = \sum_{i=1}^{N_2} \frac{\sin^2(i\pi\Delta x_2)}{2\alpha_2\Delta x_2^2 + 6\lambda_2\Delta t + (\alpha_2\Delta x_2^2 - 6\lambda_2\Delta t) \cos(i\pi\Delta x_2)}. \quad (5.17)$$

Then, the asymptotics of (5.14) for $\Delta t \rightarrow 0$ and $\Delta x_1 \rightarrow 0$ with $\Delta x_2 = r \cdot \Delta x_1$ are given by

$$\lim_{\Delta t \rightarrow 0} |\Sigma| = \frac{3\Delta x_2^2 \cdot 0}{\Delta x_1^2 \left(2\alpha_2\Delta x_2^2 - \alpha_2\Delta x_2^3 \sum_{i=1}^{N_2} \frac{3\sin^2(i\pi\Delta x_2)}{2+\cos(i\pi\Delta x_2)} \right)} = 0, \quad (5.18)$$

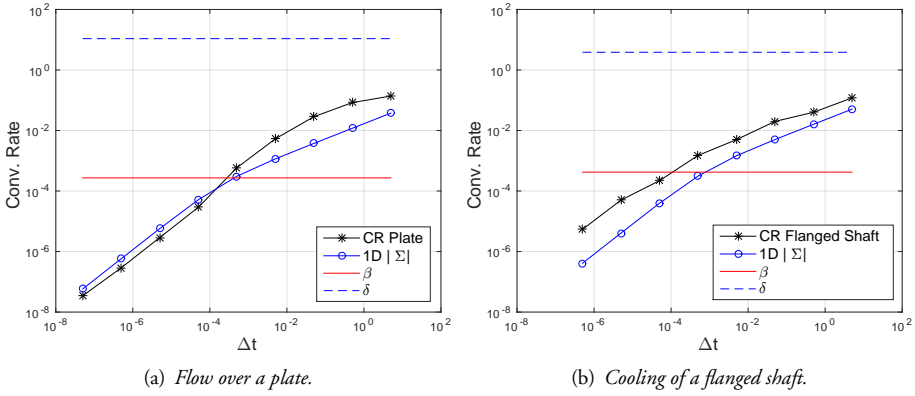


Figure 5.4: Convergence behavior of the cooling systems with respect to Δt .

$$\begin{aligned}
 \lim_{\Delta x_1 \rightarrow 0} |\Sigma| &= \frac{\lambda_1}{\lambda_2} \lim_{\Delta x_1 \rightarrow 0} \frac{3r^2 - 2r^2 \Delta x_1 \left(\frac{2(1-\Delta x_1)}{\Delta x_1} - \frac{1-2\Delta x_1}{2\Delta x_1} \right)}{2 - 2r \Delta x_1 \left(\frac{1-r\Delta x_1}{r\Delta x_1} \right)} \\
 &= \frac{\lambda_1}{\lambda_2} \lim_{\Delta x_1 \rightarrow 0} \frac{2r^2 \Delta x_1}{2r \Delta x_1} = \frac{\lambda_1}{\lambda_2} r =: \delta_r.
 \end{aligned} \tag{5.19}$$

From the result obtained in (5.19) we can conclude that the aspect ratio r also influences the behavior of the fixed point iteration, i.e, the rates will become smaller the higher the aspect ratio, e.g. the higher the Reynolds number in the fluid. This phenomenon is not unknown for PDE discretizations and is referred to as geometric stiffness. As is the case here, refining the mesh to reduce the aspect ratio would lead to faster convergence of the iterative method.

Furthermore, in paper II, we also showed numerical results where the 1D formula (5.14) also estimates the convergence rates of 2D linear examples and even of 2D nonlinear thermal FSI test cases on highly unstructured meshes. In particular, the two cooling systems introduced earlier in section 2.3.1: the cooling of a flat plate and of a flanged shaft.

In figure 5.4 (a) one observes how (5.14) predicts the rates accurately for the cooling of a flat plate. Note that the semidiscrete estimator β in (3.23) does not show any change with Δt . Remember that β is almost always constant, except for a short dynamic transition between $(\lambda_1/\lambda_2)\sqrt{D_2/D_1}$ and λ_1/λ_2 . Here, we would have to choose a Δt larger than $1e6$ to see the transition.

Finally, in figure 5.4 (b) one can see that (5.14) predicts the convergence rates for the cooling of a flanged shaft to be only slightly smaller compared to the actual performance.

This could be due to either the unstructured grids used in this case or to the non constant temperature in the structure, which varies from room temperature to $1145K$. Again, β is almost constant.

Chapter 6

Partitioned multirate approach

In this chapter we summarize the partitioned multirate approach that we have introduced in paper III for the coupling of two heterogeneous heat equations. This multirate approach was motivated because in spite of the efficient behavior of the Dirichlet-Neumann iteration in the thermal FSI framework, it has two main disadvantages. Firstly, the solvers wait for each other, and therefore, they perform sequentially. Secondly, in the time-dependent case the Dirichlet-Neumann iteration is used at each time step and consequently, both fields are solved with a common time resolution. Using instead a multirate scheme that allows for different time resolutions on each subdomain would be more efficient.

The aim of this chapter is to present a high order, parallel, multirate method for two heterogeneous coupled heat equations which could be applied to FSI problems. We use the Neumann-Neumann waveform relaxation (NNWR) method which is a waveform relaxation (WR) methods based on the classical Neumann-Neumann iteration [50, 34].

6.1 The Neumann-Neumann Waveform Relaxation algorithm

We now describe the Neumann-Neumann waveform relaxation (NNWR) algorithm [50]. The NNWR algorithm starts with an initial guess $g^0(\mathbf{x}, t)$ on the space-time interface $\Gamma \times (T_0, T_f]$, and then performs a three-step iteration.

More specifically, imposing continuity of the solution across the interface (i.e, given a common initial guess $g^0(\mathbf{x}, t)$ on $\Gamma \times (T_0, T_f)$), one can find the local solutions $u_m^{k+1}(\mathbf{x}, t)$ on Ω_m , $m = 1, 2$ through the following Dirichlet problems:

$$\begin{cases} \alpha_m \frac{\partial u_m^{k+1}(\mathbf{x}, t)}{\partial t} - \nabla \cdot (\lambda_m \nabla u_m^{k+1}(\mathbf{x}, t)) = 0, & \mathbf{x} \in \Omega_m, \\ u_m^{k+1}(\mathbf{x}, t) = 0, & \mathbf{x} \in \partial\Omega_m \setminus \Gamma, \\ u_m^{k+1}(\mathbf{x}, t) = g^k(\mathbf{x}, t), & \mathbf{x} \in \Gamma, \\ u_m^{k+1}(\mathbf{x}, 0) = u_m^0(\mathbf{x}), & \mathbf{x} \in \Omega_m. \end{cases} \quad (6.1)$$

We now add into the framework the second coupling condition which is the continuity of the heat fluxes. To this end, one solves two simultaneous Neumann problems to get the correction functions $\psi_m^{k+1}(\mathbf{x}, t)$ on Ω_m , $m = 1, 2$ where the Neumann boundary condition at the interface $\Gamma \times (T_0, T_f)$ is prescribed by the continuity of the heat fluxes of the solutions $u_m^{k+1}(\mathbf{x}, t)$ given by the Dirichlet problems:

$$\begin{cases} \alpha_m \frac{\partial \psi_m^{k+1}(\mathbf{x}, t)}{\partial t} - \nabla \cdot (\lambda_m \nabla \psi_m^{k+1}(\mathbf{x}, t)) = 0, & \mathbf{x} \in \Omega_m, \\ \psi_m^{k+1}(\mathbf{x}, t) = 0, & \mathbf{x} \in \partial\Omega_m \setminus \Gamma, \\ \lambda_m \frac{\partial \psi_m^{k+1}(\mathbf{x}, t)}{\partial \mathbf{n}_m} = \lambda_1 \frac{\partial u_1^{k+1}(\mathbf{x}, t)}{\partial \mathbf{n}_1} + \lambda_2 \frac{\partial u_2^{k+1}(\mathbf{x}, t)}{\partial \mathbf{n}_2}, & \mathbf{x} \in \Gamma, \\ \psi_m^{k+1}(\mathbf{x}, 0) = 0, & \mathbf{x} \in \Omega_m. \end{cases} \quad (6.2)$$

Finally, the interface values are updated with

$$g^{k+1}(\mathbf{x}, t) = g^k(\mathbf{x}, t) - \Theta(\psi_1^{k+1}(\mathbf{x}, t) + \psi_2^{k+1}(\mathbf{x}, t)), \quad \mathbf{x} \in \Gamma, \quad (6.3)$$

where $\Theta \in (0, 1]$ is the relaxation parameter.

The NNWR iterative method converges to the solution of the model problem (3.4) (proven in [46, chapt. 2]). Note that choosing an appropriate relaxation parameter is crucial for the good performance of the NNWR algorithm [34]. If one uses the optimal relaxation parameter, two iterations are enough.

6.2 Space-time interface interpolation

The NNWR algorithm allows the use of independent time discretization on each of the subdomains. Therefore, in the case of mismatched time grids, there exists the need to define an interface interpolation.

To this end, we consider a discrete problem in time with nonconforming time grids. Let $\tau_1 = \{t_1, t_2, \dots, t_{N_1}\}$ and $\tau_2 = \{t_1, t_2, \dots, t_{N_2}\}$ be two possibly different partitions of the time interval $[T_0, T_f]$ as shown in figure 6.1. We denote by $\Delta t_1 = (T_f - T_0)/N_1$ and $\Delta t_2 = (T_f - T_0)/N_2$ the two possibly different constant step sizes corresponding to Ω_1 and Ω_2 respectively.

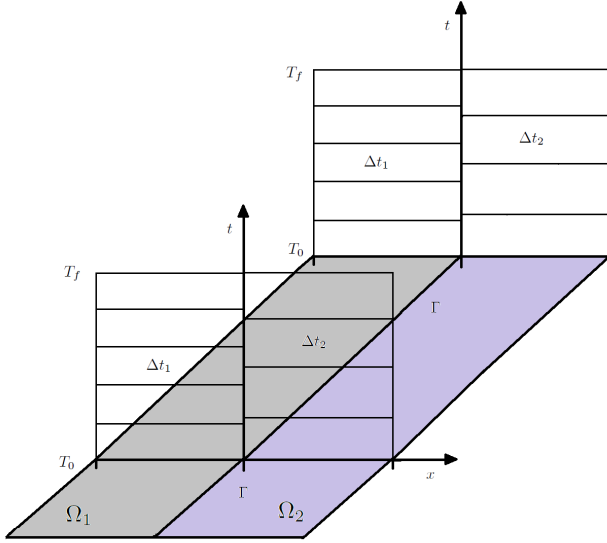


Figure 6.1: Nonconforming time grids in the two-dimensional subdomains.

In order to exchange data at the space-time interface between the different time grids, we use a linear interpolation. Given the local discrete solutions $F \in \mathbb{R}^{s \times N_1}$ and $F \in \mathbb{R}^{s \times N_2}$ at the space-time interface $\Gamma \times [T_0, T_f]$, with s being the number of grid points at Γ , we use the following procedure: For each $k = 1, 2, \dots, s$ and for each $t_i \in \tau_2$, find the subinterval in τ_1 such that $t_i \in [t_j, t_{j+1}]$. Linear interpolation through the points $(t_j, F(x_k, t_j))$ gives $G(x_k, t_j)$. We denote this by the interpolation function $G = I(\tau_2, \tau_1, F)$ summarized in algorithm 1. Conversely, we also have $F = I(\tau_1, \tau_2, G)$.

Algorithm 1 Interpolation to transfer data at the space-time interface.

```

1: procedure  $I(\tau_2, \tau_1, F)$ 
2:   for  $k = 1, 2, \dots, s$  do
3:     for  $t_i \in \tau_2$  do
4:       for  $t_j \in \tau_1$  do
5:         if  $t_i \in [t_j, t_{j+1}]$  then
6:            $G_{k,i} \leftarrow F_{k,j} + (t_i - t_j) \cdot (F_{k,j+1} - F_{k,j}) / (t_{j+1} - t_j)$ 
   return  $G$ 

```

6.3 Optimal relaxation parameter

Discretizing the problem (6.1)-(6.3) in space and applying the implicit Euler to the remaining time derivatives one gets a fully discretized purely algebraic iterative procedure. This is explained in details in paper III.

We are interested in the performance of the NNWR algorithm and therefore, it is crucial to choose an appropriate relaxation parameter Θ . If Θ is optimal, two iterations of the NNWR algorithm will be enough. In [50], a one-dimensional semidiscrete analysis is performed with constant coefficients showing that $\Theta_{opt} = 1/4$ when the two subdomains Ω_1 and Ω_2 are identical. However, the case of jumping coefficients is not covered by that analysis. For that reason, we proposed in paper III a more general fully discrete analysis for the coupling of two heterogeneous heat equations.

To that end, we derived the iteration matrix of the fully discrete NNWR algorithm with respect to the interface unknowns for one single nonmultirate time step (using implicit Euler for the time integration). We got

$$\Sigma = \mathbf{I} - \Theta \left(2\mathbf{I} + \mathbf{S}^{(1)-1}\mathbf{S}^{(2)} + \mathbf{S}^{(2)-1}\mathbf{S}^{(1)} \right), \quad (6.4)$$

where $\mathbf{S}^{(m)}$, $m = 1, 2$ are the Schur complements in (3.22).

In the one-dimensional case, the iteration matrix Σ is just a real number and thus its spectral radius is its modulus. Then, the optimal relaxation parameter Θ_{opt} in 1D is given by

$$\Theta_{opt} = \frac{1}{2 + \mathbf{S}^{(1)-1}\mathbf{S}^{(2)} + \mathbf{S}^{(2)-1}\mathbf{S}^{(1)}}. \quad (6.5)$$

And finally, we studied the iteration matrix (6.4) for the specific 1D FE discretization in section 4.1. obtaining the following optimal relaxation parameter Θ_{opt} in terms of $c := \Delta t / \Delta x^2$:

$$\Theta_{opt} = \left(2 + \frac{6\Delta t(\alpha_2 + 3\lambda_2 c) - \Delta x(\alpha_2 - 6\lambda_2 c)^2 s'_2}{6\Delta t(\alpha_1 + 3\lambda_1 c) - \Delta x(\alpha_1 - 6\lambda_1 c)^2 s'_1} + \frac{6\Delta t(\alpha_1 + 3\lambda_1 c) - \Delta x(\alpha_1 - 6\lambda_1 c)^2 s'_1}{6\Delta t(\alpha_2 + 3\lambda_2 c) - \Delta x(\alpha_2 - 6\lambda_2 c)^2 s'_2} \right)^{-1}. \quad (6.6)$$

where

$$s'_m = \sum_{i=1}^N \frac{3\Delta t \sin^2(i\pi\Delta x)}{2\alpha_m + 6\lambda_m c + (\alpha_m - 6\lambda_m c) \cos(i\pi\Delta x)}, \quad (6.7)$$

for $m = 1, 2$.

Consequently, the temporal and spatial limits of Θ_{opt} in (6.6) are

$$\Theta_{\{c \rightarrow 0\}} = \frac{\alpha_1 \alpha_2}{(\alpha_1 + \alpha_2)^2}, \quad (6.8)$$

$$\Theta_{\{c \rightarrow \infty\}} = \frac{\lambda_1 \lambda_2}{(\lambda_1 + \lambda_2)^2}. \quad (6.9)$$

The semidiscrete result in [50] is recovered by our analysis when approaching the continuous case in space in (6.9) for constant coefficients, i.e, $\lambda_1 = \lambda_2$. In that case, one gets

$$\Theta_{opt} = \frac{\lambda_1 \lambda_2}{(\lambda_1 + \lambda_2)^2} = \frac{\lambda_1^2}{4\lambda_1^2} = \frac{1}{4}. \quad (6.10)$$

6.4 Multirate SDIRK2-NNWR algorithm

The NNWR algorithm using implicit Euler mentioned in the previous section provides a first order solution. Instead we want to provide a high order solution and for that reason we also presented in paper III an alternative algorithm using as a time integration method a second order singly diagonally implicit Runge-Kutta (SDIRK2).

Consider the autonomous initial value problem

$$\dot{\mathbf{u}}(t) = \mathbf{f}(\mathbf{u}(t)), \quad \mathbf{u}(0) = \mathbf{u}_0. \quad (6.11)$$

An SDIRK method is then defined as

$$\begin{aligned} \mathbf{U}^i &= \mathbf{u}^n + \Delta t_n \sum_{k=1}^i a_{ik} \mathbf{f}(\mathbf{U}^k), \quad i = 1, \dots, j \\ \mathbf{u}^{n+1} &= \mathbf{u}^n + \Delta t_n \sum_{i=1}^j b_i \mathbf{f}(\mathbf{U}^i) \end{aligned} \quad (6.12)$$

with given coefficients a_{ik} and b_i . The two-stage method SDIRK2 is defined by the coefficients in table 6.4. As the coefficients a_{2i} and b_i for $i = 1, 2$ are identical, the second equation in (6.12) is superfluous because $\mathbf{u}^{n+1} = \mathbf{U}^2$.

The vectors $\mathbf{k}_i = \mathbf{f}(\mathbf{U}^i)$ are called stage derivatives and j is the number of stages. Since the starting vector

Table 6.1: Butcher-tableau of SDIRK2 with $a = 1 - \frac{1}{2}\sqrt{2}$ and $\hat{a} = 2 - \frac{5}{4}\sqrt{2}$.

a	a	0
1	$1 - a$	a
	$1 - a$	a
	$1 - \hat{a}$	\hat{a}

$$\mathbf{s}_i = \mathbf{u}^n + \Delta t_n \sum_{k=1}^{i-1} a_{ik} \mathbf{k}_k, \quad i = 1, \dots, j-1, \quad (6.13)$$

is known, (6.12) is just a sequence of implicit Euler steps.

After discretizing equations (6.1) and (6.2) in space, we can rewrite the resulting systems into a fully discrete form by applying SDIRK2 with time step Δt_1 on Ω_1 and Δt_2 on Ω_2 . From now on, the local approximations and the solutions at the space-time interface are given by the discrete vectors $\mathbf{u}_I^{(m),k,n_m} \in \mathbb{R}^{S_m}$ and $\mathbf{u}_\Gamma^{k,n_m} \in \mathbb{R}^s$ respectively. Here S_m is the number of spatial grid points on Ω_m and s is the number of spatial grid points at the interface Γ . Moreover, k indicates the iterate and $n_m := 1, 2, \dots, N_m$ are the time integration indices with respect to Ω_m , $m = 1, 2$. Similarly, the corrections both in the subdomains and at the interface are given by the discrete vectors $\psi_I^{(m),k,n_m} \in \mathbb{R}^{S_m}$ and $\psi_\Gamma^{(m),k,n_m} \in \mathbb{R}^s$ respectively.

Therefore, at each fixed point iteration k , let $\mathbf{s}_1^{(m)} = \mathbf{u}_I^{(m),k+1,n_m}$ and $\mathbf{s}_2^{(m)} = \mathbf{u}_I^{(m),k+1,n_m} + \Delta t_m(1-a)\mathbf{k}_1^{(m)}$ be the starting vectors. Then, one can solve a fully discretized version of the two Dirichlet problems from (6.1) for $n_m = 1, 2, \dots, N_m$, with the initial conditions $\mathbf{u}_I^{(m),k+1,0}$, $m = 1, 2$, $\mathbf{u}_\Gamma^{k+1,0}$ simultaneously:

$$\begin{aligned} \left(\frac{\mathbf{M}_{II}^{(m)}}{a\Delta t_m} + \mathbf{A}_{II}^{(m)} \right) \mathbf{U}_j^{(m)} &= \frac{\mathbf{M}_{II}^{(m)}}{a\Delta t_m} \mathbf{s}_j^{(m)} - \mathbf{M}_{I\Gamma}^{(m)} \mathbf{u}_\Gamma^{k,n_m+j-1+(2-j)a} \\ &\quad - \mathbf{A}_{I\Gamma}^{(m)} \mathbf{u}_\Gamma^{k,n_m+j-1+(2-j)a}, \quad j = 1, 2. \\ \mathbf{u}_I^{(m),k+1,n_m+1} &= \mathbf{U}_2^{(m)}, \end{aligned} \quad (6.14)$$

where $\mathbf{U}_j^{(m)}$, $\mathbf{s}_j^{(m)}$, $\mathbf{k}_j^{(m)} \in \mathbb{R}^{S_m}$ and $\mathbf{M}_{II}^{(m)}$, $\mathbf{M}_{I\Gamma}^{(m)}$, $\mathbf{A}_{II}^{(m)}$ and $\mathbf{A}_{I\Gamma}^{(m)}$ are the discretization matrices. Note that interpolation is not needed to solve the Dirichlet problems because $\mathbf{u}_I^{(1),k+1,n_1+1}$ in (6.14) is only dependent on terms related to Ω_1 . In the same way, $\mathbf{u}_I^{(2),k+1,n_2+1}$ in (6.14) only depends on n_2 . The stage derivatives are given by $\mathbf{k}_j^{(m)} =$

$\frac{1}{a\Delta t_m}(\mathbf{U}_j^{(m)} - \mathbf{s}_j^{(m)})$. Remember that the index $m = 1, 2$, denotes the subdomain and the index $j = 1, 2$, denotes the stage.

The fluxes are fully discrete versions of the normal derivatives in the third equation of (6.2). These are given by $\mathbf{F}_j^{(1),k,\tau_1} := \mathbf{f}_j^{(1),k,\tau_1} + I(\tau_1, \tau_2, \mathbf{f}_j^{(2),k,\tau_2})$, $\mathbf{F}_j^{(2),k,\tau_2} := \mathbf{f}_j^{(2),k,\tau_2} + I(\tau_2, \tau_1, \mathbf{f}_j^{(1),k,\tau_1})$ with

$$\begin{aligned} \mathbf{f}_j^{(m),k,n_m} &= \mathbf{M}_{\Gamma\Gamma}^{(m)} \mathbf{u}_{\Gamma}^{k,n_m+j-1+(2-j)a} + \mathbf{M}_{\Gamma I}^{(m)} \mathbf{k}_j^{(m)} \\ &\quad + \mathbf{A}_{\Gamma\Gamma}^{(m)} \mathbf{u}_{\Gamma}^{k,n_m+j-1+(2-j)a} + \mathbf{A}_{\Gamma I}^{(m)} \mathbf{U}_j^{(m)}, \end{aligned} \quad (6.15)$$

for $m = 1, 2$. Note that interpolation here is needed because the components of $\mathbf{F}_j^{(1),k,\tau_1}$ and $\mathbf{F}_j^{(2),k,\tau_2}$ for the two stages $j = 1, 2$ correspond to different time integrations.

Finally, one can solve a fully discretized version of the two Neumann problems in (6.2)

in terms of the vector of unknowns $\psi_m^{k+1,n_m+1} := \left(\psi_I^{(m),k+1,n_m+1T} \psi_{\Gamma}^{(m),k+1,n_m+1T} \right)^T$.

Let $\mathbf{s}_1^{(m)} = \psi_m^{k+1,n_m}$ and $\mathbf{s}_2^{(m)} = \psi_m^{k+1,n_m} + \Delta t_m(1-a)\mathbf{k}_1^{(m)}$ be the starting vectors. One then solves the two Neumann problems for $n_m = 1, 2, \dots, N_m$, with the initial conditions $\psi_m^{k+1,0} = \psi_m^{k+1}(T_0)$, $m = 1, 2$ in parallel:

$$\begin{aligned} \left(\frac{\mathbf{M}_m}{a\Delta t_m} + \mathbf{A}_m \right) \mathbf{Y}_j^{(m)} &= \frac{\mathbf{M}_m}{a\Delta t_m} \mathbf{s}_j^{(m)} + \mathbf{b}_j^{(m),k,n_m}, \quad j = 1, 2, \\ \psi_m^{k+1,n_m+1} &= \mathbf{Y}_2^{(m)}, \end{aligned} \quad (6.16)$$

Then, the interfaces values are updated respectively by

$$\mathbf{u}_{\Gamma}^{k+1,\tau_1} = \mathbf{u}_{\Gamma}^{k,\tau_1} - \Theta \left(\psi_{\Gamma}^{(1),k+1,\tau_1} + I \left(\tau_1, \tau_2, \psi_{\Gamma}^{(2),k+1,\tau_2} \right) \right), \quad (6.17)$$

$$\mathbf{u}_{\Gamma}^{k+1,\tau_2} = \mathbf{u}_{\Gamma}^{k,\tau_2} - \Theta \left(\psi_{\Gamma}^{(2),k+1,\tau_2} + I \left(\tau_2, \tau_1, \psi_{\Gamma}^{(1),k+1,\tau_1} \right) \right). \quad (6.18)$$

Here, interpolation is needed because $\psi_{\Gamma}^{(1),k+1,\tau_1}$ and $\psi_{\Gamma}^{(2),k+1,\tau_2}$ are nonconforming.

Finally, if the termination criteria $\|\mathbf{u}_{\Gamma}^{k+1,N_m} - \mathbf{u}_{\Gamma}^{k,N_m}\|$ is not small enough, one starts the process from (6.14) once more.

Algorithm 2 and 3 summarize the discrete Dirichlet solver in (6.14) and the discrete Neumann solver in (6.16) respectively. Furthermore, the complete multirate SDIRK2-NNWR algorithm is summarized in algorithm 4.

In conclusion, in paper III we suggested a new high order parallel NNWR method with nonconforming time grids for two heterogeneous coupled heat equations and studied the optimal relaxation parameter in terms of the material coefficients and the temporal and spatial resolutions Δt and Δx . The method using Θ_{opt} converges extremely fast, typically within two iterations. This is confirmed in paper III by numerical results, where we also demonstrated that the nonmultirate 1D case gives excellent estimates for the multirate 1D case and even for multirate and nonmultirate 2D examples using both implicit Euler and SDIRK2. In addition, we also showed that the NNWR method is a more efficient choice than the Dirichlet-Neumann waveform relaxation (DNWR) algorithm in the multirate case.

Algorithm 2 Solver for the Dirichlet problems in (6.14).

```

1: procedure TA-D( $\mathbf{u}_I^{k+1,n}, \mathbf{u}_\Gamma^{k,n}, \mathbf{u}_\Gamma^{k,n+1}$ )
2:    $\mathbf{u}_\Gamma^{k,n+a} \leftarrow \mathbf{u}_\Gamma^{k,n} + a \left( \mathbf{u}_\Gamma^{k,n+1} - \mathbf{u}_\Gamma^{k,n} \right)$  # linear interp. for intermediate stage
3:    $\dot{\mathbf{u}}_\Gamma^{k,n+a}, \dot{\mathbf{u}}_\Gamma^{k,n+1} \leftarrow \left( \mathbf{u}_\Gamma^{k,n+1} - \mathbf{u}_\Gamma^{k,n} \right) / \Delta t$  # forward diff. for derivatives
4:   for  $j = 1, 2$  do # loop over stages
5:      $\mathbf{s}_j \leftarrow \mathbf{u}_I^{k+1,n} + \Delta t \sum_{l=1}^{j-1} (1-a) \mathbf{k}_l$ 
6:      $\mathbf{U}_j \leftarrow \text{SOLVILINEARSYSTEM}(\mathbf{s}_j, \dot{\mathbf{u}}_\Gamma^{k,n+j-1+(2-j)a}, \mathbf{u}_\Gamma^{k,n+j-1+(2-j)a})$ 
# 1st eq in (6.14)
7:      $\mathbf{k}_j \leftarrow \frac{1}{a\Delta t} (\mathbf{U}_j - \mathbf{s}_j)$ 
8:      $\mathbf{u}_I^{k+1,n+1} \leftarrow \mathbf{U}_2$ 
9:     for  $j = 1, 2$  do # compute fluxes in (6.15)
10:       $\mathbf{f}_j^{k,n} \leftarrow \mathbf{M}_{\Gamma\Gamma} \dot{\mathbf{u}}_\Gamma^{k,n+j-1+(2-j)a} + \mathbf{M}_{\Gamma I} \mathbf{k}_j + \mathbf{A}_{\Gamma\Gamma} \mathbf{u}_\Gamma^{k,n+j-1+(2-j)a} + \mathbf{A}_{\Gamma I} \mathbf{U}_j$ 
return  $\mathbf{u}_I^{k+1,n+1}, \mathbf{f}_j^{k,n}$ 

```

Algorithm 3 Solver for the Neumann problems in (6.16).

```

1: procedure SDIRK2NEUMANN( $\psi_I^{k+1,n}, \psi_\Gamma^{k+1,n}, \mathbf{F}_j^{k,n}$ )
2:   for  $j = 1, 2$  do # loop over stages
3:      $\mathbf{s}_j \leftarrow \psi^{k+1,n} + \Delta t \sum_{l=1}^{j-1} (1-a) \mathbf{k}_l$ 
4:      $\mathbf{Y}_j \leftarrow \text{SOLVILINEARSYSTEM}(\mathbf{s}_j, \psi_I^{k+1,n}, \psi_\Gamma^{k+1,n}, \mathbf{F}_j^{k,n})$ , # 1st eq in (6.16)
5:      $\mathbf{k}_j \leftarrow \frac{1}{a\Delta t} (\mathbf{Y}_j - \mathbf{s}_j)$ 
6:      $\psi_I^{k+1,n+1}, \psi_\Gamma^{k+1,n+1} \leftarrow \mathbf{Y}_2$ 
return  $\psi_I^{k+1,n+1}, \psi_\Gamma^{k+1,n+1}$ 

```

Algorithm 4 Multirate SDIRK2-NNWR algorithm.

```

1: procedure NNWR2( $\tau_1, \tau_2, \alpha_1, \alpha_2, \lambda_1, \lambda_2, \Theta, TOL$ )
2:    $\mathbf{u}_I^{(m),k+1,0}, \mathbf{u}_\Gamma^0(\tau_m), \psi_I^{(m),k+1,n_m+1}, \psi_\Gamma^{(m),k+1,n_m+1} \leftarrow$  INITIALIZATION
3:   while  $\|\mathbf{u}_\Gamma^{k+1,N_m} - \mathbf{u}_\Gamma^{k,N_m}\| \leq TOL$  do
4:     for  $j = 1, 2$  do # loop over stages
5:       for  $t_{n_1} \in \tau_1$  do
6:          $\mathbf{u}_I^{(1),k+1,n_1+1}, \mathbf{f}_j^{(1),k,n_1} \leftarrow$  SDIRK2DIRICHLET( $\mathbf{u}_I^{(1),k+1,n_1}, \mathbf{u}_\Gamma^{k,n_1}, \mathbf{u}_\Gamma^{k,n_1+1}$ )
7:       for  $t_{n_2} \in \tau_2$  do (in parallel to 5)
8:          $\mathbf{u}_I^{(2),k+1,n_2+1}, \mathbf{f}_j^{(2),k,n_2} \leftarrow$  SDIRK2DIRICHLET( $\mathbf{u}_I^{(2),k+1,n_2}, \mathbf{u}_\Gamma^{k,n_2}, \mathbf{u}_\Gamma^{k,n_2+1}$ )
9:          $\mathbf{F}_j^{(1),k,\tau_1} \leftarrow \mathbf{f}_j^{(1),k,\tau_1} + I(\tau_1, \tau_2, \mathbf{f}_j^{(2),k,\tau_2})$ 
10:         $\mathbf{F}_j^{(2),k,\tau_2} \leftarrow \mathbf{f}_j^{(2),k,\tau_2} + I(\tau_2, \tau_1, \mathbf{f}_j^{(1),k,\tau_1})$  (in parallel to 9)
11:        for  $t_{n_1} \in \tau_1$  do
12:           $\psi_I^{k+1,n_1+1}, \psi_\Gamma^{k+1,n_1+1} \leftarrow$  SDIRK2NEUMANN( $\psi_I^{k+1,n_1}, \psi_\Gamma^{k+1,n_1}, \mathbf{F}_j^{(1),k,n_1}$ )
13:        for  $t_{n_2} \in \tau_2$  do (in parallel to 11)
14:           $\psi_I^{k+1,n_2+1}, \psi_\Gamma^{k+1,n_2+1} \leftarrow$  SDIRK2NEUMANN( $\psi_I^{k+1,n_2}, \psi_\Gamma^{k+1,n_2}, \mathbf{F}_j^{(2),k,n_2}$ )
15:         $\mathbf{u}_\Gamma^{k+1,\tau_1} \leftarrow \mathbf{u}_\Gamma^{k,\tau_1} - \Theta \left( \psi_\Gamma^{(1),k+1,\tau_1} + I(\tau_1, \tau_2, \psi_\Gamma^{(2),k+1,\tau_2}) \right)$ 
16:         $\mathbf{u}_\Gamma^{k+1,\tau_2} \leftarrow \mathbf{u}_\Gamma^{k,\tau_2} - \Theta \left( \psi_\Gamma^{(2),k+1,\tau_2} + I(\tau_2, \tau_1, \psi_\Gamma^{(1),k+1,\tau_1}) \right)$  (p. to 15)

```

Chapter 7

Partitioned time adaptive approach

The total number of time steps for each subsolver in the multirate NNWR algorithm presented above and in paper III has to be chosen beforehand. For that reason, a time adaptive version of the multirate NNWR algorithm would be a more efficient method. In this chapter we introduce a new adaptive scheme that optimizes the total number of time steps for each subsolver allowing big step sizes without increasing the error of the numerical solution. Building a variable step size multirate scheme allows each of the subsolvers to construct its own time grid independently of each other. This algorithm could be further applied to thermal FSI simulations increasing the efficiency of the coupling procedure. A time adaptive method for thermal FSI was introduced in [11]. There, the same variable time integration was used for both subsolvers performing a Dirichlet-Neumann iteration with respect to the interface unknowns at each time step. Here, we present an alternative method where each subsolver performs in parallel an independent time adaptive integration that are synchronized at the end point through a fixed point iteration.

7.1 Time adaptive method

We are going to build our partitioned time adaptive approach on the SDIRK2-NNWR algorithm introduced in the previous chapter and in paper III. To that end, an error estimate at each time step is needed to be able to choose a new step size.

In order to get an error estimate for the autonomous initial value problem (6.11) using SDIRK2, we use an embedded technique [42, chap. IV.8], i.e. the coefficients in the Butcher-array a_{ik} (see table 6.4) are the same for a lower order method, which yields

the result

$$\hat{\mathbf{u}}^{n+1} = \mathbf{u}^n + \Delta t_n \sum_{i=1}^j \hat{b}_i \mathbf{f}(\mathbf{U}^i). \quad (7.1)$$

Then, an approximation to the local error of the low order method is given by the difference of both solutions

$$\mathbf{r}^{n+1} := \mathbf{u}^{n+1} - \hat{\mathbf{u}}^{n+1} = \Delta t_n \sum_{i=1}^j (b_i - \hat{b}_i) \mathbf{f}(\mathbf{U}^i). \quad (7.2)$$

Taking the Euclidean norm throughout we consider the error estimate at each time step given by $\|\mathbf{r}^{n+1}\|_2$. We use a proportional-integral controller (PI controller) [76, 77],

$$\Delta t^{n+1} = \Delta t^n \left(\frac{TOL}{\|\mathbf{r}^{n+1}\|_2} \right)^{1/12} \left(\frac{TOL}{\|\mathbf{r}^n\|_2} \right)^{1/12}. \quad (7.3)$$

Here we use the current error estimate, as well as the previous error estimate. However, on the first step \mathbf{r}^0 is not available and for that reason, we put $\mathbf{r}^0 = TOL$. Afterwards, the recursion will start operating as intended.

In order to start the integration, one also needs to pick an initial step size. Instead of using the classic result in [41, pp. 169], we use the following formula suggested by Gustaf Söderlind and inspired by [1, pp. 682-683]

$$\Delta t^0 = \frac{|T_f - T_0| \cdot TOL^{1/2}}{100 \cdot (1 + \|\mathbf{f}(\mathbf{u}_0)\|_2)}. \quad (7.4)$$

So far, we have described the time adaptive methodology for the autonomous initial value problem in (6.11) using the time integration method SDIRK2. However, we are interested in constructing a time adaptive version of the SDIRK2-NNWR algorithm using (7.3) as controller and (7.4) as starter as we will explain below.

In our approach, we have decided to introduce time adaptive processes for the time integration of the two Dirichlet problems (6.1) which build two independent time grids τ_1 and τ_2 . The Neumann problems (6.2) and the update step (6.3) use the same time grids τ_1 and τ_2 in a multirate manner as illustrated in figure 7.1. We have chosen to have only the Dirichlet problems adaptive in this first approach because it is there where the solutions of the problem $u_1(\mathbf{x}, t)$ and $u_2(\mathbf{x}, t)$ are found. The Neumann problems instead only find the corrections $\psi_1(\mathbf{x}, t)$ and $\psi_2(\mathbf{x}, t)$. Other time adaptive strategies are possible and intended to be investigated in future research. Those could be to introduce

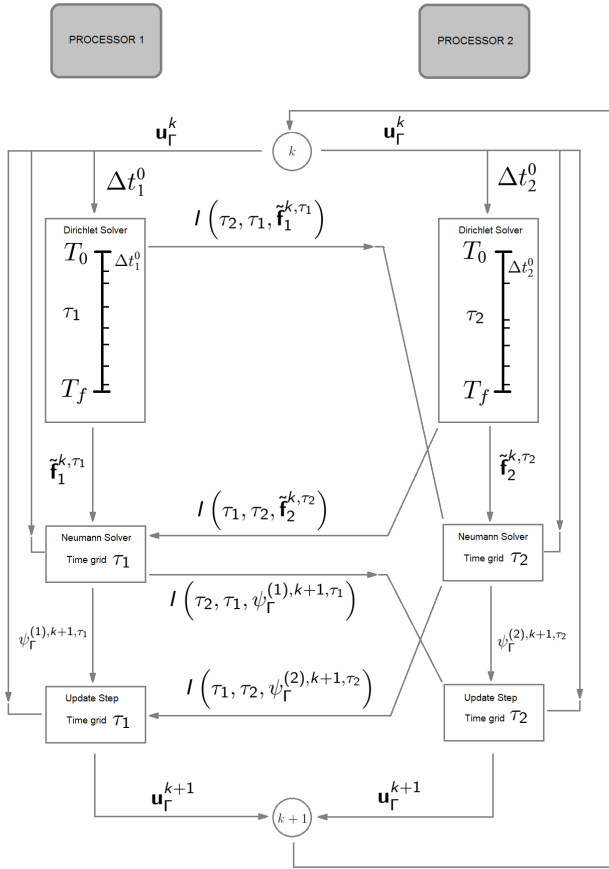


Figure 7.1: Illustration of the time adaptive SDIRK2-NNWR algorithm. The process starts with the space-time interface functions \mathbf{u}_Γ^k and the initial step sizes Δt_m^0 for $m = 1, 2$. Those are needed to run the time adaptive Dirichlet solvers in parallel getting the time grids τ_m and $\mathbf{u}_\Gamma^{(m), k+1, \tau_m}$, $m = 1, 2$. In order to run the multirate Neumann solvers for the corrections of the solution, one needs to provide the fluxes $\tilde{\mathbf{f}}_1^{k, \tau_1}$, $\tilde{\mathbf{f}}_2^{k, \tau_2}$ and their corresponding interpolations $I(\tau_2, \tau_1, \tilde{\mathbf{f}}_1^{k, \tau_1})$, $I(\tau_1, \tau_2, \tilde{\mathbf{f}}_2^{k, \tau_2})$. One can then run the Neumann problems in parallel getting the corrections $\psi_\Gamma^{(1), k+1, \tau_1}$ and $\psi_\Gamma^{(2), k+1, \tau_2}$ at the space-time interface. Finally, those and their interpolations $I(\tau_1, \tau_2, \psi_\Gamma^{(2), k+1, \tau_2})$ and $I(\tau_2, \tau_1, \psi_\Gamma^{(1), k+1, \tau_1})$ are used to update the space-time interface values. If needed, the process is restarted.

time adaptive processes on the Neumann problems instead or to build both the Dirichlet and the Neumann problems with variable step sizes.

As our time adaptive SDIRK2-NNWR algorithm contains two time adaptive Dirichlet solvers, we define the corresponding local errors as

$$\mathbf{r}_m^{n+1} = \mathbf{u}_I^{(m),n+1} - \hat{\mathbf{u}}_I^{(m),n+1}, \quad (7.5)$$

where $\hat{\mathbf{u}}_I^{(m),n+1}$ corresponds to the lower order solution of SDIRK2 and $\|\mathbf{r}_m^{n+1}\|_2$ are the corresponding error estimates for $m = 1, 2$.

Consequently, taking (7.3) and (7.4) as references

$$\Delta t_m^{n+1} = \Delta t_m^n \left(\frac{TOL_m}{\|\mathbf{r}_m^{n+1}\|_2} \right)^{1/12} \left(\frac{TOL_m}{\|\mathbf{r}_m^n\|_2} \right)^{1/12}, \quad (7.6)$$

with $\mathbf{r}_m^0 = TOL_m$ are the step size controllers with initial step sizes

$$\Delta t_m^0 = \frac{|T_f - T_0| \cdot TOL_m^{1/2}}{100 \cdot \left(1 + \|\mathbf{M}_{II}^{(m)-1} \mathbf{A}_{II}^{(m)} \mathbf{u}_I^{(m),0}\|_2 \right)}, \quad (7.7)$$

for the Dirichlet problem on the subdomain Ω_m for $m = 1, 2$ respectively.

Figure 7.1 also sketches the communication needed for the time adaptive SDIRK2-NNWR scheme just explained and summarized in algorithm 5. Note that the functions TA-D() and SDIRK2NEUMANN() were already introduced in the previous chapter, algorithm 2 and 3 respectively. The function UPDATERELAXATION() is explained in details in the next section. Furthermore, as we typically have mismatched time grids on the two space-time subdomains $\Omega_1 \times [T_0, T_f]$ and $\Omega_2 \times [T_0, T_f]$, the linear interpolation represented by the function $I()$ and explained in details in section 6 of paper III is employed. Finally, we have chosen the inner time adaptive tolerances finer than the outer tolerance used to terminate the iteration. Specifically, we take $TOL_m = TOL/5$ for $m = 1, 2$. This choice is motivated by [78] and already used in a similar context in section 6 of paper I.

7.2 Time adaptive relaxation parameter

In paper III we introduced a multirate SDIRK2-NNWR algorithm. There, we show that the choice of the relaxation parameter Θ was crucial to get a convergent iterative method. If the parameter is optimal, the algorithm will terminate after two iterations. However, if

Algorithm 5 Time adaptive SDIRK2-NNWR algorithm.

```

1: procedure TA-NNWR2( $\tau_1, \tau_2, \alpha_1, \alpha_2, \lambda_1, \lambda_2, \Delta x, TOL$ )
2:    $\mathbf{u}_I^{(m),k+1,0}, \mathbf{u}_\Gamma^0(\tau_m), \psi_I^{(m),k+1,n_m+1}, \psi_\Gamma^{(m),k+1,n_m+1} \leftarrow$  INITIALIZATION
3:    $\Theta \leftarrow (\Theta_{\{c \rightarrow 0\}} + \Theta_{\{c \rightarrow \infty\}}) / 2$  # initial guess relaxation parameter
4:    $TOL_m \leftarrow TOL/5$ 
5:    $\Delta t_m^0 \leftarrow (|T_f - T_0| \cdot TOL_m^{1/2}) / (100 \cdot (1 + \|\mathbf{M}_{II}^{(m)-1} \mathbf{A}_{II}^{(m)} \mathbf{u}_I^{(m),0}\|_2))$ 
6:   while  $\|\mathbf{u}_\Gamma^{k+1,N_m} - \mathbf{u}_\Gamma^{k,N_m}\| \leq TOL$  do
7:     for  $j = 1, 2$  do # loop over stages
8:        $t_{n_m} \leftarrow \Delta t_m^0, \mathbf{r}_m^0 \leftarrow TOL_m, \tau_m \leftarrow \{\Delta t_m^0\}$ 
9:       while  $t_{n_1} < T_f$  do
10:         $\mathbf{u}_I^{(1),k+1,n_1}, \mathbf{f}_j^{(1),k,n_1}, \mathbf{r}_1^{n_1} \leftarrow$  TA-D( $\mathbf{u}_I^{(1),k+1,n_1-1}, \mathbf{u}_\Gamma^{k,n_1-1}, \mathbf{u}_\Gamma^{k,n_1}$ )
11:         $\Delta t_1^{n_1} \leftarrow \Delta t_1^{n_1} (TOL_1 / \|\mathbf{r}_1^{n_1}\|_2)^{1/12} (TOL_1 / \|\mathbf{r}_1^{n_1-1}\|_2)^{1/12}$ 
12:         $t_{n_1} \leftarrow t_{n_1} + \Delta t_1^{n_1}, \tau_1 = \{\tau_1, t_{n_1}\}$ 
13:        while  $t_{n_2} < T_f$  do (in parallel to 8)
14:          $\mathbf{u}_I^{(2),k+1,n_2}, \mathbf{f}_j^{(2),k,n_2}, \mathbf{r}_2^{n_2} \leftarrow$  TA-D( $\mathbf{u}_I^{(2),k+1,n_2-1}, \mathbf{u}_\Gamma^{k,n_2-1}, \mathbf{u}_\Gamma^{k,n_2}$ )
15:          $\Delta t_2^{n_2} \leftarrow \Delta t_2^{n_2} (TOL_2 / \|\mathbf{r}_2^{n_2}\|_2)^{1/12} (TOL_2 / \|\mathbf{r}_2^{n_2-1}\|_2)^{1/12}$ 
16:          $t_{n_2} \leftarrow t_{n_2} + \Delta t_2^{n_2}, \tau_2 = \{\tau_2, t_{n_2}\}$ 
17:          $t_{N_m} \leftarrow T_f$  # overwrite last element time grid  $\tau_m$ 
18:          $\mathbf{u}_I^{(m),k+1,N_m}, \mathbf{f}_j^{(m),k,N_m} \leftarrow$  TA-D( $\mathbf{u}_I^{(m),k+1,N_m-1}, \mathbf{u}_\Gamma^{k,N_m-1}, \mathbf{u}_\Gamma^{k,N_m}$ )
19:          $\mathbf{F}_j^{(1),k,\tau_1} \leftarrow \mathbf{f}_j^{(1),k,\tau_1} + I(\tau_1, \tau_2, \mathbf{f}_j^{(2),k,\tau_2})$ 
20:          $\mathbf{F}_j^{(2),k,\tau_2} \leftarrow \mathbf{f}_j^{(2),k,\tau_2} + I(\tau_2, \tau_1, \mathbf{f}_j^{(1),k,\tau_1})$  (in parallel to 19)
21:         for  $t_{n_1} \in \tau_1$  do
22:           $\psi_I^{k+1,n_1}, \psi_\Gamma^{k+1,n_1} \leftarrow$  SDIRK2NEUMANN( $\psi_I^{k+1,n_1-1}, \psi_\Gamma^{k+1,n_1-1}, \mathbf{F}_j^{(1),k,n_1-1}$ )
23:          for  $t_{n_2} \in \tau_2$  do (in parallel to 21)
24:            $\psi_I^{k+1,n_2}, \psi_\Gamma^{k+1,n_2} \leftarrow$  SDIRK2NEUMANN( $\psi_I^{k+1,n_2-1}, \psi_\Gamma^{k+1,n_2-1}, \mathbf{F}_j^{(2),k,n_2-1}$ )
25:            $\mathbf{u}_\Gamma^{k+1,\tau_1} \leftarrow \mathbf{u}_\Gamma^{k,\tau_1} - \Theta \left( \psi_\Gamma^{(1),k+1,\tau_1} + I(\tau_1, \tau_2, \psi_\Gamma^{(2),k+1,\tau_2}) \right)$ 
26:            $\mathbf{u}_\Gamma^{k+1,\tau_2} \leftarrow \mathbf{u}_\Gamma^{k,\tau_2} - \Theta \left( \psi_\Gamma^{(2),k+1,\tau_2} + I(\tau_2, \tau_1, \psi_\Gamma^{(1),k+1,\tau_1}) \right)$  (p. to 25)
27:            $\Theta \leftarrow$  UPDATERELAXATION( $\tau_1, \tau_2, \alpha_1, \alpha_2, \lambda_1, \lambda_2, \Delta x$ )

```

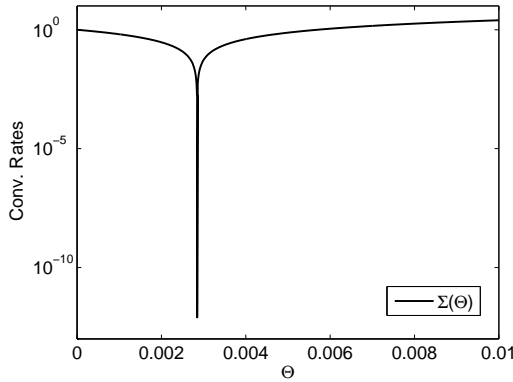


Figure 7.2: Air-water convergence rates as a function of the relaxation parameter Θ in 1D.

one deviates slightly from it, the method easily gets worse and it could even be divergent because the optimum is a singularity as illustrated in figure 7.2.

The aim here is to adapt the formula derived for Θ_{opt} in the SDIRK2-NNWR algorithm (check equation (6.6)) to the variable step size context. Θ_{opt} in (6.6) is dependent on the coupled material coefficients α_1 , α_2 , λ_1 , λ_2 , the spatial resolution Δx and the time resolution Δt through the parameter $c := \Delta t / \Delta x^2$. The main issues in the time adaptive approach is that we do not have a fixed value for Δt and we do not know beforehand how the time grids τ_1 and τ_2 look like. We propose to start the algorithm with an initial guess for Θ and update the value at each iteration once the time grids τ_1 and τ_2 have already been computed. Both the initial guess and the update to be presented are motivated by the numerical experiments shown in paper III.

Firstly, the initial guess for the relaxation parameter Θ_0 is chosen as

$$\Theta_0 := \frac{\Theta_{\{c \rightarrow 0\}} + \Theta_{\{c \rightarrow \infty\}}}{2}, \quad (7.8)$$

the arithmetic mean between the spatial and the temporal limits of Θ_{opt} . It has been observed in the non adaptive SDIRK2-NNWR scheme that the optimal relaxation parameter moves from one limit to the other in terms of the relation between Δt and Δx^2 as illustrated in figure 7.3. Initially, as Δt is unknown because there is not fixed time step in the time adaptive framework, we suggest to take an intermediate value between the two limits. Although other options were tried as the geometric mean between $\Theta_{\{c \rightarrow 0\}}$ and $\Theta_{\{c \rightarrow \infty\}}$, $\min(\Theta_{\{c \rightarrow 0\}}, \Theta_{\{c \rightarrow \infty\}})$ or $\max(\Theta_{\{c \rightarrow 0\}}, \Theta_{\{c \rightarrow \infty\}})$, the arithmetic mean was found to be the most efficient.

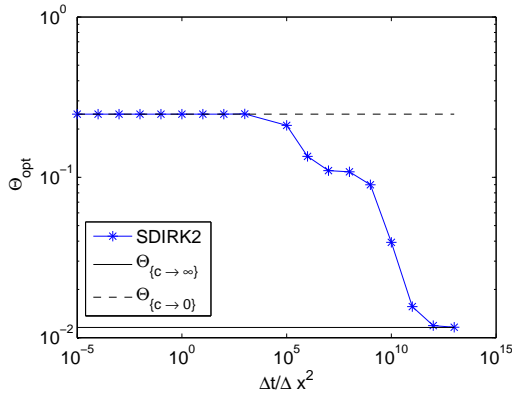


Figure 7.3: Water-steel optimal relaxation parameter Θ_{opt} as a function of the parameter $c := \Delta t / \Delta x^2$ for SDIRK2 in 1D. The constant lines $\Theta_{\{c \rightarrow \infty\}}$ and $\Theta_{\{c \rightarrow 0\}}$ represent the spatial and temporal asymptotics of Θ_{opt} in (6.6). $\Delta t = 1e-9, 1e-8, \dots, 1e8, 1e9$ and $\Delta x = 1/100$.

Secondly, in order to update the relaxation parameter after each iteration, we average the obtained variable step sizes getting the means $\bar{\Delta t}_1$ and $\bar{\Delta t}_2$ for each space-time subdomain $\Omega_1 \times [T_0, T_f]$ and $\Omega_2 \times [T_0, T_f]$. Once we have the values $\bar{\Delta t}_1$ and $\bar{\Delta t}_2$ we choose Θ using the same strategy we presented in paper III for the multirate SDIRK2-NNWR algorithm. Remember that we do not have a specific analysis for the optimal relaxation parameter in the multirate case, but we showed numerically in paper III that Θ_{opt} in (6.6) can be used as an estimate. This estimate was experimentally deduced from plotting the convergence rates with respect to the temporal ratio $\Delta t_1 / \Delta t_2$ using the relaxation parameters $\Theta_{opt}(\Delta t_1)$ and $\Theta_{opt}(\Delta t_2)$, see figure 7.4. There, we observe that one can use $\Theta_{opt}(\Delta t_2)$ when $\Delta t_1 < \Delta t_2$ and $\Theta_{opt}(\Delta t_1)$ when $\Delta t_1 > \Delta t_2$. This procedure is summarized in algorithm 6 which completes the time adaptive SDIRK2-NNWR approach in algorithm 5.

7.3 Numerical results

In this section we present numerical experiments to illustrate the behavior of the time adaptive SDIRK2-NNWR algorithm introduced above. All the results in this section have been produced by implementing algorithm 5 in Python using the one-dimensional FE discretization specified in section 4.1 and using as a initial condition the smooth function $g(x) = 900(-x^2 + 2x)$ at the interval $\Omega = \Omega_1 \cup \Omega_2 = [0, 1] \cup [1, 2]$.

Figure 7.5 shows the global error of the overall solution on Ω with respect to the

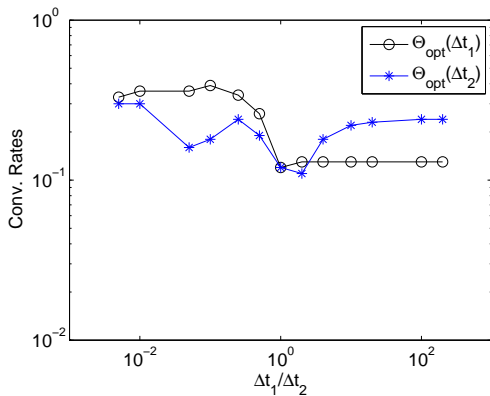


Figure 7.4: Convergence rates as a function of the temporal ratio $\Delta t_1/\Delta t_2$ for the air-water coupling for SDIRK2 in 1D. We plot the convergence rates in the multirate case ($\Delta t_1 \neq \Delta t_2$) using the relaxation parameters $\Theta_{opt}(\Delta t_1)$ and $\Theta_{opt}(\Delta t_2)$ in (6.6). $\Delta t_1/\Delta t_2 = 1e - 3/2e - 1, 2e - 3/2e - 1, 1e - 2/2e - 1, 2e - 2/2e - 1, 5e - 2/2e - 1, 1e - 1/2e - 1, 2e - 1/2e - 1, 2e - 1/1e - 1, 2e - 1/5e - 2, 2e - 1/2e - 2, 2e - 1/1e - 2, 2e - 1/2e - 3, 2e - 1/1e - 3$ and $\Delta x = 1/100$.

Algorithm 6 Algorithm to update the relaxation parameter for the time adaptive SDIRK2-NNWR scheme.

- 1: **procedure** UPDATERELAXATION($\tau_1, \tau_2, \alpha_1, \alpha_2, \lambda_1, \lambda_2, \Delta x$)
 - 2: $N_m \leftarrow \text{LENGTH}(\tau_m)$
 - 3: $\bar{\Delta t}_m \leftarrow (t_{N_m} - t_0)/N_m$ # arithmetic mean of the variable step sizes
 - 4: **if** $\bar{\Delta t}_1 < \bar{\Delta t}_2$ **then**
 - 5: $\Theta \leftarrow \Theta_{opt}(\alpha_1, \alpha_2, \lambda_1, \lambda_2, \Delta x, \bar{\Delta t}_2)$ # use eq. (6.6)
 - 6: **else**
 - 7: $\Theta \leftarrow \Theta_{opt}(\alpha_1, \alpha_2, \lambda_1, \lambda_2, \Delta x, \bar{\Delta t}_1)$ # use eq. (6.6)
 - return** Θ
-

Table 7.1: Physical properties of the materials. λ is the thermal conductivity, ρ the density, c_p the specific heat capacity and $\alpha = \rho c_p$.

Material	λ (W/mK)	ρ (kg/m ³)	c_p (J/kgK)	α (J/K m ³)
Air	0.0243	1.293	1005	1299.5
Water	0.58	999.7	4192.1	4.1908e6
Steel	48.9	7836	443	3471348

tolerance for the coupling of different materials. Physical properties of the materials are shown in table 7.1. The global error has been calculated with respect to a reference solution u_{ref} that has been computed using algorithm 5 for a very fine tolerance ($TOL = 1e - 12$ for the pairings air-steel and air-water and $TOL = 1e - 9$ for the pairing water-steel). One observes in figure 7.5 how the error decreases proportionally to the tolerance as expected in a time adaptive numerical method. Some deviations are observed for the smallest tolerances in the coupling air-water and water-steel. This could happen because the difference of the solution with respect to the reference at those points is not strong enough.

To complement these results, we have also plotted the global error with respect to the number of iterations in figure 7.6. One observes that the method does not need many iterations to achieve a really accurate solution. More specifically, five iterations are needed to get an accuracy of order $1e - 12$ for air-steel and $1e - 9$ for water-steel and ten iterations are needed to get an accuracy of order $1e - 12$ for the air-water coupling. But the most interesting aspect in figure 7.6 is the strong jump that the method performs at the first iteration. One can see that in the three material couplings presented here, the numerical solution has an error of order $1e - 5$ after the first iteration and from the second iteration, the error of the solution decreases slower. This illustrates the importance of choosing properly the relaxation parameter Θ beforehand. Remember that we suggested to use an arithmetic mean of both the spatial and temporal limits of the Θ_{opt} in (6.6). However, in view of these results one could think of increasing or decreasing the spatial resolution through Δx in order to force the numerical method into one of the two limits and use either $\Theta_{\{c \rightarrow 0\}}$ or $\Theta_{\{c \rightarrow \infty\}}$ as initial relaxation parameters. This alternative strategy might decrease the error after the first iteration even more than using the arithmetic mean of the limits.

Finally, we have included an illustration of the global error as a function of work in figure 7.7. Here we measure work with respect to the total number of step sizes computed a posteriori and as we have two space-time subdomains, $\Omega_1 \times [T_0, T_f]$ and $\Omega_2 \times [T_0, T_f]$, we get a curve for each subdomain. In order to get the relation between the number of step sizes and the global error, we measure those for a decreasing sequence of tolerances starting with $TOL = 1e - 4$ and ending with $TOL = 1e - 11$ for the couplings air-steel and

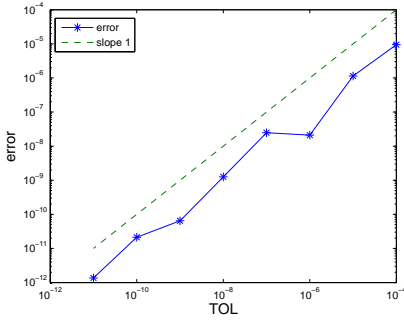
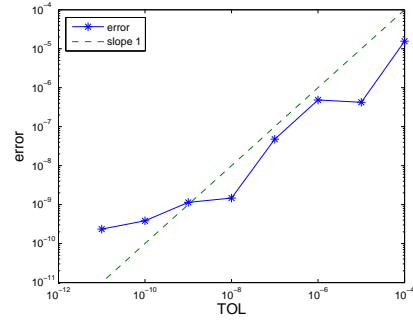
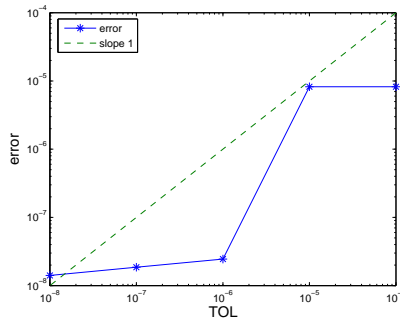
(a) *Air-steel*(b) *Air-water*(c) *Water-steel*

Figure 7.5: Global error as a function of the tolerance of the time adaptive SDIRK2-NNWR algorithm for the coupling of different materials. $\Delta x = 1/50$, $[T_0, T_f] = [0, 1]$, $TOL = 1e - 11, 1e - 10, \dots, 1e - 4$ for (a)-(b) and $TOL = 1e - 8, 1e - 7, \dots, 1e - 4$ for (c).

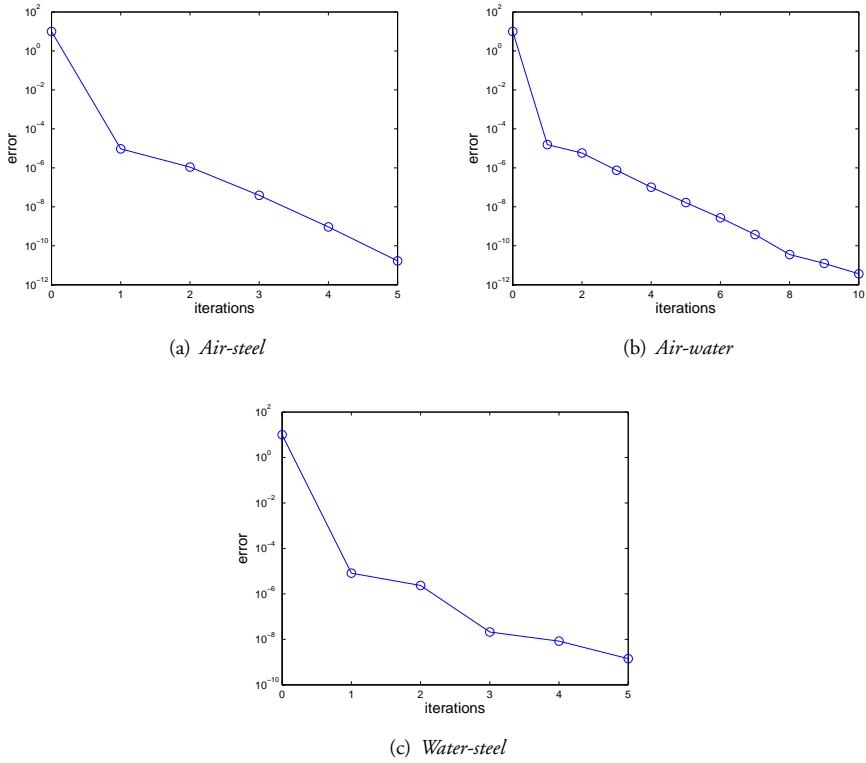
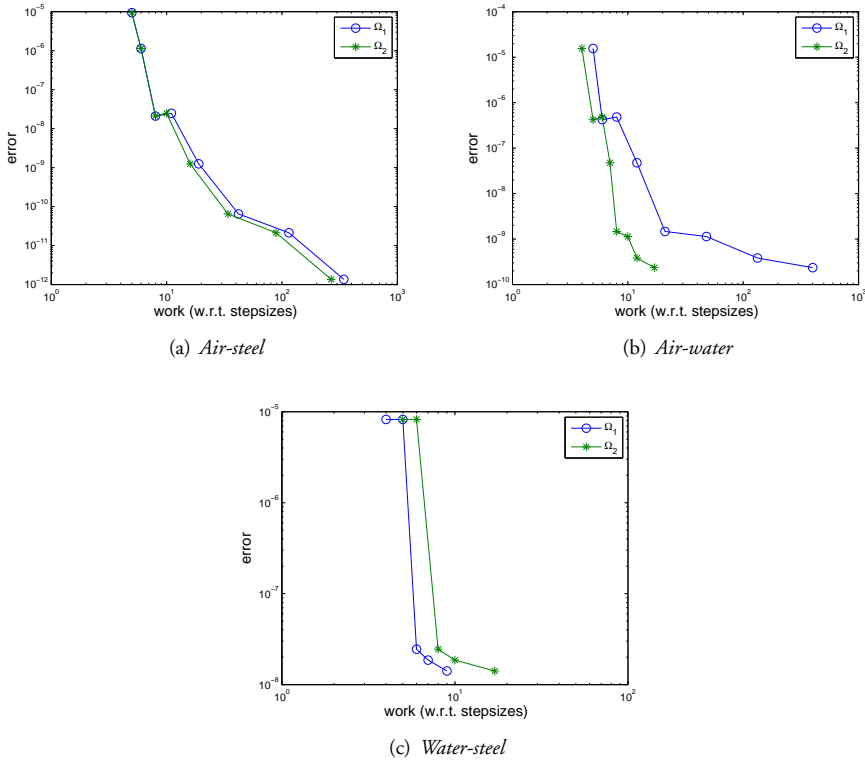


Figure 7.6: Global error as a function of the fixed point iterations of the time adaptive SDIRK2-NNWR algorithm for the coupling of different materials. $\Delta x = 1/50$, $[T_0, T_f] = [0, 1]$, $TOL = 1e - 12$ for (a)-(b) and $TOL = 1e - 9$ for (c).

Figure 7.7: Work vs error. $\Delta x = 1/50$

air-water and with $TOL = 1e - 8$ for the coupling water-steel. One observes that a very accurate solution on both subdomains is achieved in quite few step sizes illustrating the good performance of the time adaptive SDIRK2-NNWR method explained in previous sections. In the pairings air-steel and water-steel which are the ones having stronger jumps in the material coefficients across the interface we observe a good balance between the work performed on Ω_1 and Ω_2 . The example that shows a larger difference between subdomains is the air-water pairing where the air needs more step sizes than the water to achieve the same accuracy.

In conclusion, the numerical results just presented show that the time adaptive SDIRK2-NNWR algorithm introduced in this chapter is a more efficient method for the one-dimensional coupling of different combinations of materials than the multirate SDIRK2-NNWR algorithm explained in the previous chapter. The main advantage over the multirate method is that the step size controllers acting on the Dirichlet solvers allow each of

the fields to choose their own variable step sizes independent on each other. The aim of the time adaptive models is always to increase the size of the step sizes without increasing the error on the overall numerical solution. Therefore, a functional time adaptive method uses less steps than a fixed step size model gaining efficiency. Additionally, the time adaptive SDIRK2-NNWR algorithm is constructed in parallel and its implementation over the multirate SDIRK2-NNWR algorithm is very easy because only the Dirichlet solvers need to be modified as sketched in figure 7.1. Moreover, this work motivates further investigation like its natural extension to 2D or the search of alternatives for the initial relaxation parameter or the step size controller.

Chapter 8

Conclusions and Outlook

8.1 Conclusions

We have described three different partitioned approaches for the unsteady transmission problem with jumping coefficients that can be further extended for the simulation of thermal fluid-structure interaction.

Firstly, we have analyzed the convergence rates of the Dirichlet-Neumann iteration in terms of the material coefficients for the fully discrete coupled heat equations. There, we have derived an exact formula in 1D and an estimate in 2D for the convergence rates both in the FEM-FEM and FVM-FEM frameworks. Furthermore, the limits of the convergence rates when approaching the continuous case either in space or time were computed. On one hand, the spatial limit is $\delta := \lambda_1/\lambda_2$ and the temporal limit is $\gamma := \alpha_1/\alpha_2$ in the FEM-FEM case. On the other hand, in the FVM-FEM case, the rates tend to δ in the spatial limit but to zero in the temporal limit. In addition, numerical experiments show that the linear analysis is relevant for nonlinear thermal FSI problems with high aspect ratio and unstructured grids.

All these theoretical results have been confirmed by numerical results concluding that strong jumps in the coefficients of the coupled PDEs imply fast convergence. Conversely, the coupling iteration will be slow when the material coefficients are continuous over all the subdomains. For coupling of structures and compressible flows, the aspect ratio in the fluid has to be taken into account, since the convergence is proportional to it. With regards to the order comparison, we remark the importance of choosing a second order scheme to approximate the normal derivatives to preserve a second order global solution. The global error is much smaller using one-sided differences than forward differences to approximate the normal derivative.

Secondly, we have introduced a new high order multirate NNWR algorithm to solve two heterogeneous coupled heat equations in order to increase parallelization in time.

Moreover, we have studied the optimal relaxation parameter in terms of the material coefficients and the temporal and spatial resolutions Δt and Δx . To this end, we considered the coupling of two heat equations on two identical domains. We assumed structured spatial grids and conforming time grids on both subdomains to derive a formula for the optimal relaxation parameter Θ_{opt} in 1D using implicit Euler. Furthermore, we determined the limits of the optimal relaxation parameter when approaching the continuous case either in space ($\lambda_1 \lambda_2 / (\lambda_1 + \lambda_2)^2$) or time ($\alpha_1 \alpha_2 / (\alpha_1 + \alpha_2)^2$). The method using Θ_{opt} converges extremely fast, typically within two iterations. This was confirmed by numerical results, where we also demonstrated that the nonmultirate 1D case gives excellent estimates for the multirate 1D case and even for multirate and nonmultirate 2D examples using both implicit Euler and SDIRK2. In addition, we have shown that the NNWR method is a more efficient choice than the classical DNWR in the multirate case.

Thirdly, we have introduced a time adaptive version of the multirate NNWR method mentioned above. We inserted two different controllers in the Dirichlet solvers to build two independent time grids τ_1 and τ_2 increasing the efficiency of the algorithm. The numerical results show the advantages of the time adaptive method over the previous multirate approach.

8.2 Further work

There are a variety of future directions for this work. We present some of them here as possible goals.

- One future direction would be to extend the convergence analysis of the Dirichlet-Neumann iteration to higher dimensions (3D) or to generalize it to other time integration methods.
- One could also be interested in the convergence analysis of another coupling method like Robin-Neumann. However, after the highly efficient results obtained in the coupling between air and steel using the Dirichlet-Neumann iteration not much efficiency could be gained by finding an alternative sequential method. Nevertheless, this could be interesting in other FSI applications.
- One could also try to model a three-field problem where two different fluids are coupled through a common structure with a time-dependent method and estimate the convergence rates. A preliminary study of this problem was performed by Ivo Dravins in his master thesis [25] for the coupling of three elliptic problems.
- The numerical comparison performed in paper III between the NNWR and the DNWR algorithms hints that $\Theta = 1/2$ might not be the optimal relaxation parameter for the DNWR method when having strong jumps in the material coefficients for the fully discrete problem. Thus, performing an specific analysis to

find the optimal relaxation parameter of the DNWR algorithm is left for future research and it will allow us to perform a more accurate comparison between the two multirate methods.

- Another future direction would be to use both the multirate and time adaptive approaches explained in this thesis to simulate nonlinear thermal FSI test cases.
- Finally, many aspects of the time adaptive approach are left for further research. One could extend the approach to higher dimensions (2D and even 3D), investigate alternatives adding time step controllers on the Neumann problems as well, implement time adaptivity with respect to macrosteps or study the influence of the initial condition on the performance of the method.

Bibliography

- [1] C. Arévalo and G. Söderlind. Grid-independent construction of multistep methods. *J. Comp. Math.*, 35(5):670–690, 2017.
- [2] P. Argüelles et al. *European Aeronautics: A Vision for 2020*. Office for Official Publications of the European Communities, 2001.
- [3] S. Badia, A. Quaini, and A. Quarteroni. Modular vs. non-modular preconditioners for fluid-structure systems with large added-mass effect. *Comput. Methods Appl. Mech. and Engrg.*, 197:4216–4232, 2008.
- [4] Y. Bazilevs, V. Calo, Y. Zhang, and T. Hughes. Isogeometric fluid-structure interaction analysis with applications to arterial blood flow. *Comp. Mech.*, 38(4):310–322, 2006.
- [5] Y. Bazilevs, M. Hsu, Y. Zhang, W. Wang, X. Liang, T. Kvamsdal, R. Brekken, and J. Isaksen. A fully-coupled fluid-structure interaction simulation of cerebral aneurysms. *Comp. Mech.*, 46(1):3–16, 2010.
- [6] O. Bendiksen. A new approach to computational aeroelasticity. *AIAA Paper*, 91-09:1712–1727, 1991.
- [7] M. Benes, T. Krejčí, and J. Kruiš. A FETI-based mixed explicit-implicit multi-time-step method for parabolic problems. *J. Comp. Appl. Math.*, 333:247–265, 2018.
- [8] P. Birken. *Numerical Methods for the Unsteady Compressible Navier-Stokes Equations*. Habilitation Thesis. University of Kassel, 2012.
- [9] P. Birken, T. Gleim, D. Kuhl, and A. Meister. Fast Solvers for Unsteady Thermal Fluid Structure Interaction. *Int. J. Numer. Meth. Fluids*, 79(1):16–29, 2015.
- [10] P. Birken, K. Quint, S. Hartmann, and A. Meister. Choosing norms in adaptive FSI calculations. *PAMM*, 10:555–556, 2010.

- [11] P. Birken, K. Quint, S. Hartmann, and A. Meister. A time-adaptive fluid-structure interaction method for thermal coupling. *Comp. Vis. in Science*, 13(7):331–340, 2011.
- [12] R. Bishop, W. Price, and W. Yousheng. A general linear hydroelasticity theory of floating structures moving in a seaway. *Phil. Transactions of The Royal Society Bio. Sciences*, 316(1538):375–426, 1986.
- [13] D. Blom, V. Krupp, H. van Zuijlen, H. Klimach, S. Roller, and H. Bijl. *On parallel scalability aspects of strongly coupled partitioned fluid-structure-acoustics interaction*. VI International Conference on Computational Methods for Coupled Problems in Science and Engineering, COUPLED PROBLEMS 2015.
- [14] S. Bremicker-Trübelhorn and S. Ortleb. On Multirate GARK Schemes with Adaptive Micro Step Sizes for Fluid-Structure Interaction: Order Conditions and Preservation of the Geometric Conservation Law. *Aerospace*, 4(8), 2017.
- [15] D. Brown et al. *Applied Mathematics at the U.S. Department of Energy: Past, Present and a View to the Future*. tech. rep., Office of Science, U.S. Department of Energy, 2008.
- [16] J. Buchlin. Convective Heat Transfer and Infrared Thermography. *J. Appl. Fluid Mech.*, 3:55–62, 2010.
- [17] C. Carmody, G. Burriesci, I. Howard, and E. Patterson. An approach to the simulation of fluid-structure interaction in the aortic valve. *J. Biomech.*, 39(1):158–169, 2006.
- [18] P. Causin, J. Gerbeau, and F. Nobile. Added-mass effect in the design of partitioned algorithms for fluid-structure problems. *Comput. Methods Appl. Mech. Engrg.*, 194:4506–4527, 2005.
- [19] R. Cheng, Y. Lai, and K. Chandran. Three-dimensional fluid-structure interaction simulation of bileaflet mechanical heart valve flow dynamics. *Annals Biomed. Engrg.*, 32(11):1471–1483, 2004.
- [20] A. Cristiano, I. Malossi, P. Blanco, S. Deparis, and A. Quarteroni. Algorithms for the partitioned solution of weakly coupled fluid models for cardiovascular flows. *Numer. Methods Biomed. Engrg.*, 27(12):2035–2057, 2011.
- [21] G. Davis and O. Bendiksen. Transonic Panel Flutter. *AIAA Paper*, 93-1476, 1993.
- [22] J. De Hart, G. Peters, P. Schreurs, and F. Baaijens. A three-dimensional computational analysis of fluid-structure interaction in the aortic valve. *J. Biomech.*, 36(1):103–112, 2003.

- [23] S. Deparis, M. Fernández, and L. Formaggia. Acceleration of a fixed point algorithm for fluid-structure interaction using transpiration conditions. *M2AN*, 37(4):601–616, 2003.
- [24] E. Dowell, J. Thomas, and K. Hall. Transonic limit cycle oscillation analysis using reduced order aerodynamic models. *J. Fluids Struct.*, 19:17–27, 2004.
- [25] I. Dravins. *Coupled implementations of the Dirichlet-Neumann iteration. The three-field case: Methods and analyses*. Master thesis, Lund University, Sweden, 2018.
- [26] L. Evans. *Partial Differential Equations*. American Mathematical Society, 2010.
- [27] C. Farhat. CFD-based Nonlinear Computational Aeroelasticity. in *Encyclopedia of Computational Mechanics*, pages 459–480, ch. 13, 2004.
- [28] C. Farhat, M. Lesoinne, and P. Le Tallec. Load and motion transfer algorithms for fluid/structure interaction problems with non-matching discrete interfaces: Momentum and energy conservation, optimal discretization and application to aeroelasticity. *Comput. Methods Appl. Mech. Engrg.*, 157(1-2):95–114, 1998.
- [29] C. Farhat, K. van der Zee, and P. Geuzaine. Provably second-order time-accurate loosely-coupled solution algorithms for transient nonlinear computational aeroelasticity. *Comput. Methods Appl. Mech. Engrg.*, 195:1973–2001, 2006.
- [30] C. Fonseca and J. Petronilho. Explicit inverses of some tridiagonal matrices. *Linear Algebra Appl.*, 325(1-3):7–21, 2001.
- [31] S. Foundation. *The Space Report*. The Authoritative Guide to Global Space Activity, 2017.
- [32] M. Gander, L. Halpern, C. Japhet, and V. Martin. *Advection diffusion problems with pure advection approximation in subregions*. in Domain Decomposition Methods in Science and Engineering XVI. Lecture Notes in Computer Science and Engineering, vol 50, pp. 239-246. Springer, Berlin., 2007.
- [33] M. Gander, L. Halpern, and F. Nataf. Optimized Schwarz waveform relaxation for the one dimensional wave equation. *SIAM J. Numer. Anal.*, 41(5):1643–1681, 2003.
- [34] M. Gander, F. Kwok, and B. Mandal. Dirichlet-Neumann and Neumann-Neumann waveform relaxation algorithms for parabolic problems. *ETNA*, 45:424–456, 2016.
- [35] M. Gander and A. Stuart. Space-time continuous analysis of waveform relaxation for the heat equation. *SIAM J. Sci. Comput.*, 19(6):2014–2031, 1998.

- [36] E. Giladi and H. Keller. Space-time domain decomposition for parabolic problems. *Numer. Math.*, 93:279–313, 2002.
- [37] M. Giles. Stability Analysis of Numerical Interface Conditions in Fluid-Structure Thermal Analysis. *Int. J. Numer. Methods Fluids*, 25:421–436, 1997.
- [38] T. Gillebaart, D. Blom, A. van Zuijlen, and H. Bijl. Time consistent fluid structure interaction on collocated grids for incompressible flow. *Comput. Methods Appl. Mech. Engrg.*, 298:159–182, 2016.
- [39] T. Gleim, B. Schröder, and D. Kuhl. Nonlinear thermo-electromagnetic analysis of inductive heating processes. *Archive Appl. Mech.*, 85:1055–1073, 2015.
- [40] H. Guillard and C. Farhat. On the significance of the geometric conservation law for flow computations on moving meshes. *Comput. Methods Appl. Mech. Engrg.*, 190:1467–1482, 2000.
- [41] E. Hairer, S. Norsett, and G. Wanner. *Solving Ordinary Differential Equations I - Nonstiff Problems*. Springer-Verlag, 1993.
- [42] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II - Stiff and Differential-Algebraic Problems*. Springer-Verlag, 1996.
- [43] U. Heck, U. Fritsching, and B. K. Fluid flow and heat transfer in gas jet quenching of a cylinder. *Int. J. Numer. Methods Heat Fluid Flow*, 11:36–49, 2001.
- [44] W. Henshaw and K. Chand. A composite grid solver for conjugate heat transfer in fluid-structure systems. *J. Comput. Phys.*, 228:2708–3741, 2009.
- [45] M. Hinderks and R. Radespiel. Investigation of Hypersonic Gap Flow of a Reentry Nosecap with Consideration of Fluid Structure Interaction. *AIAA Paper*, 6:2708–3741, 2006.
- [46] T. Hoang. *Space-time domain decomposition methods for mixed formulations of flow and transport problems in porous media*. PhD thesis, Université Pierre et Marie Curie, Paris 6, France, 2013.
- [47] R. Kamakoti and W. Shyy. Fluid-structure interaction for aeroelastic applications. *Progress in Aerospace Sciences*, 40(8):535–558, 2004.
- [48] D. Kowollik, P. Horst, and M. Haupt. Fluid-structure interaction analysis applied to thermal barrier coated cooled rocket thrust chambers with subsequent local investigation of delamination phenomena. *Progress in Propulsion Physics*, 4:617–636, 2013.

- [49] D. Kowollik, V. Tini, S. Reese, and M. Haupt. 3D fluid-structure interaction analysis of a typical liquid rocket engine cycle based on a novel viscoplastic damage model. *Int. J. Numer. Methods Engrg.*, 94:1165–1190, 2013.
- [50] F. Kwok. *Neumann-Neumann waveform relaxation for the time-dependent heat equation*, volume 98. in Domain Decomposition in Science and Engineering XXI, J. Erhel, M.J. Gander, L. Halpern, G. Pichot, T. Sassi and O.B. Widlund, eds. Lect. Notes Comput. Sci. Eng., 2014.
- [51] P. Le Tallec and J. Mouro. Fluid structure interaction with large structural displacements. *Comput. Methods Appl. Mech. Engrg.*, 190:3039–3067, 2001.
- [52] E. Lelarasmee, A. Ruehli, and A. Sangiovanni-Vincentelli. The waveform relaxation method for time-domain analysis of large scale integrated circuits. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.*, 1(3):131–145, 1982.
- [53] L. Li, S. Sherwin, and P. Bearman. A moving frame of reference algorithm for fluid/structure interaction of rotating and translating bodies. *Int. J. Numer. Methods Fluids*, pages 187–206, 2002.
- [54] J. Lötters, W. Olthuis, P. Veltink, and P. Bergveld. The mechanical properties of the rubber elastic polymer polydimethylsiloxane for sensor applications. *J. Micromech. Microengng.*, 7(3), 1997.
- [55] R. Massjung. Discrete conservation and coupling strategies in nonlinear aeroelasticity. *Comput. Methods Appl. Mech. Engrg.*, 196:91–102, 2006.
- [56] H. Matthies, R. Niekamp, and J. Steindorf. Algorithms for strong coupling procedures. *Comput. Methods Appl. Mech. Engrg.*, 195:2028–2049, 2006.
- [57] M. Mehl, B. Uekermann, H. Bijl, D. Blom, B. Gatzhammer, and A. van Zuijlen. Parallel coupling numerics for partitioned fluid-structure interaction simulations. *Comput. Math. Appl.*, 71(4):869–891, 2016.
- [58] R. Mehta. Numerical Computation of Heat Transfer on Reentry Capsules at Mach 5. *AIAA-Paper*, 178, 2005.
- [59] C. Meyer. *Matrix Analysis and Applied Linear Algebra*. SIAM, 2000.
- [60] A. Monge and P. Birken. *Convergence analysis of coupling iterations for the unsteady transmission problem with mixed discretizations*. VII European Congress on Computational Methods in Applied Sciences and Engineering, Vol. 1, 2016.
- [61] A. Monge and P. Birken. *Convergence analysis of the Dirichlet-Neumann iteration for finite element discretizations*. PAMM 2016.

- [62] A. Monge and P. Birken. *Convergence speed of coupling iterations for the unsteady transmission problem*. VI International Conference on Computational Methods for Coupled Problems in Science and Engineering, COUPLED PROBLEMS 2015.
- [63] D. Newman and G. Karniadakis. A direct numerical simulation study of flow past a freely vibrating cable. *J. Fluids Mech.*, 344:95–136, 1997.
- [64] N. Nitin and M. Karwe. Numerical simulation and experimental investigation of conjugate heat transfer between a turbulent hot air jet impinging on a cookie-shaped object. *J. Food Sci.*, pages 59–65, 2006.
- [65] B. Ong and B. Mandal. Pipeline implementations of Neumann-Neumann and Dirichlet-Neumann waveform relaxation methods. *Numer. Algor.*, 78:1–20, 2018.
- [66] PLDSpace. Arion 1. <https://pldspace.com/new/arion-1/>. Accessed: 2018-08-05.
- [67] PLDSpace. Arion 2. <https://pldspace.com/new/arion-2/>. Accessed: 2018-08-05.
- [68] PLDSpace. Regeneratively Cooled Engine. <https://www.youtube.com/watch?v=xRm4zaoZjCI>. Accessed: 2018-08-05.
- [69] A. Quarteroni and A. Valli. *Domain Decomposition Methods for Partial Differential Equations*. Oxford Science Publications, 1999.
- [70] M. Reuter and J. Hill. An efficient, block-by-block algorithm for inverting a block tridiagonal, nearly block toeplitz matrix. *Comp. Sci. Disc.*, 5, 2012.
- [71] B. R uth, B. Uekermann, M. Mehl, and H. Bungartz. *Time stepping algorithms for partitioned multi-scale multi-physics in preCICE*. VI and VII European Conference on Computational Mechanics and Fluid Dynamics, ECCM-ECFD 2018.
- [72] D. Schwamborn, T. Gerhold, and R. Kessler. DLR-TAU Code - an Overview. In *1st ONERA/DLR Aerospace Symposium, Paris*, pages 2–10, 1999.
- [73] C. Scotti and E. Finol. Compliant biomechanics of abdominal aortic aneurysms: a fluid-structure interaction study. *Comput. Struc.*, 85:1097–1130, 2007.
- [74] D. Shiels, A. Leonard, and A. Roshko. Flow-induced vibration of a circular cylinder at limiting structural parameters. *J. Fluids Struc.*, 15(1):3–21, 2001.
- [75] J. Sigrist and S. Garreau. Dynamic analysis of fluid-structure interaction problems with modal methods using pressure-based fluid finite elements. *Finite Elements in Analysis and Design*, 43(4):287–300, 2007.

- [76] G. Söderlind. Automatic Control and Adaptive Time-Stepping. *Numer. Algor.*, 31:281–310, 2002.
- [77] G. Söderlind. Digital filters in adaptive time-stepping. *ACM Trans. Math. Software*, 29:1–26, 2003.
- [78] G. Söderlind and L. Wang. Adaptive time-stepping and computational stability. *J. Comp. Appl. Math.*, 185:225–243, 2006.
- [79] SpaceX. Background on Tonight’s Launch. <http://www.spacex.com/news/2015/12/21/background-tonights-launch>. Accessed: 2018-08-05.
- [80] SpaceX. CRS-8 Launch and Landing. <http://www.spacex.com/news/2016/04/09/crs-8-launch-and-landing>. Accessed: 2018-08-05.
- [81] SpaceX. Falcon 9 First Stage Landing. <https://www.youtube.com/watch?v=ZCBE8oc0kAQ>. Accessed: 2018-08-05.
- [82] SpaceX. First Stage Landing on Droneship. https://www.youtube.com/watch?v=sYmQQn_ZSys. Accessed: 2018-08-05.
- [83] SpaceX. Reusability: The Key to Making Human Life Multi-Planetary. <http://www.spacex.com/news/2013/03/31/reusability-key-making-human-life-multi-planetary>. Accessed: 2018-08-05.
- [84] SpaceX. The Why and How of Landing Rockets. <http://www.spacex.com/news/2015/06/24/why-and-how-landing-rockets>. Accessed: 2018-08-05.
- [85] SpaceX. World’s First Reflight of an Orbital Class Rocket. <https://www.youtube.com/watch?v=xsZSXav4wI8&feature=youtu.be>. Accessed: 2018-08-05.
- [86] K. Steinhoff, U. Weidig, and N. Saba. Investigation of plastic forming under the influence of locally and temporally variable temperature and stress states. In: Steinhoff, K., Maier, H., Biermann, D. (eds.) *Functionally Graded Materials in Industrial Mass Production. Verlag Wissenschaftliche Scripten, Auerbach*, 85:33–52, 2009.
- [87] G. Strang. *Introduction to Applied Mathematics*. Wellesley-Cambridge Press, 1986.
- [88] P. Stratton, I. Shedletsky, and M. Lee. Gas Quenching with Helium. *Solid State Phenomena*, 118:221–226, 2006.

- [89] T. Tezduyar, M. Schwaab, and S. Sathe. Sequentially-coupled arterial fluid-structure interaction (scafisi) technique. *Comput. Methods Appl. Mech. Engrg.*, 2008.
- [90] R. Torii, M. Oshima, T. Kobayashi, K. Takagi, and T. T.E. Fluid-structure interaction modeling of aneurysmal conditions with high and normal blood pressures. *Comp. Mech.*, 38(4):482–490, 2006.
- [91] A. Toselli and O. Widlund. *Domain Decomposition Methods - Algorithms and Theory*. Springer, 2004.
- [92] E. van Brummelen. Added mass effects of compressible and incompressible flows in fluid-structure interaction. *J. Appl. Mech.*, 76(2), 2009.
- [93] E. van Brummelen. Partitioned iterative solution methods for fluid-structure interaction. *Int. J. Numer. Methods Fluids*, 65(1-3):3–27, 2011.
- [94] E. van Driest. National Advisory Committee for Aeronautics (NACA) - investigation of laminar boundary layer in compressible fluids using the crocco method. 1952.
- [95] A. van Zuijlen and H. Bijl. Implicit and explicit higher order time integration schemes for structural dynamics and fluid-structure interaction computations. *Comput. Struc.*, 83:93–105, 2005.
- [96] A. van Zuijlen, A. de Boer, and H. Bijl. Higher-order time integration through smooth mesh deformation for 3d fluid-structure interaction simulations. *J. Comp. Phys.*, 224:414–430, 2007.
- [97] U. Weidig, N. Saba, and K. Steinhoff. Massivumformprodukte mit funktional gradierten Eigenschaften durch eine differenzielle thermo-mechanische Prozessführung. *WT-Online*, pages 745–752, 2007.
- [98] K. Willcox, J. Paduano, and J. Peraire. Low order aerodynamic models for aeroelastic control of turbomachines. *40th AIAA ASME ASCE AHS ASC structures, structural dynamics and materials conference, St Louis, MO*, 1999.
- [99] F. Zhang. *The Schur complement and its applications*. Springer, 2005.
- [100] H. Zhang, X. Zhang, J. Shanhong, G. Yanhu, G. Ledezma, N. Elabbasi, and H. deCougny. Recent development of fluid-structure interaction capabilities in the adina system. *Comput. Struc.*, 81(8-11):1071–1085, 2003.

Paper I

Numerical Methods for Unsteady Thermal Fluid Structure Interaction

Philipp Birken and Azahar Monge

Abstract. We discuss thermal fluid structure interaction processes, where a simulation of the time dependent temperature field is of interest. Thereby, we consider partitioned coupling schemes with a Dirichlet-Neumann method. We present an analysis of the method on a model problem of discretized coupled linear heat equations. This shows that for large quotients in the heat conductivities, the convergence rate will be very small. The time dependency makes the use of time adaptive implicit methods imperative. This gives rise to the question, how accurate the appearing nonlinear systems should be solved, which is discussed in detail for both the nonlinear and linear case. The efficiency of the resulting method is demonstrated using realistic test cases.

Keywords. Thermal Fluid Structure Interaction, Conjugate Heat Transfer, Dirichlet-Neumann method, Time adaptivity, Termination criteria.

AMS classification. 65F10, 65L04, 65M22, 74F04, 74F10.

1 Introduction

Fluid structure interaction occurs when a deformable or moving structure interacts with a surrounding or internal fluid flow. Our specific field of interest is thermal interaction between fluids and structures, also called conjugate heat transfer. Examples for thermal fluid structure interaction are cooling of gas-turbine blades, thermal anti-icing systems of airplanes [12], supersonic reentry of vehicles from space [31, 25], gas quenching, which is an industrial heat treatment of metal workpieces [23, 40] or the cooling of rocket nozzles [27, 26]. These problems are usually too complex to solve them analytically, and therefore, numerical simulations of conjugate heat transfer are essential in many applications.

The efficient numerical simulation of fluid structure interaction (FSI) models is one of the important current challenges in scientific computing as stated in [11]:

“The issue of coupling models of different events at different scales and governed by different physical laws is largely wide open and represents an enormously challenging area for future research.”

In this article, we focus on the coupling between air and an alloy. When being cooled or heated, the alloy experiences thermomechanical effects that change its internal structure. A first example are steel forging processes. One possibility is to use

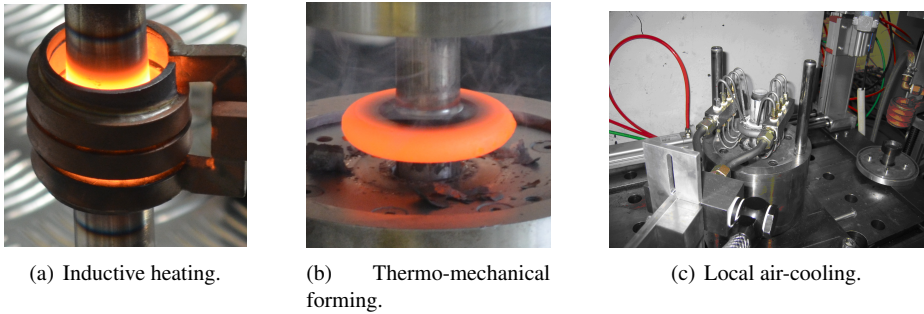


Figure 1. Gas quenching. Left and center picture: Institute of Mechanics and Dynamics, University of Kassel. Right picture: Institute of Metal Forming Technology, University of Kassel.

cold high pressured air as a cooling medium (see Figure 1). Knowledge about the time dependent temperature field is imperative to predict where martensite will be in a finished steel part. This allows to predict material properties generated by the forging process.

Another example is the cooling of rocket thrust chambers. The temperature achieved in the turbines has increased over the years due to progress in the building materials and therefore, advanced cooling methods are needed. This is both to avoid critical damage to the rocket nozzle when in use and to develop reusable rocket stages. An important case is the Ariane 5, see Figure 2a, which is used to deliver payloads into the geostationary transfer orbit (GTO) or the low Earth orbit (LEO). The first stage rocket engine for the Ariane 5 is the Vulcain 2, see figure 2b.

The first stage rocket engine is only used during the launch of the rocket. Therefore, recovering and reusing the first stage will reduce the cost of space access and the environment impact. Related to this, the company SpaceX is developing a set of new technologies for an orbital launch system that may be reused many times in a manner similar to the reusability of aircraft. The first controlled vertical splashdown of an orbital rocket stage on the ocean surface was achieved in April 2014. The next two flights in January and April 2015 attempted to land the returning first stage on a floating platform. Both of them were guided accurately to the target, but they did not succeed in landing vertically on the floating platform and were destroyed [3]. Finally, the first vertical landing was achieved on December 21, 2015, when the first stage of Falcon 9 Flight 20 successfully landed on solid ground [1]. On April 8, 2016, Flight 23 achieved the first soft landing on a drone ship in the Atlantic Ocean [2], see Figure 3.

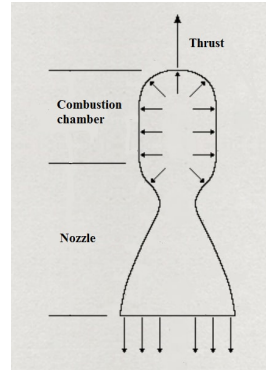
In view of these achievements, it is of interest to simulate several cycles of the combustion-cooling process of the thrust chamber. This will tell how many times the engine can be used without damaging the structure. More details about this can be found in [27, 26].



(a) Ariane 5 on the launch pad.



(b) The Vulcain engine in a museum.



(c) Sketch of the rocket thrust chamber.

Figure 2. Cooling of rocket thrust chambers. Left picture: DLR German Aerospace Center, CC BY 2.0. Center picture: Pline, CC BY-SA 3.0



(a) Unsuccessful vertical landing attempt.



(b) First stage landing vertically on solid ground in December 2015.



(c) First stage landed on autonomous droneship in April 2016.

Figure 3. SpaceX reusable launch system development program. SpaceX Photos, CC0 1.0.

Figure 2c shows a combustion chamber with the nozzle, through which hot gas can escape. The nozzle is delimited by a structure that can be damaged due to the high temperature of the gas flowing inside. In order to avoid this, a cooling fluid flows through small channels contained inside the structure. This results in a system with two thermal interactions between fluids and structures. On one hand, between the hot gas coming out from the combustion chamber and the structure recovering the nozzle. On the other hand, between the cooling fluid and the structure. A sketch of the coupling

surfaces can be consulted in Figure 4 where $\Gamma_{s,cf}$ corresponds to the interface between the structure and the cooling fluid and $\Gamma_{s,hg}$ to the interface between the structure and the hot gas.

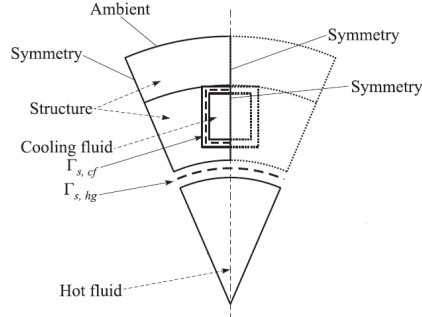


Figure 4. Sketch of the coupling surfaces. Figure taken from [27].

With regards to the space discretization, the use of the finite element method (FEM) is ubiquitous for the structure. For the fluid, typically finite volume methods (FVM) are used. However, there are some approaches to use finite elements for both problems, in particular when using a monolithic method [4].

1.1 Partitioned Coupling Methods

To simulate FSI problems there exist two main methods: monolithic and partitioned ones. In the monolithic method, a new code is tailored for the coupled equations, whereas the partitioned approach allows to reuse existing software for each sub-problem. The coupling is done by a master program which calls interface functions of the other codes [16]. If the data transfer between the subsolvers is done only once per time step, we are using a loosely coupled scheme [17]. However, for stability reasons, often a strongly coupled scheme needs to be used [28]. In this case the data exchange at every time step is repeated until a convergence criterion is satisfied. Here, we focus on partitioned methods.

At the boundary, one imposes that the temperature and the heat flux have to be continuous across the interface. To obtain such a solution, a Dirichlet-Neumann iteration can be employed. This consists of solving the fluid problem with Dirichlet boundary conditions at the interface and then the structure problem with Neumann boundary conditions at the interface, resulting in a fixed point iteration. Figure 5 illustrates this. This iteration is a basic method in both domain decomposition and fluid structure interaction.

In the domain decomposition context, it has two main problems, namely slow convergence and the need for an implementation using a red-black colouring. The slow

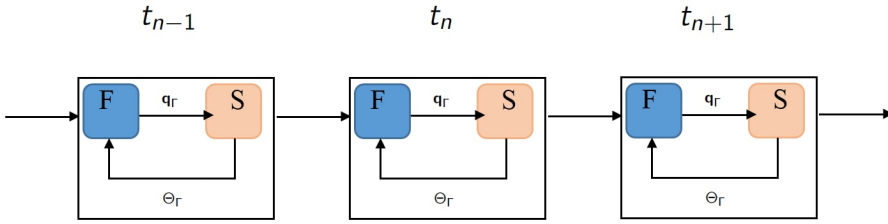


Figure 5. Illustration of the FSI solver.

convergence can be improved using a relaxation procedure [34]. In fluid structure interaction, there are only two domains, coupled along an interface, making the application straight forward. However, the convergence rate is not great for the coupling between a compressible fluid and a structure [14], which is why a lot of effort goes into convergence acceleration. On the other hand, the Dirichlet-Neumann iteration was reported to be a very fast solver for thermal fluid structure interaction [8].

As mentioned before, we study transient processes and therefore a numerical method needs to be chosen for the time discretization. In [29], the implicit midpoint rule is used in a monolithic scheme to analyze energy conservation of an aeroelasticity problem. Already in [5, 13], it is suggested to use an explicit high order Runge-Kutta scheme for both subproblems with data exchange at each stage. However, the resulting scheme has limited time steps due to the explicit nature of the method. The order of coupling schemes on moving meshes is analyzed in [22], but only first order convergence is proved for p -th order schemes. Moreover, higher order implicit Runge-Kutta schemes on moving meshes are analyzed in [43] (in 1D) and in [44] (in 3D). There, so called explicit first stage, singly diagonally implicit Runge-Kutta schemes (ESDIRK) are employed and higher order in time is proved by numerical results. The master program of the FSI procedure can be extended to SDIRK methods and furthermore, time adaptivity can be added into this framework as explained in [9, 10]. Another interesting alternative is subcycling, where in the subsolvers, different time step sizes are chosen [21, 37]. So far, this has not been brought together with time adaptive methods.

Summarizing, a partitioned method for unsteady thermal FSI involves two subsolvers (fluid and structure) and two embedded iterations. There exists an outer loop corresponding to the time integration. Then, at each timestep, a subiteration (the Dirichlet-Neumann method) couples the subsolvers, which have another nonlinear iteration so provide solutions to the subproblems.

An outline of the article now follows. In section 2, we describe the model and discretization, as well as the coupling conditions and the partitioned solution method, the Dirichlet-Neumann iteration. A model problem, namely two coupled discretized heat equations, is presented in section 3. Our analysis to determine the convergence rate of the Dirichlet-Neumann method for the model problem can be found in section

4. Numerical results are included to illustrate the theoretical analysis. In section 5, we describe how a fast solver is obtained using time adaptivity, followed by an in depth analysis of how tolerances in the various nested iterations should be chosen to avoid oversolving while giving the desired accuracy in section 6. Numerical results to demonstrate the efficiency of the obtained method are presented in section 7.

2 Treating Thermal FSI Problems

The basic setting we are in is that on a domain $\Omega_1 \subset \mathbb{R}^d$ where d corresponds to the spatial dimension, the physics is described by a fluid model, whereas on a domain $\Omega_2 \subset \mathbb{R}^d$, a different model describing the structure is used. The two domains are almost disjoint in that they are connected via an interface. The part of the interface where the fluid and the structure are supposed to interact is called the coupling interface $\Gamma \subset \partial\Omega_1 \cup \partial\Omega_2$. Note that Γ might be a true subset of the intersection, because the structure could be insulated. At the interface Γ , coupling conditions are prescribed that model the interaction between fluid and structure. For the thermal coupling problem, these conditions are that temperature and the normal component of the heat flux are continuous across the interface.

2.1 Fluid Model

We model the fluid using the time dependent Reynolds Averaged Navier-Stokes equations (URANS), which are a second order system of conservation laws (mass, momentum, energy) modeling turbulent compressible flow. We consider the two dimensional case, written in conservative variables density ρ , momentum $\mathbf{m} = \rho\mathbf{v}$ and energy per unit volume ρE , where a $\tilde{\cdot}$ denotes the Favre average and the overbar the ensemble average [6]:

$$\begin{aligned} \partial_t \bar{\rho} + \nabla \cdot \bar{\rho} \tilde{\mathbf{v}} &= 0, \\ \partial_t \bar{\rho} \tilde{\mathbf{v}} + \sum_{j=1}^2 \partial_{x_j} (\bar{\rho} \tilde{v}_i \tilde{v}_j) &= -\partial_{x_j} \bar{\rho} \delta_{ij} + \frac{1}{Re} \sum_{j=1}^2 \partial_{x_j} (\tilde{S}_{ij} + S_{ij}^R), \quad i = 1, 2 \quad (2.1) \\ \partial_t \bar{\rho} \tilde{E} + \nabla \cdot (\bar{\rho} \tilde{H} \tilde{v}_j) &= \sum_{j=1}^2 \partial_{x_j} \left(\overline{\left(\frac{1}{Re} S_{ij} - S_{ij}^R \right) v_i} - \bar{\rho} \tilde{v}_j'' + \widetilde{S_{ij} v_i''} - \bar{\rho} \tilde{v}_j'' k + \frac{\overline{W_j}}{RePr} \right). \end{aligned}$$

The Reynolds stresses

$$S_{ij}^R = -\overline{\rho v_i'' v_j''}$$

and the turbulent energy

$$k = \frac{1}{2} \sum_{j=1}^d \overline{v_j' v_j'}$$

are modelled using the Spallart-Allmaras model [39]. Furthermore, $\mathbf{q}_f = (q_1, q_2)^T$ represents the heat flux and $\mathbf{S} = (S_{ij})_{i,j=1,2}$ the viscous shear stress tensor. As the equations are dimensionless, the Reynolds number Re and the Prandtl number Pr appear. The system is closed by the equation of state for the pressure $p = (\gamma - 1)\rho e$, the Sutherland law representing the correlation between temperature and viscosity, as well as the Stokes hypothesis. Additionally, we prescribe appropriate boundary conditions at the boundary of Ω_1 except for Γ , where we have the coupling conditions. In the Dirichlet-Neumann coupling, a temperature value is enforced at Γ .

2.2 Structure Model

Regarding the structure model, we will consider heat conduction only. Thus, we have the nonlinear heat equation for the structure temperature Θ

$$\rho(\mathbf{x})c_p(\Theta)\frac{d}{dt}\Theta(\mathbf{x}, t) = -\nabla \cdot \mathbf{q}(\mathbf{x}, t), \quad (2.2)$$

where

$$\mathbf{q}_s(\mathbf{x}, t) = -\lambda(\Theta)\nabla\Theta(\mathbf{x}, t)$$

denotes the heat flux vector. For alloys, the specific heat capacity c_p and heat conductivity λ are temperature-dependent and highly nonlinear. How to model thermomechanical effects is subject of a lot of current research with important questions being how to relate the microstructure to macroscopical models.

As an example, an empirical model for the steel 51CrV4 was suggested in [35]. This was obtained simply by doing measurements and then a least squares fit to a chosen curve. The coefficient functions are then

$$\lambda(\Theta) = 40.1 + 0.05\Theta - 0.0001\Theta^2 + 4.9 \cdot 10^{-8}\Theta^3 \quad (2.3)$$

and

$$c_p(\Theta) = -10 \ln \left(\frac{e^{-c_{p1}(\Theta)/10} + e^{-c_{p2}(\Theta)/10}}{2} \right) \quad (2.4)$$

with

$$c_{p1}(\Theta) = 34.2e^{0.0026\Theta} + 421.15 \quad (2.5)$$

and

$$c_{p2}(\Theta) = 956.5e^{-0.012(\Theta-900)} + 0.45\Theta. \quad (2.6)$$

For the mass density one has $\rho = 7836 \text{ kg/m}^3$.

Finally, on the boundary, we have Neumann conditions $\mathbf{q}_s(\mathbf{x}, t) \cdot \mathbf{n}(\mathbf{x}) = q_b(\mathbf{x}, t)$.

2.3 Coupling Conditions

As mentioned before at the beginning of this section, the coupling conditions are that temperature and the normal component of the heat flux are continuous across the interface, i.e;

$$T(\mathbf{x}, t) = \Theta(\mathbf{x}, t), \quad \mathbf{x} \in \Gamma, \quad (2.7)$$

where T is the fluid temperature and Θ the structure temperature and

$$\mathbf{q}_f(\mathbf{x}, t) \cdot \mathbf{n}(x) = \mathbf{q}_s(\mathbf{x}, t) \cdot \mathbf{n}(x), \quad \mathbf{x} \in \Gamma. \quad (2.8)$$

2.4 Discretization in Space

Following the partitioned coupling approach, we discretize the two models separately in space. For the fluid, we use a finite volume method, leading to

$$\frac{d}{dt} \mathbf{u} + \mathbf{h}(\mathbf{u}, \Theta_\Gamma) = \mathbf{0}, \quad (2.9)$$

where $\mathbf{h}(\mathbf{u}, \Theta_\Gamma)$ represents the spatial discretization and its dependence on the temperatures on the discrete interface to the structure, here denoted by Θ_Γ .

Regarding structural mechanics, the use of finite element methods is ubiquitous. Therefore, we will also follow that approach here, leading to the nonlinear equation for all unknowns on Ω_2 :

$$\mathbf{M}(\Theta) \frac{d}{dt} \Theta + \mathbf{A}(\Theta) \Theta = \mathbf{q}_b^f + \mathbf{q}_b^\Gamma(\mathbf{u}). \quad (2.10)$$

Here, \mathbf{M} is the mass matrix, also called heat capacity matrix for this problem and \mathbf{A} is the heat conductivity matrix. The vector Θ consists of all discrete temperature unknowns and $\mathbf{q}_b^\Gamma(\mathbf{u})$ is the discrete heat flux vector on the coupling interface to the fluid, whereas \mathbf{q}_b^f corresponds to boundary heat fluxes independent of the fluid, for example at insulated boundaries.

2.5 Time Discretization

For sake of completeness, we now write down a discretization using the implicit Euler method. More advanced time integration schemes are discussed in Section 5. For the coupled system (2.9)-(2.10) we obtain

$$\mathbf{u}^{n+1} - \mathbf{u}^n + \Delta t_n \mathbf{h}(\mathbf{u}^{n+1}, \Theta_\Gamma^{n+1}) = \mathbf{0}, \quad (2.11)$$

$$\mathbf{M}(\Theta^{n+1})(\Theta^{n+1} - \Theta^n) + \Delta t_n \mathbf{A}(\Theta^{n+1})\Theta^{n+1} = \Delta t_n (\mathbf{q}_b^f + \mathbf{q}_b^\Gamma(\mathbf{u}^{k+1})). \quad (2.12)$$

2.6 The Dirichlet-Neumann Method

The Dirichlet-Neumann method is a basic iterative substructuring method in domain decomposition and it is a common choice for treating FSI problems. Therefore, we now employ it to solve the system (2.11)-(2.12). This corresponds to alternately solving equation (2.11) on Ω_1 with Dirichlet data on Γ and (2.12) on Ω_2 with Neumann data on Γ .

Thus, one gets for the k -th iteration the two decoupled equation systems

$$\mathbf{u}^{k+1} - \mathbf{u}^n + \Delta t_n \mathbf{h}(\mathbf{u}^{k+1}, \Theta_\Gamma^k) = \mathbf{0}, \quad (2.13)$$

$$\mathbf{M}(\Theta^{k+1})(\Theta^{k+1} - \Theta^n) + \Delta t_n \mathbf{A}(\Theta^{k+1})\Theta^{k+1} = \Delta t_n (\mathbf{q}_b^f + \mathbf{q}_b^\Gamma(\mathbf{u}^{k+1})), \quad (2.14)$$

with some initial condition Θ_Γ^0 . The iteration is terminated according to the standard criterion

$$\|\Theta_\Gamma^{k+1} - \Theta_\Gamma^k\| \leq \tau \quad (2.15)$$

where τ is a user defined tolerance.

The convergence rate of the Dirichlet-Neumann iteration is not great for the coupling between a compressible fluid and a structure [14], which is why a lot of effort goes into convergence acceleration. On the other hand, the Dirichlet-Neumann iteration was reported to be very fast solver for thermal fluid structure interaction. More specifically, in [8] the iteration is extremely efficient and achieves a very accurate solution with at most two iterations per timestep. To analyze the convergence behaviour of the Dirichlet-Neumann iteration will explain why this coupling method is very efficient in some FSI models and very inefficient in some others.

In principle, the convergence rate of the Dirichlet-Neumann method is analyzed in any standard book on domain decomposition methods, e.g. [34, 41]. There, the iteration matrix of the discretized equations is derived with respect to the interface unknowns and the convergence rate is the spectral radius of that. However, due to the nonlinearity of the thermal fluid structure interaction model explained previously, it is not possible to compute the spectral radius of the iteration matrix in this case. For this reason, we perform in the next section a convergence analysis of the coupling of two linear heat equations. We chose this model because it is a basic building block in fluid structure interaction.

3 A Model Problem: Coupled Heat Equations

In FSI models, the solid is typically discretized using finite elements. On the other hand, although the finite element method (FEM) is applicable to computational fluid dynamics, the finite volume method (FVM) is generally a better choice for the discretization of the fluid. This method guarantees the conservation of fluxes through a particular control volume. Therefore, we present here a convergence analysis of the unsteady transmission problem with mixed discretizations (FVM-FEM).

For this model problem, Henshaw and Chand provided in [24] a method to analyze stability and convergence speed of the Dirichlet-Neumann iteration in 2D based on applying the continuous Fourier transform to the semi-discretized equations. Their result depends on ratios of thermal conductivities and diffusivities of the materials. However, in the fully discrete case we have observed that the iteration behaves differently in some cases. Therefore, we propose a complementary analysis for the fully discrete case here.

3.1 Model Problem

The unsteady transmission problem is as follows, where we consider a domain $\Omega \subset \mathbb{R}^d$ which is cut into two subdomains $\Omega_1 \cup \Omega_2 = \Omega$ with transmission conditions at the interface $\Gamma = \Omega_1 \cap \Omega_2$:

$$\begin{aligned}
 \alpha_m \frac{\partial u_m(\mathbf{x}, t)}{\partial t} - \nabla \cdot (\lambda_m \nabla u_m(\mathbf{x}, t)) &= 0, \quad t \in [t_0, t_f], \quad \mathbf{x} \in \Omega_m \subset \mathbb{R}^2, \quad m = 1, 2, \\
 u_m(\mathbf{x}, t) &= 0, \quad t \in [t_0, t_f], \quad \mathbf{x} \in \partial\Omega_m \setminus \Gamma, \\
 u_1(\mathbf{x}, t) &= u_2(\mathbf{x}, t), \quad \mathbf{x} \in \Gamma, \\
 \lambda_2 \frac{\partial u_2(\mathbf{x}, t)}{\partial \mathbf{n}_2} &= -\lambda_1 \frac{\partial u_1(\mathbf{x}, t)}{\partial \mathbf{n}_1}, \quad \mathbf{x} \in \Gamma, \\
 u_m(\mathbf{x}, 0) &= u_m^0(\mathbf{x}), \quad \mathbf{x} \in \Omega_m,
 \end{aligned} \tag{3.1}$$

where \mathbf{n}_m is the outward normal to Ω_m for $m = 1, 2$.

The constants λ_1 and λ_2 describe the thermal conductivities of the materials on Ω_1 and Ω_2 respectively. D_1 and D_2 represent the thermal diffusivities of the materials and they are defined by

$$D_m = \frac{\lambda_m}{\alpha_m}, \quad \text{with } \alpha_m = \rho_m c_{p_m} \tag{3.2}$$

where ρ_m represents the density and c_{p_m} the specific heat capacity of the material placed in Ω_m , $m = 1, 2$.

We discretize this problem with a constant mesh width with respect to both spatial components ($\Delta y := \Delta x = 1/(N + 1)$) resulting in N^2 interior space discretization points in both Ω_1 and Ω_2 . We use the implicit Euler method for the time discretization.

3.2 Semidiscrete Analysis

Before we present in the next section an analysis for the fully discrete equations, we want to describe previous results which analyze the behaviour of the Dirichlet-Neumann iteration for the transmission problem in the semi discrete case.

On one hand, a one dimensional stability analysis was presented by Giles [20]. There, while using the implicit Euler method in the subsolvers, an explicit time integration method was chosen with respect to the interface unknowns.

On the other hand, Henshaw and Chand provided in [24] a method to analyze convergence speed of the Dirichlet-Neumann iteration. There, one applies the implicit Euler method for the time discretization on both equations in (3.1) but keeps the space continuous. Then, they applied the Fourier transform in space in order to transform the second order derivatives into algebraic expressions. This converts the partial differential equations into a system of purely algebraic equations. Once we have a coupled system of algebraic equations, we can insert one into the other and obtain the convergence rate, called amplification factor β in [24]. They then derive the formula

$$\beta = \left| -\frac{\lambda_1}{\lambda_2} \sqrt{\frac{D_2}{D_1}} \frac{\tanh\left(-\frac{1}{\sqrt{D_2\Delta t}}\right)}{\tanh\left(\frac{1}{\sqrt{D_1\Delta t}}\right)} \right|. \quad (3.3)$$

For Δt big enough, we have $\tanh\left(-1/\sqrt{D_2\Delta t}\right) \approx -1/\sqrt{D_2\Delta t}$ and $\tanh\left(1/\sqrt{D_1\Delta t}\right) \approx 1/\sqrt{D_1\Delta t}$ and therefore:

$$\beta \approx \frac{\lambda_1}{\lambda_2} \sqrt{\frac{D_2}{D_1}} \frac{\sqrt{D_1\Delta t}}{\sqrt{D_2\Delta t}} = \frac{\lambda_1}{\lambda_2}. \quad (3.4)$$

One observes in (3.4) that the rates of the iteration behave as the quotient of thermal conductivities when $\Delta t \rightarrow 0$. This suggests that strong jumps in the thermal conductivities of the materials cause fast convergence.

3.3 Space Discretization

We now describe a rather general space discretization of the model problem. The core property that we need is that the meshes of Ω_1 and Ω_2 are compatible on Γ (they share the same nodes on Γ) as shown in Figure 6. Furthermore, we need that there is a specific set of unknowns associated with the interface nodes.

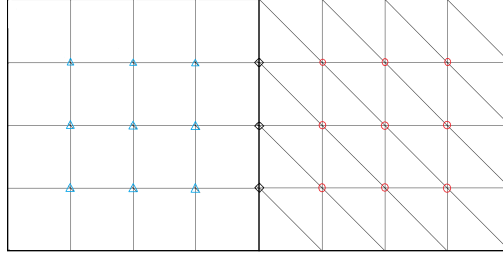


Figure 6. Splitting of Ω between finite volumes and finite elements.

Then, letting $\mathbf{u}_I^{(1)}$ correspond to the unknowns on Ω_1 and \mathbf{u}_Γ to the unknowns at the interface Γ , we can write a general discretization of the first equation in (3.1) in a compact form as:

$$\mathbf{M}_1 \dot{\mathbf{u}}_I^{(1)} + \mathbf{M}_{\Gamma}^{(1)} \dot{\mathbf{u}}_\Gamma + \mathbf{A}_1 \mathbf{u}_I^{(1)} + \mathbf{A}_{\Gamma}^{(1)} \mathbf{u}_\Gamma = \mathbf{0}. \quad (3.5)$$

On the other hand, a general discretization of the first equation in (3.1) on Ω_2 can be written as:

$$\mathbf{M}_2 \dot{\mathbf{u}}_I^{(2)} + \mathbf{M}_{\Gamma}^{(2)} \dot{\mathbf{u}}_\Gamma + \mathbf{A}_2 \mathbf{u}_I^{(2)} + \mathbf{A}_{\Gamma}^{(2)} \mathbf{u}_\Gamma = \mathbf{0}. \quad (3.6)$$

where $\mathbf{u}_I^{(2)}$ correspond to the unknowns on Ω_2 .

However, the system (3.5)-(3.6) is not enough to describe (3.1). Additionally, we need an approximation of the normal derivatives at Γ . If an FVM is used over Ω_1 , we approximate the normal derivative with respect to u_1 using second order one-sided finite differences:

$$-\lambda_1 \frac{\partial u_1}{\partial \mathbf{n}_1} \approx \frac{\lambda_1}{2\Delta x} (4u_{1,N}(t) - u_{1,N-1}(t) - 3u_\Gamma). \quad (3.7)$$

On the other hand, if an FEM is used over Ω_1 and ϕ_j is a nodal basis function for a node on Γ we observe that the normal derivative with respect to u_2 can be written as linear functionals using Green's formula [41, pp. 3]. Thus, the approximation of the normal derivative is given by

$$\begin{aligned} \lambda_2 \int_{\Gamma} \frac{\partial u_2}{\partial \mathbf{n}_2} \phi_j dS &= \lambda_2 \int_{\Omega_2} (\Delta u_2 \phi_j + \nabla u_2 \nabla \phi_j) d\mathbf{x} \\ &= \alpha_2 \int_{\Omega_2} \frac{d}{dt} u_2 \phi_j + \lambda_2 \int_{\Omega_2} \nabla u_2 \nabla \phi_j d\mathbf{x}. \end{aligned} \quad (3.8)$$

Consequently, the equation

$$\mathbf{M}_{\Gamma\Gamma}^{(2)}\dot{\mathbf{u}}_{\Gamma} + \mathbf{M}_{\Gamma I}^{(2)}\dot{\mathbf{u}}_I^{(2)} + \mathbf{A}_{\Gamma\Gamma}^{(2)}\mathbf{u}_{\Gamma} + \mathbf{A}_{\Gamma I}^{(2)}\mathbf{u}_I^{(2)} = -\mathbf{M}_{\Gamma\Gamma}^{(1)}\dot{\mathbf{u}}_{\Gamma} - \mathbf{M}_{\Gamma I}^{(1)}\dot{\mathbf{u}}_I^{(1)} - \mathbf{A}_{\Gamma\Gamma}^{(1)}\mathbf{u}_{\Gamma} - \mathbf{A}_{\Gamma I}^{(1)}\mathbf{u}_I^{(1)}, \quad (3.9)$$

is a discrete version of the fourth equation in (3.1) and completes the system (3.5)-(3.6). We now reformulate the coupled equations (3.5), (3.6) and (3.9) into an ODE for the vector of unknowns $\mathbf{u} = (\mathbf{u}_I^{(1)}, \mathbf{u}_I^{(2)}, \mathbf{u}_{\Gamma})^T$

$$\tilde{\mathbf{M}}\dot{\mathbf{u}} + \tilde{\mathbf{A}}\mathbf{u} = \mathbf{0}, \quad (3.10)$$

where

$$\tilde{\mathbf{M}} = \begin{pmatrix} \mathbf{M}_1 & \mathbf{0} & \mathbf{M}_{I\Gamma}^{(1)} \\ \mathbf{0} & \mathbf{M}_2 & \mathbf{M}_{I\Gamma}^{(2)} \\ \mathbf{M}_{\Gamma I}^{(1)} & \mathbf{M}_{\Gamma I}^{(2)} & \mathbf{M}_{\Gamma\Gamma}^{(1)} + \mathbf{M}_{\Gamma\Gamma}^{(2)} \end{pmatrix}, \quad \tilde{\mathbf{A}} = \begin{pmatrix} \mathbf{A}_1 & \mathbf{0} & \mathbf{A}_{I\Gamma}^{(1)} \\ \mathbf{0} & \mathbf{A}_2 & \mathbf{A}_{I\Gamma}^{(2)} \\ \mathbf{A}_{\Gamma I}^{(1)} & \mathbf{A}_{\Gamma I}^{(2)} & \mathbf{A}_{\Gamma\Gamma}^{(1)} + \mathbf{A}_{\Gamma\Gamma}^{(2)} \end{pmatrix}.$$

3.4 Time Discretization

Applying the implicit Euler method with time step Δt to the system (3.9), we get for the vector of unknowns $\mathbf{u}^{n+1} = (\mathbf{u}_I^{(1),n+1}, \mathbf{u}_I^{(2),n+1}, \mathbf{u}_{\Gamma}^{n+1})^T$

$$\mathbf{A}\mathbf{u}^{n+1} = \tilde{\mathbf{M}}\mathbf{u}^n, \quad (3.11)$$

where

$$\mathbf{A} = \tilde{\mathbf{M}} + \Delta t\tilde{\mathbf{A}} = \begin{pmatrix} \mathbf{M}_1 + \Delta t\mathbf{A}_1 & \mathbf{0} & \mathbf{M}_{I\Gamma}^{(1)} + \Delta t\mathbf{A}_{I\Gamma}^{(1)} \\ \mathbf{0} & \mathbf{M}_2 + \Delta t\mathbf{A}_2 & \mathbf{M}_{I\Gamma}^{(2)} + \Delta t\mathbf{A}_{I\Gamma}^{(2)} \\ \mathbf{M}_{\Gamma I}^{(1)} + \Delta t\mathbf{A}_{\Gamma I}^{(1)} & \mathbf{M}_{\Gamma I}^{(2)} + \Delta t\mathbf{A}_{\Gamma I}^{(2)} & \mathbf{M}_{\Gamma\Gamma} + \Delta t\mathbf{A}_{\Gamma\Gamma} \end{pmatrix},$$

with $\mathbf{M}_{\Gamma\Gamma} = \mathbf{M}_{\Gamma\Gamma}^{(1)} + \mathbf{M}_{\Gamma\Gamma}^{(2)}$ and $\mathbf{A}_{\Gamma\Gamma} = \mathbf{A}_{\Gamma\Gamma}^{(1)} + \mathbf{A}_{\Gamma\Gamma}^{(2)}$.

3.5 Fixed Point Iteration

We now employ a Dirichlet-Neumann iteration to solve the discrete system (3.11). This corresponds to alternately solving the discretized equations of the transmission problem (3.1) on Ω_1 with Dirichlet data on Γ and the discretization of (3.1) on Ω_2 with Neumann data on Γ .

Therefore, from (3.11) one gets for the k -th iteration the two equation systems

$$(\mathbf{M}_1 + \Delta t \mathbf{A}_1) \mathbf{u}_I^{(1),n+1,k+1} = -(\mathbf{M}_{\Gamma}^{(1)} + \Delta t \mathbf{A}_{\Gamma}^{(1)}) \mathbf{u}_{\Gamma}^{n+1,k} + \mathbf{M}_1 \mathbf{u}_I^{(1),n} + \mathbf{M}_{\Gamma}^{(1)} \mathbf{u}_{\Gamma}^n, \quad (3.12)$$

$$\hat{\mathbf{A}} \hat{\mathbf{u}}^{k+1} = \hat{\mathbf{M}} \mathbf{u}^n - \mathbf{b}^k, \quad (3.13)$$

to be solved in succession. Here,

$$\hat{\mathbf{A}} = \begin{pmatrix} \mathbf{M}_2 + \Delta t \mathbf{A}_2 & \mathbf{M}_{\Gamma}^{(2)} + \Delta t \mathbf{A}_{\Gamma}^{(2)} \\ \mathbf{M}_{\Gamma}^{(2)} + \Delta t \mathbf{A}_{\Gamma}^{(2)} & \mathbf{M}_{\Gamma}^{(2)} + \Delta t \mathbf{A}_{\Gamma}^{(2)} \end{pmatrix}, \quad \hat{\mathbf{M}} = \begin{pmatrix} \mathbf{0} & \mathbf{M}_2 & \mathbf{M}_{\Gamma}^{(2)} \\ \mathbf{M}_{\Gamma}^{(1)} & \mathbf{M}_{\Gamma}^{(2)} & \mathbf{M}_{\Gamma}^{(2)} \end{pmatrix},$$

and

$$\mathbf{b}^k = \begin{pmatrix} \mathbf{0} \\ (\mathbf{M}_{\Gamma}^{(1)} + \Delta t \mathbf{A}_{\Gamma}^{(1)}) \mathbf{u}_I^{(1),n+1,k+1} + (\mathbf{M}_{\Gamma}^{(1)} + \Delta t \mathbf{A}_{\Gamma}^{(1)}) \mathbf{u}_{\Gamma}^{n+1,k} \end{pmatrix}, \quad (3.14)$$

$$\hat{\mathbf{u}}^{k+1} = \begin{pmatrix} \mathbf{u}_I^{(2),n+1,k+1} \\ \mathbf{u}_{\Gamma}^{n+1,k+1} \end{pmatrix},$$

with some initial condition, here $\mathbf{u}_{\Gamma}^{n+1,0} = \mathbf{u}_{\Gamma}^n$. The iteration is terminated according to the standard criterion $\|\mathbf{u}_{\Gamma}^{k+1} - \mathbf{u}_{\Gamma}^k\| \leq \tau$ where τ is a user defined tolerance [7].

One way to analyze this method is to write it as a splitting method for (3.11) and try to estimate the spectral radius of that iteration. However, the results obtained in this way are much too inaccurate. For that reason, we now rewrite (3.12)-(3.13) as an iteration for $\mathbf{u}_{\Gamma}^{n+1}$ to restrict the size of the space to the dimension of \mathbf{u}_{Γ} . To this end, we isolate the term $\mathbf{u}_I^{(1),n+1,k+1}$ in (3.12) and $\mathbf{u}_I^{(2),n+1,k+1}$ in the first equation in (3.13) and we insert the resulting expressions into the second equation in (3.13). Consequently, the iteration $\mathbf{u}_{\Gamma}^{n+1,k+1} = \Sigma \mathbf{u}_{\Gamma}^{n+1,k} + \psi^n$ is obtained with iteration matrix

$$\Sigma = -\mathbf{S}^{(2)-1} \mathbf{S}^{(1)}, \quad (3.15)$$

where

$$\mathbf{S}^{(m)} = (\mathbf{M}_{\Gamma}^{(m)} + \Delta t \mathbf{A}_{\Gamma}^{(m)}) - (\mathbf{M}_{\Gamma}^{(m)} + \Delta t \mathbf{A}_{\Gamma}^{(m)}) (\mathbf{M}_m + \Delta t \mathbf{A}_m)^{-1} (\mathbf{M}_{\Gamma}^{(m)} + \Delta t \mathbf{A}_{\Gamma}^{(m)}), \quad (3.16)$$

for $m = 1, 2$ and ψ^n contains terms that depend only on the solutions at the previous time step. Notice that Σ is a discrete version of the Steklov-Poincaré operator.

Thus, the Dirichlet-Neumann iteration is a linear iteration and the rate of convergence is described by the spectral radius of the iteration matrix Σ .

4 Convergence Analysis

The derivation so far was for a rather general discretization. It is now possible to look at specific discretizations. In this section, we study the iteration matrix Σ for a specific FVM-FEM discretization.

The subdomains are here $\Omega_1 = [-1, 0] \times [0, 1]$, $\Omega_2 = [0, 1] \times [0, 1]$. An equidistant grid is chosen i.e. $\Delta x = \Delta y = 1/(N + 1)$. At each discrete point of the finite volume discretization $x_{i,j}$, we integrate over the cell $I_{i,j} = [x_{i-1/2,j}, x_{i+1/2,j}] \times [x_{i,j-1/2}, x_{i,j+1/2}]$ and use the flux function

$$F(u_L, u_R) = -\frac{\lambda_1}{\Delta x}(u_R - u_L), \quad (4.1)$$

to approximate the flux, which results in a second order scheme. For the FEM discretization, we use triangular elements distributed as sketched in Figure 7 and the following pyramidal test functions

$$\phi_k(x, y) = \begin{cases} \frac{x+y}{\Delta x} - 1, & \text{if } \mathbf{x} = (x, y) \in \text{Region 1,} \\ \frac{y}{\Delta x}, & \text{if } \mathbf{x} \in \text{Region 2,} \\ \frac{\Delta x - x}{\Delta x}, & \text{if } \mathbf{x} \in \text{Region 3,} \\ 1 - \frac{x+y}{\Delta x}, & \text{if } \mathbf{x} \in \text{Region 4,} \\ \frac{\Delta x - y}{\Delta x}, & \text{if } \mathbf{x} \in \text{Region 5,} \\ \frac{x}{\Delta x}, & \text{if } \mathbf{x} \in \text{Region 6,} \\ 0, & \text{otherwise.} \end{cases} \quad (4.2)$$

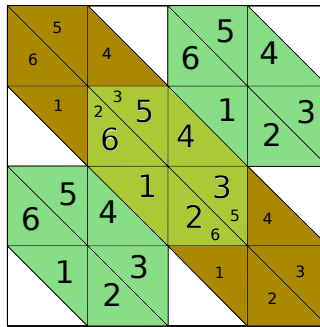


Figure 7. Sketch of the regions for the pyramidal test functions defined in (4.2).

The discretization matrices are given in this case by

$$\mathbf{A}_1 = \frac{\lambda_1}{\Delta x^2} \begin{pmatrix} -\mathbf{B} & \mathbf{I} & \mathbf{0} \\ \mathbf{I} & -\mathbf{B} & \ddots \\ & \ddots & \ddots & \mathbf{I} \\ \mathbf{0} & & \mathbf{I} & -\mathbf{B} \end{pmatrix}, \quad \mathbf{A}_2 = \frac{\lambda_2}{\Delta x^2} \begin{pmatrix} \mathbf{B} & -\mathbf{I} & \mathbf{0} \\ -\mathbf{I} & \mathbf{B} & \ddots \\ & \ddots & \ddots & -\mathbf{I} \\ \mathbf{0} & & -\mathbf{I} & \mathbf{B} \end{pmatrix},$$

where

$$\mathbf{B} = \begin{pmatrix} 4 & -1 & & 0 \\ -1 & 4 & \ddots & \\ & \ddots & \ddots & -1 \\ 0 & & -1 & 4 \end{pmatrix},$$

and $\mathbf{I} \in \mathbb{R}^{N \times N}$ is an identity matrix. Note that each block of the matrices $\mathbf{A}_m \in \mathbb{R}^{N^2 \times N^2}$ has size $N \times N$.

$$\mathbf{M}_2 = \alpha_2 \begin{pmatrix} \mathbf{N} & \tilde{\mathbf{N}} & \mathbf{0} \\ \tilde{\mathbf{N}}^T & \mathbf{N} & \ddots \\ & \ddots & \ddots & \tilde{\mathbf{N}} \\ \mathbf{0} & & \tilde{\mathbf{N}}^T & \mathbf{N} \end{pmatrix},$$

where

$$\mathbf{N} = \begin{pmatrix} 5/6 & -1/12 & & 0 \\ -1/12 & 5/6 & \ddots & \\ & \ddots & \ddots & -1/12 \\ 0 & & -1/12 & 5/6 \end{pmatrix}, \quad \tilde{\mathbf{N}} = \begin{pmatrix} -1/12 & 1/4 & & 0 \\ 0 & -1/12 & \ddots & \\ & \ddots & \ddots & 1/4 \\ 0 & & 0 & -1/12 \end{pmatrix}.$$

Each block of the matrix $\mathbf{M}_2 \in \mathbb{R}^{N^2 \times N^2}$ has size $N \times N$ as well. We consider here $\mathbf{E}_j = \left(\mathbf{0} \ \cdots \ \mathbf{0} \ \mathbf{I} \ \mathbf{0} \ \cdots \ \mathbf{0} \right)^T \in \mathbb{R}^{N^2 \times N}$ where the only nonzero block is the j -th block of size $N \times N$. Thus,

$$\mathbf{A}_{\Gamma}^{(1)} = \frac{\lambda_1}{\Delta x^2} \mathbf{E}_N, \quad \mathbf{A}_{\Gamma}^{(2)} = -\frac{\lambda_2}{\Delta x^2} \mathbf{E}_1, \quad \mathbf{M}_{\Gamma}^{(2)} = \alpha_2 \mathbf{E}_1 \tilde{\mathbf{N}},$$

$$\mathbf{M}_{\Gamma\Gamma}^{(2)} = \alpha_2 \begin{pmatrix} 5/12 & -1/24 & & 0 \\ -1/24 & 5/12 & \ddots & \\ & \ddots & \ddots & -1/24 \\ 0 & & -1/24 & 5/12 \end{pmatrix},$$

$$\mathbf{A}_{\Gamma\Gamma}^{(1)} = \frac{3\lambda_1}{2\Delta x^2} \mathbf{I}, \quad \mathbf{A}_{\Gamma\Gamma}^{(2)} = \frac{\lambda_2}{\Delta x^2} \begin{pmatrix} 2 & -1/2 & & 0 \\ -1/2 & 2 & \ddots & \\ & \ddots & \ddots & -1/2 \\ 0 & & -1/2 & 2 \end{pmatrix},$$

where $\mathbf{M}_{\Gamma\Gamma}^{(2)}$ and $\mathbf{A}_{\Gamma\Gamma}^{(m)} \in \mathbb{R}^{N \times N}$ for $m = 1, 2$.

$$\mathbf{A}_{\Gamma I}^{(1)} = \frac{\lambda_1}{2\Delta x^2} (4\mathbf{E}_N^T - \mathbf{E}_{N-1}^T), \quad \mathbf{A}_{\Gamma I}^{(2)} = -\frac{\lambda_2}{\Delta x^2} \mathbf{E}_1^T, \quad \mathbf{M}_{\Gamma I}^{(2)} = \alpha_2 \mathbf{E}_1^T \tilde{\mathbf{N}}.$$

In this specific case, $\mathbf{M}_1 = \alpha_1 \mathbf{I}$, $\mathbf{M}_{\Gamma\Gamma}^{(1)} = \mathbf{M}_{\Gamma I}^{(1)} = \mathbf{M}_{\Gamma N}^{(1)} = \mathbf{0}$. In particular,

$$\mathbf{S}^{(1)} = \Delta t \mathbf{A}_{\Gamma\Gamma}^{(1)} - \Delta t^2 \mathbf{A}_{\Gamma I}^{(1)} (\alpha_1 \mathbf{I} - \Delta t \mathbf{A}_1)^{-1} \mathbf{A}_{\Gamma\Gamma}^{(1)}, \quad (4.3)$$

$$\mathbf{S}^{(2)} = (\mathbf{M}_{\Gamma\Gamma}^{(2)} + \Delta t \mathbf{A}_{\Gamma\Gamma}^{(2)}) - (\mathbf{M}_{\Gamma I}^{(2)} + \Delta t \mathbf{A}_{\Gamma I}^{(2)}) (\mathbf{M}_2 + \Delta t \mathbf{A}_2)^{-1} (\mathbf{M}_{\Gamma N}^{(2)} + \Delta t \mathbf{A}_{\Gamma N}^{(2)}). \quad (4.4)$$

One computes $\mathbf{S}^{(1)}$ and $\mathbf{S}^{(2)}$ by inserting the corresponding matrices specified above in (4.3) and (4.4) obtaining

$$\mathbf{S}^{(1)} = \frac{3\lambda_1 \Delta t}{2\Delta x^2} \mathbf{I} - \frac{\lambda_1^2 \Delta t^2}{2\Delta x^4} (4\mathbf{E}_N^T - \mathbf{E}_{N-1}^T) (\alpha_1 \mathbf{I} - \Delta t \mathbf{A}_1)^{-1} \mathbf{E}_N, \quad (4.5)$$

$$\begin{aligned} \mathbf{S}^{(2)} = & \left(\alpha_2 \text{tridiag} \left(-\frac{1}{24}, \frac{5}{12}, -\frac{1}{24} \right) + \frac{\lambda_2 \Delta t}{\Delta x^2} \text{tridiag} \left(-\frac{1}{2}, 2, -\frac{1}{2} \right) \right) \\ & - \left(\alpha_2 \mathbf{N}_2 - \frac{\lambda_2 \Delta t}{\Delta x^2} \mathbf{I} \right) \mathbf{E}_1^T (\mathbf{M}_2 + \Delta t \mathbf{A}_2)^{-1} \mathbf{E}_1 \left(\alpha_2 \mathbf{N}_2 - \frac{\lambda_2 \Delta t}{\Delta x^2} \mathbf{I} \right). \end{aligned} \quad (4.6)$$

In the two-dimensional case, the iteration matrix Σ is a matrix of size $N \times N$. This makes the iteration matrix Σ difficult to compute for several reasons. First of all,

the matrices $\alpha_1 \mathbf{I} - \Delta t \mathbf{A}_1$ and $\mathbf{M}_2 + \Delta t \mathbf{A}_2$ are sparse block tridiagonal matrices, and consequently, their inverses are not straight forward to compute. A block-by-block algorithm for inverting a block tridiagonal matrix is explained in [36]. However, the algorithm is based on the iterative application of the Schur complement [46], and it results in a sequence of block matrices and inverses of block matrices that we did not find possible to compute exactly. Moreover, the diagonal blocks of $\alpha_1 \mathbf{I} - \Delta t \mathbf{A}_1$ and $\mathbf{M}_2 + \Delta t \mathbf{A}_2$ are tridiagonal but their inverses are full matrices [18].

Due to these difficulties, we propose here to approximate Σ . One can observe that $\alpha_1 \mathbf{I} - \Delta t \mathbf{A}_1$ and $\mathbf{M}_2 + \Delta t \mathbf{A}_2$ are strictly diagonally dominant matrices, and therefore, we propose to approximate them by their block diagonal. Thus,

$$\mathbf{S}^{(1)} \approx \frac{3\lambda_1 \Delta t}{2\Delta x^2} \mathbf{I} - \frac{2\lambda_1^2 \Delta t^2}{\Delta x^4} \left(\text{tridiag} \left(-\frac{\lambda_1 \Delta t}{\Delta x^2}, \frac{\alpha_1 \Delta x^2 + 4\lambda_1 \Delta t}{\Delta x^2}, -\frac{\lambda_1 \Delta t}{\Delta x^2} \right)^{-1} \right), \quad (4.7)$$

$$\begin{aligned} \mathbf{S}^{(2)} \approx & \left(\alpha_2 \text{tridiag} \left(-\frac{1}{24}, \frac{5}{12}, -\frac{1}{24} \right) + \frac{\lambda_2 \Delta t}{\Delta x^2} \text{tridiag} \left(-\frac{1}{2}, 2, -\frac{1}{2} \right) \right) \\ & - \left(\frac{\alpha_2 \Delta x^2 + 12\lambda_2 \Delta t}{12\Delta x^2} \right)^2 \left(\text{tridiag} (b, a, b)^{-1} \right), \end{aligned} \quad (4.8)$$

with

$$a = \frac{5\alpha_2 \Delta x^2 + 24\lambda_2 \Delta t}{6\Delta x^2}, \quad b = - \left(\frac{\alpha_2 \Delta x^2 + 12\lambda_2 \Delta t}{12\Delta x^2} \right).$$

Now, we compute the eigenvalues of the proposed approximations of $\mathbf{S}^{(1)}$ and $\mathbf{S}^{(2)}$. The eigenvalues of a tridiagonal Toeplitz matrix are known and given e.g. in [32, pp. 514-516]:

$$\mu_j^1 = \frac{3\lambda_1 \Delta t (\alpha_1 \Delta x^2 + 2\lambda_1 \Delta t (2 - \cos(j\pi \Delta x))) - 4\lambda_1^2 \Delta t^2}{2\Delta x^2 (\alpha_1 \Delta x^2 + 2\lambda_1 \Delta t (2 - \cos(j\pi \Delta x)))}, \quad (4.9)$$

$$\mu_j^2 = \frac{2(\alpha_2 \Delta x^2 (5 - \cos(j\pi \Delta x)) + 12\lambda_2 \Delta t (2 - \cos(j\pi \Delta x)))^2 - (\alpha_2 \Delta x^2 + 12\lambda_2 \Delta t)^2}{24\Delta x^2 (\alpha_2 \Delta x^2 (5 - \cos(j\pi \Delta x)) + 12\lambda_2 \Delta t (2 - \cos(j\pi \Delta x)))}, \quad (4.10)$$

for $j = 1, \dots, N$. Here μ_j^1 are the eigenvalues of the approximation of $\mathbf{S}^{(1)}$ and μ_j^2 the eigenvalues of the approximation of $\mathbf{S}^{(2)}$. Note that the eigenvectors are common for symmetric tridiagonal Toeplitz matrices, i.e, the eigenvectors do not depend on the specific entries.

Thus, we obtain an estimate of the spectral radius of the iteration matrix Σ :

$$\begin{aligned} \rho(\Sigma) &= \rho(\mathbf{S}^{(2)-1}\mathbf{S}^{(1)}) = \mu_{\max}(\mathbf{S}^{(2)-1}\mathbf{S}^{(1)}) \approx \frac{\mu_1^1}{\mu_1^2} \\ &= \left(\frac{12(\alpha_2\Delta x^2(5 - \cos(\pi\Delta x)) + 12\lambda_2\Delta t(2 - \cos(\pi\Delta x)))}{\alpha_1\Delta x^2 + 2\lambda_1\Delta t(2 - \cos(\pi\Delta x))} \right) \\ &\cdot \left(\frac{3\lambda_1\Delta t(\alpha_1\Delta x^2 + 2\lambda_1\Delta t(2 - \cos(\pi\Delta x))) - 4\lambda_1^2\Delta t^2}{2(\alpha_2\Delta x^2(5 - \cos(\pi\Delta x)) + 12\lambda_2\Delta t(2 - \cos(\pi\Delta x)))^2 - (\alpha_2\Delta x^2 + 12\lambda_2\Delta t)^2} \right) := \sigma. \end{aligned} \quad (4.11)$$

Furthermore, computing the limits of (4.11) when $\Delta t \rightarrow 0$ and $\Delta x \rightarrow 0$ we get

$$\lim_{\Delta t \rightarrow 0} \rho(\Sigma) \approx \lim_{\Delta t \rightarrow 0} \sigma = \frac{(12\alpha_2\Delta x^2(5 - \cos(\pi\Delta x))) \cdot 0}{\alpha_1\Delta x^2(2\alpha_2^2\Delta x^4(5 - \cos(\pi\Delta x))^2 - \alpha_2^2\Delta x^4)} = 0, \quad (4.12)$$

$$\lim_{\Delta x \rightarrow 0} \rho(\Sigma) \approx \lim_{\Delta x \rightarrow 0} \sigma = \frac{12^2\lambda_2\Delta t(6\lambda_1^2\Delta t^2 - 4\lambda_1^2\Delta t^2)}{2\lambda_1\Delta t(2 \cdot 12^2\lambda_2^2\Delta t^2 - 12^2\lambda_2^2\Delta t^2)} = \frac{\lambda_1}{\lambda_2} =: \delta. \quad (4.13)$$

Therefore, from the result obtained in (4.13) we expect that strong jumps in the physical properties of the materials placed in Ω_1 and Ω_2 will imply fast convergence. This is the case when modelling thermal fluid structure interaction, where often a fluid with low thermal conductivity and density is coupled with a structure having higher thermal conductivity and density.

When $c \rightarrow \infty$, $\Delta t \gg \Delta x^2$ and (3.4) matches with the asymptotic computed in (4.13). However, when $c < 1$, the semicontinuous analysis fails and the discrete analysis just presented fills the gap.

4.1 Numerical Results

In this section we present a set of numerical experiments designed to show how the validity of the approximation of $\rho(\Sigma)$ as an estimator for the rates of the coupled problem formulated above. We also show that the theoretical asymptotics deduced in (4.12) and (4.13) match with the numerical experiments.

We first compare the semidiscrete estimator β with the discrete estimator σ and the experimental convergences rates. The experimental convergence rates C_R are computed with respect to a reference solution u_{ref} over the whole domain Ω using the formula

$$C_R = \frac{\|u^3 - u_{ref}\|_2}{\|u^2 - u_{ref}\|_2},$$

where u^2 and u^3 are the second and third iterates of the Dirichlet-Neumann iteration.

Figure 8 shows a comparison between β and σ . On the left we plot β , σ and the experimental convergence rates with $c \ll 1$ and on the right we plot the same but with $c \gg 1$. We can conclude that the estimator for the convergence rates presented in the previous section is minimally better than the one proposed by the semidiscrete analysis in [24] when $c \gg 1$. Moreover, when $c \ll 1$ our estimator also predicts the rates accurately and the semidiscrete estimator deviates.

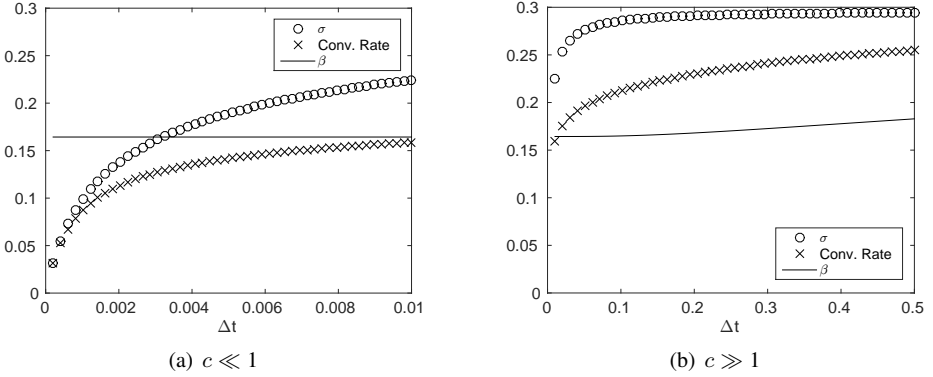


Figure 8. Here, $D_1 = 1$, $D_2 = 0.5$, $\lambda_1 = 0.3$ and $\lambda_2 = 1$. The circles correspond to σ , the crosses to the experimental convergence rates and the continuous line to β . $\Delta x = 1/20$ and on the left the curves are restricted to the discrete values $\Delta t = 1e - 2/50, 2 \cdot 1e - 2/50, \dots, 50 \cdot 1e - 2/50$ and on the right to the values $\Delta t = 1e - 2, 2 \cdot 1e - 2, \dots, 50 \cdot 1e - 2$.

We now want to illustrate how the formula (4.11) predicts the convergence rates and tends to the limits computed previously. To this end, we present two real data examples. We consider here the thermal interaction between air at $273K$ with steel at $900K$ and water at $283K$ with steel at $900K$. Physical properties of the materials and resulting asymptotics for these two cases are shown in table 1 and 2 respectively.

Table 1. Physical properties of the materials. λ is the thermal conductivity, ρ the density, C the specific heat capacity and $\alpha = \rho C$.

Material	λ (W/mK)	ρ (kg/m ³)	C (J/kgK)	α (J/K m ³)
Air	0.0243	1.293	1005	1299.5
Water	0.58	999.7	4192.1	4.1908e6
Steel	48.9	7836	443	3471348

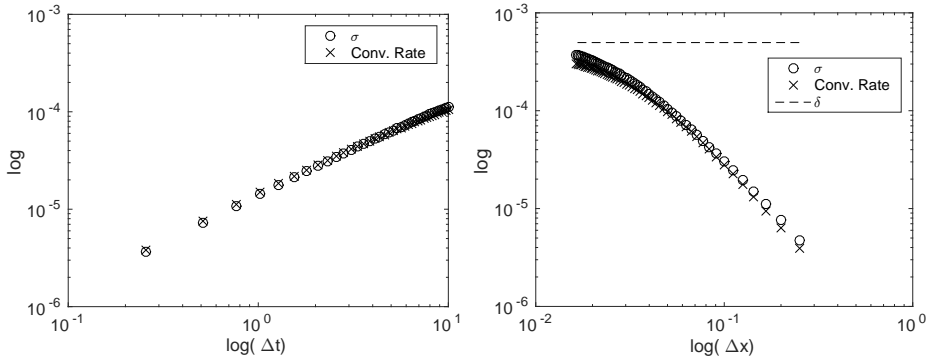
Figures 9 and 10 show the convergence rates for the interactions between air-steel and water-steel respectively. On the left we always plot the rates with respect to the

Table 2. The limits of the convergence rates when $\Delta t \rightarrow 0$ and $\Delta x \rightarrow 0$.

Case	$\Delta t \rightarrow 0$	$\Delta x \rightarrow 0$
Air-Steel	0	4.9693e-4
Water-Steel	0	0.0119

variation of Δt and for a fixed Δx . On the right we plot the behaviour of the rates for a fixed Δt and varying Δx .

We observe from figures 9 and 10 that the approximation σ predicts the convergence rates quite well because the difference with respect to the experimental rates is really small. Moreover, the rates in 9a and 10a tend to 0 as predicted in (4.12) and the rates in 9b and 10b tend to δ as predicted in (4.13).



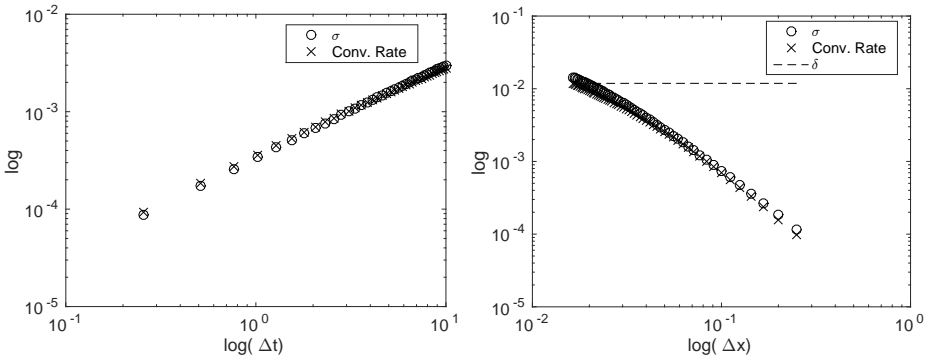
(a) The curves are restricted to the discrete values $\Delta t = 10/40, 2 \cdot 10/40, \dots, 40 \cdot 10/40$ and $\Delta x = 1/3, 1/4, \dots, 1/60$ and $\Delta t = 10, 1/20$. (b) The curves are restricted to the discrete values $\Delta x = 1/3, 1/4, \dots, 1/60$ and $\Delta t = 10, 1/20$.

Figure 9. Air-Steel thermal interaction with respect Δt on the left and Δx on the right.

From figure 9 we can observe that the convergence rates are really fast (factor of $\sim 1e-4$) when there exist strong jumps in the coefficient of the materials. For instance, when performing the thermal coupling between air and steel the Dirichlet-Neumann iteration only needs two iterations to achieve a tolerance of $1e-10$.

5 Time Adaptive Methods

The standard in FSI is to use fixed time step sizes. We find that it is much more efficient to use time adaptive methods. Thus the question arises how such a method can be implemented in a partitioned way. The method described here has been suggested in



(a) The curves are restricted to the discrete values $\Delta t = 10/40, 2 \cdot 10/40, \dots, 40 \cdot 10/40$ and $\Delta x = \Delta x = 1/3, 1/4, \dots, 1/60$ and $\Delta t = 10$.
 1/20.

Figure 10. Water-Steel thermal interaction with respect Δt on the left and Δx on the right.

α	α	0	
1	$1 - \alpha$	α	$\alpha = 1 - \sqrt{2}/2$
b_i	$1 - \alpha$	α	$\hat{\alpha} = 2 - \frac{5}{4}\sqrt{2}$
\hat{b}_i	$1 - \hat{\alpha}$	$\hat{\alpha}$	$\alpha - \hat{\alpha} = -1 + \frac{3}{4}\sqrt{2}$
$b_i - \hat{b}_i$	$\hat{\alpha} - \alpha$	$\alpha - \hat{\alpha}$	

Table 3. Butcher array for SDIRK2.

[10, 9]. Basis is a master program that does the global time stepping and calls functions available in the subsolvers.

Here, an A-stable singly diagonally implicit Runge-Kutta method (SDIRK) is used. Consider an autonomous initial value problem

$$\dot{\mathbf{u}}(t) = \mathbf{f}(\mathbf{u}(t)), \quad \mathbf{u}(0) = \mathbf{u}_0.$$

An SDIRK method is then defined as

$$\mathbf{U}^i = \mathbf{u}^n + \Delta t_n \sum_{j=1}^i a_{ij} \mathbf{f}(\mathbf{U}^j), \quad i = 1, \dots, s \quad (5.1)$$

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t_n \sum_{i=1}^s b_i \mathbf{f}(\mathbf{U}^i)$$

with given coefficients a_{ij}, b_j, c_j . Here, we use the two stage method SDIRK2, which is defined by the coefficients in the Butcher array in Table 3. For this method, $a_{si} = b_i$,

which implies that the last line is superfluous (first-is-last-property). Besides saving computational effort, this guarantees L-stability. The vectors

$$\mathbf{k}_i = \mathbf{f}(\mathbf{U}^i)$$

are called stage derivates and s is the number of stages. Since the starting vector

$$\mathbf{s}_i = \mathbf{u}^n + \Delta t_n \sum_{j=1}^{i-1} a_{ij} \mathbf{k}_j, \quad i = 1, \dots, s - 1$$

is known, (5.1) is just a sequence of implicit Euler steps.

In the following it is assumed that at time t_n , the step size Δt_n is prescribed globally. Applying a DIRK method to equations (2.9)-(2.10) results in the coupled system of equations to be solved at Runge-Kutta stage i :

$$\mathbf{u}_i - \mathbf{s}_i^{\mathbf{u}} - \Delta t_n a_{ii} \mathbf{h}(\mathbf{u}_i, \Theta_i^{\Gamma}) = \mathbf{0}, \quad (5.2)$$

$$[\mathbf{M} + \Delta t_n a_{ii} \mathbf{A}(\Theta)] \Theta_i - \mathbf{M} \mathbf{s}_i^{\Theta} - \mathbf{q}_b^f - \mathbf{q}_b^{\Gamma}(\mathbf{u}_i) = \mathbf{0}. \quad (5.3)$$

Here, $\mathbf{s}_i^{\mathbf{u}}$ and \mathbf{s}_i^{Θ} are the given starting vectors in the subproblems.

The coupled equations (5.2)-(5.3) are then solved using a Dirichlet-Neumann coupling, as would be done for the implicit Euler method. Following the analysis just presented, temperature is prescribed for the equation with smaller heat conductivity, here the fluid, and heat flux is given on Γ for the structure. Choosing these conditions the other way around leads to an unstable scheme. The canonical starting guess for the Dirichlet-Neumann iteration at stage i is the starting vector \mathbf{s}_i .

Regarding implementation, it is a safe assumption that both the fluid and the solid solver are able to carry out time steps of implicit Euler type. Then the master program of the FSI procedure can be extended to SDIRK methods very easily. The master program just has to call at stage i in iteration k the backward Euler routines with time step size $a_{ii} \Delta t_n$ and starting vectors \mathbf{s}_i and with the boundary data appropriate for iteration k .

To obtain time adaptivity, the technique of embedded Runge-Kutta methods is used. Thereby, the local error $\hat{\mathbf{I}}$ is estimated by the solvers separately:

$$\hat{\mathbf{I}}^{(i)} \approx \mathbf{I}^{(i)} = \sum_{j=1}^2 (b_j - \hat{b}_j) \mathbf{k}_j^{(i)}, \quad i = 1, 2, \quad (5.4)$$

where $\mathbf{k}_j^{(i)}$ are the stage derivatives on the subdomains for each Runge-Kutta stage. The computation of (5.4) will typically not be implemented in the subsolvers and has to be added to both codes, as well as that all stage derivatives have to be stored by the

subsolvers. These error estimates are then reported back to the master program and aggregated in the form of a weighted scaled norm [9]:

$$\|\mathbf{I}\|_{WSN} = \sqrt{\frac{1}{n} \sum_{j=1}^n \frac{l_j}{TOLu_j + TOL}} = \sqrt{\frac{1}{n} (n_1 \|\mathbf{I}^{(1)}\|_{WSN}^2 + n_2 \|\mathbf{I}^{(2)}\|_{WSN}^2)}. \quad (5.5)$$

Thus, the master solver either needs to know the number of unknowns n_1 and n_2 in the subsolvers, or instead the quantities $n_i \|\mathbf{I}^{(i)}\|_{WSN}^2$ have to be provided to it. Based on this, the new time step is chosen to comply with a user defined error tolerance TOL for the time integration:

$$\Delta t_{n+1} = \Delta t_n \cdot \|\mathbf{I}\|_{WSN}^{-1/k} \quad (5.6)$$

Finally, if the possibility of rejected time steps is taken into account, the current solution pair (\mathbf{u}, Θ) has to be stored as well.

5.1 Extrapolation from Time Integration

To find better starting values for iterative processes in implicit time integration schemes, it is common to use extrapolation based on knowledge about the trajectory of the solution of the initial value problem [15, 33]. In the spirit of partitioned solvers, it was suggested in [8] to use extrapolation of the interface temperatures only. On top, this strategy could be used as well within the subsolvers, which we will not consider here and leave that to the discretion of the developers of those codes. We now present extrapolation methods for SDIRK2 from [8].

At the first stage, we have the old time step size Δt_{n-1} with value Θ_{n-1} and the current time step size Δt_n with value Θ_n . We are looking for the value Θ_n^1 at the next stage time $t_n + c_1 \Delta t_n$. Linear extrapolation results in

$$\Theta_n^1 \approx \Theta_n + c_1 \Delta t_n (\Theta_n - \Theta_{n-1}) / \Delta t_{n-1} = \left(1 + \frac{c_1 \Delta t_n}{\Delta t_{n-1}}\right) \Theta_n - \frac{c_1 \Delta t_n}{\Delta t_{n-1}} \Theta_{n-1}. \quad (5.7)$$

At the second stage, we linearly extrapolate Θ_n at t_n and Θ_n^1 at $t_n + c_1 \Delta t$ to obtain

$$\Theta_{n+1} \approx \Theta_n + \Delta t_n (\Theta_n^1 - \Theta_n) / (c_1 \Delta t_n) = \left(1 - \frac{1}{c_1}\right) \Theta_n + \frac{1}{c_1} \Theta_n^1. \quad (5.8)$$

6 Choosing Tolerances

In the time adaptive setting, a core input parameter is a tolerance TOL . This, based on the error estimate, steers the time step size according to (5.5)-(5.6). For this to work, it is imperative that the iteration error is smaller than the time integration error, so as not to destroy the error estimate of the latter. The Dirichlet-Neumann iteration is formulated as a fixed point iteration in the interface unknowns with termination criterion

(2.15). The tolerance there thus has to be chosen that the iteration error not only on the interface, but on the whole domain is smaller than the time integration error. At the same time, oversolving should be avoided to prevent unnecessary computational cost. A good rule of thumb is to divide the tolerance for the time integration procedure by five [38], ending up with

$$\tau = TOL/5. \quad (6.1)$$

On the assumption of exact solves in the subsolvers, errors in the interface will nevertheless be translated into errors in the subproblems. The boundary condition from the interface enters the right hand side of the nonlinear equations on the subproblem. Thus, by way of the inverse operator, errors in the interface values are translated into errors in the subdomains. For parabolic problems as in heat transfer, these are bounded operators and thus the error transfer can be controlled. Right now, we assume that it is of order one and do not adjust (6.1) further. However, a more careful analysis of this point to get better estimates is required, for example using discrete trace inequalities.

Actually, it is common to use iterative solvers for the subproblems as well, typically Newton or Multigrid methods (compare [6] for an overview on the state of the art). So the question needs to be raised again, what type of termination criteria to choose and what kind of tolerance. For a nonlinear subproblem of the form

$$\mathbf{F}(\mathbf{u}) = \mathbf{0},$$

the standard would be to use a relative termination criterion

$$\|\mathbf{F}(\mathbf{u}^k)\| \leq \tau_r \|\mathbf{F}(\mathbf{u}^0)\|$$

with relative tolerance τ_r . The errors introduced from these systems into the fixed point iteration will be on the order of the residual. Thus, it makes sense to use (6.1) for the tolerances in the subsolvers as well.

In the linear case, the same reasoning applies when we use the relative termination criterion

$$\|\mathbf{A}(\mathbf{x}^k)\mathbf{x}^{k+1} - \mathbf{b}\| \leq \tau_r \|\mathbf{b}\|. \quad (6.2)$$

However, it turns out that this has some pitfalls and that in fact the use of the nonstandard relative criterion

$$\|\mathbf{A}(\mathbf{x}^k)\mathbf{x}^{k+1} - \mathbf{b}\| \leq \tau_r \|\mathbf{A}(\mathbf{x}^k)\mathbf{x}^k - \mathbf{b}\| \quad (6.3)$$

can speed up the iteration significantly. In the following analysis, taken from [7], an absolute criterion

$$\|\mathbf{A}(\mathbf{x}^k)\mathbf{x}^{k+1} - \mathbf{b}\| \leq \tau_a, \quad (6.4)$$

with an absolute tolerance τ_a is possible as well. The first criterion, in case of convergence of the iteration, leads to an automatic tightening of the tolerance over the course of the iteration.

To determine good tolerances, we will now analyze the errors arising from these more careful.

6.1 Perturbed Nested Fixed Point Iteration

Consider two functions $\mathbf{F} : \Omega_1 \rightarrow \Omega_2$ and $\mathbf{S} : \Omega_2 \rightarrow \Omega_1$ with $\Omega_1, \Omega_2 \subset \mathbb{R}^n$ closed and the fixed point equation

$$\mathbf{x} = \mathbf{S}(\mathbf{F}(\mathbf{x})) \quad (6.5)$$

which we assume to have a unique solution \mathbf{x}^* . Let then both the evaluation of \mathbf{F} and of \mathbf{S} be perturbed, namely \mathbf{S} by δ_k and \mathbf{F} by ϵ_k , and consider the iteration:

$$\mathbf{x}^{k+1} = \mathbf{S}(\mathbf{F}(\mathbf{x}^k) + \epsilon_k) + \delta_k. \quad (6.6)$$

We assume that this iteration is well defined and that this sequence has the limit \mathbf{x}_ϵ . Then, we have the following theorem [7].

Theorem 6.1. *Let \mathbf{F} and \mathbf{S} be Lipschitz continuous with Lipschitz constants L_F and L_S , respectively. Assume that $L_F L_S < 1$. Then we have, if $\epsilon_k = \delta_k = \epsilon$ for all k , that*

$$\|\mathbf{x}_\epsilon - \mathbf{x}^*\| \leq \epsilon \frac{1 + L_S}{1 - L_S L_F}. \quad (6.7)$$

In the case $\epsilon_k = \epsilon$ and $\delta_k = \delta$, we obtain

$$\|\mathbf{x}_\epsilon - \mathbf{x}^*\| \leq \frac{\epsilon L_S + \delta}{1 - L_S L_F}. \quad (6.8)$$

Finally, $\mathbf{x}_\epsilon = \mathbf{x}^*$ if and only if both δ_k and ϵ_k converge to zero.

Proof: We have due to the Lipschitz continuity

$$\begin{aligned} \|\mathbf{x}^{k+1} - \mathbf{x}^*\| &= \|\mathbf{S}(\mathbf{F}(\mathbf{x}^k) + \epsilon_k) + \delta_k - \mathbf{x}^*\| = \|\mathbf{S}(\mathbf{F}(\mathbf{x}^k) + \epsilon_k) + \delta_k - \mathbf{S}(\mathbf{F}(\mathbf{x}^*))\| \\ &\leq L_S \|\mathbf{F}(\mathbf{x}^k) - \mathbf{F}(\mathbf{x}^*) + \epsilon_k\| + \delta_k \leq L_S L_F \|\mathbf{x}^k - \mathbf{x}^*\| + L_S \epsilon_k + \delta_k \\ &\leq (L_S L_F)^2 \|\mathbf{x}^{k-1} - \mathbf{x}^*\| + L_S^2 L_F \epsilon_{k-1} + L_S L_F \delta_{k-1} + L_S \epsilon_k + \delta_k \\ &\leq (L_S L_F)^{k+1} \|\mathbf{x}^0 - \mathbf{x}^*\| + \left(\sum_{j=0}^k L_S^{j+1} L_F^j \epsilon_{k-j} \right) + \left(\sum_{j=0}^k L_S^j L_F^j \delta_{k-j} \right) \end{aligned}$$

and thus in the limit $\mathbf{x}^{k+1} \rightarrow \mathbf{x}_\epsilon$,

$$\|\mathbf{x}_\epsilon - \mathbf{x}^*\| \leq L_S \lim_{k \rightarrow \infty} \sum_{j=0}^k (L_S L_F)^j \epsilon_{k-j} + \lim_{k \rightarrow \infty} \sum_{j=0}^k (L_S L_F)^j \delta_{k-j} \quad (6.9)$$

For a constant perturbation overall, e.g. $\epsilon_k = \delta_k = \epsilon$ for all k , we obtain in the limit

$$\|\mathbf{x}_\epsilon - \mathbf{x}^*\| \leq \epsilon (1 + L_S) \lim_{k \rightarrow \infty} \sum_{j=0}^k (L_S L_F)^j = \epsilon \frac{1 + L_S}{1 - L_S L_F},$$

which proves the inequality (6.7). If we have constant but separate perturbations ϵ and δ of \mathbf{S} and \mathbf{F} , we obtain (6.8) from

$$\|\mathbf{x}_\epsilon - \mathbf{x}^*\| \leq \epsilon L_S \lim_{k \rightarrow \infty} \sum_{j=0}^k (L_S L_F)^j + \delta \lim_{k \rightarrow \infty} \sum_{j=0}^k (L_S L_F)^j = \frac{\epsilon L_S + \delta}{1 - L_S L_F}.$$

In the general case, due to positivity, the right hand side of (6.9) is zero if and only if both ϵ_k and δ_k are such that for $\phi_k = \epsilon_k$ or $\phi_k = \delta_k$,

$$\lim_{k \rightarrow \infty} \sum_{j=0}^k (L_S L_F)^j \phi_{k-j} = 0.$$

This is the case if and only if both ϵ_k and δ_k converge to zero.

6.2 Application: Dirichlet-Neumann Coupling for Transmission Problem

As an application of the theory just presented, we consider the steady transmission problem, where the Laplace equation with right hand side $f(x, y)$ on a domain Ω is cut into two domains $\Omega = \Omega_1 \cup \Omega_2$ using transmission conditions at the interface $\Gamma = \Omega_1 \cap \Omega_2$:

$$\begin{aligned} \Delta u_m(x, y) &= f(x, y), & (x, y) \in \Omega_m \subset \mathbb{R}^2, & m = 1, 2, \\ u_m(x, y) &= 0, & (x, y) \in \partial\Omega_m \Gamma, \\ u_1(x, y) &= u_2(x, y), & (x, y) \in \Gamma, \\ \frac{\partial u_2(x, y)}{\mathbf{n}_2} &= -\frac{\partial u_1(x, y)}{\mathbf{n}_1}, & (x, y) \in \Gamma. \end{aligned} \tag{6.10}$$

We now employ a standard Dirichlet-Neumann iteration to solve it. This corresponds to alternately solving the problems

$$\mathbf{A}_1 \mathbf{u}_1^{k+1} = \mathbf{b}_1(\mathbf{u}_2^k) \tag{6.11}$$

and

$$\mathbf{A}_2 \mathbf{u}_2^{k+1} = \mathbf{b}_2(\mathbf{u}_1^{k+1}), \tag{6.12}$$

where problem (6.11) originates from a linear discretization of the transmission problem (6.10) on Ω_1 only with Dirichlet data on Γ given by \mathbf{u}_2^k on the coupling interface and problem (6.12) corresponds to a linear discretization of (6.10) on Ω_2 only with Neumann data on Γ given by the discrete normal derivative of \mathbf{u}_1 on Γ .

By considering (6.11)-(6.12) as a nested iteration, we obtain a fixed point formulation

$$\mathbf{u}_\Gamma = \mathbf{S}(\mathbf{F}(\mathbf{u}_\Gamma))$$

where $\mathbf{u}_\Gamma = \mathbf{u}_2|_\Gamma$, $\mathbf{F} = \mathbf{D}_{\text{n}\Gamma} \mathbf{A}_1^{-1} \mathbf{b}_1(\mathbf{u}_\Gamma)$ and $\mathbf{S} = \mathbf{P}_\Gamma \mathbf{A}_2^{-1} \mathbf{b}_2(\mathbf{u}_1)$. Hereby $\mathbf{D}_{\text{n}\Gamma}$ is the matrix that computes the discrete normal derivatives in Ω_1 on Γ and \mathbf{P}_Γ is the discrete trace operator with respect to Γ .

In practice, the linear equation systems are solved iteratively, typically using the conjugate gradient method (CG) up to a relative tolerance of τ . Thus, we obtain a perturbed nested fixed point iteration of the form (6.6) and the question is now again if we can quantify this perturbation. We have

$$\mathbf{u}_{1_\epsilon}^{k+1} = \mathbf{A}_1^{-1} \mathbf{b}_1(\mathbf{u}_\Gamma^k) + \epsilon_k \quad (6.13)$$

and

$$\mathbf{u}_{2_\epsilon}^{k+1} = \mathbf{A}_2^{-1} \mathbf{b}_2(\mathbf{u}_{1_\epsilon}^{k+1}) + \delta_k. \quad (6.14)$$

For the iteration (6.13) we obtain

$$\|\epsilon_k\| = \|\mathbf{u}_{1_\epsilon}^{k+1} - \mathbf{A}_1^{-1} \mathbf{b}_1(\mathbf{u}_\Gamma^k)\| \leq \|\mathbf{A}_1^{-1}\| \|\mathbf{A}_1 \mathbf{u}_{1_\epsilon}^{k+1} - \mathbf{b}_1(\mathbf{u}_\Gamma^k)\|.$$

Again, the second factor is what is tested in the termination criterion. In the case of the relative criterion (6.3), this results in

$$\|\mathbf{A}_1 \mathbf{u}_{1_\epsilon}^{k+1} - \mathbf{b}_1(\mathbf{u}_\Gamma^k)\| \leq \tau_r \|\mathbf{A}_1 \mathbf{u}_{1_\epsilon}^k - \mathbf{b}_1(\mathbf{u}_\Gamma^k)\|.$$

We thus have managed to shift the index in $\mathbf{A}_1 \mathbf{u}_{1_\epsilon}^k$, but not in $\mathbf{b}_1(\mathbf{u}_\Gamma^k)$. For this we add zero and use the triangle inequality:

$$\tau_r \|\mathbf{A}_1 \mathbf{u}_{1_\epsilon}^k - \mathbf{b}_1(\mathbf{u}_\Gamma^k)\| \leq \tau_r (\|\mathbf{A}_1 \mathbf{u}_{1_\epsilon}^k - \mathbf{b}_1(\mathbf{u}_\Gamma^{k-1})\| + \|\mathbf{b}_1(\mathbf{u}_\Gamma^{k-1}) - \mathbf{b}_1(\mathbf{u}_\Gamma^k)\|).$$

We can now repeat this argument on $\|\mathbf{A}_1 \mathbf{u}_{1_\epsilon}^k - \mathbf{b}_1(\mathbf{u}_\Gamma^k)\|$, thereby multiplying again with τ_r and adding a difference of right hand sides:

$$\|\epsilon_k\| \leq \|\mathbf{A}_1^{-1}\| \left(\tau_r^k \|\mathbf{A}_1 \mathbf{u}_{1_\epsilon}^1 - \mathbf{b}_1(\mathbf{u}_\Gamma^0)\| + \sum_{j=1}^k \tau_r^j \|\mathbf{b}_1(\mathbf{u}_\Gamma^{k-j}) - \mathbf{b}_1(\mathbf{u}_\Gamma^{k-j+1})\| \right).$$

The right hand sides are equal except at the boundary and we have

$$\|\mathbf{b}_1(\mathbf{u}_\Gamma^{k-j}) - \mathbf{b}_1(\mathbf{u}_\Gamma^{k-j+1})\| = 1/\Delta x^2 \|\mathbf{u}_\Gamma^{k-j} - \mathbf{u}_\Gamma^{k-j+1}\|.$$

Furthermore, the last system is tested against the unperturbed initial data leading to

$$\|\epsilon_k\| \leq \|\mathbf{A}_1^{-1}\| \left(\tau_r^{k+1} \|\mathbf{A}_1 \mathbf{u}_1^0 - \mathbf{b}_1(\mathbf{u}_\Gamma^0)\| + \sum_{j=1}^k \tau_r^j / \Delta x^2 \|\mathbf{u}_\Gamma^{k-j} - \mathbf{u}_\Gamma^{k-j+1}\| \right).$$

For k to infinity, the first term on the right converges to zero if and only if $\tau_r < 1$ and from Lemma 1 we know that the second term on the right converges to zero if and only if $\|\mathbf{u}_\Gamma^{k-j} - \mathbf{u}_\Gamma^{k-j+1}\|$ converges to zero, which is the case if $\tau_r < 1$ and if \mathbf{u}_Γ^k is

a convergent sequence. Note that these are the perturbed iterates of the twin iteration. Thus, we have to assume that that iteration converges.

If we choose the absolute termination criterion (6.4) or the relative one based on the right hand side (6.2), we obtain a bound of the form

$$\|\epsilon_k\| \leq \|\mathbf{A}_1^{-1}\| \tau_r \|\mathbf{b}(\mathbf{u}_1^k)\|,$$

respectively

$$\|\epsilon_k\| \leq \|\mathbf{A}_1^{-1}\| \tau_a.$$

Here, we cannot make a statement on the limit of ϵ_k .

In the second case, meaning the iteration with Neumann data (6.14), we obtain

$$\|\delta_k\| = \|\mathbf{u}_{2_\epsilon}^{k+1} - \mathbf{A}_2^{-1} \mathbf{b}_2(\mathbf{u}_{1_\epsilon}^{k+1})\| \leq \|\mathbf{A}_2^{-1}\| \|\mathbf{A}_2 \mathbf{u}_{2_\epsilon}^{k+1} - \mathbf{b}_2(\mathbf{u}_{1_\epsilon}^{k+1})\|$$

and analogous arguments produce the same results for δ_k , meaning that we obtain convergence to zero under the assumptions that $\tau_r < 1$ and that the perturbations ϵ_k are convergent. By theorem 6.1, we then have that when using the relative criterion we obtain convergence to the exact solution for any $\tau_r < 1$.

6.3 Testcase: Transmission Problem

We now consider the transmission problem (6.10). Specifically, we use $\Omega_1 = [0, 1] \times [0, 1]$, $\Omega_2 = [1, 2] \times [0, 1]$ and

$$\begin{aligned} f(x, y) = & \sin \pi y^2 (\pi \cos \frac{\pi}{2} x^2 - \pi^2 x^2 \sin \frac{\pi}{2} x^2) \\ & + \sin \frac{\pi}{2} x^2 (2\pi \cos \pi y^2 - 4\pi^2 y^2 \sin \pi y^2). \end{aligned} \quad (6.15)$$

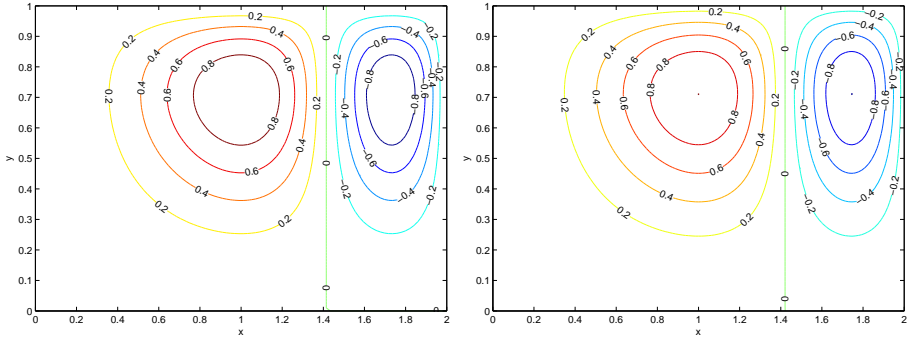
This was chosen such that the solution is

$$u(x, y) = \sin \pi y^2 \sin \frac{\pi}{2} x^2, \quad (6.16)$$

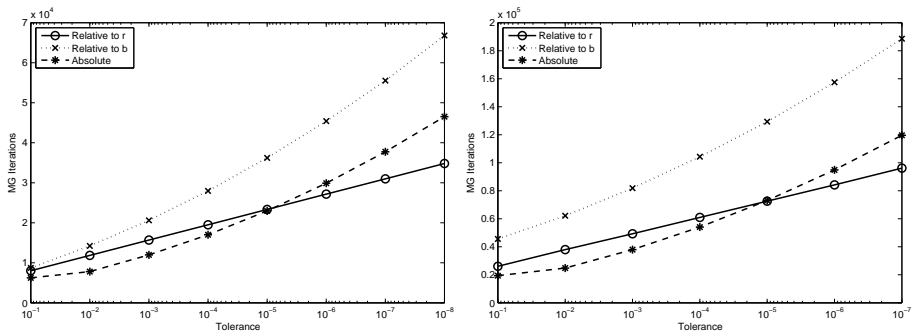
which satisfies the boundary conditions.

We discretize this problem using central differences with a constant mesh width of $\Delta x = \Delta y$. As initial guess for the Dirichlet-Neumann procedure, we employ a vector of all zeros. All linear systems are solved using a multigrid method. Specifically, we use a V cycle with exact solves for systems of dimension smaller than 16. Restriction and prolongation are standard full weighting, respectively bilinear interpolation [42], appropriately extended for the Neumann problem. The smoother is a damped Jacobi method with $\omega = 2/3$, which is applied thrice both as a pre- and a postsmoother. The exact solution and the discrete solution with $\Delta x = 1/64$ can be seen in Figure 11.

We now look at the convergence properties of the fixed point schemes for different termination criteria. Thus we choose $TOL = 1e-10$ and $\Delta x = 1/64$ and then run the

Figure 11. Exact and discrete solution with $\Delta x = 1/64$

iteration for different tolerances in multigrid. In the case of the termination criterion (6.3), the iteration recovers the reference solution for any tolerance τ_r , as predicted by the theory. For the relative termination criterion (6.2) and the absolute termination criterion (6.4) the situation is more complicated. We observe that for a given tolerance, $\|\mathbf{u}_\Gamma^{k+1} - \mathbf{u}_\Gamma^k\|$ does not converge to zero, but instead oscillates around a value that is proportional to the tolerance in multigrid. The error $\|\mathbf{u}_\Gamma - \mathbf{u}_\Gamma^*\|_2$ is then about half that value. This means that in practice, these schemes behave as if there were a constant perturbation and that the error can be controlled by choosing an appropriately small tolerance in the linear solver. Note that (6.2) is implemented in the MATLAB version of CG and that thus, a native implementation of the Dirichlet-Neumann iteration with CG in MATLAB will produce questionable results if not used with care.

Figure 12. Total multigrid iterations over tolerance for different termination criteria for the transmission problem with rhs (6.15) and $\Delta x = 1/128$ (left) and $\Delta x = 1/256$ (right).

We now compare the different termination criteria with regards to efficiency for values of TOL more relevant in practice. Hereby, we assume that the user wants to have a solution that is TOL close to the exact one. Based on the theory discussed here,

there are three choices: Using (6.3) with $\tau_r = 1e - 1$ independent of TOL, using (6.4) with $\tau_a = TOL$ or (6.2) with $\tau_r = TOL$, meaning that for the latter ones, we have to solve the inner iteration more accurately the more accurate we want the outer one. Hereby, we choose $\Delta x = 1/128$ and $\Delta x = 1/256$. As turns out, the error is in all cases about $0.4 \cdot TOL$. Since costs in the fixed point iterations not arising from solving the linear systems is negligible and the cost of one multigrid iteration is fixed, we compare the total number of multigrid iterations needed until termination. The results are depicted in Figure 12. As we can see, the scheme corresponding to (6.2) is the slowest. This is more pronounced on the finer grid, where it is about a factor of two slower than the other schemes. The only scheme where the computational effort has a linear behavior with regards to tolerance is that corresponding to (6.3). Thus, while slower for large tolerances, it beats the scheme with (6.4) at some point.

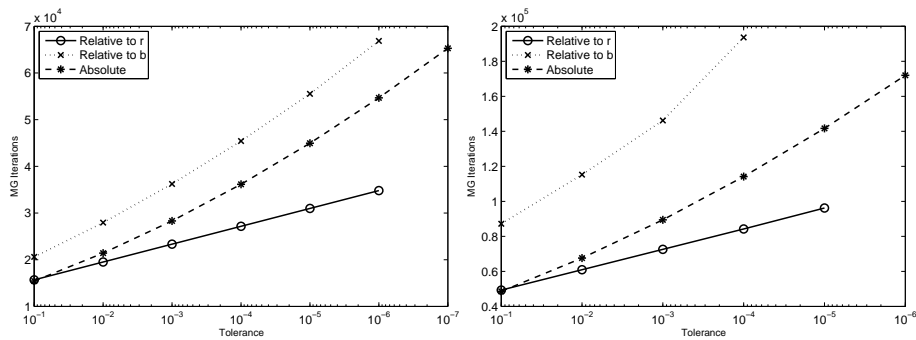


Figure 13. Total multigrid iterations over tolerance for different termination criteria for the transmission problem with rhs (6.15) multiplied by 100 and $\Delta x = 1/128$ (left) and $\Delta x = 1/256$ (right).

An absolute termination criterion can lead to oversolving. This is illustrated in the next test, where we just multiply the right hand side (6.15) with 100. The results are shown in Figure 13. Again, criterion (6.2) leads the slowest scheme. However, the scheme corresponding to (6.3) is now faster than that with criterion (6.4) for $TOL < 1e - 1$.

7 Numerical Results

The following results illustrate the performance of the complete solver described so far for realistic test cases of a steel-air coupling. Thus, the models from Section 2 are employed and discretized using FEM in the structure and FVM in the fluid. In particular, the DLR TAU-Code is employed [19], which is a cell-vertex-type finite volume method with AUSMDV as flux function and a linear reconstruction to increase the order of accuracy. The finite element code uses quadratic finite elements [47] and is the

inhouse code Native of the Institute for Static and Dynamic at the University of Kassel. The coupling between the solvers is done using the Component Template Library (CTL) of the University of Braunschweig [30]. The following numerical results are taken from [8].

7.1 Flow over a Plate

As a first test case, the cooling of a flat plate resembling a simple work piece is considered. The work piece is initially at a much higher temperature than the fluid and then cooled by a constant air stream, see Figure 14.

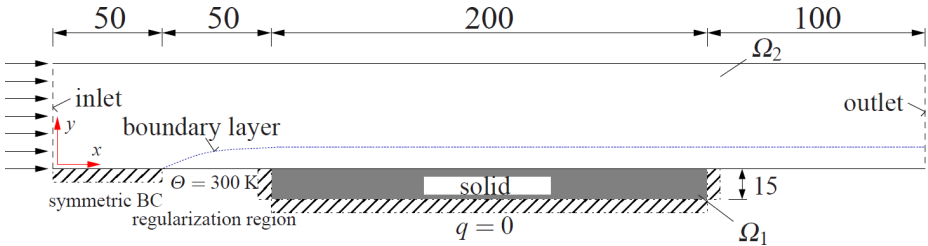


Figure 14. Test case for the coupling method

The inlet is given on the left, where air enters the domain with an initial velocity of $Ma_\infty = 0.8$ in horizontal direction and a temperature of 273 K. Then, there are two succeeding regularization regions of 50 mm to obtain an unperturbed boundary layer. In the first region, $0 \leq x \leq 50$, symmetry boundary conditions, $v_y = 0$, $q = 0$, are applied. In the second region, $50 \leq x \leq 100$, a constant wall temperature of 300 K is specified. Within this region the velocity boundary layer fully develops. The third part is the solid (work piece) of length 200 mm, which exchanges heat with the fluid, but is assumed insulated otherwise, thus $q_b = 0$. Therefore, Neumann boundary conditions are applied throughout. Finally, the fluid domain is closed by a second regularization region of 100 mm with symmetry boundary conditions and the outlet.

Regarding the initial conditions in the structure, a constant temperature of 900 K at $t = 0$ s is chosen throughout. To specify reasonable initial conditions within the fluid, a steady state solution of the fluid with a constant wall temperature $\Theta = 900$ K is computed.

The grid is chosen cartesian and equidistant in the structural part, where in the fluid region the thinnest cells are on the boundary and then become coarser in y -direction (see Figure 15). To avoid additional difficulties from interpolation, the points of the primary fluid grid, where the heat flux is located in the fluid solver, and the nodes of the structural grid are chosen to match on the interface Γ .

We now compare the different schemes for a whole simulation of 100 seconds real time. If not mentioned otherwise, the initial time step size is $\Delta t = 0.5$ s. To first give an impression on the effect of the time adaptive method, we look at fixed time step

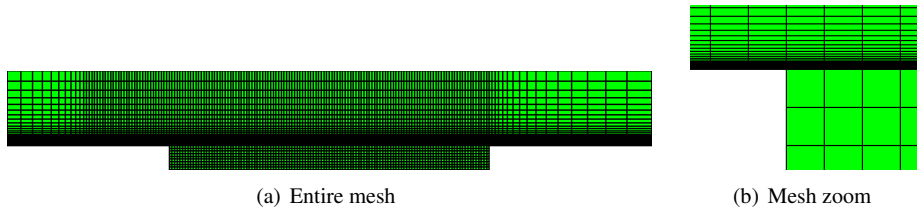


Figure 15. Full grid (left) and zoom into coupling region (right)

TOL	Fixed time step size		Time adapt.	Time adapt. with lin. extr.
10^{-2}	$\Delta t = 5s$	64	31	19
10^{-3}	$\Delta t = 5s$	82	39	31
10^{-4}	$\Delta t = 0.5s$	802	106	73
10^{-5}	$\Delta t = 0.5s$	1014	857	415

Table 4. Total number of iterations for 100 secs of real time. Fixed time step sizes versus adaptive steering. For time adaptive calculations, the initial time step is $\Delta t_0 = 0.5s$.

versus adaptive computations in Table 4. Thus, the different tolerances for the time adaptive case lead to different time step sizes and tolerances for the nonlinear system over the course of the algorithm, whereas in the fixed time step size, they steer only how accurate the nonlinear systems are solved. For the fixed time step case, we chose $\Delta t = 0.5s$ and $\Delta t = 5s$, which roughly corresponds to an error of 10^{-2} and 10^{-3} , respectively 10^{-4} . Thus, computations in one line of table 4 correspond to similar errors. As it can be seen, the time adaptive method is in the worst case a factor two faster and in the best case a factor of eight. Thus the time adaptive computation serves from now on as the base method for the construction of a fast solver.

Finally, we consider extrapolation based on the time integration scheme. As can be seen in Table 4, linear extrapolation speeds up the computations between 20% and 50%. Overall, we are thus able to simulate 100 seconds of real time for this problem for an engineering tolerance using only 19 calls to fluid and the structure solver each.

7.2 Cooling of a Flanged Shaft

To illustrate a realistic example of gas quenching, the cooling of a flanged shaft by cold high pressured air is used as a second test case. The complete process consists of the inductive heating of a steel rod, the forming of the hot rod into a flanged shaft, a transport to a cooling unit and the cooling process. Here, we consider only the cooling, meaning that we have a hot flanged shaft that is cooled by cold high pressured air coming out of small tubes [45]. We perform a two dimensional cut through the domain and assume symmetry along the vertical axis, resulting in one half of the flanged shaft and

two tubes blowing air at it, see figure 16. Since the air nozzles are evenly distributed around the flanged shaft, we use an axisymmetric model in the structure. The heat flux from the two-dimensional simulation of the fluid at the boundary of the flanged shaft is impressed axially symmetrical on the structure.

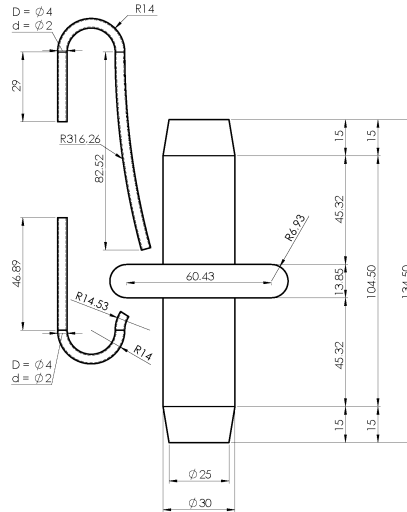


Figure 16. Sketch of the flanged shaft

We assume that the air leaves the tube in a straight and uniform way at a Mach number of 1.2. Furthermore, we assume a freestream in x -direction of Mach 0.005. This is mainly to avoid numerical difficulties at Mach 0, but could model a draft in the workshop. The Reynolds number is $Re = 2500$ and the Prandtl number $Pr = 0.72$.

The grid consists of 279212 cells in the fluid, which is the dual grid of an unstructured grid of quadrilaterals in the boundary layer and triangles in the rest of the domain, and 1997 quadrilateral elements in the structure. It is illustrated in Figure 17.

To obtain initial conditions for the subsequent tests, we use the following procedure: We define a first set of initial conditions by setting the flow velocity to zero throughout and choose the structure temperatures at the boundary points to be equal to temperatures that have been measured by a thermographic camera. Then, setting the y -axis on the axis of revolution of the flange, we set the temperature at each horizontal slice to the temperature at the corresponding boundary point. Finally, to determine the actual initial conditions, we compute 10^{-5} seconds of real time using the coupling solver with a fixed time step size of $\Delta t = 10^{-6} s$. This means, that the high pressured air is coming out of the tubes and the first front has already hit the flanged shaft. This solution is illustrated in Figure 18 (left).

Now, we compute 1 second of real time using the time adaptive algorithm with different tolerances and an initial time step size of $\Delta t = 10^{-6} s$. This small initial step

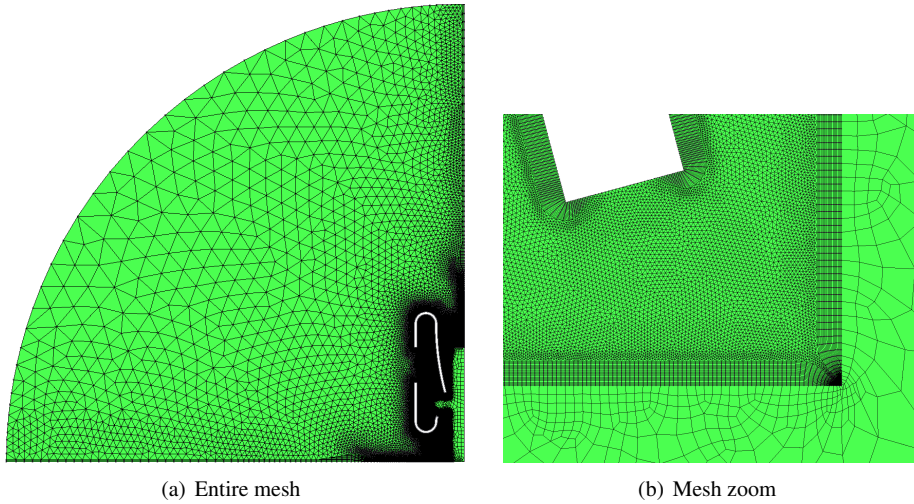


Figure 17. Full grid (left) and zoom into shaft region (right)

size is necessary to prevent instabilities in the fluid solver. During the course of the computation, the time step size is increased until it is on the order of $\Delta t = 0.1s$, which demonstrates the advantages of the time adaptive algorithm and reaffirms that it is this algorithm that we need to compare to. In total, the time adaptive method needs 22, 41, 130 and 850 time steps to reach $t = 1s$ for the different tolerances, compared to the 10^6 steps the fixed time step method would need. The solution at the final time is depicted in Figure 18 (right). As can be seen, the stream of cold air is deflected by the shaft.

Finally, we consider extrapolation based on the time integration scheme. In Table 5, the total number of iterations for 1 second of real time is shown. As before, the extrapolation causes a noticeable decrease in the total number of fixed point iterations. The speedup from linear extrapolation is between 18% and 34%, compared to the results obtained without extrapolation.

<i>TOL</i>	none	lin.
10^{-2}	51	42
10^{-3}	126	97
10^{-4}	414	309
10^{-5}	2768	1805

Table 5. Total number of iterations for 1 sec of real time for different extrapolation methods in time.

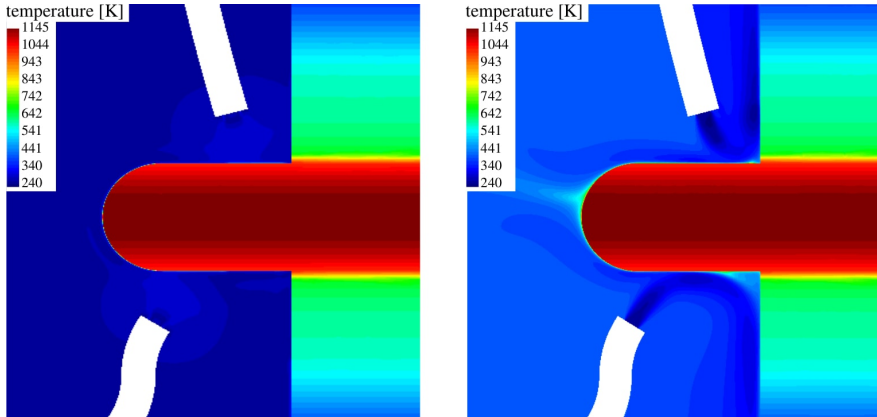


Figure 18. Temperature distribution in fluid and structure at $t = 0s$ (left) and $t = 1s$ (right)

8 Summary and Conclusions

We considered numerical methods for unsteady thermal fluid structure interaction. As a basic paradigm, we employed a partitioned method with the use of the Dirichlet-Neumann iteration. In this framework, a time adaptive SDIRK2 method with linear extrapolation of initial values is a fast solver. We cite three reasons for this: the time adaptivity, the very fast convergence rate of the Dirichlet-Neumann iteration for this problem and a smart choice of tolerances in the various nested solvers.

The fast convergence rate can be explained by estimating the spectral radius of the iteration matrix based on a model problem. This shows that the quotients of material properties play a crucial role. An analysis of the tolerances to be chosen shows that in the linear case, a nonstandard relative termination criterion based on the current residual leads to better performance compared to a standard termination criterion and to avoid pitfalls with possible wrong limits.

Bibliography

- [1] *Background on Tonight's Launch*, <http://www.spacex.com/news/2015/12/21/background-tonights-launch>, Accessed: 2016-08-23.
- [2] *CRS-8 Launch and Landing*, <http://www.spacex.com/news/2016/04/09/crs-8-launch-and-landing>, Accessed: 2016-08-23.
- [3] *The Why and How of Landing Rockets*, <http://www.spacex.com/news/2015/06/24/why-and-how-landing-rockets>, Accessed: 2016-08-23.
- [4] Y. Bazilevs, K. Takizawa and T. E. Tezduyar, *Computational Fluid-Structure Interaction: Methods and Applications*, Wiley, 2013.

-
- [5] O.O. Bendiksen, A new approach to computational aeroelasticity, *AIAA Paper* 91-09 (1991), 1712–1727.
- [6] P. Birken, *Numerical Methods for the Unsteady Compressible Navier-Stokes Equations*, Habilitation Thesis, University of Kassel, 2012.
- [7] ———, Termination criteria for inexact fixed point schemes, *Numer. Linear Algebra Appl.* (2015).
- [8] P. Birken, T. Gleim, D. Kuhl and A. Meister, Fast Solvers for Unsteady Thermal Fluid Structure Interaction, *Int. J. Numer. Meth. Fluids* 79(1) (2015), 16–29.
- [9] P. Birken, K.J. Quint, S. Hartmann and A. Meister, Choosing norms in adaptive FSI calculations, *PAMM* 10 (2010), 555–556.
- [10] ———, A time-adaptive fluid-structure interaction method for thermal coupling, *Comp. Vis. in Science* 13(7) (2011), 331–340.
- [11] D.L. Brown et al., *Applied Mathematics at the U.S. Department of Energy: Past, Present and a View to the Future*, tech. rep., Office of Science, U.S. Department of Energy, 2008.
- [12] J.M. Buchlin, Convective Heat Transfer and Infrared Thermography, *J. Appl. Fluid Mech.* 3 (2010), 55–62.
- [13] G.A. Davis and O.O. Bendiksen, Transonic Panel Flutter, *AIAA Paper* 93-1476 (1993).
- [14] S. Deparis, M.A. Fernández and L. Formaggia, Acceleration of a fixed point algorithm for fluid-structure interaction using transpiration conditions, *M2AN* 37(4) (2003), 601–616.
- [15] P. Erbts and A. Düster, Accelerated staggered coupling schemes for problems of thermoelasticity at finite strains, *Comp. & Math. with Appl.* 64 (2012), 2408–2430.
- [16] C. Farhat, CFD-based Nonlinear Computational Aeroelasticity, in *Encyclopedia of Computational Mechanics* (2004), 459–480, ch. 13.
- [17] C. Farhat, K.G. van der Zee and P. Geuzaine, Provably second-order time-accurate loosely-coupled solution algorithms for transient nonlinear computational aeroelasticity, *Comput. Methods Appl. Mech. Engrg.* 195 (2006), 1973–2001.
- [18] C.M. Fonseca and J. Petronilho, Explicit inverses of some tridiagonal matrices, *Linear Algebra Appl.* 325(1-3) (2001), 7–21.
- [19] T. Gerhold, O. Friedrich, J. Evans and M. Galle, Calculation of Complex Three-Dimensional Configurations Employing the DLR-TAU-Code, *AIAA Paper* 97-0167 (1997).
- [20] M.B. Giles, Stability Analysis of Numerical Interface Conditions in Fluid-Structure Thermal Analysis, *Int. J. Numer. Meth. Fluids* 25 (1997), 421–436.
- [21] M. Guenther, A. Kvaerno and P. Rentrop, Multirate partitioned Runge-Kutta methods, *BIT* 41 (2001), 504–514.
- [22] H. Guillard and C. Farhat, On the significance of the geometric conservation law for flow computations on moving meshes, *Comput. Methods Appl. Mech. Engrg.* 190 (2000), 1467–1482.

-
- [23] U. Heck, U. Fritsching and Bauckhage K., Fluid flow and heat transfer in gas jet quenching of a cylinder, *Int. J. Numer. Methods Heat Fluid Flow* 11 (2001), 36–49.
- [24] W.D. Henshaw and K.K. Chand, A composite grid solver for conjugate heat transfer in fluid-structure systems, *J. Comput. Phys.* 228 (2009), 2708–3741.
- [25] M. Hinderks and R. Radespiel, Investigation of Hypersonic Gap Flow of a Reentry Nosecap with Consideration of Fluid Structure Interaction, *AIAA Paper* 6 (2006), 2708–3741.
- [26] D. Kowollik, V. Tini, S. Reese and M. Haupt, 3D fluid-structure interaction analysis of a typical liquid rocket engine cycle based on a novel viscoplastic damage model, *Int. J. Numer. Methods Engrg.* 94 (2013), 1165–1190.
- [27] D.S.C. Kowollik, P. Horst and M.C. Haupt, Fluid-structure interaction analysis applied to thermal barrier coated cooled rocket thrust chambers with subsequent local investigation of delamination phenomena, *Progress in Propulsion Physics* 4 (2013), 617–636.
- [28] P. Le Tallec and J. Mouro, Fluid structure interaction with large structural displacements, *Comput. Methods Appl. Mech. Engrg.* 190 (2001), 3039–3067.
- [29] R. Massjung, Discrete conservation and coupling strategies in nonlinear aeroelasticity, *Comput. Methods Appl. Mech. Engrg.* 196 (2006), 91–102.
- [30] H. G. Matthies, R. Niekamp and J. Steindorf, Algorithms for strong coupling procedures, *Comput. Methods Appl. Mech. Engrg.* 195 (2006), 2028–2049.
- [31] R.C. Mehta, Numerical Computation of Heat Transfer on Reentry Capsules at Mach 5, *AIAA-Paper* 178 (2005).
- [32] C.D. Meyer, *Matrix Analysis and Applied Linear Algebra*, 2000.
- [33] H. Olsson and G. Söderlind, Stage value predictors and efficient Newton iterations in implicit Runge-Kutta methods, *SIAM J. Sci. Comput.* 20 (1998), 185–202.
- [34] A. Quarteroni and A. Valli, *Domain Decomposition Methods for Partial Differential Equations*, Oxford Science Publications, 1999.
- [35] K. J. Quint, S. Hartmann, S. Rothe, N. Saba and K. Steinhoff, Experimental validation of high-order time integration for non-linear heat transfer problems, *Comput. Mech.* 48 (2011), 81–96.
- [36] M.G. Reuter and J.C. Hill, An efficient, block-by-block algorithm for inverting a block tridiagonal, nearly block Toeplitz matrix, *Comp. Sci. Disc.* 5 (2012).
- [37] V. Savcenco, W. Hundsdorfer and J. G. Verwer, A multirate time stepping strategy for stiff ordinary differential equations, *BIT Numerical Mathematics* 47 (2007), 137–155.
- [38] G. Söderlind and L. Wang, Adaptive time-stepping and computational stability, *J. Comp. Appl. Math.* 185 (2006), 225 – 243.
- [39] P. R. Spalart and S. R. Allmaras, A One-Equation Turbulence Model for Aerodynamic Flows, *AIAA 30th Aerospace Science Meeting* 92–0439 (1992).
- [40] P. Stratton, I. Shedletsky and M. Lee, Gas Quenching with Helium, *Solid State Phenomena* 118 (2006), 221–226.

- [41] A. Toselli and O. Widlund, *Domain Decomposition Methods - Algorithms and Theory*, Springer, 2004.
- [42] U. Trottenberg, C. W. Oosterlee and S. Schüller, *Multigrid*, Elsevier Academic Press, 2001.
- [43] A. van Zuijlen and H. Bijl, Implicit and explicit higher order time integration schemes for structural dynamics and fluid-structure interaction computations, *Comp. & Struct.* 83 (2005), 93–105.
- [44] A. van Zuijlen, A. de Boer and H. Bijl, Higher-order time integration through smooth mesh deformation for 3D fluid-structure interaction simulations, *J. Comp. Phys.* 224 (2007), 414–430.
- [45] U. Weidig, N. Saba and K. Steinhoff, Massivumformprodukte mit funktional gradierten Eigenschaften durch eine differenzielle thermo-mechanische Prozessführung, *WT-Online* (2007), 745–752.
- [46] F. Zhang, *The Schur complement and its applications*, Springer, 2005.
- [47] O.C. Zienkiewicz and R.L. Taylor, *The Finite Element Method*, Butterworth Heinemann, 2000.

Author information

Philipp Birken, Lund University, Centre for the Mathematical Sciences, Numerical Analysis, Box 118, 22100 Lund, Sweden.

E-mail: philipp.birken@na.lu.se

Azahar Monge, Lund University, Centre for the Mathematical Sciences, Numerical Analysis, Box 118, 22100 Lund, Sweden.

E-mail: azahar.monge@na.lu.se

Paper II



On the convergence rate of the Dirichlet–Neumann iteration for unsteady thermal fluid–structure interaction

Azahar Monge¹ · Philipp Birken¹

Received: 11 May 2017 / Accepted: 3 November 2017
© The Author(s) 2017. This article is an open access publication

Abstract

We consider the Dirichlet–Neumann iteration for partitioned simulation of thermal fluid–structure interaction, also called conjugate heat transfer. We analyze its convergence rate for two coupled fully discretized 1D linear heat equations with jumps in the material coefficients across the interface. The heat equations are discretized using an implicit Euler scheme in time, whereas a finite element method on one domain and a finite volume method with variable aspect ratio on the other one are used in space. We provide an exact formula for the spectral radius of the iteration matrix. The formula indicates that for large time steps, the convergence rate is the aspect ratio times the quotient of heat conductivities and that decreasing the time step will improve the convergence rate. Numerical results confirm the analysis and show that the 1D formula is a very good estimator in 2D and even for nonlinear thermal FSI applications.

Keywords Solid–fluid interaction · Finite element · Stability · Converging · Numerical methods

1 Introduction

The Dirichlet–Neumann iteration is a basic method in both domain decomposition and fluid–structure interaction (FSI). In the latter case, the iteration arises in a partitioned approach [11], where different codes for the sub-problems are reused and the coupling is done by a master program which calls interface functions of the segregated codes. This allows to reuse existing software for each sub-problem, in contrast to a monolithic approach, where a new code is tailored for the coupled equations. To satisfy coupling conditions at the interface, the subsolvers are iterated by providing Dirichlet and Neumann data for the other solver in a sequential manner, giving rise to its name.

In the domain decomposition context, the iteration has two main problems, namely slow convergence and the need for an implementation using a red-black colouring. The slow convergence can be slightly improved using a relaxation procedure. In fluid–structure interaction, there are typically only two domains, coupled along an interface, making the application straight forward. The convergence rate for the

interaction of a flexible structure with a fluid has been analyzed in [35]. There, the added mass effect is proven to be dependent on the step size for compressible flows and independent for incompressible flows. However, the convergence rate is not great for some FSI couplings, which is why a lot of effort goes into convergence acceleration [10]. An alternative are optimized Schwarz methods [13,15,29] and the CHAMP scheme (Conjugate Heat transfer Advanced Multi-domain Partitioned) which uses a generalized Robin (mixed) condition at the interface to accelerate the iteration [24], but overlapping domains. Furthermore, for incompressible fluids it is known that the ratio of densities of the materials plays an important role [1,9]. Finally, the Dirichlet–Neumann iteration was reported to be a very fast solver for thermal fluid–structure interaction between air and steel [4].

Our prime motivation here is thermal interaction between fluids and structures, also called conjugate heat transfer. In particular, we consider two domains with jumps in the material coefficients across the connecting interface. Conjugate heat transfer plays an important role in many applications and its simulation has proved essential [2]. Examples for thermal fluid–structure interaction are cooling of gas-turbine blades, thermal anti-icing systems of airplanes [8], supersonic reentry of vehicles from space [19,23], gas quenching, which is an industrial heat treatment of metal workpieces [17,33] or the cooling of rocket nozzles [20,21].

Azahar Monge
azahar.monge@na.lu.se

¹ Centre for Mathematical Sciences, Lund University, Box 118, 22100 Lund, Sweden

A convergence analysis of the Dirichlet–Neumann iteration for an FE–FE discretization of a steady heat transfer problem can be found in [32]. Asymptotically, it is found to be the quotient of heat conductivities. In this paper, we present a convergence analysis of the unsteady heat transfer problem. For this model, a 1D stability analysis was presented by Giles [16]. There, an explicit time integration method was chosen with respect to the interface unknowns. On the other hand, Henshaw and Chand provided in [18] a method to analyze stability and convergence speed of the Dirichlet–Neumann iteration in 2D based on applying the continuous Fourier transform to the semi-discretized equations. They show that the ratios of thermal conductivities and diffusivities of the materials play an important role. This is similar to the behavior mentioned above for classical FSI with incompressible fluids where the performance is affected by the ratio of densities of the materials [1,9].

However, in the fully discrete case we observe in some cases that the iteration behaves differently, because some aspects of the problem are not taken into account by the semidiscrete analysis: the effect of Δt is not accurately represented and neither are possibly different mesh widths in the two problems. This matters particularly for compressible fluids where a high aspect ratio grid is needed to accurately represent the boundary layer. This leads to geometric stiffness that significantly influences the convergence rate, as we will show.

For the fully discrete case, the convergence rate is in principle analyzed in any standard book on domain decomposition methods, e.g. [30,34]. There, the iteration matrix is derived in terms of the stiffness and mass matrices of finite element discretizations and the convergence rate is the spectral radius of that. However, this does not provide a quantitative answer, since the spectral radius is unknown. Computing it is in general a nontrivial task. In our context, the material properties are discontinuous across the interface and as a consequence, computing the spectral radius of the iteration matrix is even more difficult.

In [26,28], a convergence analysis of the Dirichlet–Neumann iteration for the unsteady transmission problem using finite element methods (FEM) on both subdomains is presented. A similar analysis using finite differences (FDM) on one domain and FEM on the other one can be found in [27]. In addition, the corresponding analysis when coupling finite volumes (FVM) with FEM is described in [5,26]. All these results assume equal mesh sizes on both subdomains, i.e. the aspect ratio is equal to one.

Thus, we present here a complete discretization of the coupled problem using FVM in space on one domain and FEM on the other one with variable aspect ratio r . We consider this to be a relevant case, because these are the standard discretizations for the subproblems. The implicit Euler method is used for the temporal discretization. Then, we derive the spectral

radius of the iteration matrix exactly in terms of the eigendecomposition of the resulting matrices for the one dimensional case. The asymptotic convergence rates when approaching the continuous case in either time or space are also determined. In the spatial limit, the convergence rate turns out to be proportional to the aspect ratio r , whereas in the temporal limit, we obtain 0. Note that the temporal limit turns out to be the ratio of the product of densities and heat capacities of the materials for FEM–FEM couplings instead. Moreover, we also include numerical results where it is shown that the one dimensional formula is a very good estimator for a 2D version of the coupled heat equations and for two nonlinear FSI models, namely the cooling of a flat plate and the cooling of a flanged shaft.

An outline of the paper now follows. In Sect. 2, we describe the model and discretization, as well as the coupling conditions and the Dirichlet–Neumann iteration. Two thermal FSI test cases are introduced in Sect. 3: the cooling of a flat plate and of a flanged shaft. For these, we present numerical convergence rates, motivating further analysis. A model problem, consisting of two coupled discretized heat equations, is presented in Sect. 4 and then analyzed in 1D in Sect. 5. In Sect. 6, an extension of the analysis to 2D and different discretizations are discussed. In Sect. 7, the analytical results are compared to linear and nonlinear numerical results.

2 Thermal FSI methodology

We consider the basic setting where on a domain $\Omega_1 \subset \mathbb{R}^d$ where d corresponds to the spatial dimension, the physics is described by a fluid model, whereas on a domain $\Omega_2 \subset \mathbb{R}^d$, a different model describing the structure is used. The two domains are connected via an interface. The part of the interface where the fluid and the structure are supposed to interact is called the coupling interface $\Gamma \subset \partial\Omega_1 \cap \partial\Omega_2$. Note that Γ might be a true subset of the intersection, because the structure could be insulated. At the interface Γ , coupling conditions are prescribed that model the interaction between fluid and structure. For the thermal coupling problem, these conditions are that temperature and the normal component of the heat flux are continuous across the interface.

2.1 Fluid model

We model the fluid using the time dependent compressible Navier–Stokes equations, which are a second order system of conservation laws (mass, momentum, energy) modeling compressible flow. We consider the two dimensional case, written in conservative variables density ρ , momentum $\mathbf{m} = \rho\mathbf{v}$ and energy per unit volume ρE as:

$$\begin{aligned} \partial_t \rho + \nabla \cdot \rho \mathbf{v} &= 0, \\ \partial_t \rho v_i + \sum_{j=1}^2 \partial_{x_j} (\rho v_i v_j + p \delta_{ij}) &= \frac{1}{Re} \sum_{j=1}^2 \partial_{x_j} S_{ij}, \quad i = 1, 2, \\ \partial_t \rho E + \nabla \cdot (\rho H v_j) &= \frac{1}{Re} \sum_{j=1}^2 \partial_{x_j} \left(S_{ij} v_i + \frac{q_j}{Pr} \right). \end{aligned} \tag{1}$$

Here, enthalpy is given by $H = E + p/\rho$ with $p = (\gamma - 1)\rho(E - 1/2|v|^2)$ being the pressure and $\gamma = 1.4$ the adiabatic index for an ideal gas. Furthermore, $\mathbf{q}_f = (q_1, q_2)^T$ represents the heat flux and $\mathbf{S} = (S_{ij})_{i,j=1,2}$ the viscous shear stress tensor with $S_{ij} = \mu ((\partial_{x_j} v_i + \partial_{x_i} v_j) - \frac{2}{3} \delta_{ij} \nabla \cdot \mathbf{v})$. The nondimensional dynamic viscosity μ is given by the Sutherland law

$$\mu = T^{\frac{3}{2}} \left(\frac{1 + Su}{T + Su} \right)$$

with the Sutherland constant $Su = \frac{110\text{K}}{273\text{K}}$. As the equations are dimensionless, the Reynolds number Re and the Prandtl number Pr appear. The system is closed by the equation of state for the pressure $p = (\gamma - 1)\rho e$, the Sutherland law representing the correlation between temperature and viscosity, as well as the Stokes hypothesis. Additionally, we prescribe appropriate boundary conditions at the boundary of Ω_1 except for Γ , where we have the coupling conditions. In the Dirichlet–Neumann coupling, a temperature value is enforced at Γ .

2.2 Structure model

Regarding the structure model, we will consider heat conduction only. Thus, we have the nonlinear heat equation for the structure temperature Θ

$$\rho(\mathbf{x})c_p(\Theta) \frac{d}{dt} \Theta(\mathbf{x}, t) = -\nabla \cdot \mathbf{q}(\mathbf{x}, t), \tag{2}$$

where

$$\mathbf{q}_s(\mathbf{x}, t) = -\lambda(\Theta)\nabla\Theta(\mathbf{x}, t)$$

denotes the heat flux vector. For alloys, the specific heat capacity c_p and heat conductivity λ are temperature-dependent and highly nonlinear.

As an example, we will consider the empirical model for the steel 51CrV4 suggested in [31]. This was obtained from measurements and a least squares fit. The coefficient functions are then

$$\lambda(\Theta) = 40.1 + 0.05\Theta - 0.0001\Theta^2 + 4.9 \cdot 10^{-8}\Theta^3 \tag{3}$$

and

$$c_p(\Theta) = -10 \ln \left(\frac{e^{-c_{p1}(\Theta)/10} + e^{-c_{p2}(\Theta)/10}}{2} \right) \tag{4}$$

with

$$c_{p1}(\Theta) = 34.2e^{0.0026\Theta} + 421.15 \tag{5}$$

and

$$c_{p2}(\Theta) = 956.5e^{-0.012(\Theta-900)} + 0.45\Theta. \tag{6}$$

For the mass density, one obtains $\rho = 7836 \text{ kg/m}^3$.

Finally, on the boundary, we have Neumann conditions $\mathbf{q}_s(\mathbf{x}, t) \cdot \mathbf{n}(\mathbf{x}) = q_b(\mathbf{x}, t)$.

2.3 Coupling conditions

As mentioned before, the coupling conditions are that temperature and the normal component of the heat flux are continuous across the interface, i.e.;

$$T(\mathbf{x}, t) = \Theta(\mathbf{x}, t), \quad \mathbf{x} \in \Gamma, \tag{7}$$

where T is the fluid temperature and Θ the structure temperature and

$$\mathbf{q}_f(\mathbf{x}, t) \cdot \mathbf{n}(\mathbf{x}) = \mathbf{q}_s(\mathbf{x}, t) \cdot \mathbf{n}(\mathbf{x}), \quad \mathbf{x} \in \Gamma. \tag{8}$$

2.4 Discretization in space

Following the partitioned coupling approach, we discretize the two models separately in space. For the fluid model (1), we use a finite volume method, leading to the following equation for all unknowns on Ω_1 , collected in the vector $\mathbf{u} \in \mathbb{R}^{N_f}$:

$$\frac{d}{dt} \mathbf{u} + \mathbf{h}(\mathbf{u}, \Theta_\Gamma) = \mathbf{0}, \tag{9}$$

where $\mathbf{h}(\mathbf{u}, \Theta_\Gamma)$ represents the nonlinear finite element spatial discretization and its dependence on the temperatures on the discrete interface to the structure, here denoted by Θ_Γ .

In structural mechanics, the use of finite element methods is ubiquitous. Therefore, we will also follow that approach here. Using quadratic finite element, one obtains the following nonlinear equation for all unknowns on Ω_2 :

$$\mathbf{M}(\Theta) \frac{d}{dt} \Theta + \mathbf{A}(\Theta)\Theta = \mathbf{q}_b^f + \mathbf{q}_b^r(\mathbf{u}). \tag{10}$$

Here, \mathbf{M} is the mass matrix, also called heat capacity matrix for this problem and \mathbf{A} is the heat conductivity and stiffness matrix, respectively. The vector $\Theta \in \mathbb{R}^{N_s}$ consists of all

discrete temperature unknowns and $\mathbf{q}_b^{\Gamma}(\mathbf{u})$ is the discrete heat flux vector on the coupling interface to the fluid, whereas \mathbf{q}_b^f corresponds to boundary heat fluxes independent of the fluid, for example at insulated boundaries.

2.5 Time discretization

In time, we use the implicit Euler method with constant time step Δt . For the system (9)–(10) we obtain

$$\mathbf{u}^{n+1} - \mathbf{u}^n + \Delta t \mathbf{h}(\mathbf{u}^{n+1}, \Theta_r^{n+1}) = \mathbf{0}, \tag{11}$$

$$\begin{aligned} \mathbf{M}(\Theta^{n+1})(\Theta^{n+1} - \Theta^n) + \Delta t \mathbf{A}(\Theta^{n+1})\Theta^{n+1} \\ = \Delta t (\mathbf{q}_b^f + \mathbf{q}_b^{\Gamma}(\mathbf{u}^{n+1})). \end{aligned} \tag{12}$$

2.6 The Dirichlet–Neumann method

The Dirichlet–Neumann method is a basic iterative substructuring method in domain decomposition and it is a common choice for treating FSI problems. Therefore, we now employ it to solve the system (11)–(12). This corresponds to alternately solving equation (11) on Ω_1 with Dirichlet data on Γ and (12) on Ω_2 with Neumann data on Γ .

Thus, one gets for the k th iteration the two decoupled equation systems

$$\mathbf{u}^{n+1,k+1} - \mathbf{u}^n + \Delta t \mathbf{h}(\mathbf{u}^{n+1,k+1}, \Theta_r^{n+1,k}) = \mathbf{0}, \tag{13}$$

$$\begin{aligned} \mathbf{M}(\Theta^{n+1,k+1})(\Theta^{n+1,k+1} - \Theta^n) + \Delta t \mathbf{A}(\Theta^{n+1,k+1})\Theta^{n+1,k+1} \\ = \Delta t (\mathbf{q}_b^f + \mathbf{q}_b^{\Gamma}(\mathbf{u}^{n+1,k+1})), \end{aligned} \tag{14}$$

with some initial condition $\Theta(t = 0)|_{\Gamma} = \Theta_r^0$. The iteration is terminated according to the standard criterion

$$\|\Theta_r^{k+1} - \Theta_r^k\| \leq \tau \tag{15}$$

where τ is a user defined tolerance.

3 Thermal FSI test cases

In this section we present two thermal FSI test cases that are solved using the methodology explained in the previous section. The aim of this paper is to estimate the convergence rate

of the Dirichlet–Neumann iteration used as a solver for thermal FSI problems. Therefore, we first want to illustrate the behavior for two examples before proceeding to the convergence analysis in the next section. We consider the cooling of a flat plate and the cooling of a flanged shaft. For the first problem, structured grids are used and for the second, unstructured grids.

For the coupling, the Dirichlet–Neumann method as presented in (13)–(14) is used. A fixed tolerance of $1e-8$ is chosen for all involved equation solvers. The coupling code used has been developed in a series of papers [4,6,7]. Its main feature is time adaptivity, which is not employed here. The coupling between the solvers is done using the Component Template Library (CTL) [22]. In the fluid, the DLR TAU-Code in its 2014.2 version is employed [14], which is a cell-vertex-type finite volume method with AUSMDV as flux function and a linear reconstruction to increase the order of accuracy. The finite element code uses quadratic finite elements and is the inhouse code Native of the Institute for Static and Dynamic at the University of Kassel.

3.1 Flow over a plate

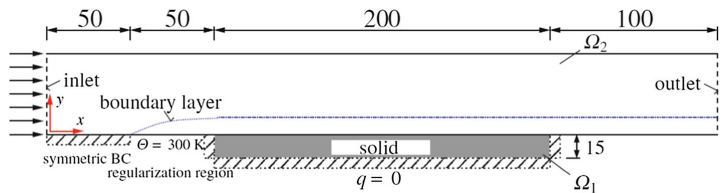
The first test case is the cooling of a flat steel plate resembling a simple work piece [7]. It is initially at a much higher temperature than the fluid and then cooled by a constant laminar air stream, see Fig. 1.

The inlet is located on the left, where air enters the domain with an initial velocity of $Ma_{\infty} = 0.8$ in horizontal direction and a temperature of 273 K. Regarding the initial condition in the structure, a constant temperature of 900 K at $t = 0$ is chosen throughout. To determine the Reynolds number, a reference length of $\hat{x}_{ref} = 0.2$ m is chosen.

The grid is chosen Cartesian and equidistant in the structural part. In the fluid region the thinnest cells touch the boundary and then get coarser in y -direction with a maximal aspect ratio of $r = 1.7780e5$. The points of the primary fluid grid and the nodes of the structural grid match on the interface Γ and there are 9660 cells in the fluid region and $n_x \times n_y = 120 \times 9 = 1080$ elements with $121 \times 10 = 1210$ nodes in the region of the structure.

The left plot in Fig. 4 shows the convergence behaviour of the Dirichlet–Neumann iteration against the time step Δt . One observes that the convergence rates is roughly propor-

Fig. 1 Sketch of the cooling of a flat plate



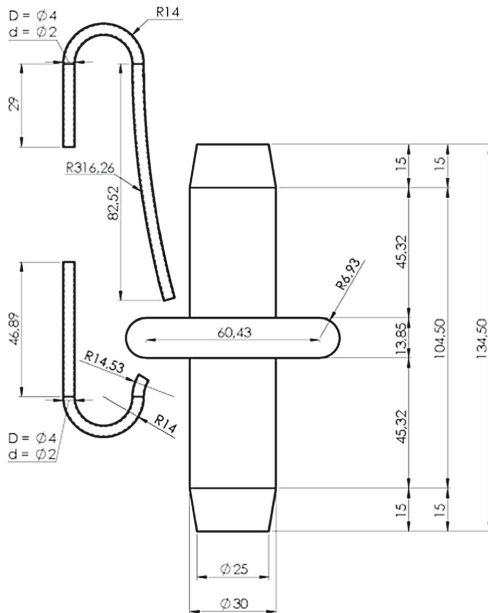


Fig. 2 Sketch of the cooling of the flanged shaft

tional to the time step Δt . Furthermore, even for $\Delta t = 1$ a reduction of the error by a factor of ten per iteration is achieved.

3.2 Cooling of a flanged shaft

The second test case is the cooling of a flanged steel shaft by cold high pressured air (this process is also known as gas quenching) [36]. Here, we have a hot flanged shaft that is cooled by cold high pressured air coming out of small tubes, see Fig. 2. We assume symmetry along the vertical axis in order to consider one half of the flanged shaft and two tubes blowing air at it. We also assume that the air leaves the tube in a straight and uniform way at a Mach number of 1.2, as well as a freestream in x -direction of Mach 0.005. The Reynolds number, based on a reference length of $x_{ref} = 0.02$ m is $Re = 2500$ and the Prandtl number $Pr = 0.72$.

The grid, see Fig. 3, consists of 279,212 cells in the fluid, which is the dual grid of an unstructured grid of quadrilaterals in the boundary layer and triangles in the rest of the domain, and 1997 quadrilateral elements in the structure. Regarding the initial conditions, we use the procedure from [4]: first freestream values are set overall in the fluid and temperatures from a thermographic camera in the structure. Then 10^{-5} s of real time are computed using a time step of $\Delta t = 10^{-6}$ s.

The right plot in Fig. 4 shows the convergence behaviour of the Dirichlet–Neumann iteration against the time step Δt . The convergence rate is again about proportional to the time step size and the iteration is again convergent even for very large time steps. If we compare the rates for the two problems, we observe that for a given Δt , the iteration is about a factor ten faster for the plate.

Summarizing, the Dirichlet–Neumann iteration is a very fast solver for thermal FSI cases with strong jumps in the material coefficients, as the ones presented here. To understand this better, we perform in the next section a convergence analysis for the case of two coupled linear heat equations.

4 A model problem: coupled heat equations

We present here a convergence analysis of the unsteady transmission problem with mixed discretizations. In particular, we choose a finite volume method (FVM) on the first subdomain and a finite element method (FEM) on the second subdomain.

4.1 Model problem

The unsteady transmission problem reads as follows, where we consider a domain $\Omega \subset \mathbb{R}^d$ which is cut into two subdomains $\Omega_1 \cup \Omega_2 = \Omega$ with transmission conditions at the interface $\Gamma = \partial\Omega_1 \cap \partial\Omega_2$:

$$\begin{aligned} \alpha_m \frac{\partial u_m(\mathbf{x}, t)}{\partial t} - \nabla \cdot (\lambda_m \nabla u_m(\mathbf{x}, t)) &= 0, \\ \mathbf{x} \in \Omega_m \subset \mathbb{R}^d, \quad m &= 1, 2, \\ u_m(\mathbf{x}, t) &= 0, \quad \mathbf{x} \in \partial\Omega_m \setminus \Gamma, \\ u_1(\mathbf{x}, t) &= u_2(\mathbf{x}, t), \quad \mathbf{x} \in \Gamma, \\ \lambda_2 \frac{\partial u_2(\mathbf{x}, t)}{\partial \mathbf{n}_2} &= -\lambda_1 \frac{\partial u_1(\mathbf{x}, t)}{\partial \mathbf{n}_1}, \quad \mathbf{x} \in \Gamma, \\ u_m(\mathbf{x}, 0) &= u_m^0(\mathbf{x}), \quad \mathbf{x} \in \Omega_m, \end{aligned} \tag{16}$$

where $t \in [t_0, t_f]$ and \mathbf{n}_m is the outward normal to Ω_m for $m = 1, 2$.

The constants λ_1 and λ_2 describe the thermal conductivities of the materials on Ω_1 and Ω_2 , respectively. D_1 and D_2 represent the thermal diffusivities of the materials and they are defined by

$$D_m = \frac{\lambda_m}{\alpha_m}, \quad \text{with } \alpha_m = \rho_m c_{p_m} \tag{17}$$

where ρ_m represents the density and c_{p_m} the specific heat capacity of the material placed in Ω_m , $m = 1, 2$.

Fig. 3 Full grid (left) and zoom into shaft region (right)

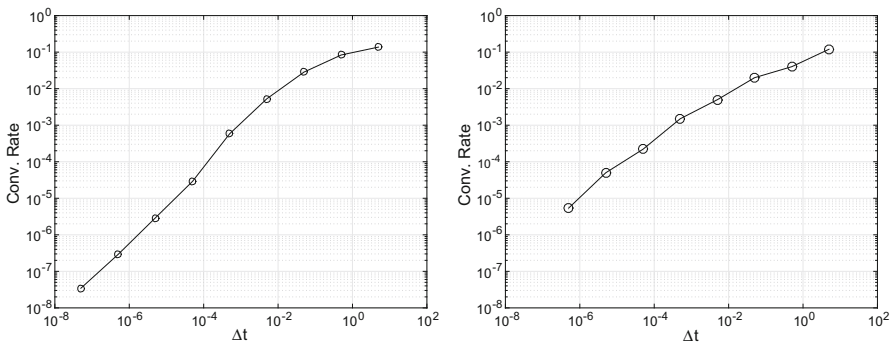
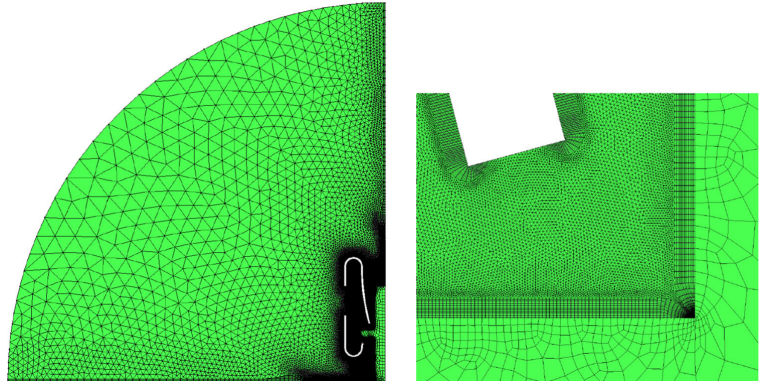


Fig. 4 Convergence behavior of the cooling systems with respect to Δt . Left: Test case 1: flow over a plate. Right: Test case 2: cooling of a flanged shaft

4.2 Semidiscrete analysis

Before we present in the next section an analysis for the fully discrete equations, we want to describe previous results about the behaviour of the Dirichlet–Neumann iteration for the transmission problem in the semidiscrete case.

Henshaw and Chand applied in [18] the implicit Euler method for the time discretization on both equations in (16) but kept the space continuous. Then, they applied the Fourier transform in space (with dual variable k) in order to transform the second order derivatives into algebraic expressions. Once they have a coupled system of algebraic equations, they insert one into the other and obtain the Dirichlet–Neumann convergence rate β :

$$\beta = \left| \frac{\lambda_1 \sqrt{\frac{1}{(D_1 \Delta t) + k^2} \tanh\left(\frac{-\sqrt{1/(D_2 \Delta t) + k^2}}{1/(D_2 \Delta t) + k^2}\right)}}{\lambda_2 \sqrt{\frac{1}{(D_2 \Delta t) + k^2} \tanh\left(\frac{\sqrt{1/(D_1 \Delta t) + k^2}}{1/(D_1 \Delta t) + k^2}\right)}} \right|. \tag{18}$$

In the one dimensional case, the transverse Fourier mode k is zero. Then, for Δt small enough, we have $\tanh\left(-1/\sqrt{D_2 \Delta t}\right) \approx -1$ and $\tanh\left(1/\sqrt{D_1 \Delta t}\right) \approx 1$ and therefore:

$$\beta \approx \frac{\lambda_1}{\lambda_2} \sqrt{\frac{D_2}{D_1}}. \tag{19}$$

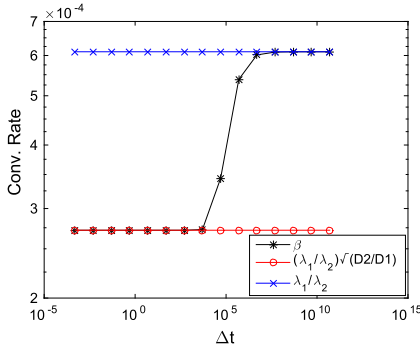


Fig. 5 Semidiscrete estimator β in (18) against Δt in 1D

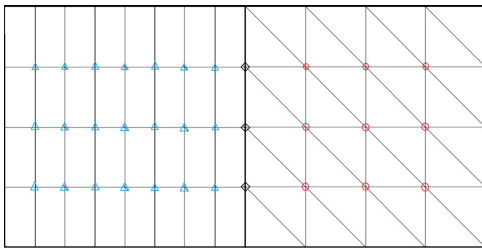


Fig. 6 Splitting of Ω between finite volumes and finite elements

On the other hand, for Δt big enough, we have $\tanh(-1/\sqrt{D_2\Delta t}) \approx -1/\sqrt{D_2\Delta t}$ and $\tanh(1/\sqrt{D_1\Delta t}) \approx 1/\sqrt{D_1\Delta t}$ and therefore:

$$\beta \approx \frac{\lambda_1}{\lambda_2} \sqrt{\frac{D_2}{D_1}} \frac{\sqrt{D_1\Delta t}}{\sqrt{D_2\Delta t}} = \frac{\lambda_1}{\lambda_2}. \tag{20}$$

Figure 5 shows β as a function of Δt for $k = 0$. It is almost constant, except for a short dynamic transition between $(\lambda_1/\lambda_2)\sqrt{D_2/D_1}$ and λ_1/λ_2 .

Finally, one observes in (20) that the convergence rates of the Dirichlet–Neumann iteration are given by the quotient of thermal conductivities for Δt large. This suggests that strong jumps in the thermal conductivities of the materials give fast convergence.

4.3 Space discretization

We now describe a rather general space discretization of the model problem. The core property we need is that the meshes of Ω_1 and Ω_2 share the same nodes on Γ as shown in Fig. 6. Furthermore, we assume that there is a specific set of unknowns associated with the interface nodes. Otherwise, we allow at this point for arbitrary meshes on both sides.

Then, letting $\mathbf{u}_I^{(m)}$ correspond to the unknowns on Ω_m , $m = 1, 2$, and \mathbf{u}_Γ to the unknowns at the interface Γ , we can write a general discretization of the first equation in (16) in a compact form as:

$$\mathbf{M}_m \dot{\mathbf{u}}_I^{(m)} + \mathbf{M}_{\Gamma\Gamma}^{(m)} \dot{\mathbf{u}}_\Gamma + \mathbf{A}_m \mathbf{u}_I^{(m)} + \mathbf{A}_{\Gamma\Gamma}^{(m)} \mathbf{u}_\Gamma = \mathbf{0}. \tag{21}$$

To close the system, we need an approximation of the normal derivatives on Γ . For the FVM on Ω_1 , we approximate the normal derivative of u_1 with respect to the interface using second order one-sided finite differences:

$$-\lambda_1 \frac{\partial u_1}{\partial \mathbf{n}_1} \approx \frac{\lambda_1}{2\Delta x} (4u_{1,N}(t) - u_{1,N-1}(t) - 3u_\Gamma). \tag{22}$$

Now, let ϕ_j be a nodal FE basis function on Ω_2 for a node on Γ we observe that the normal derivative of u_2 with respect to the interface can be written as a linear functional using Green’s formula [34, pp. 3]. Thus, the approximation of the normal derivative is given by

$$\begin{aligned} \lambda_2 \int_\Gamma \frac{\partial u_2}{\partial \mathbf{n}_2} \phi_j dS &= \lambda_2 \int_{\Omega_2} (\Delta u_2 \phi_j + \nabla u_2 \nabla \phi_j) dx \\ &= \alpha_2 \int_{\Omega_2} \frac{d}{dt} u_2 \phi_j + \lambda_2 \int_{\Omega_2} \nabla u_2 \nabla \phi_j dx. \end{aligned} \tag{23}$$

Consequently, the equation

$$\begin{aligned} \mathbf{M}_{\Gamma\Gamma}^{(2)} \dot{\mathbf{u}}_\Gamma + \mathbf{M}_{\Gamma I}^{(2)} \dot{\mathbf{u}}_I^{(2)} + \mathbf{A}_{\Gamma\Gamma}^{(2)} \mathbf{u}_\Gamma + \mathbf{A}_{\Gamma I}^{(2)} \mathbf{u}_I^{(2)} \\ = -\mathbf{M}_{\Gamma\Gamma}^{(1)} \dot{\mathbf{u}}_\Gamma - \mathbf{M}_{\Gamma I}^{(1)} \dot{\mathbf{u}}_I^{(1)} - \mathbf{A}_{\Gamma\Gamma}^{(1)} \mathbf{u}_\Gamma - \mathbf{A}_{\Gamma I}^{(1)} \mathbf{u}_I^{(1)} \end{aligned} \tag{24}$$

is a discrete version of the fourth equation in (16) and completes the system (21). Notice that the left hand side of (24) comes from (23) and the right hand side from (22). We can now write the coupled equations (21) and (24) as an ODE for the vector of unknowns $\mathbf{u} = (\mathbf{u}_I^{(1)}, \mathbf{u}_I^{(2)}, \mathbf{u}_\Gamma)^T$

$$\tilde{\mathbf{M}} \dot{\mathbf{u}} + \tilde{\mathbf{A}} \mathbf{u} = \mathbf{0}, \tag{25}$$

where

$$\begin{aligned} \tilde{\mathbf{M}} &= \begin{pmatrix} \mathbf{M}_1 & \mathbf{0} & \mathbf{M}_{\Gamma\Gamma}^{(1)} \\ \mathbf{0} & \mathbf{M}_2 & \mathbf{M}_{\Gamma\Gamma}^{(2)} \\ \mathbf{M}_{\Gamma I}^{(1)} & \mathbf{M}_{\Gamma I}^{(2)} & \mathbf{M}_{\Gamma\Gamma}^{(1)} + \mathbf{M}_{\Gamma\Gamma}^{(2)} \end{pmatrix}, \\ \tilde{\mathbf{A}} &= \begin{pmatrix} \mathbf{A}_1 & \mathbf{0} & \mathbf{A}_{\Gamma\Gamma}^{(1)} \\ \mathbf{0} & \mathbf{A}_2 & \mathbf{A}_{\Gamma\Gamma}^{(2)} \\ \mathbf{A}_{\Gamma I}^{(1)} & \mathbf{A}_{\Gamma I}^{(2)} & \mathbf{A}_{\Gamma\Gamma}^{(1)} + \mathbf{A}_{\Gamma\Gamma}^{(2)} \end{pmatrix}. \end{aligned}$$

4.4 Time discretization

Applying the implicit Euler method with time step Δt to the system (24), we get for the vector of unknowns $\mathbf{u}^{n+1} =$

$$(\mathbf{u}_I^{(1),n+1}, \mathbf{u}_I^{(2),n+1}, \mathbf{u}_I^{n+1})^T$$

$$\mathbf{A}\mathbf{u}^{n+1} = \tilde{\mathbf{M}}\mathbf{u}^n, \tag{26}$$

where

$$\mathbf{A} = \tilde{\mathbf{M}} + \Delta t \tilde{\mathbf{A}}$$

$$= \begin{pmatrix} \mathbf{M}_1 + \Delta t \mathbf{A}_1 & \mathbf{0} & \mathbf{M}_{I\Gamma}^{(1)} + \Delta t \mathbf{A}_{I\Gamma}^{(1)} \\ \mathbf{0} & \mathbf{M}_2 + \Delta t \mathbf{A}_2 & \mathbf{M}_{I\Gamma}^{(2)} + \Delta t \mathbf{A}_{I\Gamma}^{(2)} \\ \mathbf{M}_{\Gamma I}^{(1)} + \Delta t \mathbf{A}_{\Gamma I}^{(1)} & \mathbf{M}_{\Gamma I}^{(2)} + \Delta t \mathbf{A}_{\Gamma I}^{(2)} & \mathbf{M}_{\Gamma\Gamma} + \Delta t \mathbf{A}_{\Gamma\Gamma} \end{pmatrix},$$

with $\mathbf{M}_{\Gamma\Gamma} = \mathbf{M}_{\Gamma\Gamma}^{(1)} + \mathbf{M}_{\Gamma\Gamma}^{(2)}$ and $\mathbf{A}_{\Gamma\Gamma} = \mathbf{A}_{\Gamma\Gamma}^{(1)} + \mathbf{A}_{\Gamma\Gamma}^{(2)}$.

4.5 Dirichlet–Neumann iteration

We now employ a Dirichlet–Neumann iteration to solve the discrete system (26). This corresponds to alternately solving the discretized equations of the transmission problem (16) on Ω_1 with Dirichlet data on Γ and the discretization of (16) on Ω_2 with Neumann data on Γ .

Therefore, from (26) one obtains for the k th iteration the two equation systems

$$(\mathbf{M}_1 + \Delta t \mathbf{A}_1)\mathbf{u}_I^{(1),n+1,k+1} = -(\mathbf{M}_{I\Gamma}^{(1)} + \Delta t \mathbf{A}_{I\Gamma}^{(1)})\mathbf{u}_I^{n+1,k} + \mathbf{M}_1\mathbf{u}_I^{(1),n} + \mathbf{M}_{I\Gamma}^{(1)}\mathbf{u}_I^n, \tag{27}$$

$$\hat{\mathbf{A}}\hat{\mathbf{u}}^{k+1} = \hat{\mathbf{M}}\mathbf{u}^n - \mathbf{b}^k, \tag{28}$$

to be solved in succession. Here,

$$\hat{\mathbf{A}} = \begin{pmatrix} \mathbf{M}_2 + \Delta t \mathbf{A}_2 & \mathbf{M}_{I\Gamma}^{(2)} + \Delta t \mathbf{A}_{I\Gamma}^{(2)} \\ \mathbf{M}_{\Gamma I}^{(2)} + \Delta t \mathbf{A}_{\Gamma I}^{(2)} & \mathbf{M}_{\Gamma\Gamma} + \Delta t \mathbf{A}_{\Gamma\Gamma} \end{pmatrix},$$

$$\hat{\mathbf{M}} = \begin{pmatrix} \mathbf{0} & \mathbf{M}_2 & \mathbf{M}_{I\Gamma}^{(2)} \\ \mathbf{M}_{\Gamma I}^{(1)} & \mathbf{M}_{\Gamma I}^{(2)} & \mathbf{M}_{\Gamma\Gamma} \end{pmatrix},$$

and

$$\mathbf{b}^k = \begin{pmatrix} \mathbf{0} \\ (\mathbf{M}_{\Gamma I}^{(1)} + \Delta t \mathbf{A}_{\Gamma I}^{(1)})\mathbf{u}_I^{(1),n+1,k+1} + (\mathbf{M}_{\Gamma\Gamma}^{(1)} + \Delta t \mathbf{A}_{\Gamma\Gamma}^{(1)})\mathbf{u}_I^{n+1,k} \end{pmatrix},$$

$$\hat{\mathbf{u}}^{k+1} = \begin{pmatrix} \mathbf{u}_I^{(2),n+1,k+1} \\ \mathbf{u}_I^{n+1,k+1} \end{pmatrix},$$

with some initial condition, here $\mathbf{u}_I^{n+1,0} = \mathbf{u}_I^n$. The iteration is terminated according to the standard criterion $\|\mathbf{u}_I^{k+1} - \mathbf{u}_I^k\| \leq \tau$ where τ is a user defined tolerance [3].

One way to analyze this method is to write it as a splitting method for (26) and try to estimate the spectral radius of that iteration by a norm. However, the results obtained in this way are much too inaccurate. For that reason, we now rewrite (27)–(28) as an iteration for \mathbf{u}_I^{n+1} to restrict the size of the space to the dimension of \mathbf{u}_I which is much smaller. To this end, we isolate the term $\mathbf{u}_I^{(1),n+1,k+1}$ in (27)

and $\mathbf{u}_I^{(2),n+1,k+1}$ in the first equation in (28) and we insert the resulting expressions into the second equation in (28). Consequently, the iteration $\mathbf{u}_I^{n+1,k+1} = \Sigma \mathbf{u}_I^{n+1,k} + \psi^n$ is obtained with iteration matrix

$$\Sigma = -\mathbf{S}^{(2)-1}\mathbf{S}^{(1)}, \tag{29}$$

where

$$\mathbf{S}^{(m)} = (\mathbf{M}_{I\Gamma}^{(m)} + \Delta t \mathbf{A}_{I\Gamma}^{(m)}) - (\mathbf{M}_{\Gamma I}^{(m)} + \Delta t \mathbf{A}_{\Gamma I}^{(m)})(\mathbf{M}_m + \Delta t \mathbf{A}_m)^{-1}(\mathbf{M}_{I\Gamma}^{(m)} + \Delta t \mathbf{A}_{I\Gamma}^{(m)}), \tag{30}$$

for $m = 1, 2$ and ψ^n contains terms that depend only on the solutions at the previous time step. Notice that Σ is a discrete version of the Steklov–Poincaré operator.

Thus, the Dirichlet–Neumann iteration is a linear iteration and the rate of convergence is described by the spectral radius of the iteration matrix Σ .

5 One-dimensional convergence analysis

So far, the derivation was performed for a rather general discretization. In this section, we study the iteration matrix Σ for a specific FVM–FEM discretization in 1D. We will give an exact formula for the convergence rates. The behaviour of the rates when approaching both the continuous case in time and space is also given.

Specifically, we use $\Omega_1 = [-1, 0]$, $\Omega_2 = [0, 1]$. For the FVM discretization, we consider a primal grid, i.e. we discretize Ω_1 into $N_1 + 1$ equal sized grid cells of size $\Delta x_1 = 1/(N_1 + 1)$, and define $x_i = i\Delta x_1$, so that x_i is the center of the cell i , see Fig. 7. The edges of cell i are then $x_{i-1/2}$ and $x_{i+1/2}$ and they form the corresponding dual grid. Moreover, we use the flux function

$$F(u_L, u_R) = -\frac{\lambda_1}{\Delta x_1}(u_{1,i} - u_{1,i-1}), \tag{31}$$

to approximate the flux, which results in a second order scheme. For the FEM discretization, we use the standard piecewise-linear polynomials as test functions. Here we discretize Ω_2 into $N_2 + 1$ equal sized cells of size $\Delta x_2 = 1/(N_2 + 1)$.

For the coupling between a compressible fluid and a structure, there would typically be a boundary layer in the fluid, meaning that the mesh would be very fine in direction normal to the boundary, implying $\Delta x_1 \ll \Delta x_2$.

With $\mathbf{e}_{m,j} = (0 \dots 0 \ 1 \ 0 \dots 0)^T \in \mathbb{R}^{N_m}$ where the only nonzero entry is located at the j th position, the discretization matrices are given by

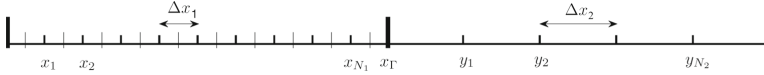


Fig. 7 Grid cells over Ω_1 and Ω_2 for the finite volume discretization and the finite element discretization respectively

$$\begin{aligned}
 \mathbf{A}_1 &= \frac{\lambda_1}{\Delta x_1^2} \begin{pmatrix} -2 & 1 & & 0 \\ 1 & -2 & \ddots & \\ & \ddots & \ddots & 1 \\ 0 & & 1 & -2 \end{pmatrix}, \\
 \mathbf{A}_2 &= \frac{\lambda_2}{\Delta x_2^2} \begin{pmatrix} 2 & -1 & & 0 \\ -1 & 2 & \ddots & \\ & \ddots & \ddots & -1 \\ 0 & & -1 & 2 \end{pmatrix}, \\
 \mathbf{M}_2 &= \frac{\alpha_2}{6} \begin{pmatrix} 4 & 1 & & 0 \\ 1 & 4 & \ddots & \\ & \ddots & \ddots & 1 \\ 0 & & 1 & 4 \end{pmatrix}, \\
 \mathbf{A}_{\Gamma\Gamma}^{(1)} &= \frac{3\lambda_1}{2\Delta x_1^2}, \quad \mathbf{A}_{\Gamma\Gamma}^{(2)} = \frac{\lambda_2}{\Delta x_2^2}, \quad \mathbf{M}_{\Gamma\Gamma}^{(2)} = \frac{2\alpha_2}{6}, \\
 \mathbf{A}_{\Gamma I}^{(1)} &= \frac{\lambda_1}{\Delta x_1^2} \mathbf{e}_{1,N_1}, \quad \mathbf{A}_{\Gamma I}^{(2)} = -\frac{\lambda_2}{\Delta x_2^2} \mathbf{e}_{2,1}, \quad \mathbf{M}_{\Gamma I}^{(2)} = \frac{\alpha_2}{6} \mathbf{e}_{2,1}, \\
 \mathbf{A}_{I\Gamma}^{(1)} &= \frac{\lambda_1}{2\Delta x_1^2} (4\mathbf{e}_{1,N_1}^T - \mathbf{e}_{1,N_1-1}^T), \\
 \mathbf{A}_{I\Gamma}^{(2)} &= -\frac{\lambda_2}{2\Delta x_2^2} \mathbf{e}_{2,1}^T, \quad \mathbf{M}_{I\Gamma}^{(2)} = \frac{\alpha_2}{6} \mathbf{e}_{2,1}^T.
 \end{aligned}$$

where $\mathbf{A}_m, \mathbf{M}_m \in \mathbb{R}^{N_m \times N_m}$, $\mathbf{A}_{\Gamma I}^{(m)}, \mathbf{M}_{\Gamma I}^{(m)} \in \mathbb{R}^{N_m \times 1}$ and $\mathbf{A}_{I\Gamma}^{(m)}, \mathbf{M}_{I\Gamma}^{(m)} \in \mathbb{R}^{1 \times N_m}$ for $m = 1, 2$.

In this case, $\mathbf{M}_1 = \alpha_1 \mathbf{I}$, $\mathbf{M}_{\Gamma\Gamma}^{(1)} = \mathbf{M}_{\Gamma I}^{(1)} = \mathbf{M}_{I\Gamma}^{(1)} = \mathbf{0}$. Thus,

$$\mathbf{S}^{(1)} = \Delta t \mathbf{A}_{\Gamma\Gamma}^{(1)} - \Delta t^2 \mathbf{A}_{\Gamma I}^{(1)} (\alpha_1 \mathbf{I} - \Delta t \mathbf{A}_1)^{-1} \mathbf{A}_{I\Gamma}^{(1)}, \quad (32)$$

$$\begin{aligned}
 \mathbf{S}^{(2)} &= (\mathbf{M}_{\Gamma\Gamma}^{(2)} + \Delta t \mathbf{A}_{\Gamma\Gamma}^{(2)}) \\
 &\quad - (\mathbf{M}_{\Gamma I}^{(2)} + \Delta t \mathbf{A}_{\Gamma I}^{(2)}) (\mathbf{M}_2 + \Delta t \mathbf{A}_2)^{-1} (\mathbf{M}_{I\Gamma}^{(2)} + \Delta t \mathbf{A}_{I\Gamma}^{(2)}). \quad (33)
 \end{aligned}$$

Note that the iteration matrix Σ is just a real number in this case and thus its spectral radius is its modulus. One computes $\mathbf{S}^{(1)}$ and $\mathbf{S}^{(2)}$ by inserting the corresponding matrices specified above in (32) and (33) obtaining

$$\begin{aligned}
 \mathbf{S}^{(1)} &= \Delta t \frac{3\lambda_1}{2\Delta x_1^2} - \Delta t^2 \frac{\lambda_1^2}{2\Delta x_1^4} (4\mathbf{e}_{1,N_1}^T - \mathbf{e}_{1,N_1-1}^T) (\alpha_1 \mathbf{I} - \Delta t \mathbf{A}_1)^{-1} \mathbf{e}_{1,N_1} \\
 &= \Delta t \frac{3\lambda_1}{2\Delta x_1^2} - \Delta t^2 \frac{\lambda_1^2}{2\Delta x_1^4} (4\alpha_{N_1 N_1}^1 - \alpha_{N_1-1 N_1}^1), \quad (34)
 \end{aligned}$$

$$\mathbf{S}^{(2)} = \left(\frac{\alpha_2}{3} + \Delta t \frac{\lambda_2}{\Delta x_2^2} \right) - \left(\frac{\alpha_2}{6} - \Delta t \frac{\lambda_2}{\Delta x_2^2} \right) \mathbf{e}_{2,1}^T (\mathbf{M}_2 + \Delta t \mathbf{A}_2)^{-1} \mathbf{e}_{2,1}$$

$$= \left(\frac{\alpha_2}{3} + \Delta t \frac{\lambda_2}{\Delta x_2^2} \right) - \left(\frac{\alpha_2}{6} - \Delta t \frac{\lambda_2}{\Delta x_2^2} \right) \alpha_{11}^2, \quad (35)$$

where α_{ij}^1 represents the entries of the matrix $(\alpha_1 \mathbf{I} - \Delta t \mathbf{A}_1)^{-1}$ and α_{ij}^2 the entries of $(\mathbf{M}_2 + \Delta t \mathbf{A}_2)^{-1}$ for $i, j = 1, \dots, N_1$ and $i, j = 1, \dots, N_2$ respectively. Observe that the matrices $(\alpha_1 \mathbf{I} - \Delta t \mathbf{A}_1)$ and $(\mathbf{M}_2 + \Delta t \mathbf{A}_2)$ are tridiagonal Toeplitz matrices but their inverses are full matrices. The computation of the exact inverses could be performed based on the recursive formula presented in [12] which runs over the entries of the matrix and consequently, it is non trivial to compute $\alpha_{N_1 N_1}^1$, $\alpha_{N_1-1 N_1}^1$ and α_{11}^2 this way.

Due to these difficulties, we propose to rewrite the matrices $(\alpha_1 \mathbf{I} - \Delta t \mathbf{A}_1)^{-1}$ and $(\mathbf{M}_2 + \Delta t \mathbf{A}_2)^{-1}$ in terms of their eigendecomposition:

$$\begin{aligned}
 (\alpha_1 \mathbf{I} - \Delta t \mathbf{A}_1)^{-1} &= \left[\text{tridiag} \left(-\frac{\lambda_1 \Delta t}{\Delta x_1^2}, \frac{\alpha_1 \Delta x_1^2 + 2\lambda_1 \Delta t}{\Delta x_1^2}, -\frac{\lambda_1 \Delta t}{\Delta x_1^2} \right) \right]^{-1} \\
 &= \mathbf{V}_{N_1} \Lambda_1^{-1} \mathbf{V}_{N_1}, \quad (36)
 \end{aligned}$$

$$\begin{aligned}
 (\mathbf{M}_2 + \Delta t \mathbf{A}_2)^{-1} &= \left[\text{tridiag} \left(\frac{\alpha_2 \Delta x_2^2 - 6\lambda_2 \Delta t}{6\Delta x_2^2}, \frac{2\alpha_2 \Delta x_2^2 + 6\lambda_2 \Delta t}{3\Delta x_2^2}, \frac{\alpha_2 \Delta x_2^2 - 6\lambda_2 \Delta t}{6\Delta x_2^2} \right) \right]^{-1} \\
 &= \mathbf{V}_{N_2} \Lambda_2^{-1} \mathbf{V}_{N_2}, \quad (37)
 \end{aligned}$$

where the matrix \mathbf{V}_N has the eigenvectors of any symmetric tridiagonal Toeplitz matrix of dimension N as columns. The entries of \mathbf{V}_{N_1} and \mathbf{V}_{N_2} are not dependent on the entries of $\alpha_1 \mathbf{I} - \Delta t \mathbf{A}_1$ or $\mathbf{M}_2 + \Delta t \mathbf{A}_2$ due to their symmetry. Moreover, the matrices Λ_1 and Λ_2 are diagonal matrices having the eigenvalues of $\alpha_1 \mathbf{I} - \Delta t \mathbf{A}_1$ or $\mathbf{M}_2 + \Delta t \mathbf{A}_2$ as entries respectively. These are known and given e.g. in [25, pp. 514–516]:

$$\begin{aligned}
 v_{ij}^m &= \frac{1}{\sqrt{\sum_{k=1}^{N_m} \sin^2 \left(\frac{k\pi}{N_m+1} \right)}} \sin \left(\frac{ij\pi}{N_m+1} \right) \\
 &\quad \text{for } i, j = 1, \dots, N_m, \quad m = 1, 2, \\
 \mu_{1,j} &= \frac{1}{\Delta x_1^2} \left(\alpha_1 \Delta x_1^2 + 2\lambda_1 \Delta t - 2\lambda_1 \Delta t \cos \left(\frac{j\pi}{N_1+1} \right) \right) \\
 &\quad \text{for } j = 1, \dots, N_1, \\
 \mu_{2,j} &= \frac{1}{3\Delta x_2^2} \left(2\alpha_2 \Delta x_2^2 + 6\lambda_2 \Delta t + (\alpha_2 \Delta x_2^2 - 6\lambda_2 \Delta t) \cos \left(\frac{j\pi}{N_2+1} \right) \right) \quad \text{for } j = 1, \dots, N_2. \quad (38)
 \end{aligned}$$

The entries $\alpha_{N_1 N_1}^1$, $\alpha_{N_1-1 N_1}^1$ and α_{11}^2 of the matrices $(\alpha_1 \mathbf{I} - \Delta t \mathbf{A}_1)^{-1}$ and $(\mathbf{M}_2 + \Delta t \mathbf{A}_2)^{-1}$, respectively, are now computed through their eigendecomposition resulting in

$$\alpha_{N_1-1 N_1}^1 = \frac{\Delta x_1^2 s_0}{\sum_{i=1}^{N_1} \sin^2(i\pi \Delta x_1)}, \tag{39}$$

$$\alpha_{N_1 N_1}^1 = \frac{\Delta x_1^2 s_1}{\sum_{i=1}^{N_1} \sin^2(i\pi \Delta x_1)}, \tag{40}$$

$$\alpha_{11}^2 = \frac{3\Delta x_2^2 s_2}{\sum_{i=1}^{N_2} \sin^2(i\pi \Delta x_2)}, \tag{41}$$

with

$$s_0 = \sum_{i=1}^{N_1} \frac{\sin(i\pi \Delta x_1) \sin(2i\pi \Delta x_1)}{\alpha_1 \Delta x_1^2 + 2\lambda_1 \Delta t (1 - \cos(i\pi \Delta x_1))}, \tag{42}$$

$$s_1 = \sum_{i=1}^{N_1} \frac{\sin^2(i\pi \Delta x_1)}{\alpha_1 \Delta x_1^2 + 2\lambda_1 \Delta t (1 - \cos(i\pi \Delta x_1))}, \tag{43}$$

$$s_2 = \sum_{i=1}^{N_2} \frac{\sin^2(i\pi \Delta x_2)}{2\alpha_2 \Delta x_2^2 + 6\lambda_2 \Delta t + (\alpha_2 \Delta x_2^2 - 6\lambda_2 \Delta t) \cos(i\pi \Delta x_2)}. \tag{44}$$

Now, inserting (39), (40) and (41) into (34) and (35) we get for $\mathbf{S}^{(1)}$ and $\mathbf{S}^{(2)}$:

$$\mathbf{S}^{(1)} = \frac{3\lambda_1 \Delta t}{2\Delta x_1^2} - \frac{\lambda_1^2 \Delta t^2}{2\Delta x_1^2} \frac{4s_1 - s_0}{\sum_{i=1}^{N_1} \sin^2(i\pi \Delta x_1)}, \tag{45}$$

$$\mathbf{S}^{(2)} = \left(\frac{\alpha_2 \Delta x_2^2 + 3\lambda_2 \Delta t}{3\Delta x_2^2} \right) - \frac{(\alpha_2 \Delta x_2^2 - 6\lambda_2 \Delta t)^2}{12\Delta x_2^2} \frac{s_2}{\sum_{i=1}^{N_2} \sin^2(i\pi \Delta x_2)}. \tag{46}$$

With this we obtain an explicit formula for the spectral radius of the iteration matrix Σ as a function of Δx_1 , Δx_2 and Δt :

$$\begin{aligned} \rho(\Sigma) &= |\Sigma| = |\mathbf{S}^{(2)-1} \mathbf{S}^{(1)}| \\ &= \left(\frac{\alpha_2 \Delta x_2^2 + 3\lambda_2 \Delta t}{3\Delta x_2^2} - \frac{(\alpha_2 \Delta x_2^2 - 6\lambda_2 \Delta t)^2}{12\Delta x_2^2} \frac{s_2}{\sum_{i=1}^{N_2} \sin^2(i\pi \Delta x_2)} \right)^{-1} \\ &\quad \cdot \left(\frac{3\lambda_1 \Delta t}{2\Delta x_1^2} - \frac{\lambda_1^2 \Delta t^2}{2\Delta x_1^2} \frac{4s_1 - s_0}{\sum_{i=1}^{N_1} \sin^2(i\pi \Delta x_1)} \right). \end{aligned} \tag{47}$$

To simplify this, the finite sums $\sum_{i=1}^{N_1} \sin^2(i\pi \Delta x_1)$ and $\sum_{i=1}^{N_2} \sin^2(i\pi \Delta x_2)$ can be computed. We first rewrite the sum of squared sine terms into a sum of cosine terms using the identity $\sin^2(x/2) = (1 - \cos(x))/2$. Then, the resulting

sum can be converted into a geometric sum using Euler’s formula. We thus obtain after some calculations:

$$\begin{aligned} \sum_{j=1}^{N_1} \sin^2(j\pi \Delta x_1) &= \frac{1 - \Delta x_1}{2\Delta x_1} \\ &\quad - \frac{1}{2} \sum_{j=1}^{N_1} \cos(2j\pi \Delta x_1) = \frac{1}{2\Delta x_1}, \end{aligned} \tag{48}$$

$$\sum_{j=1}^{N_2} \sin^2(j\pi \Delta x_2) = \frac{1}{2\Delta x_2}. \tag{49}$$

Inserting (48) and (49) into (47) we get after some manipulations

$$|\Sigma| = \frac{3\Delta x_2^2 (3\lambda_1 \Delta t - 2\lambda_1^2 \Delta x_1 \Delta t^2 (4s_1 - s_0))}{\Delta x_1^2 (2(\alpha_2 \Delta x_2^2 + 3\lambda_2 \Delta t) - \Delta x_2 (\alpha_2 \Delta x_2^2 - 6\lambda_2 \Delta t)^2 s_2)}. \tag{50}$$

We could not find a way of simplifying the finite sum (44) because Δx_2 depends on N_2 (i.e., $\Delta x_2 = 1/(N_2 + 1)$). However, (50) is a computable expression that gives the exact convergence rates of the Dirichlet–Neumann iteration for given Δt , Δx_m , α_m and λ_m , $m = 1, 2$.

We are now interested in the asymptotics of (50) with respect to both spatial and temporal resolutions. This corresponds to the computation of two different limits: $\Delta t \rightarrow 0$ for a fixed Δx_1 and $\Delta x_1 \rightarrow 0$ for a fixed Δt . As an alternative, one could reformulate (50) in terms of $c := \Delta t / \Delta x_1^2$ and compute the limits $c \rightarrow 0$ and $c \rightarrow \infty$. Both choices give the same results because for a fixed Δx_1 , if $\Delta t \rightarrow 0$, then $c \rightarrow 0$ and for a fixed Δt , if $\Delta x_1 \rightarrow 0$, then $c \rightarrow \infty$.

For simplicity, we compute the asymptotics of (50) for $\Delta t \rightarrow 0$ and $\Delta x_1 \rightarrow 0$ with $\Delta x_2 = r \cdot \Delta x_1$ where $r := \Delta x_2 / \Delta x_1$ is a fixed aspect ratio. This is motivated by the assumption that we have matching nodes at the interface. Thus, the resolution in the fluid in direction tangential to the wall is the same as the resolution in the structure. This means that the aspect ratio of the left subdomain cells in 2D corresponds to the ratio of grid spacings between the two subdomains in 1D. This is illustrated in Fig. 8. We obtain:

$$\lim_{\Delta t \rightarrow 0} |\Sigma| = \frac{3\Delta x_2^2 \cdot 0}{\Delta x_1^2 (2\alpha_2 \Delta x_2^2 - \alpha_2 \Delta x_2^3 \sum_{i=1}^{N_2} \frac{3 \sin^2(i\pi \Delta x_2)}{2 + \cos(i\pi \Delta x_2)})} = 0. \tag{51}$$

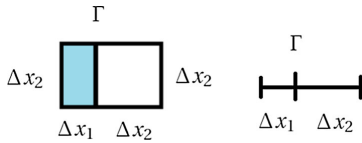


Fig. 8 Relation between the aspect ratio of the left subdomain cells in 2D and the ratio of grid spacings between both subdomains in 1D

$$\begin{aligned}
 \lim_{\Delta x_1 \rightarrow 0} |\Sigma| &= \lim_{\Delta x_1 \rightarrow 0} \frac{9\lambda_1 r^2 \Delta t - 6\lambda_1 r^2 \Delta x_1 \Delta t \left(\sum_{i=1}^{N_1} \frac{\sin^2(i\pi \Delta x_1)(2 - \cos(i\pi \Delta x_1))}{1 - \cos(i\pi \Delta x_1)} \right)}{6\lambda_2 \Delta t - 6\lambda_2 r \Delta t \Delta x_1 \left(\sum_{i=1}^{N_2} \frac{\sin^2(i\pi r \Delta x_1)}{1 - \cos(i\pi r \Delta x_1)} \right)} \\
 &= \frac{\lambda_1}{\lambda_2} \lim_{\Delta x_1 \rightarrow 0} \frac{3r^2 - 2r^2 \Delta x_1 \left(\sum_{i=1}^{N_1} 2 + \sum_{i=1}^{N_1} \cos(i\pi \Delta x_1) - \sum_{i=1}^{N_1} \cos^2(i\pi \Delta x_1) \right)}{2 - 2r \Delta x_1 \left(\sum_{i=1}^{N_2} 1 + \sum_{i=1}^{N_2} \cos(i\pi r \Delta x_1) \right)}. \tag{52}
 \end{aligned}$$

To simplify (52), it is well known that the finite sums $\sum_{i=1}^{N_1} \cos(i\pi \Delta x_1)$, $\sum_{i=1}^{N_2} \cos(i\pi r \Delta x_1)$ and $\sum_{i=1}^{N_1} \cos^2(i\pi \Delta x_1)$ can be computed by using Euler’s formula to convert them into geometric sums. We thus obtain after some calculations:

$$\begin{aligned}
 \sum_{j=1}^{N_2} \cos(j\pi r \Delta x_1) &= \operatorname{Re} \left(\sum_{j=1}^{N_2} e^{ij\pi r \Delta x_1} \right) \\
 &= \operatorname{Re} \left(\frac{e^{i\pi r \Delta x_1} (1 - e^{iN_2 \pi r \Delta x_1})}{1 - e^{i\pi r \Delta x_1}} \right) = 0. \tag{53}
 \end{aligned}$$

In order to compute the third sum, we rewrite the sum of squared cosine terms into a sum of sine terms using the identity $\cos^2(x/2) = (1 + \cos(x))/2$ and then apply the same technique:

$$\begin{aligned}
 \sum_{j=1}^{N_1} \cos^2(j\pi \Delta x_1) &= \frac{1 - \Delta x_1}{2\Delta x_1} + \frac{1}{2} \sum_{j=1}^{N_1} \cos(2j\pi \Delta x_1) \\
 &= \frac{1 - 2\Delta x_1}{2\Delta x_1}. \tag{54}
 \end{aligned}$$

Inserting (53) and (54) into (52) we get

$$\begin{aligned}
 \lim_{\Delta x_1 \rightarrow 0} |\Sigma| &= \frac{\lambda_1}{\lambda_2} \lim_{\Delta x_1 \rightarrow 0} \frac{3r^2 - 2r^2 \Delta x_1 \left(\frac{2(1 - \Delta x_1)}{\Delta x_1} - \frac{1 - 2\Delta x_1}{2\Delta x_1} \right)}{2 - 2r \Delta x_1 \left(\frac{1 - r \Delta x_1}{r \Delta x_1} \right)} \\
 &= \frac{\lambda_1}{\lambda_2} \lim_{\Delta x_1 \rightarrow 0} \frac{2r^2 \Delta x_1}{2r \Delta x_1} = \frac{\lambda_1}{\lambda_2} r =: \delta_r. \tag{55}
 \end{aligned}$$

From the result obtained in (51) we can conclude that the convergence rate goes to zero when the time step decreases and therefore, the iteration will be fast for Δt small and can always be made to converge by decreasing Δt . This is consistent with the behavior of the cooling of a flat plate and the flanged shaft presented earlier in Fig. 4.

On the other hand, from the spatial asymptotics (55) we can observe that strong jumps in the thermal conductivities of the materials placed in Ω_1 and Ω_2 will imply fast convergence. This is often the case when modelling thermal fluid–structure interaction, since fluids typically have lower thermal conductivities than structures.

Finally, the aspect ratio r also influences the behavior of the fixed point iteration, i.e., the rates will become smaller the higher the aspect ratio, e.g. the higher the Reynolds number in the fluid. This phenomenon is not unknown for PDE discretizations and is referred to as geometric stiffness. As is the case here, refining the mesh to reduce the aspect ratio would lead to faster convergence of the iterative method.

Before presenting numerical results we want to show the results obtained for different space discretization combinations with the same constant mesh width on both subdomains.

6 Extension of the analysis

In this section we want to extend the results presented in the previous section by reviewing similar analysis for other choices of space discretizations. In particular, FEM–FEM coupling and 2D FVM–FEM with $r = 1$.

Firstly, when one uses a linear FEM discretization for the fluid in 1D and the same mesh width on both subdomains (i.e. $r = 1$) and applies the same analysis as in the previous section, the corresponding limits for the spectral radius of the iteration matrix Σ are given by [26,28]:

$$\lim_{\Delta t \rightarrow 0} \rho(\Sigma) = \frac{\alpha_1}{\alpha_2}, \tag{56}$$

$$\lim_{\Delta x \rightarrow 0} \rho(\Sigma) = \frac{\lambda_1}{\lambda_2}. \tag{57}$$

When we compare these with the asymptotics obtained with FVM–FEM discretizations (51)–(55), we observe that while the spatial limit is the same, the temporal limit does not match. This arises from differences in the matrix $\mathbf{S}^{(1)}$ in (30). In the FEM–FEM context, the matrices $\mathbf{S}^{(1)}$ and $\mathbf{S}^{(2)}$ lead to the same expression with only different material coefficients ($\alpha_1, \alpha_2, \lambda_1, \lambda_2$). Because of this, the limits of $\rho(\Sigma)$ are quotients of those coefficients. However, the situation is different in the FVM–FEM context. There, the matrix $\mathbf{S}^{(1)}$ in (32) is missing several mass matrices if we compare it with $\mathbf{S}^{(2)}$ in (33). This unsymmetry between $\mathbf{S}^{(1)}$ and $\mathbf{S}^{(2)}$ causes that the

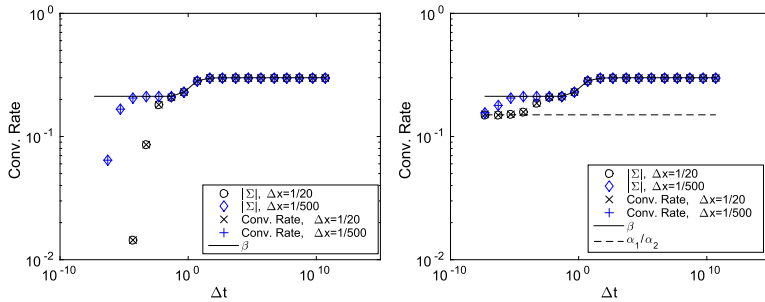


Fig. 9 Semidiscrete estimator β , exact rate Σ and numerical rates over Δt in 1D. $D_1 = 1$, $D_2 = 0.5$, $\lambda_1 = 0.3$, $\lambda_2 = 1$, $\Delta x = 1/20$ and $\Delta t = 5e-8, 5e-7, \dots, 5e10$. Left: FVM-FEM. Right: FEM-FEM

limit of $\rho(\Sigma)$ when $\Delta t \rightarrow 0$ is not balanced between the numerator and the denominator, resulting in 0.

This implies that, opposed to the FVM-FEM case, where convergence can always be achieved by decreasing the time step, for an FEM-FEM coupling, a situation can occur where $\alpha_1/\alpha_2 > \lambda_1/\lambda_2$ and therefore, a decrease in time step can even cause divergence. This is for example the case for an air-water coupling [26].

Secondly, for an aspect ratio of $r = 1$, we were able to extend the 1D results for both FVM-FEM and FEM-FEM to 2D in the following sense (see [5,26]). In 2D, the iteration matrix Σ is not easy to compute for several reasons. First of all, the matrices $\mathbf{M}_1 + \Delta t \mathbf{A}_1$ and $\mathbf{M}_2 + \Delta t \mathbf{A}_2$ are sparse block tridiagonal matrices, and consequently their inverses are not straight forward to compute. Moreover, the diagonal blocks of the same matrices are tridiagonal but their inverses are full matrices.

Due to these difficulties, we approximated the strictly diagonally dominant matrices $\mathbf{M}_1 + \Delta t \mathbf{A}_1$ and $\mathbf{M}_2 + \Delta t \mathbf{A}_2$ by their diagonal. Thus, we obtained an estimate of the spectral radius of the iteration matrix Σ . This estimator tends to the exact same limits as for the 1D case for both combination of discretizations.

We did not find a way to further extend these results to the high aspect ratio case. However, we will show now by numerical experiments that already the 1D formula (50) is a good estimator for convergence rates in 2D.

7 Numerical results

We now present numerical experiments designed to illustrate the validity of the theoretical results of the previous sections. Firstly, we will confirm that the theoretical formula for $|\Sigma|$ in (50) predicts the convergence rates in the 1D case. Secondly, we will show the validity of (50) as an estimator for the rates in the 2D case, we will also show that the theoretical asymp-

otics deduced in (51) and (55) match with the numerical experiments. Finally, we illustrate the validity of (50) as an estimator for the nonlinear thermal FSI test cases introduced in Sect. 3.

7.1 Results in 1D

We first compare the semidiscrete estimator β in (18) with the discrete formula $|\Sigma|$ in 1D in (50) and experimental convergence rates. The latter are obtained from implementing the Dirichlet-Neumann method (27)-(28). The results are then compared to a reference solution u_{ref} over the whole domain Ω , obtained by choosing a tolerance of $1e-10$ as a termination criterion.

Figure 9 shows a comparison between β and $|\Sigma|$ for $r = 1$, $\Delta x = 1/20$ and $\Delta x = 1/500$ and varying Δt . On the left we plot β , $|\Sigma|$ and the experimental convergence rates for the FVM-FEM approach described in Sect. 5 and on the right for the FEM-FEM approach mentioned in Sect. 6. As can be seen, the experimental convergence rate matches exactly with the exact formula (50). Observe that β is almost constant and presents the same behavior as in Fig. 5. We can conclude that the formulas for the convergence rates in 1D presented in Sect. 5 match the semidiscrete one proposed in [18] when $\Delta t/\Delta x^2 \gg 1$. In the, less relevant case, $\Delta t/\Delta x^2 \ll 1$ our formula also predicts the rates accurately, while the semidiscrete estimator deviates according to (19). Finally, Fig. 9 also illustrates the differences in the temporal limit when employing different combinations of spatial discretizations as explained in previous section. In the FVM-FEM case the limit is 0 [see (51)] and in the FEM-FEM case it is α_1/α_2 [see (56)].

The difference to the semidiscrete analysis in [18] stems from different limits taking place. The semidiscrete analysis implicitly assumes that first a limit Δx to zero has taken place for Δt fixed. Thus, a limit first Δt , then Δx to zero is not addressed by it. This can be seen in Fig. 9 in the follow-

Table 1 Physical properties of the materials. λ is the thermal conductivity, ρ the density, c_p the specific heat capacity and $\alpha = \rho c_p$

Material	λ (W/mK)	ρ (kg/m ³)	c_p (J/kg K)	α (J/K m ³)
Air	0.0243	1.293	1005	1299.5
Water	0.58	999.7	4192.1	4.1908e6
Steel	48.9	7836	443	3471348

Table 2 Temporal and spatial asymptotics of (50) for the thermal interaction of air at 273 K with steel at 900 K, water at 283 K with steel and air with water

Case	$\Delta t \rightarrow 0$	$\Delta x \rightarrow 0$
Air–steel	0	$4.9693e-4 \cdot r$
Water–steel	0	$0.0119 \cdot r$
Air–water	0	$0.0419 \cdot r$

ing way: for fixed Δx , letting Δt become very small causes the convergence rate to move into the direction predicted by

the fully discrete analysis. However, then keeping this very small Δt fixed and decreasing Δx moves that rate back in the vicinity of β .

We now want to illustrate how $|\Sigma|$ in (50) gives the convergence rates and tends to the limits computed previously in (51) and (55). To this end, we present two real data examples. We consider here the thermal interaction between air at 273 K with steel at 900 K and water at 283 K with steel at 900 K. Physical properties of the materials and resulting asymptotics for these two cases are shown in Tables 1 and 2 respectively.

Figures 10 and 11 show the convergence rates for the interactions between air and steel and between water and steel, respectively. On the left we have always fixed Δx_1 and r and vary Δt , whereas on the right we have fixed Δt and r , and vary Δx_1 . Each plot includes graphs for two different values of r . In Fig. 10 we choose $r = 1$ and $r = 100$ to illustrate the effect of a neutral or a high aspect ratio. In Fig. 11 we use $r = 0.01$ and $r = 1$ to illustrate how the rates are affected by a small or a neutral aspect ratio.

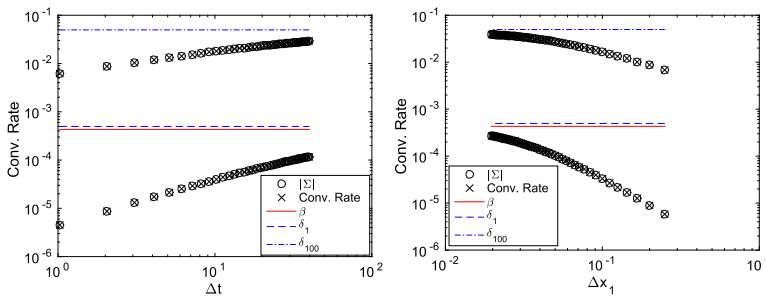


Fig. 10 Air–steel thermal interaction with respect Δt on the left and Δx_1 on the right in 1D. Left: $\Delta t = 40/39, 2 \cdot 40/39, \dots, 39 \cdot 40/39$, $\Delta x_1 = 1/1100$ and $r = 100$ (top curves) or $r = 1$ (bottom

curves). Right: $\Delta x_1 = 1/3, 1/4, \dots, 1/50$, $\Delta t = 10$ and $r = 100$ (top curves) or $r = 1$ (bottom curves)

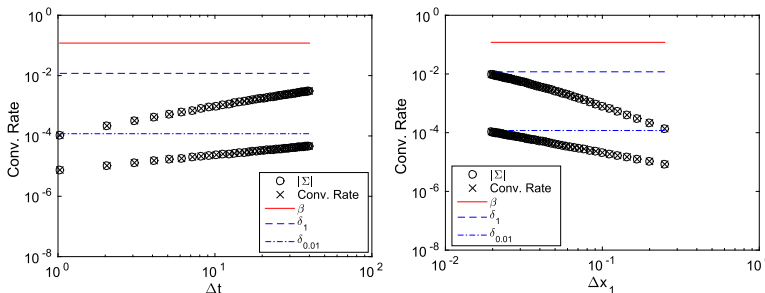


Fig. 11 Water–steel thermal interaction with respect Δt on the left and Δx_1 on the right in 1D. Left: $\Delta t = 1/39, 2 \cdot 1/39, \dots, 39 \cdot 1/39$, $\Delta x_1 = 1/1100$ and $r = 1$ (top curves) or $r = 0.01$ (bottom curves). Right:

$\Delta x_1 = 1/3, 1/4, \dots, 1/50$, $\Delta t = 10$ and $r = 1$ (top curves) or $r = 0.01$ (bottom curves)

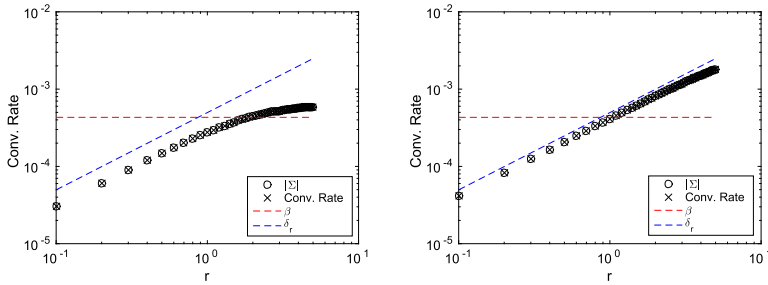


Fig. 12 Air–steel thermal interaction with respect to the aspect ratio r in 1D. $r = 5/50, 2 \cdot 5/50, \dots, 50 \cdot 5/50$ and $\Delta t = 10$. Left: $\Delta x_1 = 1/50$. Right: $\Delta x_1 = 1/200$

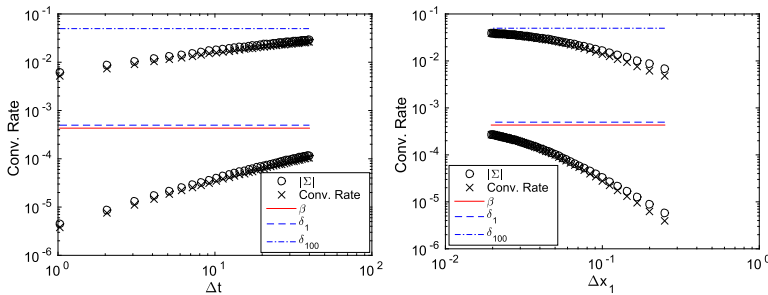


Fig. 13 2D Air–steel thermal interaction. Observed and estimated convergence rates over Δt (left) and Δx_1 (right). Left: $\Delta t = 40/39, 2 \cdot 40/39, \dots, 39 \cdot 40/39$, $\Delta x_1 = 1/1100$ and $r = 100$ (top curves) or

$r = 1$ (bottom curves). Right: $\Delta x_1 = 1/3, 1/4, \dots, 1/50$, $\Delta t = 10$ and $r = 100$ (top curves) or $r = 1$ (bottom curves)

Again, $|\Sigma|$ gives the exact convergence rates. Moreover, one observes that the rates on the left plots in Figs. 10 and 11 tend to 0 as predicted in (51) and on the right plots in Figs. 10 and 11 to δ_r , as predicted in (55).

Before ending this subsection, we want to illustrate the relation between the convergence rates and the aspect ratio r . To this end, Fig. 12 shows the convergence rates for the interaction between air and steel. In the left plot we have chosen Δx_1 to be coarse and on the right one to be fine. This explains why the convergence rates on the right plot are closer to the spatial limit δ_r . Furthermore, there is a roughly proportional relation between the convergence rate and the aspect ratio. For coupling with compressible flows, we typically have a high aspect ratio and therefore, the Dirichlet–Neumann iteration will be slowed down. Furthermore, this shows that it is very important to take the aspect ratio into account to make a reasonable prediction of the convergence rate at all.

7.2 2D FVM–FEM results

We now want to demonstrate that the 1D formula (50) is a good estimator for the convergence rates in 2D. Thus, we

now consider a 2D version of (16) consisting of two coupled linear heat equations on two identical unit squares, e.g. $\Omega_1 = [-1, 0] \times [0, 1]$ and $\Omega_2 = [0, 1] \times [0, 1]$. We use a non equidistant cartesian grid with aspect ratio r on Ω_1 and an equidistant grid on Ω_2 . In order to use (50) as an estimator we decided to take the equidistant mesh width on Ω_2 as Δx_2 and the mesh width in x -direction on Ω_1 as Δx_1 .

As before, we present two real data examples described in Tables 1 and 2, namely the thermal interaction between air at 273 K with steel at 900 K and air at 273 K with water at 283 K.

Figures 13 and 14 show the convergence rates for the interactions between air and steel and between air and water in 2D respectively. On the left we always plot the rates for fixed Δx_1 and r with variable Δt , whereas on the right we have fixed Δt and r and varying Δx_1 . As before, each plot includes two different values of r . In Fig. 13 we choose $r = 1$ and $r = 100$ as in the 1D case (see Fig. 10) and in Fig. 14 we use $r = 1$ and $r = 1000$ to illustrate the effect of a neutral or a high aspect ratio. To compute β we use the transverse Fourier mode $k = i \Delta y$, $i = 0, 1, 2, \dots, N_y$ that maximizes it [see (18)]. One observes that the convergence rates predicted

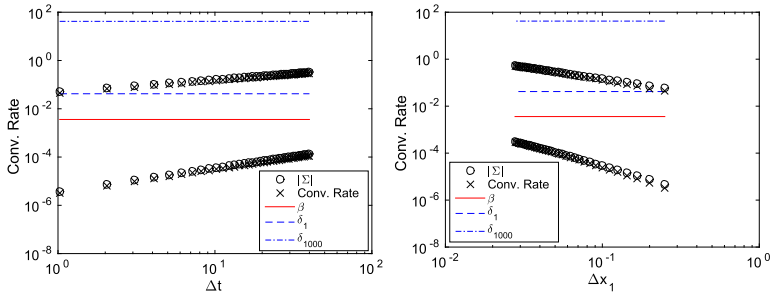


Fig. 14 2D Air–water thermal interaction. Observed and estimated convergence rates over Δt (left) and Δx_1 (right). Left: $\Delta t = 40/39, 2 \cdot 40/39, \dots, 39 \cdot 40/39$, $\Delta x_1 = 1/1100$ and $r = 1000$ (top curves) or

$r = 1$ (bottom curves). Right: $\Delta x_1 = 1/3, 1/4, \dots, 1/35$, $\Delta t = 10$ and $r = 1000$ (top curves) or $r = 1$ (bottom curves)

by the one-dimensional formula (50) are almost exactly the ones observed in 2D. Thus, the 1D case gives a very good estimator for the 2D model problem.

7.3 Thermal FSI test cases

Finally, we want to relate the results for the two nonlinear applications (the two cooling systems introduced in Sects. 3.1 and 3.2: the cooling of a flat plate and of a flanged shaft) to our analysis. The left plot in Fig. 15 shows the convergence behaviour for the flat plate and the right one for the flanged shaft. We plot the experimental convergence rates, the one-dimensional formula (50), the semidiscrete estimator (18) for the maximizing Fourier mode and the spatial limit δ_r specified in (55).

In order to apply the 1D formula (50) here, some assumptions need to be made, since we partly have unstructured meshes and nonuniform temperatures. Thus, we assume air at 273 K on the first subdomain with steel at 900 K on the second subdomain for the cooling of a flat plate and air at 273 K with steel at 1145 K for the cooling of a flanged shaft. The density, heat capacity and heat conductivity of air and the density of steel are given in Table 1. In addition, the heat conductivities and heat capacities of steel at 900 and 1145 K are obtained from the nonlinear coefficient functions (3) and (4) by inserting $\Theta = 900$ K or $\Theta = 1145$ K respectively. This gives $\lambda = 39.82$ and $c_p = 1.3684e3$ for steel at 900 K and $\lambda = 39.8$ and $c_p = 572.75$ for steel at 1145 K.

Furthermore, for the cooling of a flat plate, we take $\Delta x_1 = 9.3736e-5$ which is the width of the fluid cells touching the interface in the y -direction and $\Delta x_2 = 1.6667$ which is the width of the structure cells in both directions. Thus, we have an aspect ratio of $r = 1.7780e4$. On the other hand, choosing Δx_1 and Δx_2 for the cooling of a flanged shaft is more difficult due to the unstructured grids. In order to get an upper bound for the aspect ratio r , we choose

$\Delta x_1 = 1.6538e-4$ which is the minimum width of all the fluid cells touching the interface in direction normal to the wall and $\Delta x_2 = 1.1364$ which is the maximum width of all the structure cells touching the interface tangential to the wall. This gives $r = 6.8713e3$.

From the left plot in Fig. 15 one observes with these choices that (50) predicts the rates accurately for the cooling of a flat plate. Note that the semidiscrete estimator β does not show any change with Δt . Remember that β is almost always constant, except for a short dynamic transition between $(\lambda_1/\lambda_2)\sqrt{D_2/D_1}$ and λ_1/λ_2 as shown in Fig. 5. Here, we would have to choose a Δt larger than $1e6$ to see the transition.

Finally, on the right plot in Fig. 15 one can see that (50) predicts the convergence rates for the cooling of a flanged shaft to be only slightly smaller compared to the actual performance. This could be due to either the unstructured grids used or to the nonconstant temperature in the structure, which varies from room temperature to 1145 K. Again, β is almost constant.

8 Summary and conclusions

We considered the Dirichlet–Neumann iteration for thermal FSI and studied the convergence rates. To this end, we considered the coupling of two heat equations on two identical domains. We assumed structured grids on both subdomains, but allowed for high aspect ratio grids in one domain. An exact formula for the convergence rates was derived for the 1D case. Furthermore, we determined the limits of the convergence rates when approaching the continuous case either in space ($r\lambda_1/\lambda_2$) or time (0). This was confirmed by numerical results, where we also demonstrated that the 1D case gives excellent estimates for the 2D case. In addition, numer-

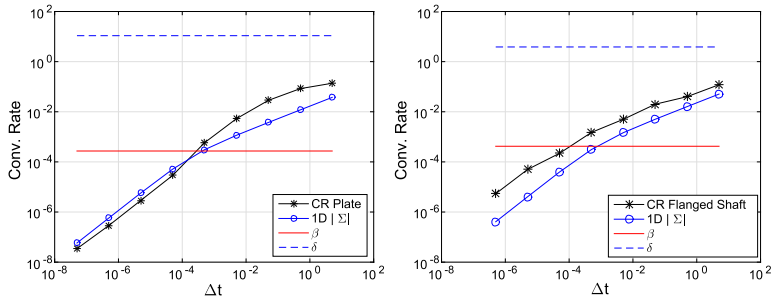


Fig. 15 Convergence behavior of the cooling systems with respect to Δt . Left: Test case 1: flow over a plate. Right: Test case 2: Cooling of a flanged shaft

ical experiments show that the linear analysis is relevant for nonlinear thermal FSI problems.

All in all, strong jumps in the coefficients of the coupled PDEs imply fast convergence. On the other hand, the coupling iteration will be slow when the material coefficients are continuous over all the subdomains, i.e., $\lambda_1 = \lambda_2$, and therefore $\delta_1 \sim 1$. For coupling of structures and compressible flows, the aspect ratio in the fluid has to be taken into account, since the convergence rate is proportional to it. For the nonlinear cooling problems considered here, the convergence rate was still around 0.1 for large Δt . When encountering divergence anyhow, this can be solved by reducing the time step. Note that in a time adaptive setting, it is standard to allow for a feedback loop between the nonlinear solver and the time stepper.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

- Badia S, Nobile F, Vergara C (2008) Fluid–structure partitioned procedures based on Robin transmission conditions. *J Comput Phys* 227:7027–7051
- Banka A (2005) Practical Applications of CFD in heat processing. Heat Treating Progress
- Birken P (2015) Termination criteria for inexact fixed point schemes. *Numer Linear Algebra Appl* 22(4):702–716
- Birken P, Gleim T, Kuhl D, Meister A (2015) Fast solvers for unsteady thermal fluid–structure interaction. *Int J Numer Methods Fluids* 79(1):16–29
- Birken P, Monge A (2017) A numerical methods for unsteady thermal fluid–structure interaction. In: Frei S, Holm B, Richter T, Wick T, Yang H (eds) Fluid–structure interaction. Modeling, adaptive discretisations and solvers, contributions in mathematical and computational sciences. Springer, Berlin
- Birken P, Quint K, Hartmann S, Meister A (2010) Choosing norms in adaptive FSI calculations. *PAMM* 10:555–556
- Birken P, Quint K, Hartmann S, Meister A (2011) A time-adaptive fluid–structure interaction method for thermal coupling. *Comput Vis Sci* 13(7):331–340
- Buchlin J (2010) Convective heat transfer and infrared thermography. *J Appl Fluid Mech* 3:55–62
- Causin P, Gerbeau J, Nobile F (2005) Added-mass effect in the design of partitioned algorithms for fluid–structure problems. *Comput Methods Appl Mech Eng* 194:4506–4527
- Deparis S, Fernández M, Formaggia L (2003) Acceleration of a fixed point algorithm for fluid–structure interaction using transpiration conditions. *M2AN* 37(4):601–616
- Farhat C (2004) CFD-based nonlinear computational aeroelasticity, ch. 13. In: Encyclopedia of computational mechanics, pp 459–480
- Fonseca C, Petronilho J (2001) Explicit inverses of some tridiagonal matrices. *Linear Algebra Appl* 325(1–3):7–21
- Gander M (2006) Optimized Schwarz methods. *SIAM J Numer Anal* 44(2):699–731
- Gerhold T, Friedrich O, Evans J, Galle M (1997) Calculation of complex three-dimensional configurations employing the DLR-TAU-Code. In: AIAA Paper 97-0167
- Gigante G, Vergara C (2016) Optimized Schwarz method for the fluid–structure interaction with cylindrical interfaces. In: Domain decomposition methods in science and engineering XXII. Springer, Berlin, pp 521–529
- Giles M (1997) Stability analysis of numerical interface conditions in fluid–structure thermal analysis. *Int J Numer Methods Fluids* 25:421–436
- Heck U, Fritsching U, Bauchhage K (2001) Fluid flow and heat transfer in gas jet quenching of a cylinder. *Int J Numer Methods Heat Fluid Flow* 11:36–49
- Henshaw W, Chand K (2009) A composite grid solver for conjugate heat transfer in fluid–structure systems. *J Comput Phys* 228:2708–3741
- Hinderks M, Radespiel R (2006) Investigation of hypersonic gap flow of a reentry nosecone with consideration of fluid–structure interaction. In AIAA Paper 6, pp 2708–3741
- Kowolik D, Horst P, Haupt M (2013) Fluid–structure interaction analysis applied to thermal barrier coated cooled rocket thrust chambers with subsequent local investigation of delamination phenomena. *Prog Propuls Phys* 4:617–636
- Kowolik D, Tini V, Reese S, Haupt M (2013) 3D fluid–structure interaction analysis of a typical liquid rocket engine cycle based on a novel viscoplastic damage model. *Int J Numer Methods Eng* 94:1165–1190

22. Matthies HG, Niekamp R, Steindorf J (2006) Algorithms for strong coupling procedures. *Comput Methods Appl Mech Eng* 195:2028–2049
23. Mehta R (2005) Numerical computation of heat transfer on reentry capsules at mach 5. In: *AIAA-Paper 178*
24. Meng F, Banks J, Henshaw W, Schwendeman D (2017) A stable and accurate partitioned algorithm for conjugate heat transfer. *J Comput Phys* 344:51–85
25. Meyer C (2000) *Matrix analysis and applied linear algebra*. SIAM, Philadelphia
26. Monge A (2016) *The Dirichlet–Neumann iteration for unsteady thermal fluid–structure interaction*, Licentiate Thesis. Lund University
27. Monge A, Birken P (2016) Convergence analysis of coupling iterations for the unsteady transmission problem with mixed discretizations. In: *VII European Congress on computational methods in applied sciences and engineering*, vol 1. ECCOMAS Congress
28. Monge A, Birken P (2016) Convergence analysis of the Dirichlet–Neumann iteration for finite element discretizations. *PAMM* 16:733–734
29. Peet Y, Fischer P (2012) Stability analysis of interface temporal discretization in grid overlapping methods. *SIAM J Numer Anal* 50(6):3375–3401
30. Quarteroni A, Valli A (1999) *Domain decomposition methods for partial differential equations*. Oxford Science Publications, Oxford
31. Quint KJ, Hartmann S, Rothe S, Saba N, Steinhoff K (2011) Experimental validation of high-order time integration for non-linear heat transfer problems. *Comput Mech* 48:81–96
32. Roux F (2008) Domain decomposition methodology with Robin interface matching conditions for solving strongly coupled. In: *Computational science, ICCS*
33. Stratton P, Shedletsky I, Lee M (2006) Gas quenching with helium. *Solid State Phenom* 118:221–226
34. Toselli A, Widlund O (2004) *Domain decomposition methods. Algorithms and theory*. Springer, Berlin
35. van Brummelen E (2009) Added mass effects of compressible and incompressible flows in fluid–structure interaction. *J Appl Mech* 76(2):021206
36. Weidig U, Saba N, Steinhoff K (2007) Massivumformprodukte mit funktional gradierten Eigenschaften durch eine differenzielle thermo-mechanische Prozessführung, *WT-Online*, pp 745–752

Paper III

A multirate Neumann-Neumann waveform relaxation method for heterogeneous coupled heat equations

Azahar Monge* and Philipp Birken*

**Centre for Mathematical Sciences,
Lund University,
Box 118, 22100, Lund, Sweden*

August 5, 2018

Abstract

An important challenge when coupling two different time dependent problems is to increase parallelization in time. We suggest a multirate Neumann-Neumann waveform relaxation algorithm to solve two heterogeneous coupled heat equations. In order to fix the mismatch produced by the multirate feature at the space-time interface a linear interpolation is constructed. The heat equations are discretized using a finite element method in space, whereas two alternative time integration methods are used: implicit Euler and SDIRK2. We perform a one-dimensional convergence analysis for the nonmultirate fully discretized heat equations using implicit Euler to find the optimal relaxation parameter in terms of the material coefficients, the stepsize and the mesh resolution. This gives a very efficient method which needs only two iterations. Numerical results confirm the analysis and show that the 1D nonmultirate optimal relaxation parameter is a very good estimator for the multirate 1D case and even for multirate and nonmultirate 2D examples using both implicit Euler and SDIRK2.

Keywords: Fluid-Structure Interaction, Coupled Problems, Transmission Problem, Domain Decomposition, Neumann-Neumann Method, Multirate, Thermal

1 Introduction

The main goal of this work is to describe a partitioned algorithm to solve two heterogeneous coupled heat equations allowing parallelization in time. In a partitioned approach different codes for the sub-problems are reused and the coupling is done by a master program which calls interface functions of the segregated codes [6, 7]. These algorithms are currently an active research topic driven by certain multiphysics applications where multiple physical models or multiple simultaneous physical phenomena involve solving coupled systems of partial differential equations (PDEs). An example of this is fluid structure interaction (FSI) [30, 4]. Moreover, we want that the time parallelization performed at the subsolvers works for different time grids. This is handled through multirate methods which are a classical field of research, see [10].

Our prime motivation here is thermal interaction between fluids and structures, also called conjugate heat transfer. There are two domains with jumps in the material coefficients across the connecting interface. Conjugate heat transfer plays an important role in many applications and its simulation has proved essential [1]. Examples for thermal fluid structure interaction are cooling of gas-turbine blades, thermal anti-icing systems of airplanes [5], supersonic reentry of vehicles from space [24, 17], gas quenching, which is an industrial heat treatment of metal workpieces [16, 28] or the cooling of rocket nozzles [19, 20].

The classical way of parallelizing the numerical solution of PDEs is to use domain decomposition (DD) methods. These split the computational domain into subdomains and coordinate the coupling between the subdomains in an iterative manner. For an introduction to DD methods and their basic convergence results see [27, 29]. The Dirichlet-Neumann iteration is a standard DD method to find solutions of the coupled problem. The PDEs are solved sequentially using Dirichlet-, respectively Neumann boundary with data given from the solution of the other

* e-mail: azahar.monge@na.lu.se; web page: <http://www.maths.lu.se/staff/azahar-monge>

problem. Previous numerical experiments [2] showed that this iteration is fast for thermal FSI, and a convergence analysis of two heterogeneous linear heat equations showed that the fast behavior was a consequence of the strong jumps in the material coefficients [26].

In spite of the efficient behavior of the Dirichlet-Neumann iteration in the thermal FSI framework, it has two main disadvantages. Firstly, the subsolvers wait for each other, and therefore, they perform sequentially. Secondly, in the time dependent case the Dirichlet-Neumann iteration is used at each time step and consequently, both fields are solved with a common time resolution. Using instead a multirate scheme that allows for different time resolutions on each subdomain would be more efficient.

The aim of this work is to present a high order, parallel, multirate method for two heterogeneous coupled heat equations which could be applied to FSI problems. We use the Neumann-Neumann waveform relaxation (NNWR) method which is a waveform relaxation (WR) methods based on the classical Neumann-Neumann iteration [21, 13]. For time discretization we consider two alternatives, the implicit Euler method and a second order singly diagonally implicit Runge-Kutta method (SDIRK2). The WR methods were originally introduced by [22] for ordinary differential equation (ODE) systems, and they were used for the first time to solve time dependent PDEs in [14, 15]. They allow the use of different spatial and time discretizations for each subdomain which is specially useful in problems with strong jumps in the material coefficients [12] or the coupling of different models for the subdomains [11]. A time adaptive partitioned approach for thermal FSI was presented in [3]. In [23], two new iterative partitioned coupling methods that allow for the simultaneous execution of flow and structure solvers were introduced. However, parallelization in time for the coupling of heterogeneous materials was not considered.

Our algorithm has to take care of two aspects. On one hand, an interpolation procedure needs to be chosen to communicate data between the subdomains through the space-time interface in the multirate case. We want that the interpolation preserves a second order numerical solution of the coupled problem when using SDIRK2. On the other hand, the choice of the relaxation parameter for the NNWR method is crucial because when choosing the relaxation parameter right, two iterations are sufficient. In [21], a one-dimensional semidiscrete analysis shows that $\Theta = 1/4$ is the optimal relaxation parameter for two homogeneous coupled heat equations on two identical subdomains.

In this paper, we perform a fully discrete analysis of the NNWR algorithm for two heterogeneous coupled one-dimensional heat equations to find the optimal relaxation parameter in terms of the material coefficients. More specifically, we choose finite element methods (FEM) in space for both subdomains and implicit Euler method for the temporal discretization. Then, we derive the iteration matrix of the fully discrete NNWR algorithm with respect to the interface unknowns. In addition, we calculate the spectral radius of the iteration matrix through its eigendecomposition in order to estimate the optimal relaxation parameter Θ_{opt} which is dependent on the material coefficients, the time and space resolutions. In the case of homogeneous materials, $\Theta_{opt} = 1/4$ recovering the result in [21]. Furthermore, the asymptotic optimal relaxation parameters when approaching the continuous case in either time or space are also determined. In the spatial limit, the relaxation parameter turns out to be dependent on the heat conductivities, whereas in the temporal limit, we obtain dependency of the densities and the heat capacities.

In addition, we include numerical results where it is shown that the parallel, multirate method for two heterogeneous coupled heat equations introduced in this paper is extremely fast when choosing the right relaxation parameter. Moreover, we also show that the one-dimensional formula is a very good estimate for the multirate 1D case and even for multirate and nonmultirate 2D examples using both implicit Euler and SDIRK2. Finally, we also include a numerical comparison that shows that the NNWR method is a more efficient choice than the Dirichlet-Neumann waveform relaxation (DNWR) in the multirate case.

An outline of the paper now follows. In section 2, 3 and 4, we describe the model problem, the DNWR and the NNWR methods respectively. The FE space discretization is specified in section 5. In section 6, we describe the linear interpolation that needs to be performed at the space-time interface to get a multirate algorithm. Both time integration methods used in this paper are explained in section 7, these are implicit Euler and SDIRK2. In section 8, we present a derivation of the iteration matrix for a rather general discretization which is then applied to a specific 1D case in section 9. Numerical results are presented in section 10 and conclusions can be found in the last section.

2 Model problem

The unsteady transmission problem reads as follows, where we consider a domain $\Omega \subset \mathbb{R}^d$ which is cut into two subdomains $\Omega = \Omega_1 \cup \Omega_2$ with transmission conditions at the interface $\Gamma = \partial\Omega_1 \cap \partial\Omega_2$:

$$\begin{cases} \alpha_m \frac{\partial u_m(\mathbf{x}, t)}{\partial t} - \nabla \cdot (\lambda_m \nabla u_m(\mathbf{x}, t)) = 0, & \mathbf{x} \in \Omega_m \subset \mathbb{R}^d, \quad m = 1, 2, \\ u_m(\mathbf{x}, t) = 0, & \mathbf{x} \in \partial\Omega_m \setminus \Gamma, \\ u_1(\mathbf{x}, t) = u_2(\mathbf{x}, t), & \mathbf{x} \in \Gamma, \\ \lambda_2 \frac{\partial u_2(\mathbf{x}, t)}{\partial \mathbf{n}_2} = -\lambda_1 \frac{\partial u_1(\mathbf{x}, t)}{\partial \mathbf{n}_1}, & \mathbf{x} \in \Gamma, \\ u_m(\mathbf{x}, 0) = u_m^0(\mathbf{x}), & \mathbf{x} \in \Omega_m, \end{cases} \quad (1)$$

where $t \in [T_0, T_f]$ and \mathbf{n}_m is the outward normal to Ω_m for $m = 1, 2$.

The constants λ_1 and λ_2 describe the thermal conductivities of the materials on Ω_1 and Ω_2 respectively. D_1 and D_2 represent the thermal diffusivities of the materials and they are defined by

$$D_m = \frac{\lambda_m}{\alpha_m}, \quad \text{with } \alpha_m = \rho_m c_{p_m} \quad (2)$$

where ρ_m represents the density and c_{p_m} the specific heat capacity of the material placed in Ω_m , $m = 1, 2$.

3 The Dirichlet-Neumann Waveform Relaxation algorithm

The Dirichlet-Neumann waveform relaxation (DNWR) method is a basic iterative substructuring method in domain decomposition. The PDEs are solved sequentially using Dirichlet-, respectively Neumann boundary with data given from the solution of the other problem introduced in [13].

It starts with an initial guess $g^0(\mathbf{x}, t)$ on the interface $\Gamma \times (T_0, T_f]$, and then performs a three-step iteration. At each iteration k , imposing continuity of the solution across the interface, one first finds the local solution $u_1^{k+1}(\mathbf{x}, t)$ on Ω_1 by solving the Dirichlet problem:

$$\begin{cases} \alpha_1 \frac{\partial u_1^{k+1}(\mathbf{x}, t)}{\partial t} - \nabla \cdot (\lambda_1 \nabla u_1^{k+1}(\mathbf{x}, t)) = 0, & \mathbf{x} \in \Omega_1, \\ u_1^{k+1}(\mathbf{x}, t) = 0, & \mathbf{x} \in \partial\Omega_1 \setminus \Gamma, \\ u_1^{k+1}(\mathbf{x}, t) = g^k(\mathbf{x}, t), & \mathbf{x} \in \Gamma, \\ u_1^{k+1}(\mathbf{x}, 0) = u_1^0(\mathbf{x}), & \mathbf{x} \in \Omega_1. \end{cases} \quad (3)$$

Then, imposing continuity of the heat fluxes across the interface, one finds the local solution $u_2^{k+1}(\mathbf{x}, t)$ on Ω_2 by solving the Neumann problem:

$$\begin{cases} \alpha_2 \frac{\partial u_2^{k+1}(\mathbf{x}, t)}{\partial t} - \nabla \cdot (\lambda_2 \nabla u_2^{k+1}(\mathbf{x}, t)) = 0, & \mathbf{x} \in \Omega_2, \\ u_2^{k+1}(\mathbf{x}, t) = 0, & \mathbf{x} \in \partial\Omega_2 \setminus \Gamma, \\ \lambda_2 \frac{\partial u_2^{k+1}(\mathbf{x}, t)}{\partial \mathbf{n}_2} = -\lambda_1 \frac{\partial u_1^{k+1}(\mathbf{x}, t)}{\partial \mathbf{n}_1}, & \mathbf{x} \in \Gamma, \\ u_2^{k+1}(\mathbf{x}, 0) = u_2^0(\mathbf{x}), & \mathbf{x} \in \Omega_2. \end{cases} \quad (4)$$

Finally, the interface values are updated with

$$g^{k+1}(\mathbf{x}, t) = \Theta u_2^{k+1}(\mathbf{x}, t) + (1 - \Theta)g^k(\mathbf{x}, t), \quad \mathbf{x} \in \Gamma, \quad (5)$$

where $\Theta \in (0, 1]$ is the relaxation parameter. The optimal relaxation parameter for the DNWR algorithm has been proved to be $\Theta = 1/2$ in [13] for the choice $\lambda_1 = \lambda_2 = \alpha_1 = \alpha_2 = 1$.

4 The Neumann-Neumann Waveform Relaxation algorithm

We now describe the Neumann-Neumann waveform relaxation (NNWR) algorithm [21]. The solution given by the NNWR method corresponds to the solution of the model problem (1) (proved in [18, chapt. 2]). The main advantage of the NNWR method is that it allows to find the solution on the subdomains in parallel.

The NNWR algorithm starts with an initial guess $g^0(\mathbf{x}, t)$ on the space-time interface $\Gamma \times (T_0, T_f]$, and then performs a three-step iteration. At each iteration k , one first solves two Dirichlet problems on Ω_1 and Ω_2 simultaneously, then two Neumann problems are solved simultaneously again on Ω_1 and Ω_2 , and finally, an update is performed to get a new guess $g^{k+1}(\mathbf{x}, t)$ on the interface $\Gamma \times (T_0, T_f]$.

More specifically, imposing continuity of the solution across the interface (i.e. given a common initial guess $g^0(\mathbf{x}, t)$ on $\Gamma \times (T_0, T_f]$), one can find the local solutions $u_m^{k+1}(\mathbf{x}, t)$ on Ω_m , $m = 1, 2$ through the following Dirichlet problems:

$$\begin{cases} \alpha_m \frac{\partial u_m^{k+1}(\mathbf{x}, t)}{\partial t} - \nabla \cdot (\lambda_m \nabla u_m^{k+1}(\mathbf{x}, t)) = 0, & \mathbf{x} \in \Omega_m, \\ u_m^{k+1}(\mathbf{x}, t) = 0, & \mathbf{x} \in \partial\Omega_m \setminus \Gamma, \\ u_m^{k+1}(\mathbf{x}, t) = g^k(\mathbf{x}, t), & \mathbf{x} \in \Gamma, \\ u_m^{k+1}(\mathbf{x}, 0) = u_m^0(\mathbf{x}), & \mathbf{x} \in \Omega_m. \end{cases} \quad (6)$$

We now add into the framework the second coupling condition which is the continuity of the heat fluxes. To this end, one solves two simultaneous Neumann problems to get the correction functions $\psi_m^{k+1}(\mathbf{x}, t)$ on Ω_m , $m = 1, 2$ where the Neumann boundary condition at the interface $\Gamma \times (T_0, T_f]$ is prescribed by the continuity of the heat fluxes of the solutions $u_m^{k+1}(\mathbf{x}, t)$ given by the Dirichlet problems:

$$\begin{cases} \alpha_m \frac{\partial \psi_m^{k+1}(\mathbf{x}, t)}{\partial t} - \nabla \cdot (\lambda_m \nabla \psi_m^{k+1}(\mathbf{x}, t)) = 0, & \mathbf{x} \in \Omega_m, \\ \psi_m^{k+1}(\mathbf{x}, t) = 0, & \mathbf{x} \in \partial\Omega_m \setminus \Gamma, \\ \lambda_m \frac{\partial \psi_m^{k+1}(\mathbf{x}, t)}{\partial \mathbf{n}_m} = \lambda_1 \frac{\partial u_1^{k+1}(\mathbf{x}, t)}{\partial \mathbf{n}_1} + \lambda_2 \frac{\partial u_2^{k+1}(\mathbf{x}, t)}{\partial \mathbf{n}_2}, & \mathbf{x} \in \Gamma, \\ \psi_m^{k+1}(\mathbf{x}, 0) = 0, & \mathbf{x} \in \Omega_m. \end{cases} \quad (7)$$

Finally, the interface values are updated with

$$g^{k+1}(\mathbf{x}, t) = g^k(\mathbf{x}, t) - \Theta(\psi_1^{k+1}(\mathbf{x}, t) + \psi_2^{k+1}(\mathbf{x}, t)), \quad \mathbf{x} \in \Gamma, \quad (8)$$

where $\Theta \in (0, 1]$ is the relaxation parameter. Note that choosing an appropriate relaxation parameter is crucial for the good performance of the NNWR algorithm [13]. If one uses the optimal relaxation parameter, two iterations are enough.

5 Semidiscrete method

We now describe a rather general space discretization of the problem (6)-(8). The core property we need is that the meshes of Ω_1 and Ω_2 share the same nodes on Γ as shown in figure 1. Furthermore, we assume that there is a specific set of unknowns associated with the interface nodes. Otherwise, we allow at this point for arbitrary meshes on both sides.

Then, letting $\mathbf{u}_I^{(m)}, \psi_I^{(m)} : [T_0, T_f] \rightarrow \mathbb{R}^{S_m}$ where S_m is the number of grid points on Ω_m , $m = 1, 2$, and $\mathbf{u}_\Gamma, \psi_\Gamma^{(1)}, \psi_\Gamma^{(2)} : [T_0, T_f] \rightarrow \mathbb{R}^s$, where s is the number of grid points at the interface Γ , we can write a general discretization of the first equation in (6) and (7), respectively, in a compact form as:

$$\mathbf{M}_{II}^{(m)} \dot{\mathbf{u}}_I^{(m), k+1}(t) + \mathbf{A}_{II}^{(m)} \mathbf{u}_I^{(m), k+1}(t) = -\mathbf{M}_{I\Gamma}^{(m)} \dot{\mathbf{u}}_\Gamma^k(t) - \mathbf{A}_{I\Gamma}^{(m)} \mathbf{u}_\Gamma^k(t), \quad (9)$$

$$\mathbf{M}_{II}^{(m)} \dot{\psi}_I^{(m), k+1}(t) + \mathbf{M}_{I\Gamma}^{(m)} \dot{\psi}_\Gamma^{(m), k+1}(t) + \mathbf{A}_{II}^{(m)} \psi_I^{(m), k+1}(t) + \mathbf{A}_{I\Gamma}^{(m)} \psi_\Gamma^{(m), k+1}(t) = \mathbf{0}, \quad (10)$$

where the initial conditions $\mathbf{u}_I^{(m)}(T_0), \psi_I^{(m)}(T_0) \in \mathbb{R}^{S_m}$ and $\mathbf{u}_\Gamma(T_0), \psi_\Gamma^{(m)}(T_0) \in \mathbb{R}^s$ for $m = 1, 2$ are known.

To close the system, we need an approximation of the normal derivatives on Γ . Letting ϕ_j be a nodal FE basis function on Ω_m for a node on Γ we observe that the normal derivative of u_m with respect to the interface can be written as a linear functional using Green's formula [29, pp. 3]. Thus, the approximation of the normal derivative is given by

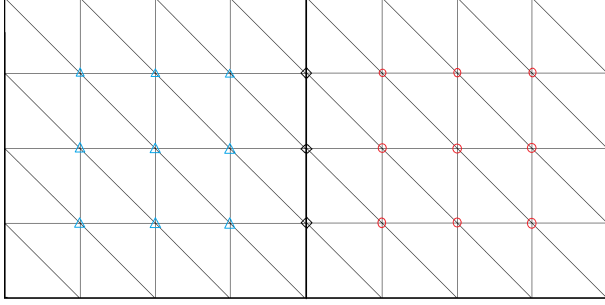


Figure 1: Splitting of Ω and finite element triangulation.

$$\begin{aligned}
 \lambda_m \int_{\Gamma} \frac{\partial u_m}{\partial \mathbf{n}_m} \phi_j dS &= \lambda_m \int_{\Omega_m} (\Delta u_m \phi_j + \nabla u_m \nabla \phi_j) d\mathbf{x} \\
 &= \alpha_m \int_{\Omega_m} \frac{d}{dt} u_m \phi_j + \lambda_m \int_{\Omega_m} \nabla u_m \nabla \phi_j d\mathbf{x}, \quad m = 1, 2.
 \end{aligned} \tag{11}$$

Consequently, the equation

$$\begin{aligned}
 &\mathbf{M}_{\Gamma\Gamma}^{(m)} \dot{\psi}_{\Gamma}^{(m),k+1}(t) + \mathbf{M}_{\Gamma I}^{(m)} \dot{\psi}_I^{(m),k+1}(t) + \mathbf{A}_{\Gamma\Gamma}^{(m)} \psi_{\Gamma}^{(m),k+1}(t) + \mathbf{A}_{\Gamma I}^{(m)} \psi_I^{(m),k+1}(t) \\
 &= \sum_{i=1}^2 \left(\mathbf{M}_{\Gamma\Gamma}^{(i)} \dot{\mathbf{u}}_{\Gamma}^k(t) + \mathbf{M}_{\Gamma I}^{(i)} \dot{\mathbf{u}}_I^{(i),k+1}(t) + \mathbf{A}_{\Gamma\Gamma}^{(i)} \mathbf{u}_{\Gamma}^k(t) + \mathbf{A}_{\Gamma I}^{(i)} \mathbf{u}_I^{(i),k+1}(t) \right), \quad m = 1, 2,
 \end{aligned} \tag{12}$$

is a discrete version of the third equation in (7) and completes the system (10).

We can now write a semidiscrete version of the NNWR algorithm using an ODE system. At each iteration k , one first solves the two Dirichlet problems in (9) obtaining $\mathbf{u}_I^{(m),k+1}(t)$ for $m = 1, 2$. Then, for the vector of unknowns $\psi_m^{k+1}(t) = \left(\psi_I^{(m),k+1}(t)^T \psi_{\Gamma}^{(m),k+1}(t)^T \right)^T$, one solves the following two Neumann problems in parallel that correspond to equations (10)-(12):

$$\mathbf{M}_m \dot{\psi}_m^{k+1}(t) + \mathbf{A}_m \psi_m^{k+1}(t) = \mathbf{b}^k, \quad m = 1, 2, \tag{13}$$

where

$$\mathbf{M}_m = \begin{pmatrix} \mathbf{M}_{II}^{(m)} & \mathbf{M}_{I\Gamma}^{(m)} \\ \mathbf{M}_{\Gamma I}^{(m)} & \mathbf{M}_{\Gamma\Gamma}^{(m)} \end{pmatrix}, \quad \mathbf{A}_m = \begin{pmatrix} \mathbf{A}_{II}^{(m)} & \mathbf{A}_{I\Gamma}^{(m)} \\ \mathbf{A}_{\Gamma I}^{(m)} & \mathbf{A}_{\Gamma\Gamma}^{(m)} \end{pmatrix}, \quad \mathbf{b}^k = \begin{pmatrix} \mathbf{0} \\ \mathbf{F}^k \end{pmatrix}, \tag{14}$$

with

$$\mathbf{F}^k = \sum_{i=1}^2 \left(\mathbf{M}_{\Gamma\Gamma}^{(i)} \dot{\mathbf{u}}_{\Gamma}^k(t) + \mathbf{M}_{\Gamma I}^{(i)} \dot{\mathbf{u}}_I^{(i),k+1}(t) + \mathbf{A}_{\Gamma\Gamma}^{(i)} \mathbf{u}_{\Gamma}^k(t) + \mathbf{A}_{\Gamma I}^{(i)} \mathbf{u}_I^{(i),k+1}(t) \right). \tag{15}$$

Finally, the interfaces values are updated by

$$\mathbf{u}_{\Gamma}^{k+1}(t) = \mathbf{u}_{\Gamma}^k(t) - \Theta \left(\psi_{\Gamma}^{(1),k+1}(t) + \psi_{\Gamma}^{(2),k+1}(t) \right). \tag{16}$$

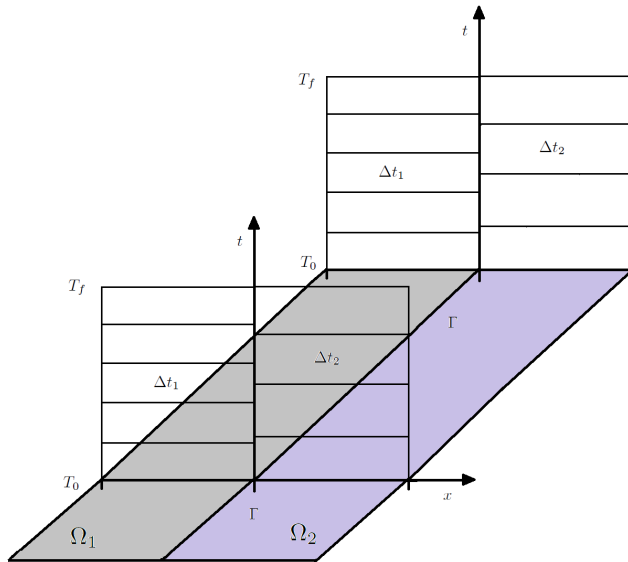


Figure 2: Nonconforming time grids in the two-dimensional subdomains.

The iteration starts with some initial condition $\mathbf{u}_\Gamma^0(t)$ and a termination criterion must be chosen. One option would be $\|\mathbf{u}_\Gamma^{k+1}(t) - \mathbf{u}_\Gamma^k(t)\| \leq TOL$ where TOL is a user defined tolerance. However, this option is memory consuming because it saves the solutions for all $t \in [T_0, T_f]$. Moreover, an extra interpolation step is needed in the multirate case, i.e., when having two nonconforming time grids. As we expect the error to be largest at the end point T_f and because it simplifies the analysis to be presented for finding the optimal relaxation parameter, we propose the criterion $\|\mathbf{u}_\Gamma^{k+1}(T_f) - \mathbf{u}_\Gamma^k(T_f)\| \leq TOL$ where T_f is the synchronization endpoint of the macrostep.

6 Space-time interface interpolation

The NNWR algorithm for parabolic problems was first introduced in [21, 13], but they do not consider the possibility of using two different step sizes on the two subdomains. In addition, their analysis does not include the coupling of two different materials. For those reasons, the goal of this paper is to introduce a parallel multirate method for the coupling of two heterogeneous heat equations and analyze its performance in the fully discrete case. This would be especially useful when coupling two different materials, where typically the field with higher heat conductivity needs a finer resolution than the other and therefore, efficiency will be gained by using a multirate method.

Note that both the Dirichlet and the Neumann problems in (9) and (13) allow the use of independent time discretization on each of the subdomains. Therefore, in the case of mismatched time grids, there exists the need to define an interface interpolation.

To this end, we consider a discrete problem in time with nonconforming time grids. Let $\tau_1 = \{t_1, t_2, \dots, t_{N_1}\}$ and $\tau_2 = \{t_1, t_2, \dots, t_{N_2}\}$ be two possibly different partitions of the time interval $[T_0, T_f]$ as shown in figure 2. We denote by $\Delta t_1 = (T_f - T_0)/N_1$ and $\Delta t_2 = (T_f - T_0)/N_2$ the two possibly different constant stepsizes corresponding to Ω_1 and Ω_2 respectively.

In order to exchange data at the space-time interface between the different time grids, we use a linear interpolation. For instance, if we want to interpolate the local discrete solution $G := (G_1, G_2, \dots, G_s) \in \mathbb{R}^{s \times N_2}$ from a given local discrete solution $F := (F_1, F_2, \dots, F_s) \in \mathbb{R}^{s \times N_1}$ at the space-time interface $\Gamma \times [T_0, T_f]$, with s being the number of grid points at Γ , we use the following procedure: for each $k = 1, 2, \dots, s$ we consider the discrete vector $F_k := (F_{k,1}, F_{k,2}, \dots, F_{k,N_1}) \in \mathbb{R}^{N_1}$. Then, for each $t_i \in \tau_2$, $i = 1, \dots, N_2$, we find the subinterval $[t_j, t_{j+1}]$, $j = 0, \dots, N_1 - 1$ such that $t_i \in [t_j, t_{j+1}]$. We then define the linear interpolation polynomial $p_j(t)$ through the points

$(t_j, F_{k,j})$ and $(t_{j+1}, F_{k,j+1})$, i.e. $p_j(t) = F_{k,j} + (t - t_j) \cdot (F_{k,j+1} - F_{k,j}) / (t_{j+1} - t_j)$. Finally, we evaluate p_j at t_i . Repeating the procedure for all the elements of τ_2 we get the discrete vector G_k whose components are given by,

$$G_{k,i} = \{p_j(t_i) \mid p_j : [t_j, t_{j+1}] \rightarrow \mathbb{R}, \forall t \in [t_j, t_{j+1}], j = 0, \dots, N_1 - 1\}, \quad (17)$$

for $i = 1, \dots, N_2$.

Thus, to interpolate G from F we use the interpolation function $G = I(\tau_2, \tau_1, F)$ explained above and summarized in algorithm 1. The same procedure can be applied to interpolate the discrete solution F from a given discrete solution G .

Algorithm 1 Interpolation to transfer data at the space-time interface.

```

1: procedure I( $\tau_2, \tau_1, F$ )
2:   for  $k = 1, 2, \dots, s$  do
3:     for  $t_i \in \tau_2$  do
4:       for  $t_j \in \tau_1$  do
5:         if  $t_i \in [t_j, t_{j+1}]$  then
6:            $G_{k,i} \leftarrow F_{k,j} + (t_i - t_j) \cdot (F_{k,j+1} - F_{k,j}) / (t_{j+1} - t_j)$ 
       return  $G$ 

```

7 Time integration

In this section we present a time discretized version of the NNWR method presented in equations (9), (13) and (16). In order to get a multirate algorithm we use a certain time integration method with time step $\Delta t_1 := (T_f - T_0) / N_1$ on Ω_1 and with time step $\Delta t_2 := (T_f - T_0) / N_2$ on Ω_2 and the interpolation presented in the previous section will be used to transfer data from one domain to the other. We let $n_m := 1, 2, \dots, N_m$ be the time integration indices with respect to Ω_m and t_{n_m} defines any time point of the grid for $m = 1, 2$. We have chosen two alternative time integration schemes as a basis to construct the multirate algorithm: the implicit Euler method and a second order singly diagonally implicit Runge-Kutta method (SDIRK2).

7.1 Implicit Euler

Applying the implicit Euler method with time step Δt_1 on Ω_1 and with time step Δt_2 on Ω_2 we can write the systems (9), (13) and (16) in a fully discrete form. At each fixed point iteration k , one first performs the time integration of the Dirichlet problems in parallel. Secondly, the interpolation explained in the previous section is used for the boundary conditions in order to solve the Neumann problems in parallel. Once the Neumann problems are solved, interpolation is again used to match the components of the update step. Finally, if the termination criterion is not fulfilled, one starts the process once more.

The local approximations and the solutions at the space-time interface are given by the vectors $\mathbf{u}_I^{(m),k,n_m} \approx \mathbf{u}_I^{(m),k}(t_{n_m}) \in \mathbb{R}^{S_m}$ and $\mathbf{u}_\Gamma^{k,n_m} \approx \mathbf{u}_\Gamma^k(t_{n_m}) \in \mathbb{R}^s$ respectively. Remember that S_m is the number of spatial grid points on Ω_m and s is the number of spatial grid points at the interface Γ . Similarly, the corrections both in the subdomains and at the interface are given by the vectors $\psi_I^{(m),k,n_m} \approx \psi_I^{(m),k}(t_{n_m}) \in \mathbb{R}^{S_m}$ and $\psi_\Gamma^{(m),k,n_m} \approx \psi_\Gamma^{(m),k}(t_{n_m}) \in \mathbb{R}^s$, respectively.

At each iteration k , one first solves the two Dirichlet problems from (9) for $n_m = 1, 2, \dots, N_m$, with $\mathbf{u}_I^{(m),k+1,0} \approx \mathbf{u}_I^{(m)}(T_0)$, $m = 1, 2$ and $\mathbf{u}_\Gamma^{k+1,0} \approx \mathbf{u}_\Gamma(T_0)$ simultaneously:

$$\begin{aligned} \left(\frac{\mathbf{M}_{II}^{(m)}}{\Delta t_m} + \mathbf{A}_{II}^{(m)} \right) \mathbf{u}_I^{(m),k+1,n_m+1} = & - \left(\frac{\mathbf{M}_{I\Gamma}^{(m)}}{\Delta t_m} + \mathbf{A}_{I\Gamma}^{(m)} \right) \mathbf{u}_\Gamma^{k,n_m+1} \\ & + \frac{\mathbf{M}_{II}^{(m)}}{\Delta t_m} \mathbf{u}_I^{(m),k+1,n_m} + \frac{\mathbf{M}_{I\Gamma}^{(m)}}{\Delta t_m} \mathbf{u}_\Gamma^{k,n_m}, \end{aligned} \quad (18)$$

for $m = 1, 2$. Note that interpolation is not needed to solve the Dirichlet problems because $\mathbf{u}_I^{(1),k+1,n_1+1}$ in (18) is only dependent on terms related to Ω_1 . In the same way, $\mathbf{u}_I^{(2),k+1,n_2+1}$ in (18) only depends on n_2 .

We compute now the fluxes $\tilde{\mathbf{F}}_1^{k,\tau_1} := \tilde{\mathbf{f}}_1^{k,\tau_1} + I(\tau_1, \tau_2, \tilde{\mathbf{f}}_2^{k,\tau_2})$ and $\tilde{\mathbf{F}}_2^{k,\tau_2} := \tilde{\mathbf{f}}_2^{k,\tau_2} + I(\tau_2, \tau_1, \tilde{\mathbf{f}}_1^{k,\tau_1})$ in (15) with

$$\begin{aligned} \tilde{\mathbf{f}}_m^{k,n_m} = & \left(\frac{\mathbf{M}_{\Gamma\Gamma}^{(m)}}{\Delta t_m} + \mathbf{A}_{\Gamma\Gamma}^{(m)} \right) \mathbf{u}_{\Gamma}^{k,n_m+1} + \left(\frac{\mathbf{M}_{\Gamma I}^{(m)}}{\Delta t_m} + \mathbf{A}_{\Gamma I}^{(m)} \right) \mathbf{u}_I^{(m),k+1,n_m+1} \\ & - \frac{\mathbf{M}_{\Gamma\Gamma}^{(m)}}{\Delta t_m} \mathbf{u}_{\Gamma}^{k,n_m} - \frac{\mathbf{M}_{\Gamma I}^{(m)}}{\Delta t_m} \mathbf{u}_I^{(m),k+1,n_m}, \end{aligned} \quad (19)$$

where $n_m = 1, \dots, N_m$ and $\tau_m = \{t_1, t_2, \dots, t_{N_m}\}$ are the corresponding time grids on Ω_m for $m = 1, 2$. Note that unlike in the Dirichlet problems, we need to use the interpolation described in the previous section. We use it to calculate $\tilde{\mathbf{F}}_1^{k,\tau_1}$ and $\tilde{\mathbf{F}}_2^{k,\tau_2}$ because their components run over different time integrations (one indicated by n_1 and the other by n_2).

One can now rewrite the Neumann problems in (13) in terms of the vector of unknowns $\psi_m^{k+1,n_m+1} := \left(\psi_I^{(m),k+1,n_m+1T} \psi_{\Gamma}^{(m),k+1,n_m+1T} \right)^T$. One then solves the two Neumann problems for $n_m = 1, 2, \dots, N_m$, with $\psi_m^{k+1,0} \approx \psi_m(T_0)$, $m = 1, 2$ in parallel:

$$\left(\frac{\mathbf{M}_m}{\Delta t_m} + \mathbf{A}_m \right) \psi_m^{k+1,n_m+1} = \frac{\mathbf{M}_m}{\Delta t_m} \psi_m^{k+1,n_m} + \tilde{\mathbf{b}}^{k,n_m}, \quad (20)$$

where $\tilde{\mathbf{b}}^{k,n_m} = \left(\mathbf{0}^T \tilde{\mathbf{F}}_m^{k,n_mT} \right)^T$.

Then, the interfaces values are updated respectively by

$$\mathbf{u}_{\Gamma}^{k+1,\tau_1} = \mathbf{u}_{\Gamma}^{k,\tau_1} - \Theta \left(\psi_{\Gamma}^{(1),k+1,\tau_1} + I \left(\tau_1, \tau_2, \psi_{\Gamma}^{(2),k+1,\tau_2} \right) \right), \quad (21)$$

$$\mathbf{u}_{\Gamma}^{k+1,\tau_2} = \mathbf{u}_{\Gamma}^{k,\tau_2} - \Theta \left(\psi_{\Gamma}^{(2),k+1,\tau_2} + I \left(\tau_2, \tau_1, \psi_{\Gamma}^{(1),k+1,\tau_1} \right) \right). \quad (22)$$

Here, interpolation is needed to perform the additions because $\psi_{\Gamma}^{(1),k+1,\tau_1}$ and $\psi_{\Gamma}^{(2),k+1,\tau_2}$ correspond to different time integrations.

Finally, if the termination criteria $\|\mathbf{u}_{\Gamma}^{k+1,N_m} - \mathbf{u}_{\Gamma}^{k,N_m}\| \approx \|\mathbf{u}_{\Gamma}^{k+1}(T_f) - \mathbf{u}_{\Gamma}^k(T_f)\|$ is not small enough, one starts the process from (18) once more.

Summarizing, figure 3 sketches the communication needed for the NNWR algorithm just explained. Algorithm 2 and 3 summarize the discrete Dirichlet solver in (18) and the discrete Neumann solver in (20) respectively. Furthermore, the complete NNWR algorithm is summarized in algorithm 4.

Algorithm 2 Solver for the Dirichlet problems in (18).

- 1: **procedure** SOLVEDIRICHLET($\mathbf{u}_I^{k+1,n}, \mathbf{u}_{\Gamma}^{k,n}, \mathbf{u}_{\Gamma}^{k,n+1}$)
 - 2: $\mathbf{u}_I^{k+1,n+1} \leftarrow \text{SOLVILINEARSYSTEM}(\mathbf{u}_I^{k+1,n}, \mathbf{u}_{\Gamma}^{k,n}, \mathbf{u}_{\Gamma}^{k,n+1})$, # solve (18)
 - 3: $\tilde{\mathbf{f}}^{k,n} \leftarrow \left(\frac{\mathbf{M}_{\Gamma\Gamma}}{\Delta t} + \mathbf{A}_{\Gamma\Gamma} \right) \mathbf{u}_{\Gamma}^{k,n+1} + \left(\frac{\mathbf{M}_{\Gamma I}}{\Delta t} + \mathbf{A}_{\Gamma I} \right) \mathbf{u}_I^{k+1,n+1} - \frac{\mathbf{M}_{\Gamma\Gamma}}{\Delta t} \mathbf{u}_{\Gamma}^{k,n} - \frac{\mathbf{M}_{\Gamma I}}{\Delta t} \mathbf{u}_I^{k+1,n}$, # compute (19)
- return** $\mathbf{u}_I^{k+1,n+1}, \tilde{\mathbf{f}}^{k,n}$
-

Algorithm 3 Solver for the Neumann problems in (20).

- 1: **procedure** SOLVENEUMANN($\psi_I^{k+1,n}, \psi_{\Gamma}^{k+1,n}, \tilde{\mathbf{F}}^{k,n}$)
 - 2: $\psi_I^{k+1,n+1}, \psi_{\Gamma}^{k+1,n+1} \leftarrow \text{SOLVILINEARSYSTEM}(\psi_I^{k+1,n}, \psi_{\Gamma}^{k+1,n}, \tilde{\mathbf{F}}^{k,n})$, # solve (20)
- return** $\psi_I^{k+1,n+1}, \psi_{\Gamma}^{k+1,n+1}$
-

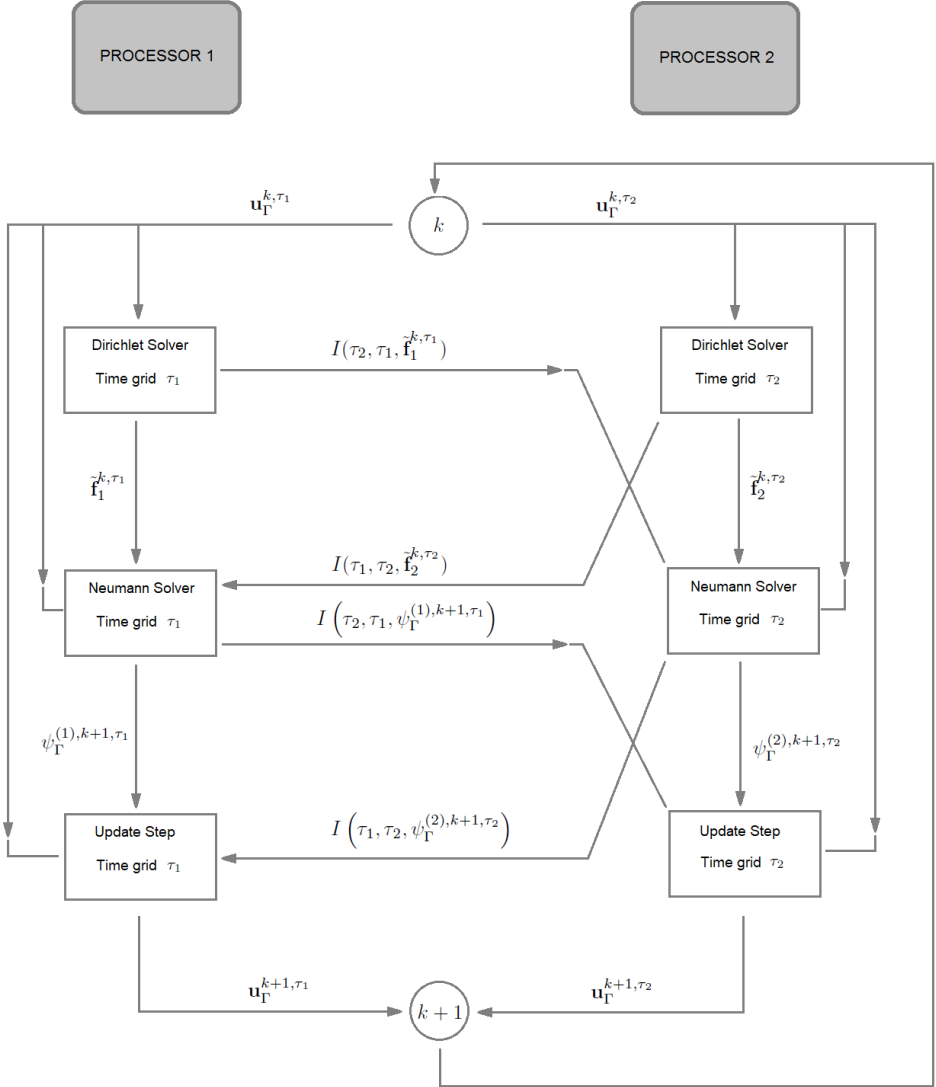


Figure 3: Illustration of the NNWR algorithm using implicit Euler. The process starts with the space-time interface functions $\mathbf{u}_\Gamma^{k, \tau_m}$, $\tau_m = \{t_1, t_2, \dots, t_{N_m}\}$ for $m = 1, 2$ corresponding to the two nonconforming time grids. Those are needed to run the Dirichlet solvers in parallel getting $\mathbf{u}_\Gamma^{(m), k+1, \tau_m}$, $m = 1, 2$. In order to run the Neumann solvers for the corrections of the solution, one needs to provide the fluxes $\tilde{\mathbf{f}}_1^{k, \tau_1}$, $\tilde{\mathbf{f}}_2^{k, \tau_2}$ and their corresponding interpolations $I(\tau_2, \tau_1, \tilde{\mathbf{f}}_1^{k, \tau_1})$, $I(\tau_1, \tau_2, \tilde{\mathbf{f}}_2^{k, \tau_2})$. One can then run the Neumann problems in parallel getting the corrections $\psi_\Gamma^{(1), k+1, \tau_1}$ and $\psi_\Gamma^{(2), k+1, \tau_2}$ at the space-time interface. Finally, those and their interpolations $I(\tau_1, \tau_2, \psi_\Gamma^{(2), k+1, \tau_2})$ and $I(\tau_2, \tau_1, \psi_\Gamma^{(1), k+1, \tau_1})$ are used to update the space-time interface values. If needed, the process is restarted.

Algorithm 4 NNWR algorithm using implicit Euler.

```

1: procedure NNWR( $\tau_1, \tau_2, \alpha_1, \alpha_2, \lambda_1, \lambda_2, \Theta, TOL$ )
2:    $\mathbf{u}_I^{(m),k+1,0}, \mathbf{u}_I^0(\tau_m), \psi_I^{(m),k+1,n_m+1}, \psi_I^{(m),k+1,n_m+1} \leftarrow$  INITIALIZATION
3:   while  $\|\mathbf{u}_I^{k+1,N_m} - \mathbf{u}_I^{k,N_m}\| \leq TOL$  do
4:     for  $t_{n_1} \in \tau_1$  do
5:        $\mathbf{u}_I^{(1),k+1,n_1+1}, \tilde{\mathbf{f}}_1^{k,n_1} \leftarrow$  SOLVEDIRICHLET( $\mathbf{u}_I^{(1),k+1,n_1}, \mathbf{u}_I^{k,n_1}, \mathbf{u}_I^{k,n_1+1}$ )
6:     for  $t_{n_2} \in \tau_2$  do (in parallel to 4)
7:        $\mathbf{u}_I^{(2),k+1,n_2+1}, \tilde{\mathbf{f}}_2^{k,n_2} \leftarrow$  SOLVEDIRICHLET( $\mathbf{u}_I^{(2),k+1,n_2}, \mathbf{u}_I^{k,n_2}, \mathbf{u}_I^{k,n_2+1}$ )
8:        $\tilde{\mathbf{F}}_1^{k,\tau_1} \leftarrow \tilde{\mathbf{f}}_1^{k,\tau_1} + \mathbf{I}(\tau_1, \tau_2, \tilde{\mathbf{f}}_2^{k,\tau_2})$ 
9:        $\tilde{\mathbf{F}}_2^{k,\tau_2} \leftarrow \tilde{\mathbf{f}}_2^{k,\tau_2} + \mathbf{I}(\tau_2, \tau_1, \tilde{\mathbf{f}}_1^{k,\tau_1})$  (in parallel to 8)
10:      for  $t_{n_1} \in \tau_1$  do
11:         $\psi_I^{(1),k+1,n_1+1}, \psi_I^{(1),k+1,n_1+1} \leftarrow$  SOLVENEUMANN( $\psi_I^{(1),k+1,n_1}, \psi_I^{(1),k+1,n_1}, \tilde{\mathbf{F}}_1^{k,n_1}$ )
12:      for  $t_{n_2} \in \tau_2$  do (in parallel to 10)
13:         $\psi_I^{(2),k+1,n_2+1}, \psi_I^{(2),k+1,n_2+1} \leftarrow$  SOLVENEUMANN( $\psi_I^{(2),k+1,n_2}, \psi_I^{(2),k+1,n_2}, \tilde{\mathbf{F}}_2^{k,n_2}$ )
14:         $\mathbf{u}_I^{k+1,\tau_1} \leftarrow \mathbf{u}_I^{k,\tau_1} - \Theta \left( \psi_I^{(1),k+1,\tau_1} + \mathbf{I}(\tau_1, \tau_2, \psi_I^{(2),k+1,\tau_2}) \right)$ 
15:         $\mathbf{u}_I^{k+1,\tau_2} \leftarrow \mathbf{u}_I^{k,\tau_2} - \Theta \left( \psi_I^{(2),k+1,\tau_2} + \mathbf{I}(\tau_2, \tau_1, \psi_I^{(1),k+1,\tau_1}) \right)$  (in p. to 14)

```

7.2 SDIRK2

As an alternative, we introduce here a higher order version of the multirate algorithm presented above. Specifically, we consider the second order singly diagonally implicit Runge-Kutta (SDIRK2) as a basis to discretize the systems (9), (13) and (16) in time. Consider an autonomous initial value problem

$$\dot{\mathbf{u}}(t) = \mathbf{f}(\mathbf{u}(t)), \quad \mathbf{u}(0) = \mathbf{u}_0. \quad (23)$$

An SDIRK method is then defined as

$$\begin{aligned} \mathbf{U}^i &= \mathbf{u}^n + \Delta t_n \sum_{k=1}^i a_{ik} \mathbf{f}(\mathbf{U}^k), \quad i = 1, \dots, j \\ \mathbf{u}^{n+1} &= \mathbf{u}^n + \Delta t_n \sum_{i=1}^j b_i \mathbf{f}(\mathbf{U}^i) \end{aligned} \quad (24)$$

with given coefficients a_{ik} and b_i . The two-stage method SDIRK2 is defined by the coefficients in the following Butcher array:

$$\begin{array}{c|cc} a & a & 0 \\ 1 & 1-a & a \\ \hline & 1-a & a \end{array}$$

with $a = 1 - \frac{1}{2}\sqrt{2}$. As the coefficients a_{2i} and b_i for $i = 1, 2$ are identical, the second equation in (24) is superfluous because $\mathbf{u}^{n+1} = \mathbf{U}^2$.

The vectors $\mathbf{k}_i = \mathbf{f}(\mathbf{U}^i)$ are called stage derivatives and j is the number of stages. Since the starting vector

$$\mathbf{s}_i = \mathbf{u}^n + \Delta t_n \sum_{k=1}^{i-1} a_{ik} \mathbf{k}_k, \quad i = 1, \dots, j-1, \quad (25)$$

is known, (24) is just a sequence of implicit Euler steps.

Applying SDIRK2 with time step Δt_1 on Ω_1 and with time step Δt_2 on Ω_2 we can write the systems (9), (13) and (16) in a fully discrete form. This algorithm preserves more or less the same structure as the one presented

above for implicit Euler. The main difference lies in the fact that now both the Dirichlet and the Neumann solvers have to take into account the two stages of SDIRK2 as well as the interpolation has to be applied for each stage.

Therefore, at each fixed point iteration k , let $\mathbf{s}_1^{(m)} = \mathbf{u}_I^{(m),k+1,n_m}$ and $\mathbf{s}_2^{(m)} = \mathbf{u}_I^{(m),k+1,n_m} + \Delta t_m(1-a)\mathbf{k}_1^{(m)}$ be the starting vectors. Then, one first solves the two Dirichlet problems for $n_m = 1, 2, \dots, N_m$, with $\mathbf{u}_I^{(m),k+1,0}$, $m = 1, 2$, $\mathbf{u}_\Gamma^{k+1,0}$ simultaneously:

$$\begin{aligned} \left(\frac{\mathbf{M}_{II}^{(m)}}{a\Delta t_m} + \mathbf{A}_{II}^{(m)} \right) \mathbf{U}_j^{(m)} &= \frac{\mathbf{M}_{II}^{(m)}}{a\Delta t_m} \mathbf{s}_j^{(m)} - \mathbf{M}_{I\Gamma}^{(m)} \mathbf{u}_\Gamma^{k,n_m+j-1+(2-j)a} \\ &\quad - \mathbf{A}_{I\Gamma}^{(m)} \mathbf{u}_\Gamma^{k,n_m+j-1+(2-j)a}, \quad j = 1, 2, \\ \mathbf{u}_I^{(m),k+1,n_m+1} &= \mathbf{U}_2^{(m)}, \end{aligned} \quad (26)$$

where $\mathbf{u}_I^{(m),k,n_m}$, $\mathbf{U}_j^{(m)}$, $\mathbf{s}_j^{(m)}$, $\mathbf{k}_j^{(m)} \in \mathbb{R}^{S_m}$ and $\mathbf{u}_\Gamma^{k,n_m} \in \mathbb{R}^s$. The stage derivatives are given by $\mathbf{k}_j^{(m)} = \frac{1}{a\Delta t_m}(\mathbf{U}_j^{(m)} - \mathbf{s}_j^{(m)})$. Note that the index $m = 1, 2$, denotes the subdomain and the index $j = 1, 2$, denotes the stage.

We compute now the fluxes $\mathbf{F}_j^{(1),k,\tau_1} := \mathbf{f}_j^{(1),k,\tau_1} + I(\tau_1, \tau_2, \mathbf{f}_j^{(2),k,\tau_2})$, $\mathbf{F}_j^{(2),k,\tau_2} := \mathbf{f}_j^{(2),k,\tau_2} + I(\tau_2, \tau_1, \mathbf{f}_j^{(1),k,\tau_1})$ in (15) with

$$\begin{aligned} \mathbf{f}_j^{(m),k,n_m} &= \mathbf{M}_{I\Gamma}^{(m)} \mathbf{u}_\Gamma^{k,n_m+j-1+(2-j)a} + \mathbf{M}_{II}^{(m)} \mathbf{k}_j^{(m)} \\ &\quad + \mathbf{A}_{I\Gamma}^{(m)} \mathbf{u}_\Gamma^{k,n_m+j-1+(2-j)a} + \mathbf{A}_{II}^{(m)} \mathbf{U}_j^{(m)}, \end{aligned} \quad (27)$$

for $m = 1, 2$. Note that interpolation here is needed because the components of $\mathbf{F}_j^{(1),k,\tau_1}$ and $\mathbf{F}_j^{(2),k,\tau_2}$ for the two stages $j = 1, 2$ correspond to different time integrations.

One can now rewrite the Neumann problems in (13) in terms of the vector of unknowns $\psi_m^{k+1,n_m+1} := \left(\psi_I^{(m),k+1,n_m+1T} \right)^T$ where $\psi_I^{(m),k+1,n_m+1} \in \mathbb{R}^{S_m}$ and $\psi_\Gamma^{(m),k+1,n_m+1} \in \mathbb{R}^s$. Let $\mathbf{s}_1^{(m)} = \psi_m^{k+1,n_m}$ and $\mathbf{s}_2^{(m)} = \psi_m^{k+1,n_m} + \Delta t_m(1-a)\mathbf{k}_1^{(m)}$ be the starting vectors. One then solves the two Neumann problems for $n_m = 1, 2, \dots, N_m$, with $\psi_m^{k+1,0} = \psi_m^{k+1}(T_0)$, $m = 1, 2$ in parallel:

$$\begin{aligned} \left(\frac{\mathbf{M}_m}{a\Delta t_m} + \mathbf{A}_m \right) \mathbf{Y}_j^{(m)} &= \frac{\mathbf{M}_m}{a\Delta t_m} \mathbf{s}_j^{(m)} + \mathbf{b}_j^{(m),k,n_m}, \quad j = 1, 2, \\ \psi_m^{k+1,n_m+1} &= \mathbf{Y}_2^{(m)}, \end{aligned} \quad (28)$$

where $\mathbf{Y}_j^{(m)}$, $\mathbf{s}_j^{(m)}$, $\mathbf{b}_j^{(m),k,n_m}$, $\mathbf{k}_j^{(m)} \in \mathbb{R}^{S_m+s}$, $\mathbf{k}_j^{(m)} = \frac{1}{a\Delta t_m}(\mathbf{Y}_j^{(m)} - \mathbf{s}_j^{(m)})$ and $\mathbf{b}_j^{(m),k,n_m} = \left(\mathbf{0}^T \mathbf{F}_j^{(m),k,n_mT} \right)^T$.

Then, the interfaces values are updated respectively by

$$\mathbf{u}_\Gamma^{k+1,\tau_1} = \mathbf{u}_\Gamma^{k,\tau_1} - \Theta \left(\psi_\Gamma^{(1),k+1,\tau_1} + I \left(\tau_1, \tau_2, \psi_\Gamma^{(2),k+1,\tau_2} \right) \right), \quad (29)$$

$$\mathbf{u}_\Gamma^{k+1,\tau_2} = \mathbf{u}_\Gamma^{k,\tau_2} - \Theta \left(\psi_\Gamma^{(2),k+1,\tau_2} + I \left(\tau_2, \tau_1, \psi_\Gamma^{(1),k+1,\tau_1} \right) \right). \quad (30)$$

Here, interpolation is needed because $\psi_\Gamma^{(1),k+1,\tau_1}$ and $\psi_\Gamma^{(2),k+1,\tau_2}$ are nonconforming.

Finally, if the termination criteria $\|\mathbf{u}_\Gamma^{k+1,N_m} - \mathbf{u}_\Gamma^{k,N_m}\|$ is not small enough, one starts the process from (26) once more.

We use a linear interpolation through the points $(t_{n_m}, \mathbf{u}_\Gamma^{k,n_m})$ and $(t_{n_m} + \Delta t_m, \mathbf{u}_\Gamma^{k,n_m+1})$ in order to approximate $\mathbf{u}_\Gamma^{k,n_m+a}$ in the first equation of (26) and in the fluxes (27), i.e:

$$\mathbf{u}_\Gamma^{k,n_m+a} \approx \mathbf{u}_\Gamma^{k,n_m} + a \left(\mathbf{u}_\Gamma^{k,n_m+1} - \mathbf{u}_\Gamma^{k,n_m} \right). \quad (31)$$

Furthermore, there are first order time derivatives in the first equation of (26) and in (27). We use forward differences to approximate all the remaining first order derivatives:

$$\dot{\mathbf{u}}_{\Gamma}^{k,n_m+j-1+(2-j)a} \approx \frac{\mathbf{u}_{\Gamma}^{k,n_m+1} - \mathbf{u}_{\Gamma}^{k,n_m}}{\Delta t_m}, \quad (32)$$

for $j = 1, 2$ and $m = 1, 2$.

Summarizing, the SDIRK2-NNWR algorithm just presented has the same structure as the implicit Euler NNWR algorithm described previously and sketched in figure 3. The main difference is that the whole procedure is repeated twice, once for each stage.

Algorithm 5 and 6 summarize the discrete Dirichlet solver in (26) and the discrete Neumann solver in (28) respectively. Furthermore, the complete SDIRK2-NNWR algorithm is summarized in algorithm 7.

Algorithm 5 Solver for the Dirichlet problems in (26).

```

1: procedure SDIRK2DIRICHLET( $\mathbf{u}_I^{k+1,n}$ ,  $\mathbf{u}_{\Gamma}^{k,n}$ ,  $\mathbf{u}_{\Gamma}^{k,n+1}$ )
2:    $\mathbf{u}_{\Gamma}^{k,n+a} \leftarrow \mathbf{u}_{\Gamma}^{k,n} + a \left( \mathbf{u}_{\Gamma}^{k,n+1} - \mathbf{u}_{\Gamma}^{k,n} \right)$ 
3:    $\dot{\mathbf{u}}_{\Gamma}^{k,n+a}$ ,  $\dot{\mathbf{u}}_{\Gamma}^{k,n+1} \leftarrow \left( \mathbf{u}_{\Gamma}^{k,n+1} - \mathbf{u}_{\Gamma}^{k,n} \right) / \Delta t$ 
4:   for  $j = 1, 2$  do # loop over stages
5:      $\mathbf{s}_j \leftarrow \mathbf{u}_I^{k+1,n} + \Delta t \sum_{l=1}^{j-1} (1-a) \mathbf{k}_l$ 
6:      $\mathbf{U}_j \leftarrow \text{SOLVELINEARSYSTEM}(\mathbf{s}_j, \dot{\mathbf{u}}_{\Gamma}^{k,n+j-1+(2-j)a}, \mathbf{u}_{\Gamma}^{k,n+j-1+(2-j)a}),$  # solve 1st eq in (26)
7:      $\mathbf{k}_j \leftarrow \frac{1}{a\Delta t} (\mathbf{U}_j - \mathbf{s}_j)$ 
8:      $\mathbf{u}_I^{k+1,n+1} \leftarrow \mathbf{U}_2$ 
9:     for  $j = 1, 2$  do # compute fluxes in (27)
10:       $\mathbf{f}_j^{k,n} \leftarrow \mathbf{M}_{\Gamma\Gamma} \dot{\mathbf{u}}_{\Gamma}^{k,n+j-1+(2-j)a} + \mathbf{M}_{\Gamma I} \mathbf{k}_j + \mathbf{A}_{\Gamma\Gamma} \mathbf{u}_{\Gamma}^{k,n+j-1+(2-j)a} + \mathbf{A}_{\Gamma I} \mathbf{U}_j$ 
   return  $\mathbf{u}_I^{k+1,n+1}$ ,  $\mathbf{f}_j^{k,n}$ 

```

Algorithm 6 Solver for the Neumann problems in (28).

```

1: procedure SDIRK2NEUMANN( $\psi_I^{k+1,n}$ ,  $\psi_{\Gamma}^{k+1,n}$ ,  $\mathbf{F}_j^{k,n}$ )
2:   for  $j = 1, 2$  do # loop over stages
3:      $\mathbf{s}_j \leftarrow \psi_I^{k+1,n} + \Delta t \sum_{l=1}^{j-1} (1-a) \mathbf{k}_l$ 
4:      $\mathbf{Y}_j \leftarrow \text{SOLVELINEARSYSTEM}(\mathbf{s}_j, \psi_I^{k+1,n}, \psi_{\Gamma}^{k+1,n}, \mathbf{F}_j^{k,n}),$  # solve 1st eq in (28)
5:      $\mathbf{k}_j \leftarrow \frac{1}{a\Delta t} (\mathbf{Y}_j - \mathbf{s}_j)$ 
6:      $\psi_I^{k+1,n+1}$ ,  $\psi_{\Gamma}^{k+1,n+1} \leftarrow \mathbf{Y}_2$ 
   return  $\psi_I^{k+1,n+1}$ ,  $\psi_{\Gamma}^{k+1,n+1}$ 

```

8 Derivation of the iteration matrix

We are interested in the performance of the NNWR algorithm. As the rate of convergence of a linear iteration is given by the spectral radius of its iteration matrix, we derive in this section the iteration matrix with respect to the set of unknowns at the space-time interface for implicit Euler. A similar analysis to find the convergence rates of the Dirichlet-Neumann iteration for the unsteady transmission problem can be found in [26]. We intentionally avoid a derivation for SDIRK2 and we will show in the numerical results section that NNWR-SDIRK2 behaves as predicted by the analysis of implicit Euler. From now on we assume that we have conforming time grids, i.e., $\Delta t := \Delta t_1 = \Delta t_2$. We will see later in how far the analysis performed for the nonmultirate case is applicable to the multirate case.

The goal now is to find the iteration matrix Σ with respect to the final synchronization point $\mathbf{u}_{\Gamma}^{N_m} \approx \mathbf{u}_{\Gamma}(T_f)$ because the global error over the time window $[T_0, T_f]$ is assumed to be increasing, having its maximum at the final time T_f . Thus, we will find Σ such that

Algorithm 7 NNWR algorithm using SDIRK2.

```

1: procedure NNWR2( $\tau_1, \tau_2, \alpha_1, \alpha_2, \lambda_1, \lambda_2, \Theta, TOL$ )
2:    $\mathbf{u}_I^{(m),k+1,0}, \mathbf{u}_I^0(\tau_m), \psi_I^{(m),k+1,n_m+1}, \psi_\Gamma^{(m),k+1,n_m+1} \leftarrow$  INITIALIZATION
3:   while  $\|\mathbf{u}_\Gamma^{k+1,N_m} - \mathbf{u}_\Gamma^{k,N_m}\| \leq TOL$  do
4:     for  $j = 1, 2$  do # loop over stages
5:       for  $t_{n_1} \in \tau_1$  do
6:          $\mathbf{u}_I^{(1),k+1,n_1+1}, \mathbf{f}_j^{(1),k,n_1} \leftarrow$  SDIRK2DIRICHLET( $\mathbf{u}_I^{(1),k+1,n_1}, \mathbf{u}_\Gamma^{k,n_1}, \mathbf{u}_\Gamma^{k,n_1+1}$ )
7:       for  $t_{n_2} \in \tau_2$  do (in parallel to 5)
8:          $\mathbf{u}_I^{(2),k+1,n_2+1}, \mathbf{f}_j^{(2),k,n_2} \leftarrow$  SDIRK2DIRICHLET( $\mathbf{u}_I^{(2),k+1,n_2}, \mathbf{u}_\Gamma^{k,n_2}, \mathbf{u}_\Gamma^{k,n_2+1}$ )
9:          $\mathbf{F}_j^{(1),k,\tau_1} \leftarrow \mathbf{f}_j^{(1),k,\tau_1} + \mathbf{I}(\tau_1, \tau_2, \mathbf{f}_j^{(2),k,\tau_2})$ 
10:         $\mathbf{F}_j^{(2),k,\tau_2} \leftarrow \mathbf{f}_j^{(2),k,\tau_2} + \mathbf{I}(\tau_2, \tau_1, \mathbf{f}_j^{(1),k,\tau_1})$  (in parallel to 9)
11:        for  $t_{n_1} \in \tau_1$  do
12:           $\psi_I^{k+1,n_1+1}, \psi_\Gamma^{k+1,n_1+1} \leftarrow$  SDIRK2NEUMANN( $\psi_I^{k+1,n_1}, \psi_\Gamma^{k+1,n_1}, \mathbf{F}_j^{(1),k,n_1}$ )
13:        for  $t_{n_2} \in \tau_2$  do (in parallel to 11)
14:           $\psi_I^{k+1,n_2+1}, \psi_\Gamma^{k+1,n_2+1} \leftarrow$  SDIRK2NEUMANN( $\psi_I^{k+1,n_2}, \psi_\Gamma^{k+1,n_2}, \mathbf{F}_j^{(2),k,n_2}$ )
15:         $\mathbf{u}_\Gamma^{k+1,\tau_1} \leftarrow \mathbf{u}_\Gamma^{k,\tau_1} - \Theta \left( \psi_\Gamma^{(1),k+1,\tau_1} + \mathbf{I}(\tau_1, \tau_2, \psi_\Gamma^{(2),k+1,\tau_2}) \right)$ 
16:         $\mathbf{u}_\Gamma^{k+1,\tau_2} \leftarrow \mathbf{u}_\Gamma^{k,\tau_2} - \Theta \left( \psi_\Gamma^{(2),k+1,\tau_2} + \mathbf{I}(\tau_2, \tau_1, \psi_\Gamma^{(1),k+1,\tau_1}) \right)$  (p. to 15)

```

$$\mathbf{u}_\Gamma^{k+1,N_m} = \Sigma \mathbf{u}_\Gamma^{k,N_m} + \sum_{i=1}^2 (\varphi^{k+1,\tilde{\tau}_i} + \varphi^{k,\tau_i}), \quad (33)$$

where φ^{k,τ_m} are terms dependent on solutions at the previous fixed point iteration k for the time grids $\tau_m = \{t_1, t_2, \dots, t_{N_m}\}$, $m = 1, 2$ and $\varphi^{k+1,\tilde{\tau}_m}$ are terms dependent on solutions at the current iteration $k+1$ but for the time grids $\tilde{\tau}_m = \{t_1, t_2, \dots, t_{N_{m-1}}\} \subset \tau_m$, $m = 1, 2$. To perform the analysis, we neglect all the solutions at previous time steps (indicated by $\varphi^{k+1,\tilde{\tau}_m}$). Thus, we do not to find the exact rate of convergence when having more than one single time step, but instead a good estimate.

We now rewrite (18), (20) and (21)-(22) as an iteration for $\mathbf{u}_\Gamma^{k+1,N_m}$. As we chose above, we omit all the terms in (33) except for the first two. We isolate the term $\mathbf{u}_I^{(m),k+1,N_m}$ from (18) and $\psi_I^{(m),k+1,N_m}$ from the first equation in (20) leading to

$$\mathbf{u}_I^{(m),k+1,N_m} = - \left(\frac{\mathbf{M}_{II}^{(m)}}{\Delta t} + \mathbf{A}_{II}^{(m)} \right)^{-1} \left(\frac{\mathbf{M}_{I\Gamma}^{(m)}}{\Delta t} + \mathbf{A}_{I\Gamma}^{(m)} \right) \mathbf{u}_\Gamma^{k,N_m}, \quad (34)$$

$$\psi_I^{(m),k+1,N_m} = - \left(\frac{\mathbf{M}_{II}^{(m)}}{\Delta t} + \mathbf{A}_{II}^{(m)} \right)^{-1} \left(\frac{\mathbf{M}_{I\Gamma}^{(m)}}{\Delta t} + \mathbf{A}_{I\Gamma}^{(m)} \right) \psi_\Gamma^{(m),k+1,N_m}. \quad (35)$$

Inserting (34) and (35) into the second equation of (20) we get

$$\psi_\Gamma^{(m),k+1,N_m} = \mathbf{S}^{(m)-1} \sum_{i=1}^2 \mathbf{S}^{(i)} \mathbf{u}_\Gamma^{k,N_m}, \quad (36)$$

with

$$\mathbf{S}^{(m)} := \left(\frac{\mathbf{M}_{I\Gamma}^{(m)}}{\Delta t} + \mathbf{A}_{I\Gamma}^{(m)} \right) - \left(\frac{\mathbf{M}_{II}^{(m)}}{\Delta t} + \mathbf{A}_{II}^{(m)} \right) \left(\frac{\mathbf{M}_{I\Gamma}^{(m)}}{\Delta t} + \mathbf{A}_{I\Gamma}^{(m)} \right)^{-1} \left(\frac{\mathbf{M}_{I\Gamma}^{(m)}}{\Delta t} + \mathbf{A}_{I\Gamma}^{(m)} \right). \quad (37)$$

Finally, inserting (36) into (21) or (22) one gets $\mathbf{u}_\Gamma^{k+1, N_m} = \Sigma \mathbf{u}_\Gamma^{k, N_m}$ with

$$\Sigma = \mathbf{I} - \Theta \left(2\mathbf{I} + \mathbf{S}^{(1)-1} \mathbf{S}^{(2)} + \mathbf{S}^{(2)-1} \mathbf{S}^{(1)} \right). \quad (38)$$

In the one-dimensional case, the iteration matrix Σ is just a real number and thus its spectral radius is its modulus. Then, the optimal relaxation parameter Θ_{opt} in 1D is given by

$$\Theta_{opt} = \frac{1}{2 + \mathbf{S}^{(1)-1} \mathbf{S}^{(2)} + \mathbf{S}^{(2)-1} \mathbf{S}^{(1)}}. \quad (39)$$

9 One-dimensional convergence analysis

So far, the derivation was performed for a rather general discretization. In this section, we study the iteration matrix Σ for a specific FE discretization in 1D. We will give a formula for the convergence rates. The behaviour of the rates when approaching both the continuous case in time and space is also given.

Specifically, we use $\Omega_1 = [-1, 0]$, $\Omega_2 = [0, 1]$. For the FE discretization, we use the standard piecewise-linear polynomials as test functions. Here we discretize Ω_m into $N+1$ equal sized cells of size $\Delta x = 1/(N+1)$ for $m = 1, 2$.

With $\mathbf{e}_j = (0 \ \cdots \ 0 \ 1 \ 0 \ \cdots \ 0)^T \in \mathbb{R}^N$ where the only nonzero entry is located at the j -th position, the discretization matrices are given by

$$\mathbf{A}_{II}^{(m)} = \frac{\lambda_m}{\Delta x^2} \begin{pmatrix} 2 & -1 & & 0 \\ -1 & 2 & \ddots & \\ & \ddots & \ddots & -1 \\ 0 & & -1 & 2 \end{pmatrix}, \quad \mathbf{M}_{II}^{(m)} = \frac{\alpha_m}{6} \begin{pmatrix} 4 & 1 & & 0 \\ 1 & 4 & \ddots & \\ & \ddots & \ddots & 1 \\ 0 & & 1 & 4 \end{pmatrix},$$

$$\begin{aligned} \mathbf{A}_{I\Gamma}^{(m)} &= \frac{\lambda_m}{\Delta x^2}, \quad \mathbf{A}_{I\Gamma}^{(1)} = -\frac{\lambda_1}{\Delta x^2} \mathbf{e}_N, \quad \mathbf{A}_{I\Gamma}^{(2)} = -\frac{\lambda_2}{\Delta x^2} \mathbf{e}_1, \quad \mathbf{A}_{\Gamma I}^{(1)} = -\frac{\lambda_1}{\Delta x^2} \mathbf{e}_N^T, \quad \mathbf{A}_{\Gamma I}^{(2)} = -\frac{\lambda_2}{\Delta x^2} \mathbf{e}_1^T, \\ \mathbf{M}_{I\Gamma}^{(m)} &= \frac{2\alpha_m}{6}, \quad \mathbf{M}_{I\Gamma}^{(1)} = \frac{\alpha_1}{6} \mathbf{e}_N, \quad \mathbf{M}_{I\Gamma}^{(2)} = \frac{\alpha_2}{6} \mathbf{e}_1, \quad \mathbf{M}_{\Gamma I}^{(1)} = \frac{\alpha_1}{6} \mathbf{e}_N^T, \quad \mathbf{M}_{\Gamma I}^{(2)} = \frac{\alpha_2}{6} \mathbf{e}_1^T. \end{aligned}$$

where $\mathbf{A}_{II}^{(m)}, \mathbf{M}_{II}^{(m)} \in \mathbb{R}^{N \times N}$, $\mathbf{A}_{I\Gamma}^{(m)}, \mathbf{M}_{I\Gamma}^{(m)} \in \mathbb{R}^{N \times 1}$ and $\mathbf{A}_{\Gamma I}^{(m)}, \mathbf{M}_{\Gamma I}^{(m)} \in \mathbb{R}^{1 \times N}$ for $m = 1, 2$.

One computes $\mathbf{S}^{(m)}$ for $m = 1, 2$, by inserting the corresponding matrices specified above in (37) obtaining

$$\begin{aligned} \mathbf{S}^{(1)} &= \left(\frac{\alpha_1}{3\Delta t} + \frac{\lambda_1}{\Delta x^2} \right) - \left(\frac{\alpha_1}{6\Delta t} - \frac{\lambda_1}{\Delta x^2} \right)^2 \mathbf{e}_N^T \left(\frac{\mathbf{M}_{II}^{(1)}}{\Delta t} + \mathbf{A}_{II}^{(1)} \right)^{-1} \mathbf{e}_N \\ &= \left(\frac{\alpha_1}{3\Delta t} + \frac{\lambda_1}{\Delta x^2} \right) - \left(\frac{\alpha_1}{6\Delta t} - \frac{\lambda_1}{\Delta x^2} \right)^2 \alpha_{1N}^1, \end{aligned} \quad (40)$$

$$\begin{aligned} \mathbf{S}^{(2)} &= \left(\frac{\alpha_2}{3\Delta t} + \frac{\lambda_2}{\Delta x^2} \right) - \left(\frac{\alpha_2}{6\Delta t} - \frac{\lambda_2}{\Delta x^2} \right)^2 \mathbf{e}_1^T \left(\frac{\mathbf{M}_{II}^{(2)}}{\Delta t} + \mathbf{A}_{II}^{(2)} \right)^{-1} \mathbf{e}_1 \\ &= \left(\frac{\alpha_2}{3\Delta t} + \frac{\lambda_2}{\Delta x^2} \right) - \left(\frac{\alpha_2}{6\Delta t} - \frac{\lambda_2}{\Delta x^2} \right)^2 \alpha_{11}^2, \end{aligned} \quad (41)$$

where α_{ij}^m represent the entries of the matrices $\left(\frac{\mathbf{M}_{II}^{(m)}}{\Delta t} + \mathbf{A}_{II}^{(m)} \right)^{-1}$ for $i, j = 1, \dots, N$, $m = 1, 2$. Observe that the matrices $\left(\frac{\mathbf{M}_{II}^{(1)}}{\Delta t} + \mathbf{A}_{II}^{(1)} \right)$ and $\left(\frac{\mathbf{M}_{II}^{(2)}}{\Delta t} + \mathbf{A}_{II}^{(2)} \right)$ are tridiagonal Toeplitz matrices but their inverses are full matrices. The computation of the exact inverses could be performed based on the recursive formula presented in [9] which runs over the entries of the matrix and consequently, it is non trivial to compute α_{1N}^1 and α_{11}^2 this way.

Due to these difficulties, we rewrite them in terms of their eigendecomposition:

$$\left(\frac{\mathbf{M}_{II}^{(m)}}{\Delta t} + \mathbf{A}_{II}^{(m)} \right)^{-1} = \left[\text{tridiag} \left(\frac{\alpha_m \Delta x^2 - 6\lambda_m \Delta t}{6\Delta t \Delta x^2}, \frac{2\alpha_m \Delta x^2 + 6\lambda_m \Delta t}{3\Delta t \Delta x^2}, \frac{\alpha_m \Delta x^2 - 6\lambda_m \Delta t}{6\Delta t \Delta x^2} \right) \right]^{-1} = \mathbf{V} \Lambda_m^{-1} \mathbf{V},$$

for $m = 1, 2$ where the matrix \mathbf{V} has the eigenvectors of any symmetric tridiagonal Toeplitz matrix of dimension N as columns. The entries of \mathbf{V} are not dependent on the entries of $\frac{\mathbf{M}_{II}^{(m)}}{\Delta t} + \mathbf{A}_{II}^{(m)}$ due to their symmetry. Moreover, the matrices Λ_m are diagonal matrices having the eigenvalues of $\frac{\mathbf{M}_{II}^{(m)}}{\Delta t} + \mathbf{A}_{II}^{(m)}$ as entries. These are known and given e.g. in [25, pp. 514-516]:

$$v_{ij} = \frac{1}{\sqrt{\sum_{k=1}^N \sin^2 \left(\frac{k\pi}{N+1} \right)}} \sin \left(\frac{ij\pi}{N+1} \right),$$

$$\lambda_j^m = \frac{1}{3\Delta t \Delta x^2} \left(2\alpha_m \Delta x^2 + 6\lambda_m \Delta t + (\alpha_m \Delta x^2 - 6\lambda_m \Delta t) \cos \left(\frac{j\pi}{N+1} \right) \right),$$

for $i, j = 1, \dots, N$ and $m = 1, 2$.

The entries α_{NN}^1 and α_{11}^2 of the matrices $\left(\frac{\mathbf{M}_{II}^{(1)}}{\Delta t} + \mathbf{A}_{II}^{(1)} \right)^{-1}$ and $\left(\frac{\mathbf{M}_{II}^{(2)}}{\Delta t} + \mathbf{A}_{II}^{(2)} \right)^{-1}$, respectively, are now computed through their eigendecomposition resulting in

$$\alpha_{NN}^1 = \frac{\sum_{i=1}^N \frac{1}{\lambda_i^1} \sin^2 \left(\frac{i\pi N}{N+1} \right)}{\sum_{i=1}^N \sin^2 \left(\frac{i\pi}{N+1} \right)} = \frac{s_1}{\sum_{i=1}^N \sin^2(i\pi \Delta x)}, \quad (42)$$

$$\alpha_{11}^2 = \frac{\sum_{i=1}^N \frac{1}{\lambda_i^2} \sin^2 \left(\frac{i\pi}{N+1} \right)}{\sum_{i=1}^N \sin^2 \left(\frac{i\pi}{N+1} \right)} = \frac{s_2}{\sum_{i=1}^N \sin^2(i\pi \Delta x)}, \quad (43)$$

with

$$s_m = \sum_{i=1}^N \frac{3\Delta t \Delta x^2 \sin^2(i\pi \Delta x)}{2\alpha_m \Delta x^2 + 6\lambda_m \Delta t + (\alpha_m \Delta x^2 - 6\lambda_m \Delta t) \cos(i\pi \Delta x)}, \quad (44)$$

for $m = 1, 2$.

To simplify this, the finite sum $\sum_{i=1}^N \sin^2(i\pi \Delta x)$ can be computed. We first rewrite the sum of squared sinus terms into a sum of cosinus terms using the identity $\sin^2(x/2) = (1 - \cos(x))/2$. Then, the resulting sum can be converted into a geometric sum using Euler's formula:

$$\sum_{i=1}^N \sin^2(i\pi \Delta x) = \frac{1 - \Delta x}{2\Delta x} - \frac{1}{2} \sum_{i=1}^N \cos(2i\pi \Delta x) = \frac{1}{2\Delta x}. \quad (45)$$

Now, inserting (45) into (42) and (43) and these two into (40) and (41) we get for $\mathbf{S}^{(m)}$ for $m = 1, 2$,

$$\mathbf{S}^{(m)} = \frac{6\Delta t \Delta x (\alpha_m \Delta x^2 + 3\lambda_m \Delta t) - (\alpha_m \Delta x^2 - 6\lambda_m \Delta t)^2 s_m}{18\Delta t^2 \Delta x^3}, \quad (46)$$

With this we obtain an explicit formula for the optimal relaxation parameter Θ_{opt} in (39):

$$\Theta_{opt} = \left(2 + \frac{6\Delta t \Delta x (\alpha_2 \Delta x^2 + 3\lambda_2 \Delta t) - (\alpha_2 \Delta x^2 - 6\lambda_2 \Delta t)^2 s_2}{6\Delta t \Delta x (\alpha_1 \Delta x^2 + 3\lambda_1 \Delta t) - (\alpha_1 \Delta x^2 - 6\lambda_1 \Delta t)^2 s_1} + \frac{6\Delta t \Delta x (\alpha_1 \Delta x^2 + 3\lambda_1 \Delta t) - (\alpha_1 \Delta x^2 - 6\lambda_1 \Delta t)^2 s_1}{6\Delta t \Delta x (\alpha_2 \Delta x^2 + 3\lambda_2 \Delta t) - (\alpha_2 \Delta x^2 - 6\lambda_2 \Delta t)^2 s_2} \right)^{-1}. \quad (47)$$

We could not find a way of simplifying the finite sum (44) because Δx depends on N (i.e., $\Delta x = 1/(N+1)$). However, (47) is a computable expression that gives the optimal relaxation parameter Θ_{opt} of the NNWR algorithm using implicit Euler for given Δx , Δt , α_m and λ_m , $m = 1, 2$.

We are now interested in the asymptotics of (47) with respect to both spatial and temporal resolutions. To this end, we rewrite (47) in terms of $c := \Delta t/\Delta x^2$:

$$\Theta_{opt} = \left(2 + \frac{6\Delta t(\alpha_2 + 3\lambda_2 c) - \Delta x(\alpha_2 - 6\lambda_2 c)^2 s'_2}{6\Delta t(\alpha_1 + 3\lambda_1 c) - \Delta x(\alpha_1 - 6\lambda_1 c)^2 s'_1} + \frac{6\Delta t(\alpha_1 + 3\lambda_1 c) - \Delta x(\alpha_1 - 6\lambda_1 c)^2 s'_1}{6\Delta t(\alpha_2 + 3\lambda_2 c) - \Delta x(\alpha_2 - 6\lambda_2 c)^2 s'_2} \right)^{-1}, \quad (48)$$

where

$$s'_m = \sum_{i=1}^N \frac{3\Delta t \sin^2(i\pi\Delta x)}{2\alpha_m + 6\lambda_m c + (\alpha_m - 6\lambda_m c) \cos(i\pi\Delta x)}, \quad (49)$$

for $m = 1, 2$.

Finally, we compute the limits $c \rightarrow 0$ and $c \rightarrow \infty$ of the iteration matrix Σ :

$$\begin{aligned} \lim_{c \rightarrow 0} \Sigma &= \lim_{c \rightarrow 0} \left(1 - \Theta \left(2 + \frac{6\alpha_2 \Delta t - \alpha_2 \Delta t \Delta x \sum_{i=1}^N \frac{3 \sin(i\pi\Delta x)^2}{2 + \cos(i\pi\Delta x)}}{6\alpha_1 \Delta t - \alpha_1 \Delta t \Delta x \sum_{i=1}^N \frac{3 \sin(i\pi\Delta x)^2}{2 + \cos(i\pi\Delta x)}} + \frac{6\alpha_1 \Delta t - \alpha_1 \Delta t \Delta x \sum_{i=1}^N \frac{3 \sin(i\pi\Delta x)^2}{2 + \cos(i\pi\Delta x)}}{6\alpha_2 \Delta t - \alpha_2 \Delta t \Delta x \sum_{i=1}^N \frac{3 \sin(i\pi\Delta x)^2}{2 + \cos(i\pi\Delta x)}} \right) \right) \\ &= 1 - \Theta \left(2 + \frac{\alpha_2}{\alpha_1} + \frac{\alpha_1}{\alpha_2} \right) = 1 - \Theta \left(\frac{(\alpha_1 + \alpha_2)^2}{\alpha_1 \alpha_2} \right), \end{aligned}$$

and consequently

$$\Theta_{\{c_1 \rightarrow 0\}} = \frac{\alpha_1 \alpha_2}{(\alpha_1 + \alpha_2)^2}. \quad (50)$$

$$\begin{aligned} \lim_{c \rightarrow \infty} \Sigma &= \lim_{c \rightarrow \infty} \left(1 - \Theta \left(2 + \frac{18\lambda_2 \Delta t c - 18\lambda_2 \Delta t \Delta x c \sum_{i=1}^N \frac{\sin(i\pi\Delta x)^2}{1 - \cos(i\pi\Delta x)}}{18\lambda_1 \Delta t c - 18\lambda_1 \Delta t \Delta x c \sum_{i=1}^N \frac{\sin(i\pi\Delta x)^2}{1 - \cos(i\pi\Delta x)}} + \frac{18\lambda_1 \Delta t c - 18\lambda_1 \Delta t \Delta x c \sum_{i=1}^N \frac{\sin(i\pi\Delta x)^2}{1 - \cos(i\pi\Delta x)}}{18\lambda_2 \Delta t c - 18\lambda_2 \Delta t \Delta x c \sum_{i=1}^N \frac{\sin(i\pi\Delta x)^2}{1 - \cos(i\pi\Delta x)}} \right) \right) \\ &= 1 - \Theta \left(2 + \frac{\lambda_2}{\lambda_1} + \frac{\lambda_1}{\lambda_2} \right) = 1 - \Theta \left(\frac{(\lambda_1 + \lambda_2)^2}{\lambda_1 \lambda_2} \right), \end{aligned}$$

and consequently

$$\Theta_{\{c_1 \rightarrow \infty\}} = \frac{\lambda_1 \lambda_2}{(\lambda_1 + \lambda_2)^2}. \quad (51)$$

The result obtained in (51) is consistent with the one-dimensional semidiscrete analysis performed in [21]. There, a convergence analysis for the NNWR method in (6), (7) and (8) with constant coefficients using Laplace transforms shows that $\Theta_{opt} = 1/4$ when the two subdomains Ω_1 and Ω_2 are identical. Their result is recovered by our analysis when one approaches the continuous case in space in (51) for constant coefficients, i.e., $\lambda_1 = \lambda_2$. In that case, one gets

$$\Theta_{opt} = \frac{\lambda_1 \lambda_2}{(\lambda_1 + \lambda_2)^2} = \frac{\lambda_1^2}{4\lambda_1^2} = \frac{1}{4}. \quad (52)$$

10 Numerical results

We now present numerical experiments to illustrate the validity of the theoretical results of the previous sections. All the results in this section have been produced by implementing algorithms 4 and 7 in Python using the FE discretization specified in the previous section. Firstly, we will show numerically that the NNWR algorithm using implicit Euler preserves first order and using SDIRK2 second order. Secondly, we will show the validity of (47) as estimator for the optimal relaxation parameter Θ_{opt} of the NNWR algorithm using implicit Euler. We will also show that (47) is a good estimator for the multirate case both using implicit Euler and SDIRK2 and also for 2D examples. Furthermore, we will also show that the theoretical asymptotics deduced both in (50) and (51) match with the numerical experiments. Finally, a comparison between the Dirichlet-Neumann and the Neuman-Neumann methods is included.

10.1 NNWR results

Figure 4 shows the error plots of the NNWR algorithm for the coupling of different materials using both implicit Euler and SDIRK2. Physical properties of the materials are shown in table 1. We have considered two initial time grids (for $\Delta t_c = 1/10$ and $\Delta t_f = 1/100$ given), which we then refine several times by a factor of 2:

- (C-C): Two coarse conforming time grids with $\Delta t_1 = \Delta t_2 = \Delta t_c$.
- (C-F): Nonconforming time grids with $\Delta t_1 = \Delta t_c$ and $\Delta t_2 = \Delta t_f$.

Table 1: Physical properties of the materials. λ is the thermal conductivity, ρ the density, c_p the specific heat capacity and $\alpha = \rho c_p$.

Material	λ (W/mK)	ρ (kg/m ³)	c_p (J/kgK)	α (J/K m ³)
Air	0.0243	1.293	1005	1299.5
Water	0.58	999.7	4192.1	4.1908e6
Steel	48.9	7836	443	3471348

When coupling two different materials in the multirate case, we always assign the finer grid to the material that has higher heat conductivity because it performs the heat changes faster. In space, we fix $\Delta x = 1/200$ and we compute a reference solution by solving problem (1) directly on a very fine time grid, with $\Delta t = 1/1000$. One observes in figure 4 that first and second order convergence is obtained in the nonconforming case for implicit Euler and SDIRK2 respectively. Moreover, the errors obtained in the multirate case (C-F) are nearly the same as in the coarser nonmultirate case (C-C). Thus, the accuracy of the multirate case is determined by its coarser rate. This is consistent with [31, 8] where the convergence of the discrete multirate WR algorithm is independent of the ratio of timesteps.

Figure 5 compares the behavior of the algorithm described in this paper using implicit Euler (left plot) and SDIRK2 (right plot). It shows the convergence rates in terms of the relaxation parameter Θ for the one-dimensional thermal coupling between air and water. We have plotted $\Sigma(\Theta)$ in (38) with the 1D space discretization specified in section 9 and the experimental convergence rates for both the multirate and nonmultirate cases. The relevance of the analysis presented above is illustrated in figure 5 because the algorithm is extremely fast at Θ_{opt} (converging in 2 iterations), but if one deviates slightly from Θ_{opt} , we get a divergent method. As can be seen in the left plot, the experimental convergence rates for the nonmultirate case (C-C) are exactly predicted by the theory. Moreover, our formula also predicts where the convergence rate of the NNWR algorithm in the multirate case (C-F) is minimal. They are not identical because the linear interpolation performed at the space-time interface in the multirate case is neglected in (33) to simplify the theoretical analysis. One can also observe in the right plot that $\Sigma(\Theta)$ using implicit Euler estimates quite well the optimal relaxation parameter of the NNWR algorithm using SDIRK2 for both the multirate and nonmultirate cases.

In order to illustrate the behavior of the NNWR method in the multirate case ($\Delta t_1 \neq \Delta t_2$), we have plotted in figure 6 the convergence rates using the relaxation parameters $\Theta_{opt}(\Delta t_1)$ and $\Theta_{opt}(\Delta t_2)$ in (47) with respect to the variation of $\Delta t_1/\Delta t_2$ for the air-water coupling. In 6 we have chosen $\Delta t_1/\Delta t_2 = 1e - 3/2e - 1, 2e - 3/2e - 1, 1e - 2/2e - 1, 2e - 2/2e - 1, 5e - 2/2e - 1, 1e - 1/2e - 1, 2e - 1/2e - 1, 2e - 1/1e - 1, 2e - 1/5e - 2, 2e - 1/2e - 2, 2e - 1/1e - 2, 2e - 1/2e - 3, 2e - 1/1e - 3$ and $\Delta x = 1/100$. One observes that the multirate method converges fast using

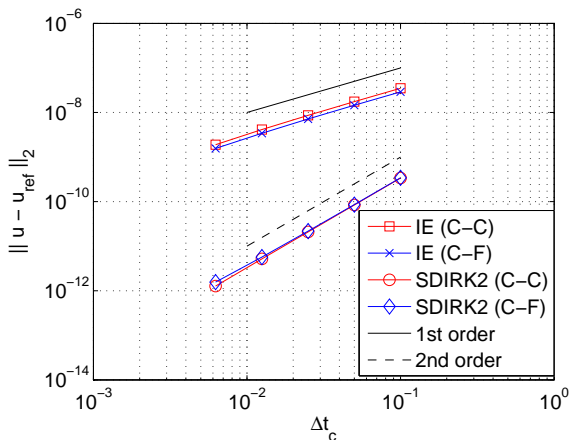


Figure 4: Error plot of the NNWR algorithm for three different time grids. $\Delta x = 1/200$, $[T_0, T_f] = [0, 1]$ and $TOL = 1e-15$. Air-steel thermal interaction using implicit Euler and air-water thermal interaction using SDIRK2.

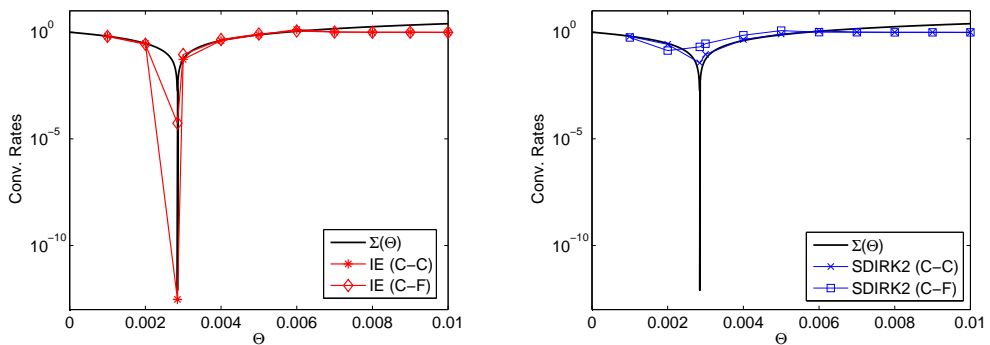


Figure 5: Air-water convergence rates as a function of the relaxation parameter Θ in 1D. $\Delta x = 1/100$, $\Delta t_e = 100$ and $\Delta t_f = 1$. *Left*: $\Sigma(\Theta)$ in (38) and the experimental convergence rates both for the multirate (C-F) and nonmultirate (C-C) cases using implicit Euler. *Right*: $\Sigma(\Theta)$ in (38) using implicit Euler and the experimental convergence rates both for the multirate (C-F) and nonmultirate (C-C) cases using SDIRK2.

any of the two relaxation parameters for both implicit Euler and SDIRK2. Nevertheless, one can also see in figure 6 that even though we have not performed a specific analysis for the optimal relaxation parameter in the multirate case, the Θ_{opt} in (47) can be used as an estimator. More specifically, we conclude from figure 6 that one can use $\Theta_{opt}(\Delta t_2)$ when $\Delta t_1 < \Delta t_2$ and $\Theta_{opt}(\Delta t_1)$ when $\Delta t_1 > \Delta t_2$.

Figure 7 shows the optimal relaxation parameter Θ_{opt} with respect to the parameter $c := \Delta t/\Delta x^2$ using both implicit Euler and SDIRK2. We have chosen $\Delta t = 1e-9, 1e-8, \dots, 1e8, 1e9$ and $\Delta x = 1/100$. For implicit Euler, we have plotted the function $\Theta_{opt}(c)$ in (48). For SDIRK2, we have plotted a sister function $\Theta_{opt}(c)$ that can be found applying exactly the derivation presented in sections 8 and 9 to the discretized SDIRK2-NNWR method introduced in section 7.2. One can see that the two time discretization methods have a similar behavior when varying c . This illustrates why the optimal relaxation parameter Θ_{opt} computed in (47) for implicit Euler is also valid for SDIRK2 as observed in figure 5. Furthermore, in 7 we observe that the optimal relaxation parameter for any given Δt and

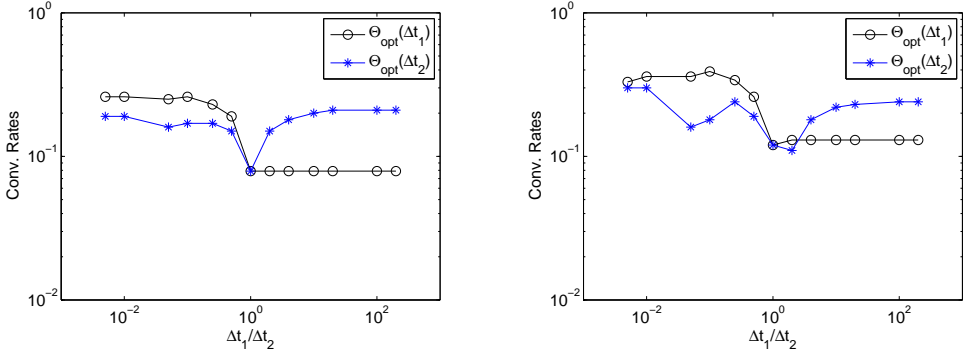


Figure 6: Convergence rates as a function of the temporal ratio $\Delta t_1/\Delta t_2$ for the air-water coupling in 1D. We plot the convergence rates in the multirate case ($\Delta t_1 \neq \Delta t_2$) using the relaxation parameters $\Theta_{opt}(\Delta t_1)$ and $\Theta_{opt}(\Delta t_2)$ in (47). $\Delta t_1/\Delta t_2 = 1e - 3/2e - 1, 2e - 3/2e - 1, 1e - 2/2e - 1, 2e - 2/2e - 1, 5e - 2/2e - 1, 1e - 1/2e - 1, 2e - 1/2e - 1, 2e - 1/1e - 1, 2e - 1/5e - 2, 2e - 1/2e - 2, 2e - 1/1e - 2, 2e - 1/2e - 3, 2e - 1/1e - 3$ and $\Delta x = 1/100$. *Left*: Implicit Euler. *Right*: SDIRK2.

Δx is always between the bounds of the theoretical asymptotics deduced both in (50) and (51), tending to them in the temporal and spatial limits respectively.

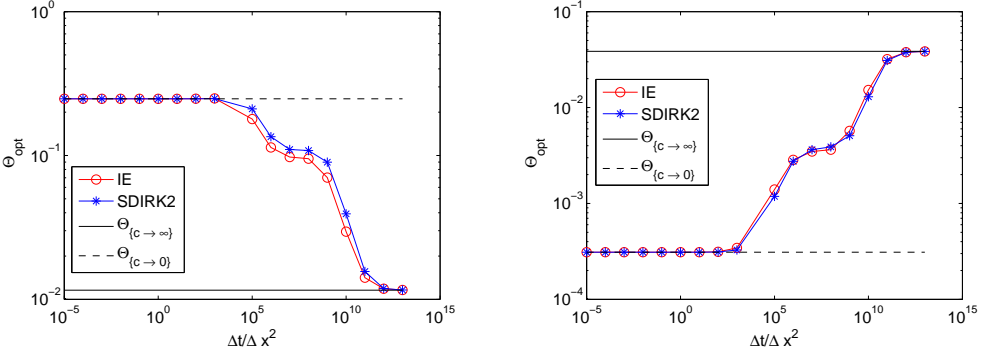


Figure 7: Optimal relaxation parameter Θ_{opt} as a function of the parameter $c := \Delta t/\Delta x^2$ for both implicit Euler and SDIRK2 in 1D. The constant lines $\Theta_{\{c \rightarrow \infty\}}$ and $\Theta_{\{c \rightarrow 0\}}$ represent the spatial and temporal asymptotics of Θ_{opt} in (48). $\Delta t = 1e - 9, 1e - 8, \dots, 1e8, 1e9$ and $\Delta x = 1/100$. *Left*: Water-steel coupling. *Right*: Air-Water coupling.

We now want to demonstrate that the 1D formula (47) is a good estimator for the optimal relaxation parameter Θ_{opt} in 2D. Thus, we now consider a 2D version of the model problem consisting of two coupled linear heat equations on two identical unit squares, e.g. $\Omega_1 = [-1, 0] \times [0, 1]$ and $\Omega_2 = [0, 1] \times [0, 1]$.

Figure 8 shows the convergence rates in terms of the relaxation parameter Θ for 2D examples. On the left we have the thermal coupling between air and steel and on the right between air and water. One can observe that the convergence rates of the NNWR method using Θ_{opt} from (47) in the 2D examples are worse than in 1D, but still optimal. Hence, we suggest to use Θ_{opt} in 2D as well, otherwise the method is divergent.

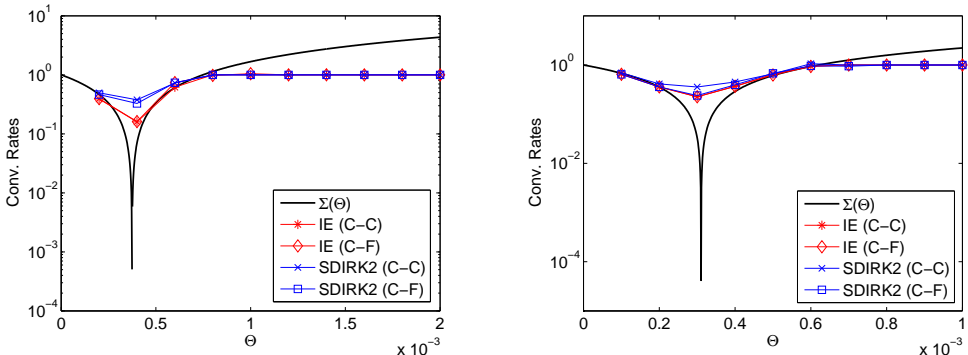


Figure 8: Convergence rates as a function of the relaxation parameter Θ in 2D. $\Sigma(\Theta)$ in (47) using implicit Euler and the experimental convergence rates both for the multirate (C-F) and the nonmultirate (C-C) cases using implicit Euler and SDIRK2. $\Delta x = 1/10$, $\Delta t_c = 1/10$ and $\Delta t_f = 1/50$. *Left*: Air-steel coupling. *Right*: Air-water coupling.

10.2 NNWR - DNWR Comparison

Finally, we compare the Dirichlet-Neumann and the Neumann-Neumann couplings. We consider the FE discretization specified in section 9 and the implicit Euler as a time integration method for both DNWR and NNWR with $\Delta x = 1/500$. In addition, we will use $\Theta = 1/2$ as the optimal relaxation parameter for the DNWR algorithm as suggested in [13] for constant material coefficients. Note that the optimal relaxation parameter for the NNWR method is $\Theta_{opt} = 1/4$ (see (47)) in the case of constant coefficients because $\lambda_1 = \lambda_2$ and $\alpha_1 = \alpha_2$.

Table 2 shows the time needed to solve the 1D steel-steel coupling in the nonmultirate case. The number of fixed point iterations needed to achieve a chosen tolerance of $1e - 8$ is also given. One can see that the DNWR method is slightly more efficient than the NNWR method. Moreover, the NNWR algorithm runs in parallel on two different processors using double the amount of computational power than the DNWR. Thus, the DNWR method is a better option for this case because it is cheaper and faster.

Table 2: Computational effort of DNWR and NNWR for the 1D steel-steel coupling in the nonmultirate case. $\Delta x = 1/500$ and $TOL = 1e - 8$. Number of fixed point iterations in brackets.

Δt	Comp. Time - NNWR (s)	Comp. Time - DNWR (s)
1	0.07 (2 iterations)	0.05 (2 iterations)
1/10	0.42 (2 iterations)	0.3 (2 iterations)
1/50	1.86 (2 iterations)	1.25 (2 iterations)
1/100	3.74 (2 iterations)	2.47 (2 iterations)

However, the NNWR algorithm beats the DNWR algorithm by far when we move to the multirate environment. This is illustrated in table 3 where the computational effort used to solve the 1D steel-steel coupling in the multirate case is shown. There, one can see how the number of fixed point iterations needed to achieve a tolerance of $1e - 8$ using DNWR grows exponentially when the difference between Δt_1 and Δt_2 increases. On the contrary, the NNWR method is very efficient even when there is a huge difference between Δt_1 and Δt_2 . Thus, we recommend the NNWR algorithm when coupling two fields with nonconforming time grids.

Finally, we have included a comparison for the 1D air-steel coupling in the multirate case. This interaction between air and steel has the particularity of strong jumps in the material coefficients across the space interface. In this case, we have chosen $\Theta = 1/2$ for DNWR because even though in [13] it is only proved optimal for the case of constant coefficients, they show in the numerical results section that also applies to an example where the diffusion coefficient varies spatially. Moreover, Θ_{opt} in (47) is chosen for the NNWR method. Table 4 shows a comparison of the computational time employed to solve the 1D air-steel coupling in the multirate case. One can see that

Table 3: Computational effort comparison of DNWR and NNWR for the 1D steel-steel coupling in the multirate case. $\Delta x = 1/500$ and $TOL = 1e - 8$. Number of fixed point iterations in brackets.

$\Delta t_1 - \Delta t_2$	Comp. Time - NNWR (s)	Comp. Time - DNWR (s)
1/5 - 1/10	0.48 (3 iterations)	0.70 (6 iterations)
1/5 - 1/50	1.5 (3 iterations)	26.98 (71 iterations)
1/5 - 1/100	2.74 (3 iterations)	Not convergent

Table 4: Computational effort comparison of DNWR and NNWR for the 1D air-steel coupling in the multirate case. $\Delta x = 1/500$ and $TOL = 1e - 8$. Number of fixed point iterations in brackets.

$\Delta t_1 - \Delta t_2$	Comp. Time - NNWR (s)	Comp. Time - DNWR (s)
1/5 - 1/10	0.47 (3 iterations)	1.24 (12 iterations)
1/5 - 1/50	2.20 (4 iterations)	4.65 (12 iterations)
1/5 - 1/100	3.95 (4 iterations)	8.77 (12 iterations)

the NNWR method is more efficient than the DNWR method because it needs way less fixed point iterations to achieve the same tolerance. Note that the number of iterations increases when the time resolution decreases for the NNWR method. This happens because the analysis performed in section 8 to find the optimal relaxation parameter takes into account only one single time step (see (33)). Therefore, in the case of multiple time steps, Θ_{opt} in (47) is a very good choice, but it is not optimal. Besides that, the large amount of iterations performed by the DNWR algorithm hints that $\Theta = 1/2$ might not be the optimal relaxation parameter when having strong jumps in the material coefficients for the fully discrete problem. Thus, performing an specific analysis to find the optimal relaxation parameter of the DNWR algorithm is left for future research.

11 Summary and conclusions

We suggested a new high order parallel NNWR method with nonconforming time grids for two heterogeneous coupled heat equations and studied the optimal relaxation parameter in terms of the material coefficients and the temporal and spatial resolutions Δt and Δx . To this end, we considered the coupling of two heat equations on two identical domains. We assumed structured spatial grids and conforming time grids on both subdomains to derive a formula for the optimal relaxation parameter Θ_{opt} in 1D using implicit Euler. Furthermore, we determined the limits of the optimal relaxation parameter when approaching the continuous case either in space ($\lambda_1 \lambda_2 / (\lambda_1 + \lambda_2)^2$) or time ($\alpha_1 \alpha_2 / (\alpha_1 + \alpha_2)^2$). The method using Θ_{opt} converges extremely fast, typically within two iterations. This was confirmed by numerical results, where we also demonstrated that the nonmultirate 1D case gives excellent estimates for the multirate 1D case and even for multirate and nonmultirate 2D examples using both implicit Euler and SDIRK2. In addition, we have shown that the NNWR method is a more efficient choice than the classical DNWR in the multirate case.

References

- [1] A. BANKA, *Practical Applications of CFD in heat processing*, Heat Treating Progress., (2005).
- [2] P. BIRKEN, T. GLEIM, D. KUHL, AND A. MEISTER, *Fast Solvers for Unsteady Thermal Fluid Structure Interaction*, Int. J. Numer. Meth. Fluids, 79(1) (2015), pp. 16–29.
- [3] P. BIRKEN, K. QUINT, S. HARTMANN, AND A. MEISTER, *A time-adaptive fluid-structure interaction method for thermal coupling*, Comp. Vis. in Science, 13(7) (2011), pp. 331–340.
- [4] S. BREMICKER-TRÜBELHORN AND S. ORTLEB, *On Multirate GARK Schemes with Adaptive Micro Step Sizes for Fluid-Structure Interaction: Order Conditions and Preservation of the Geometric Conservation Law*, Aerospace, 4(8) (2017).

- [5] J. BUCHLIN, *Convective Heat Transfer and Infrared Thermography*, J. Appl. Fluid Mech., 3 (2010), pp. 55–62.
- [6] P. CAUSIN, J. GERBEAU, AND F. NOBILE, *Added-mass effect in the design of partitioned algorithms for fluid-structure problems*, Comp. Meth. Appl. Mech. Eng., 194 (2005), pp. 4506–4527.
- [7] A. CRISTIANO, I. MALOSSI, P. BLANCO, S. DEPARIS, AND A. QUARTERONI, *Algorithms for the partitioned solution of weakly coupled fluid models for cardiovascular flows*, Numer. Meth. Biomed. Eng., 27(12) (2011), pp. 2035–2057.
- [8] M. CROW AND M. ILIC, *The waveform relaxation method for systems of differential/algebraic equations*, in 29th IEEE Conference on Decision and Control, vol. 2, 1990, pp. 453–458.
- [9] C. FONSECA AND J. PETRONILHO, *Explicit inverses of some tridiagonal matrices*, Linear Algebra Appl., 325(1-3) (2001), pp. 7–21.
- [10] M. GANDER, *50 years of time parallel time integration*, in Multiple Shooting and Time Domain Decomposition Methods, T. Carraro, M. Geiger, S. Körkel, and R. Rannacher, eds., Springer, Heidelberg., 2015.
- [11] M. GANDER, L. HALPERN, C. JAPHET, AND V. MARTIN, *Advection diffusion problems with pure advection approximation in subregions*, in Domain Decomposition Methods in Science and Engineering XVI. Lecture Notes in Computer Science and Engineering, vol 50, pp. 239–246. Springer, Berlin., 2007.
- [12] M. GANDER, L. HALPERN, AND F. NATAF, *Optimized Schwarz waveform relaxation for the one dimensional wave equation*, SIAM J. Numer. Anal., 41(5) (2003), pp. 1643–1681.
- [13] M. GANDER, F. KWOK, AND B. MANDAL, *Dirichlet-Neumann and Neumann-Neumann waveform relaxation algorithms for parabolic problems*, ETNA, 45 (2016), pp. 424–456.
- [14] M. GANDER AND A. STUART, *Space-time continuous analysis of waveform relaxation for the heat equation*, SIAM J. Sci. Comput., 19(6) (1998), pp. 2014–2031.
- [15] E. GILADI AND H. KELLER, *Space-time domain decomposition for parabolic problems*, Numer. Math., 93 (2002), pp. 279–313.
- [16] U. HECK, U. FRITSCHING, AND B. K., *Fluid flow and heat transfer in gas jet quenching of a cylinder*, Int. J. Numer. Methods Heat Fluid Flow, 11 (2001), pp. 36–49.
- [17] M. HINDERKS AND R. RADESPIEL, *Investigation of Hypersonic Gap Flow of a Reentry Nosecap with Consideration of Fluid Structure Interaction*, AIAA Paper, 6 (2006), pp. 2708–3741.
- [18] T. HOANG, *Space-time domain decomposition methods for mixed formulations of flow and transport problems in porous media*, PhD thesis, Universitée Pierre et Marie Curie, Paris 6, France, 2013.
- [19] D. KOWOLLIK, P. HORST, AND M. HAUPT, *Fluid-structure interaction analysis applied to thermal barrier coated cooled rocket thrust chambers with subsequent local investigation of delamination phenomena*, Progress in Propulsion Physics, 4 (2013), pp. 617–636.
- [20] D. KOWOLLIK, V. TINI, S. REESE, AND M. HAUPT, *3D fluid-structure interaction analysis of a typical liquid rocket engine cycle based on a novel viscoplastic damage model*, Int. J. Numer. Methods Engrg., 94 (2013), pp. 1165–1190.
- [21] F. KWOK, *Neumann-Neumann waveform relaxation for the time-dependent heat equation*, vol. 98, in Domain Decomposition in Science and Engineering XXI, J. Erhel, M.J. Gander, L. Halpern, G. Pichot, T. Sassi and O.B. Widlund, eds. Lect. Notes Comput. Sci. Eng., 2014.
- [22] E. LELARASMEE, A. RUEHLI, AND A. SANGIOVANNI-VINCENTELLI, *The waveform relaxation method for time-domain analysis of large scale integrated circuits*, IEEE Trans. Comput. Aided Des. Integr. Circuits Syst., 1(3) (1982), pp. 131–145.
- [23] M. MEHL, B. UEKERMANN, H. BIJL, D. BLOM, B. GATZHAMMER, AND A. VAN ZUIJLEN, *Parallel coupling numerics for partitioned fluid-structure interaction simulations*, Comput. Math. Appl., 71(4) (2016), pp. 869–891.

- [24] R. MEHTA, *Numerical Computation of Heat Transfer on Reentry Capsules at Mach 5*, AIAA-Paper, 178 (2005).
- [25] C. MEYER, *Matrix Analysis and Applied Linear Algebra*, 2000.
- [26] A. MONGE AND P. BIRKEN, *On the convergence rate of the Dirichlet-Neumann iteration for unsteady thermal fluid-structure interaction*, Computational Mechanics, (2017).
- [27] A. QUARTERONI AND A. VALLI, *Domain Decomposition Methods for Partial Differential Equations*, Oxford Science Publications, 1999.
- [28] P. STRATTON, I. SHEDLETSKY, AND M. LEE, *Gas Quenching with Helium*, Solid State Phenomena, 118 (2006), pp. 221–226.
- [29] A. TOSELLI AND O. WIDLUND, *Domain Decomposition Methods - Algorithms and Theory*, Springer, 2004.
- [30] E. VAN BRUMMELEN, *Partitioned iterative solution methods for fluid-structure interaction*, International Journal for Numerical Methods in Fluids, 65(1-3) (2011), pp. 3–27.
- [31] J. K. WHITE, *Multirate Integration Properties of Waveform Relaxation with Application to Circuit Simulation and Parallel Computation*, PhD thesis, EECS Department, University of California, Berkeley, 1985.

