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.

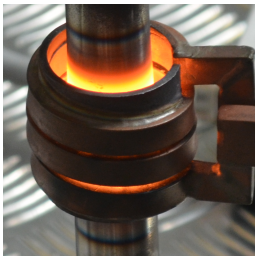
1 Introduction

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]:

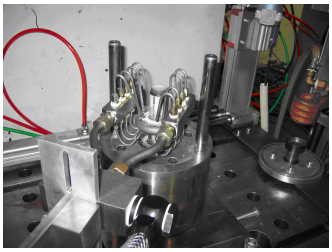
> "The issue of coupling models of different events at different scales and governed by different physical laws is largely wide open and represents an enormously challenging area for future research."

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



(a) Inductive heating.

(b) Thermo-mechanical forming.



(c) Local air-cooling.

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.

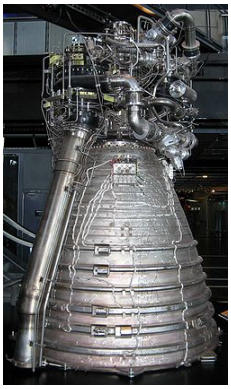
cold high pressured air as a cooling medium (see Figure 1). Knowledge about the time dependent temperature field is imperative to predict where martensite will be in a finished steel part. This allows to predict material properties generated by the forging process.

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.

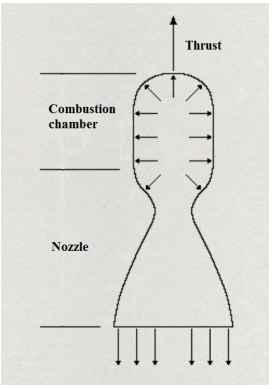
The first stage rocket engine is only used during the launch of the rocket. Therefore, recovering and reusing the first stage will reduce the cost of space access and the environment impact. Related to this, the company SpaceX is developing a set of new technologies for an orbital launch system that 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.

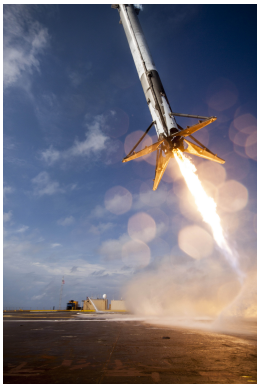


(b) The Vulcain engine in a museum.



(c) Sketch of the rocket thurst 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) Unsuccessul 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.

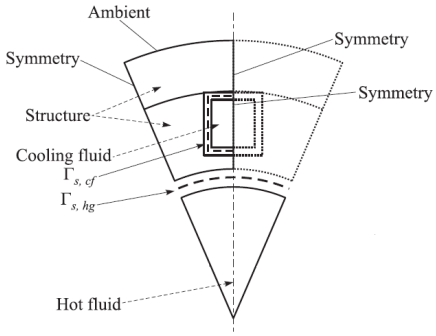


Figure 4. Sketch of the coupling surfaces. Figure taken from [27].

With regards to the space discretization, the use of the finite element method (FEM) is ubiquitious 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 accross 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

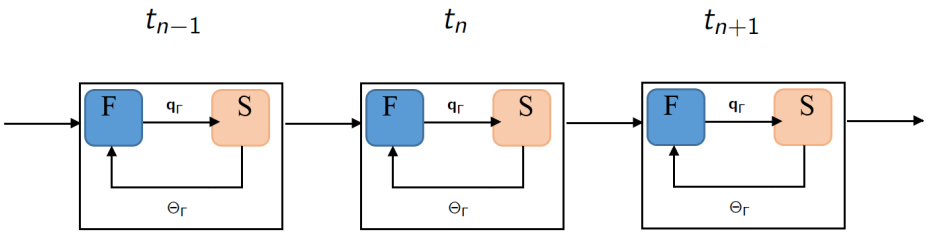


Figure 5. Illustration of the FSI solver.

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 $\tilde{}$ denotes the Favre average and the overbar the ensemble average [6]:

$$\partial_t \overline{\rho} + \nabla \cdot \overline{\rho}\tilde{\mathbf{v}} = 0,$$

$$\partial_t \overline{\rho}\tilde{\mathbf{v}} + \sum_{j=1}^{2} \partial_{x_j}(\overline{\rho}\tilde{v}_i\tilde{v}_j) = -\partial_{x_j}\overline{p}\delta_{ij} + \frac{1}{Re}\sum_{j=1}^{2}\partial_{x_j}\left(\tilde{S}_{ij} + S_{ij}^R\right), \quad i = 1,2 \tag{2.1}$$

$$\partial_t \overline{\rho}\tilde{E} + \nabla \cdot (\overline{\rho}\tilde{H}\tilde{v}_j) = \sum_{j=1}^{2}\partial_{x_j}\left(\overline{(\frac{1}{Re}S_{ij} - S_{ij}^R)v_i} - \overline{\rho}\widetilde{v_j''} + \widetilde{S_{ij}v_i''} - \overline{\rho}\widetilde{v_j''k} + \frac{\overline{W}_j}{RePr}\right).$$

The Reynolds stresses

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

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

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

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

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

and

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

with

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

and

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

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

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;

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

where T is the fluid temperature and $\mathbf{\Theta}$ the structure temperature and

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

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

$$\frac{d}{dt}\mathbf{u} + \mathbf{h}(\mathbf{u}, \mathbf{\Theta}_\Gamma) = \mathbf{0}, \tag{2.9}$$

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

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

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

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 $\mathbf{\Theta}$ 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

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

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

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

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

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

with some initial condition $\mathbf{\Theta}_\Gamma^0$. The iteration is terminated according to the standard criterion

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

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 $\Gamma = \Omega_1 \cap \Omega_2$:

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

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

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

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

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

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:

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

One observes in (3.4) that the rates of the iteration behave as the quotient of thermal conductivities when $\Delta t \to 0$. 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.

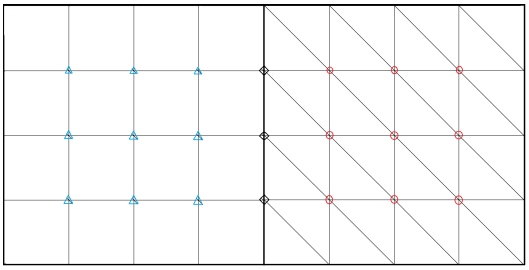


Figure 6. Splitting of Ω between finite volumes and finite elements.

Then, letting $\mathbf{u}_I^{(1)}$ correspond to the unknowns on Ω_1 and $\mathbf{u}_\Gamma$ 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:

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

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

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

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:

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

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

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

Consequently, the equation

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

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 $\mathbf{u} = \left(\mathbf{u}^{(1)}_I, \mathbf{u}^{(2)}_I, \mathbf{u}_\Gamma\right)^T$

$$\tilde{\mathbf{M}}\dot{\mathbf{u}} + \tilde{\mathbf{A}}\mathbf{u} = \mathbf{0}, \tag{3.10}$$

where

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

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 $\mathbf{u}^{n+1} = (\mathbf{u}^{(1),n+1}_I, \mathbf{u}^{(2),n+1}_I, \mathbf{u}^{n+1}_\Gamma)^T$

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

where

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

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

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

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

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

to be solved in succession. Here,

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

and

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

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

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 $\mathbf{u}_\Gamma$. 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

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

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

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

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

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

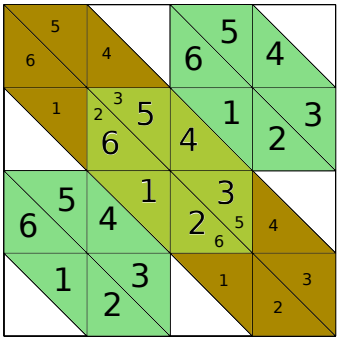


Figure 7. Sketch of the regions for the pyramidal test functions defined in (4.2).

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 $\mathbf{E}_j = \begin{pmatrix} \mathbf{0} & \cdots & \mathbf{0} & \mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} \end{pmatrix}^T \in \mathbb{R}^{N^2\times N}$ where the only nonzero block is the j-th block of size $N \times N$. Thus,

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

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

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

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

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

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

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

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

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

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

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

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,

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

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

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 $j = 1, .., N$. 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:

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

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

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

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

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 $c \to \infty$, $\Delta t \gg \Delta x^2$ 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

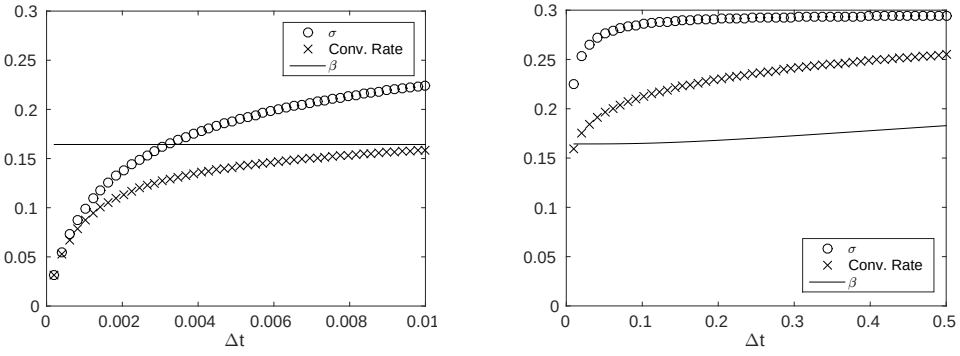
In this section we present a set of numerical experiments designed to show how the validity of the approximation of $\rho(\Sigma)$ as an estimator for the rates of the coupled problem formulated above. We also show that the theoretical asymptotics deduced in (4.12) and (4.13) match with the numerical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



(a) $c \ll 1$ (b) $c \gg 1$

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 $\Delta t = 1e-2/50, 2 \cdot 1e-2/50, ..., 50 \cdot 1e-2/50$ and on the right to the values $\Delta t = 1e-2, 2 \cdot 1e-2, ..., 50 \cdot 1e-2$.

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

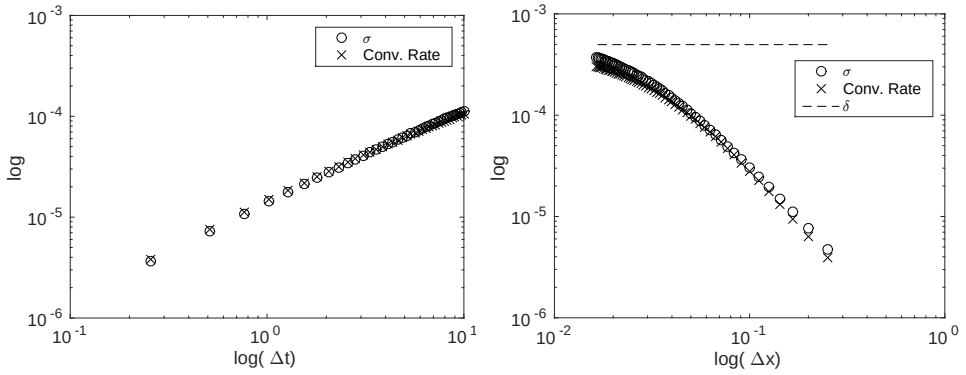
Figures 9 and 10 show the convergence rates for the interactions between air-steel and water-steel respectively. On the left we always plot the rates with respect to the

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

Case	$\Delta t \to 0$	$\Delta x \to 0$
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 $\Delta t = 10/40, 2 \cdot 10/40, ..., 40 \cdot 10/40$ and $\Delta x = 1/20$.

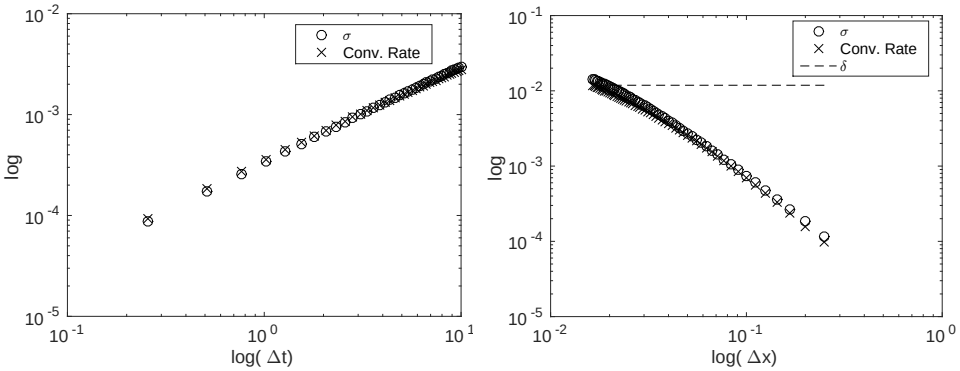
(b) The curves are restricted to the discrete values $\Delta x = 1/3, 1/4, ..., 1/60$ and $\Delta t = 10$.

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 $\Delta t = 10/40, 2 \cdot 10/40, ..., 40 \cdot 10/40$ and $\Delta x = 1/20$.

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

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

α	α	0
1	$1-\alpha$	α
b_i	$1-\alpha$	α
$\hat{b}_i$	$1-\hat{\alpha}$	$\hat{\alpha}$
$b_i - \hat{b}_i$	$\hat{\alpha}-\alpha$	$\alpha-\hat{\alpha}$

$$\alpha = 1 - \sqrt{2}/2$$
$$\hat{\alpha} = 2 - \tfrac{5}{4}\sqrt{2}$$
$$\alpha - \hat{\alpha} = -1 + \tfrac{3}{4}\sqrt{2}$$

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

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

An SDIRK method is then defined as

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

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

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

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

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:

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

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

Here, $\mathbf{s}_i^{\mathbf{u}}$ and $\mathbf{s}_i^{\Theta}$ 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 $\hat{\mathbf{l}}$ is estimated by the solvers separately:

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

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]:

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

Thus, the master solver either needs to know the number of unknowns n_1 and n_2 in the subsolvers, or instead the quantities $n_i\|\mathbf{l}^{(i)}\|^2_{WSN}$ 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:

$$\Delta t_{n+1} = \Delta t_n \cdot \|\mathbf{l}\|^{-1/k}_{WSN} \tag{5.6}$$

Finally, if the possibility of rejected time steps is taken into account, the current solution pair $(\mathbf{u}, \mathbf{\Theta})$ 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^1_n at the next stage time $t_n + c_1\Delta t_n$. Linear extrapolation results in

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

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

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

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

$$\tau = TOL/5. \tag{6.1}$$

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

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

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

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

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

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

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

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

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:

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

We assume that this iteration is well defined and that this sequence has the limit $\mathbf{x}_\epsilon$. 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*

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

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

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

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 $\mathbf{x}^{k+1} \to \mathbf{x}_\epsilon$,

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

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

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

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

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

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 $\phi_k = \delta_k$,

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

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 $\Gamma = \Omega_1 \cap \Omega_2$:

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

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

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

and

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

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 $\mathbf{u}_\Gamma^k$ 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

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

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

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

This was chosen such that the solution is

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

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

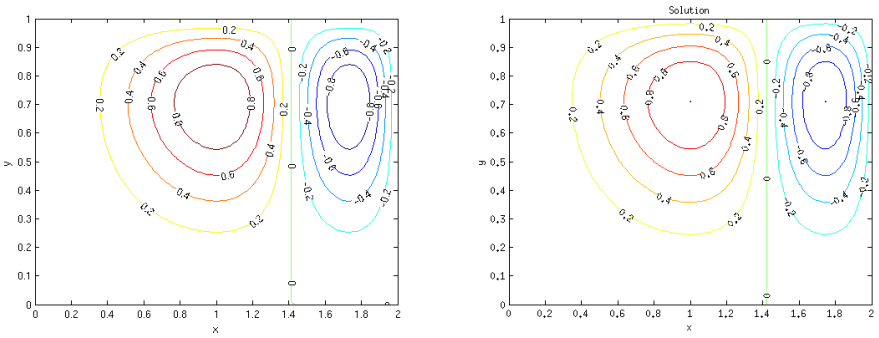


Figure 11. Exact and discrete solution with $\Delta x = 1/64$

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.

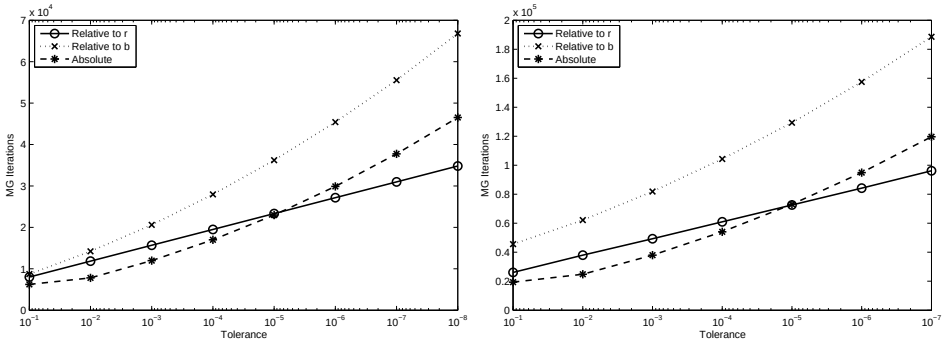


Figure 12. Total multigrid iterations over tolerance for different termination criteria for the transmission problem with rhs (6.15) and $\Delta x = 1/128$ (left) and $\Delta x = 1/256$ (right).

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.

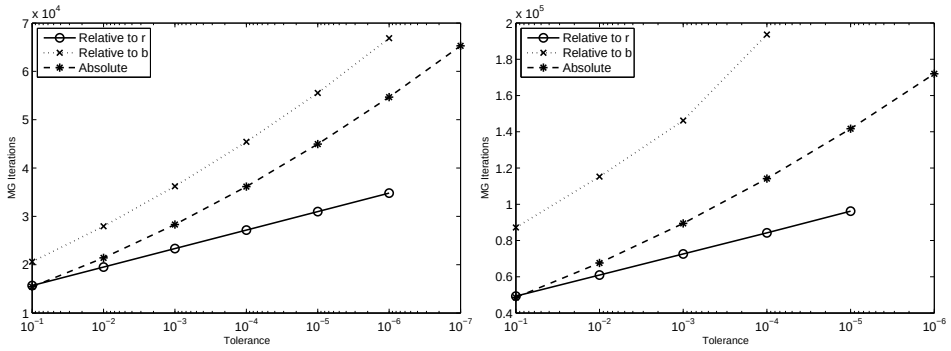


Figure 13. Total multigrid iterations over tolerance for different termination criteria for the transmission problem with rhs (6.15) multiplied by 100 and $\Delta x = 1/128$ (left) and $\Delta x = 1/256$ (right).

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.

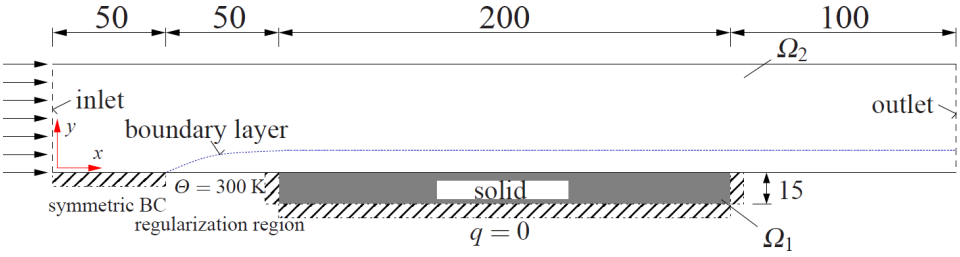


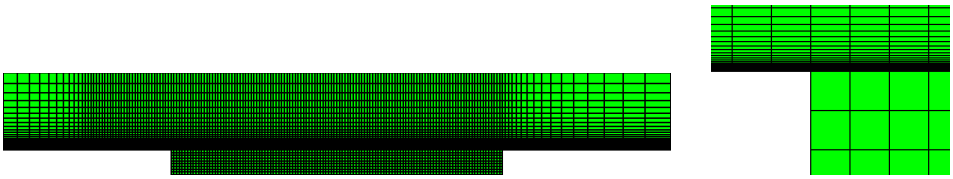
Figure 14. Test case for the coupling method

The inlet is given on the left, where air enters the domain with an initial velocity of $\mathrm{Ma}_\infty = 0.8$ 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 $\Delta t = 0.5s$. To first give an impression on the effect of the time adaptive method, we look at fixed time step



(a) Entire mesh (b) Mesh zoom

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.

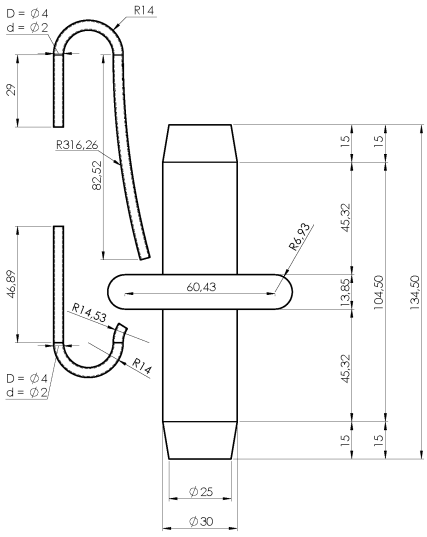


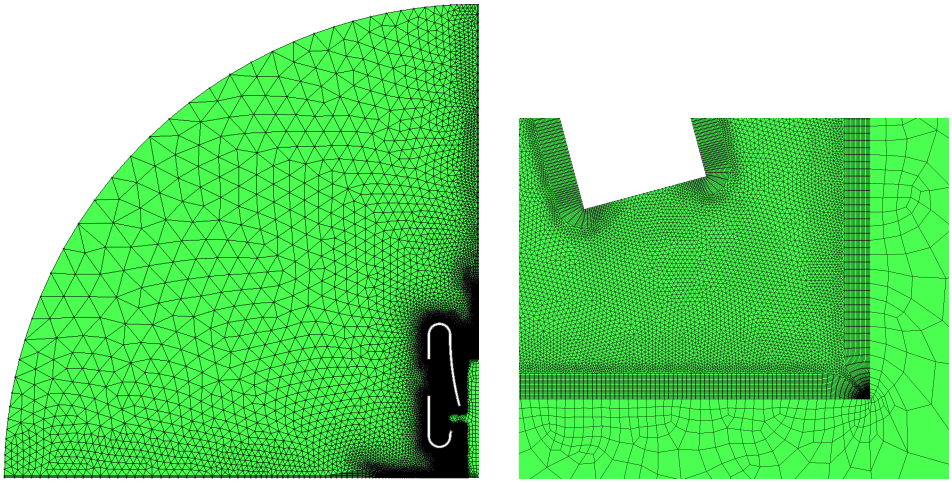
Figure 16. Sketch of the flanged shaft

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



(a) Entire mesh (b) Mesh zoom

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 noticable 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.

TOL	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.

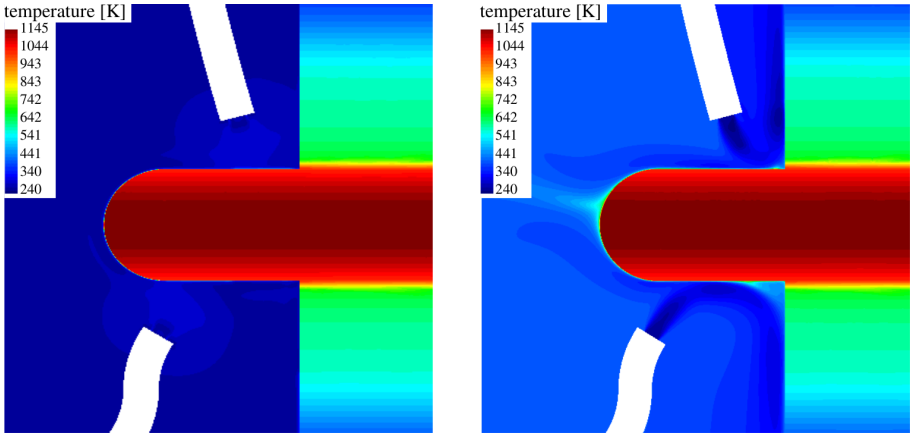


Figure 18. Temperature distribution in fluid and structure at $t = 0s$ (left) and $t = 1s$ (right)

8 Summary and Conclusions

We considered numerical methods for unsteady thermal fluid structure iteraction. 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