TERM PAPER

Harmonic Balance Time Discretization in Finite Element Problems

Author:
Benedikt Hofer

Matriculation-No:
03726169

Supervisors (UoC):
Dr. Alexandros Kontogiannis, Ph. D.
Prof. Matthew Juniper, Ph. D.
Supervisors (TUM):
M. Sc. Philipp Brokof
Dr. Gregoire Varillon, Ph. D.
Prof. Wolfgang Polifke, Ph. D.

04.10.2024

Technische Universität München
Fakultät für Maschinenwesen
Professur für Thermofluiddynamik
Prof. Wolfgang Polifke, Ph. D.

Erklärung

Hiermit versichere ich, die vorliegende Arbeit selbstständig verfasst zu haben. Ich habe Teile dr Arbeit nach der Fertigstellung rein sprachlich mit Grammarly korrigiert. Dabei wurden die inhaltlichen Aussagen in keiner Weise verändert. Dieser Vorgang geschah in Absprache und mit ausdrücklicher Genehmigung seitens der Betreuer. Ich habe keine anderen Quellen und Hilfsmittel als die angegebenen verwendet.

Ort, Datum

Benedikt Hofer

Abstract

The current work focuses on the harmonic balance (HB) time discretization for solving boundary value problems that are periodic in time. The HB method is integrated into finite element (FE) solvers to account for the problems' spatial dependency and all corresponding weak forms are formally derived.

A heat equation problem with periodic wall boundary conditions (BCs) constitutes the first considered problem. Two different formulations of the thermal diffusivity leading to linear and nonlinear formulations are examined. For the nonlinear setting, a Newton algorithm is developed. The linear cases are validated against an Euler time discretization.

Furthermore, an existing two-dimensional steady-state diffuse domain Poisson solver is extended with the HB approach. The diffuse domain method allows for an implicit geometry definition using a phase field function. This function takes on values of unity inside the domain of interest and zero outside of it when the mesh size tends to zero. After adjusting the weak form, the PDEs can be solved on a background tessellation and no body-fitted mesh is required. The essential BCs are weakly enforced using Nitsche's method. The velocity profile is accurately reproduced when testing the solver on a pulsatile flow with a known analytical solution.

Finally, a steady-state Stokes solver, which relies on the diffuse domain method, is adapted to feature the HB. Numerical stabilization terms are introduced to prevent unstable behavior and the Dirichlet BCs are weakly enforced using Nitsche's method. A three-dimensional pulsatile pipe flow for which an analytical solution is available serves as a test case. Promising results are achieved when comparing the velocity profiles with the exact solution.

Contents

Nomenclature

Roman Symbols

$\boldsymbol{n}$	Unit Normal Vector on Domain Boundary	
$\mathbb{T}$	Time Interval	s
$\mathcal{A}$	Kronecker Product Finite Element Matrix	
$\mathpzc{a}$	Thermal Diffusivity Model Parameter	
$\mathcal{B}$	Kronecker Product Finite Element Matrix	
$\mathpzc{c}$	Temporal Expansion Coefficient	
$\mathcal{D}$	Kronecker Product Finite Element Matrix	
$\mathcal{F}$	Cell Facet	
$\mathpzc{f}$	Arbitrary Function	
$\mathpzc{z}$	Signed Distance Function	m
$\mathcal{S}$	Trigonometric Functions Set	
$\mathcal{T}$	Background Tessellation	
$\mathcal{U}$	Solution Matrix	
$\mathpzc{u}$	Physical Quantity	
∇f_0	Jacobian matrix of f_0	
$\partial\mathcal{T}$	Background Tessellation Boundary	
F	Set of Cell Facets	
A	Finite Element Matrix	
$\boldsymbol{a}$	Thermal Diffusivity	m^2/s

a_{I-V}	Velocity-Velocity Bilinear Form	
B	Finite Element Matrix	
$b_{\vec{u},I-II}$	Velocity-Pressure Bilinear Form	
$b_{p,I-II}$	Velocity-Pressure Bilinear Form	
bc	Essential Boundary Condition Form	
D	Finite Element Matrix	
E	Finite Element Matrix	
F	Right Hand Side Matrix	
f	Right Hand Side Vector	
F_0	Residual Matrix	
f_0	Root Finding Problem Function	
g	Boundary Value Function	
h	Mesh Size	m
H_e^1	H^1 Error	
H^k	Space of Square-Integrable Functions with k Square-Integrable Derivatives	
i	Imaginary Number	
j	Boundary Condition Linear Form	
J_0	Bessel Function of First Kind and Order Zero	
K	Simplex	
L	Length	m
L^2	Space of Square-Integrable Functions	
L_e^2	L^2 Error	
n	Number of Harmonics	
n_r	Additional Mesh Refinement Levels	
n_x	Number of Degrees of Freedom	

P	Pressure Matrix	kg/(m s^2)
p	Pressure	kg/(m s^2)
P'_n	Pressure Gradient Magnitude of the n-th Harmonic	kg/(m^2 s^2)
Q	Pressure Test Function Space	
q	Pressure Test Function	kg/(m s^2)
R	Radius	m
r	Radial Coordinate	m
S	Schur Complement	
T	Time Period	s
t	Time	s
C_{ij}	Harmonic Balance Matrix	
c_{ij}	Harmonic Balance Coefficient	
x	Generic Space Variable	m

Non-Dimensional Numbers

Re	Reynolds number	
Wo	Womersley Number	

Greek Symbols

α	Diffuse Domain Parameter	
χ	Characteristic Function	
ϵ	Diffuse Domain Boundary Parameter	
Γ	Boundary Part	
γ	Numerical Parameter	
μ	Viscosity	kg/(m s)
ν	Kinematic Viscosity	m^2/s
Ω	Domain	
ω	Angular Frequency	s^{-1}

$\partial\Omega$	Domain Boundary	
Φ	Phase Field Function	
ϕ	Basis Function	
Ψ	Test Function Space	
ψ	Test Function	
ρ	Density	$\mathrm{kg/m^3}$
τ	Step Size	
θ	Temporal Expansion Function	

Modifiers

$\bullet_{\mathscr{F}}$	Quantity of a Facet
$\lvert\lvert\nabla\bullet_h\rvert\rvert$	Jump in Gradient of a Finite Element Discretized Quantity
$\bullet$	Temporal Quantity
$\bullet^{(i)}$	Quantity at Time Instant t_i
$\bullet'$	Perturbation Quantity
$\bullet_{\mathrm{HB}}$	Harmonic Balance Quantity
$\bullet_a$	Axial Quantity
$\bullet_f$	Forcing Function Quantity
$\bullet_g$	Ghost Penalty Quantity
$\bullet_h$	Quantity in the Finite Element Discretization Space
$\bullet_N$	Natural Boundary Quantity
$\bullet_p$	Pressure Quantity
$\bullet_r$	Radial Quantity
$\bullet_{(m)}$	Quantity at Iteration m
$\bullet_0$	Zero Function Space Quantity
$\bullet_{\mathrm{E}}$	Euler Time Discretized Quantity
$\bullet_{\max}$	Maximum of a Quantity

$\bullet_{\min}$	Minimum of a Quantity
$\bullet_{\mathrm{N}}$	Nitsche Boundary Parameter Quantity
$\bullet_{\vec{u}}$	Velocity Quantity
$\bullet_{ac}$	Active Part of a Quantity
$\bullet_{W}$	Wall Quantity
$\mathcal{R}(\bullet)$	Real Part of a Quantity
$\nabla^{s}\bullet$	Strain Rate Tensor
$\tilde{\bullet}$	Approximation
$\hat{\bullet}_{c}$	Discrete Fourier Spectrum Cosine Coefficient
$\hat{\bullet}_{s}$	Discrete Fourier Spectrum Sine Coefficient

Acronyms

ASM	Additive Schwarz Method
BC	Boundary Condition
CIP	Continuous Interior Penalty
DE	Differential Equation
DFT	Discrete Fourier Transform
DOF	Degree of Freedom
FE	Finite Element
FEM	Finite Element Method
HB	Harmonic Balance
MRV	Magnetic Resonance Velocimetry
PDE	Partial Differential Equation
PINN	Physics-Informed Neural Network
RHS	Right Hand Side
SNR	Signal-to-Noise Ratio

1 Introduction

1.1 Motivation in the Context of Current Advances

Measuring complex three-dimensional flow fields is a commonly encountered task in multiple fields of interest. For example, knowledge of hemodynamics is essential in medical care due to its role in diagnosing cardiovascular diseases [1, 2]. In engineering flows, possible applications include the analysis of turbine cooling passages, multiphase bio- or microreactors [3–5]. 4D magnetic resonance velocimetry (MRV) represents one noninvasive option for obtaining three-dimensional velocity fields without the use of optical access or flow markers [4]. However, the signal-to-noise ratio (SNR) is inversely proportional to the spatial resolution, implying the need for temporal averaging. As a consequence, a long signal acquisition duration is required, constituting a substantial challenge associated with this method [4, 6, 7].

Compressed sensing (CS) presents one possible way of dealing with sparsely sampled data, allowing for shorter measurement times [8, 9]. The underlying idea is to recover the original signal given the sparse measurements by solving an optimization problem using a reconstruction algorithm [10]. These algorithms can be regularized by employing prior knowledge about the data structure using techniques such as L^1-norm regularization [7, 11]. Furthermore, the physics of the problem can be leveraged for the regularization [7, 12].

When dealing with flow field reconstruction, insights into the physics are given by the Navier-Stokes equations. As a consequence, the task of reconstructing and segmenting a flow image takes the form of an inverse Navier-Stokes problem with unknown geometry, boundary conditions (BCs) and fluid properties. These parameters are inferred on the model to ensure an accurate reconstruction of the flow field measurements, which adheres to the physics. One key advantage of this method is its ability to simultaneously approximate additional information, such as pressure, which is inherently challenging to quantify [7, 13].

The work of Funke et al. [14] revolves around cardiovascular conditions caused by locally abnormal wall shear stresses. They investigated three- and four-dimensional synthetic and measured data on unsteady pulsatile flow in cerebral aneurysms. In their framework, the initial conditions and Dirichlet BCs are the inferred parameters identified by the inverse Navier-Stokes problem. The velocity and wall shear stress results accurately match the data well when appropriate regularization is applied even in the limit of significant noise levels [14].

Koltukluoğlu and Blanco [15] conducted a study on reduced inverse steady Navier-Stokes problems, where the Dirichlet inlet BC is inferred on the model. The main focus is on near-wall regions, aiming to reveal more information about wall shear stresses. Three different three-dimensional setups are analyzed, including a Hagen-Poiseuille flow and both numerical and experimental MRV data of a glass replica of the human aorta. The result achieved using the introduced model corresponds well with the data from all three cases and the model itself demonstrates robust behavior [15]. Later, in [16] and [17], the approach is extended to periodic flows, where the harmonic balance (HB) method is utilized to account for the time dependency of the unsteady Navier-Stokes problem. This method is validated with the same experimental setup as in [15], but with four-dimensional data. Solutions acquired with approximately ten harmonics are compared to the data and a good agreement is found. Additionally, the HB method is evaluated against a standard second-order numerical time discretization scheme, precisely backward differentiation, and performs about 15 times faster while achieving a similar reconstruction accuracy [16, 17].

All studies mentioned above treat the flow domain as a known quantity. In practice, the information on boundary position, time-averaged or instantaneous, stems from additional experiments [13]. This constitutes a notable constraint restraining such techniques, as measurement uncertainties of the domain are introduced [7, 13, 18, 19]. These deviations comprise the predictions of the wall shear stress [7, 13, 20, 21], which often is the main quantity of interest [14, 15]. Alternatively, there are approaches where parts of the boundary or the whole boundary are incorporated as unknowns in the inverse problem [7, 13, 22].

In [22], the focus is set on valve heart diseases by reconstructing two and three-dimensional images with an open valve. A formulation of an inverse steady Navier-Stokes problem is introduced, where the flow obstacle is replaced with a permeability parameter, which follows Brinkman's law and is incorporated into the momentum equations. The derived model, whose inferred parameters are the permeability parameter and the maximum velocity of a prescribed inlet velocity profile, is applied to synthetic two and three-dimensional data. In the case of noise-free data, the approach performs well. However, the utilized threshold criterion fails to recover the space between the valves accurately with increasing noise levels [22].

Kontogiannis et al. [7] investigated the reconstruction of noisy 2D-images of a steady incompressible flow by solving a Bayesian-inverse Navier-Stokes problem, whose well-posedness has been shown in [23]. In their method, the flow domain, inlet and outlet boundary conditions and kinematic viscosity are the unknown parameters inferred on the model. This model was tested on synthetic data of a converging channel and an abdominal aortic aneurysm with a low SNR and experimental data obtained using MRV with a slightly higher SNR. The authors showed that the proposed approach can accurately reconstruct velocity and reduced pressure in all three test cases [7]. Recently, Kontogiannis et al. [13] further developed and optimized their previous work towards reconstructing three-dimensional steady laminar flow images. The implemented fictitious domain cut cell finite element method (FEM) is tested on MRV images depicting the flow in a replica of the human aorta. The model's performance is evaluated on four different flows, including high and low SNR flow images at two different Reynolds numbers, Re. Excellent agreement between the reconstruction and the data is achieved at lower Re regardless of the SNR. For the higher Re, small but notable reconstruction errors are reached. These discrepancies are attributed to the near-critical Re, leading to errors in the averaging and the possibility of local turbulence [13].

The current study serves as an intermediate step in extending the work in [13] towards periodic flows. With this future goal in mind, the HB time discretization approach proposed in [16, 17] is further analyzed. The focus is on replacing the relatively loosely temporally coupled numerical implementation in [16, 17] with a more robust variant. Due to the intricacy of the inverse problem and the importance of an accurate forward model [24, 25], this work limits its scope to forward problems. The proposed approach is initially validated on a heat equation with periodic BCs and subsequently combined with two existing FEM solvers for Poisson and Stokes problems.

Before advancing with the current work, one more rapidly growing field in the context of flow reconstruction is addressed: the use of physics-informed neural networks (PINNs). Numerous studies, for example, [26–28] have explored their abilities in the given context. As Kontogiannis et al. [13] point out, solving an inverse Navier-Stokes problem offers clear advantages over PINNs. One general issue with PINNs is that the physics are treated as a soft constraint of a minimization problem. In comparison, the studies mentioned earlier aim to find actual solutions to the Navier-Stokes problem. The inverse models can be understood by humans and are mathematically analyzable. Additionally, the derived inverse techniques can be applied to unknown Navier-Stokes flows without the need to provide training data [13].

1.2 Outline

This study comprises six chapters. After the introduction, which includes a brief overview of current research in the context of flow field reconstruction, Chap. 2 explains the theoretical background of the employed FEM and HB approach. Based on this knowledge, Chap. 3 focuses on deriving one possible coupling of both methods and testing it on a simple test case in the form of the heat equation. In Chap. 4 and 5, the derived method is extended by combining it with diffuse domain Poisson and Stokes flow solvers developed by Matthias Henssler and Dr. Alexandros Kontogiannis at the Energy Research Group of the Department of Engineering at the University of Cambridge. Lastly, a conclusion of the findings is drawn and further research possibilities are presented in Chap. 6.

2 Theoretical Background

2.1 Finite Element Method

Although the governing equations of many problems, such as fluid flow through arbitrary shapes or heat transfer with complex material properties, are well-known, analytical solutions are often not obtainable due to the intricacy of these problems. As a result, numerical methods are widely used to find approximations of these solutions at a discrete number of points [29, 30]. In the current work, a numerical approach, referred to as the finite element method, is used, and this will be briefly introduced in this section. Note that $L^2(\Omega)$ denotes the space of square-integrable functions in the domain Ω and $H^k(\Omega)$ is the space of square-integrable functions with k square-integrable derivatives.

The basic idea of FEM is to divide the domain into a system of small elements and recast the governing equations in a form that can be solved numerically. Subsequently, the governing equations of each element are combined into a global system, yielding the entire solution [30, 31]. One possible approach for finding such an expression of the differential equations (DEs) is called the weak form [31, 32]. To illustrate this concept, the weak form of a two-dimensional Poisson problem is derived. Let the domain $\Omega \subset \mathbb{R}^2$ be an open set, which is enclosed by its boundary $\partial\Omega = \Gamma_1 \cup \Gamma_2$. The behavior of a physical quantity, u, is given by

$$\begin{cases} \nabla \cdot (\nabla u) = 1 & \text{in } \Omega, \\ u = 1 & \text{on } \Gamma_1, \\ \frac{\partial u}{\partial \boldsymbol{n}} = 1 & \text{on } \Gamma_2, \end{cases} \tag{2.1}$$

where $\boldsymbol{n}$ is the unit normal vector on $\partial\Omega$. The procedure to finding the weak form is multiplying Eq. (2.1) with an arbitrary test function ψ from the test space

$$\Psi = \left\{ \psi \in H^1(\Omega) \cdot \psi|_{\Gamma_1} = 0 \right\} \tag{2.2}$$

and integrating the expression over Ω, which results in [32–34]:

$$-\int_\Omega \nabla\psi \cdot \nabla u + \int_\Gamma \left(\psi \nabla u\right) \cdot \boldsymbol{n} = \int_\Omega \psi. \tag{2.3}$$

Since $\psi = 0$ on Γ_1 and $\nabla u \cdot \boldsymbol{n} = 1$ on Γ_2, Eq. 2.3 becomes

$$-\int_\Omega \nabla\psi \cdot \nabla u = \int_\Omega \psi - \int_{\Gamma_2} \psi, \tag{2.4}$$

which is the weak form of Eq. 2.1 [32, 33].

The problem to solve is [35]:

$$\text{find } u \text{ in } H^1(\Omega) \text{ such that } -\int_\Omega \nabla\psi\cdot\nabla u = \int_\Omega \psi - \int_{\Gamma_2}\psi \quad \forall\psi\in\Psi.$$

To achieve this goal, a finite-dimensional approximation subspace $\Psi_h \subset \Psi$, which consists of continuous piecewise polynomial functions defined on a subdivision of Ω, is considered. The finite element (FE) approximation of the problem then reads [35]:

$$\text{find } u_h \text{ in } H^1(\Omega)_h \text{ such that } -\int_\Omega \nabla\psi_h\cdot\nabla u_h = \int_\Omega \psi_h - \int_{\Gamma_2}\psi_h \quad \forall\psi_h\in\Psi_h,$$

where the subscript $\bullet_h$ denotes a quantity in the finite element discretization space. By assuming that

$$\dim\Psi_h = n_x \quad \text{and} \quad \Psi_h = \operatorname{span}\{\phi_1,\phi_2,\dots,\phi_{n_x}\}, \tag{2.5}$$

where n_x is the number of total degrees of freedom (DOF) at which the solution is approximated and ϕ_i are the linearly independent basis functions with compact support, the solution in the approximation space can be written using the basis functions

$$u_h(x) = \sum_{i=1}^{n_x} u_{h,i}\,\phi_i(x). \tag{2.6}$$

Finally, the following problem is reached [35]:

$$\text{find } (u_{h,1},u_{h,2},\dots,u_{h,n_x})^T \text{ in } \mathbb{R}^{n_x} \text{ such that}$$

$$\textstyle\sum_{i=1}^{n_x} -u_{h,i}\int_\Omega \nabla\phi_j\cdot\nabla\phi_i = \int_\Omega\phi_j - \int_{\Gamma_2}\phi_j, \quad j=1,2,\dots,n_x.$$

Since the basis has compact support, a sparse matrix is obtained, where the entry at position (i,j) is given by $-\int_\Omega\nabla\phi_j\cdot\nabla\phi_i$. The right hand side (RHS) of the problem becomes a vector with the entry $\int_\Omega\phi_j - \int_{\Gamma_2}\phi_j$ at position j and thus, a linear system of equations for $(u_{h,1},u_{h,2},\dots,u_{h,n_x})^T$ is found [35].

2.2 Harmonic Balance Time Discretization

In this work, a time discretization method, which is referred to as harmonic balance, is employed to solve time-dependent partial differential equations (PDEs) describing a physical quantity u [16]. A formal introduction to this method does not require a precise definition of spatial domain or its dimensionality and for now, x serves as a generic space variable.

Using an arbitrary function, f, the examined PDEs, which in the context of the current work are limited to first-order in time, t, read

$$\partial_t u = f(u,t). \tag{2.7}$$

Further, the solution, $u(t,x)$, is assumed to be a real-valued periodic function with a known period T in the time interval $\mathbb{T} := [0,T]$. A sequence of real valued basis functions θ_k and corresponding coefficients $c_k(x)$ allow for an approximation of u, denoted with $\tilde{u}$, in space and time [16, 17]:

$$u(t,x) \approx \tilde{u} = \sum_{k=1}^{2n+1} c_k(x)\theta_k(t), \tag{2.8}$$

where the basis θ_k is taken from a set consisting of trigonometric functions

$$\mathcal{S} := \{1, \cos(\omega t), \sin(\omega t), \cos(2\omega t), \sin(2\omega t), ..., \cos(n\omega t), \sin(n\omega t)\}. \tag{2.9}$$

Here, $\mathcal{S}$ is a complete L^2-orthogonal system on $\mathbb{T}$ and $\omega = \frac{2\pi}{T}$ is the angular frequency [16, 17]. Consequently, a Fourier polynomial of degree n for the desired approximation is obtained:

$$\tilde{u}(t,x) = \frac{\hat{\tilde{u}}_{c_0}}{2} + \sum_{k=1}^{n} \left(\hat{\tilde{u}}_{c_k}(x)\cos(k\omega t) + \hat{\tilde{u}}_{s_k}(x)\sin(k\omega t)\right), \tag{2.10}$$

where $\hat{\tilde{u}}_{c_k}(x)$ for $k = 0, 1, ..., n$ and $\hat{\tilde{u}}_{s_k}(x)$ for $k = 1, 2, ..., n$ construct the discrete Fourier spectrum of $\tilde{u}$ and n is the number of harmonics [16, 17, 36].

This spectrum comprises $2n+1$ unknowns and thus, $2n+1$ collocation points, t_j, in time at which Eq. (2.7) holds true are gathered. These points are chosen to be equidistant within $\mathbb{T}$:

$$t_j := \frac{jT}{2n+1}, \quad j = 1, 2, ..., 2n+1. \tag{2.11}$$

Substituting Eq. (2.10) evaluated at the collocation points into the initial PDE (2.7) results in a system consisting of $2n+1$ equations. However, instead of solving this set of equations for the discrete Fourier spectrum coefficients, a discrete Fourier transform (DFT) is used to obtain $\tilde{u}$ at the time instants given by Eq. 2.11 [16, 17].

For this purpose, a harmonically balanced system of equations is derived. Let $\tilde{u}^{(i)} := \tilde{u}(t_i, x)$ be the approximation, $\tilde{u}$, at time instant t_i. The discrete cosine and sine transforms

$$\hat{\tilde{u}}_{c_k} = \frac{2}{2n+1} \sum_{i=1}^{2n+1} \tilde{u}^{(i)} \cos(k\omega t^{(i)}) \qquad \text{and} \qquad \hat{\tilde{u}}_{s_k} = \frac{2}{2n+1} \sum_{i=1}^{2n+1} \tilde{u}^{(i)} \sin(k\omega t^{(i)}) \tag{2.12}$$

are substituted into Eq. (2.10) and evaluated at all times t_j [16, 17]:

$$\begin{aligned} \tilde{u}^{(j)}(t^{(j)}, x) = \frac{2}{2n+1} \sum_{i=1}^{2n+1} \tilde{u}^{(i)} \Bigg(\frac{1}{2} + \sum_{k=1}^{n} \Big(\cos\left(k\omega t^{(i)}\right)\cos\left(k\omega t^{(j)}\right) \\ + \sin\left(k\omega t^{(i)}\right)\sin\left(k\omega t^{(j)}\right)\Big)\Bigg), \quad j = 1, 2, ..., 2n+1. \end{aligned} \tag{2.13}$$

Expression (2.13) can be differentiated with respect to t and inserted into the initial PDE, see (2.7), resulting in

$$\frac{2\omega}{2n+1} \sum_{i=1}^{2n+1} \sum_{k=1}^{n} k\tilde{u}^{(i)} \sin\left(k\omega\left(t^{(i)} - t^{(j)}\right)\right) = f\left(\tilde{u}^{(j)}, t^{(j)}\right), \quad j = 1, 2, ..., 2n+1. \tag{2.14}$$

Finally, defining a harmonic balance coefficient c_{ij},

$$c_{ij} := \frac{2\omega}{2n+1} \sum_{k=1}^{n} k \sin\left(k\omega\left(t^{(i)} - t^{(j)}\right)\right), \quad i,j = 1,2,\ldots,2n+1 \tag{2.15}$$

yields the following compact form of Eq. (2.14), which can be solved for $\tilde{u}$ at all $2n+1$ collocation points [16, 17]:

$$\sum_{i=1}^{2n+1} \tilde{u}^{(i)} c_{ij} = f\left(\tilde{u}^{(j)}, t^{(j)}\right), \quad j = 1,2,\ldots,2n+1. \tag{2.16}$$

By defining a matrix $C_{ij} \in \mathbb{R}^{(2n+1)\times(2n+1)}$, whose entries at positions (i,j) for $i,j = 1,2,\ldots,2n+1$ are given by Eq. (2.15), following expression is found

$$\frac{\overrightarrow{\partial u}}{\partial t} = \vec{u}\, C_{ij}, \tag{2.17}$$

where each row of the vectors $\frac{\overrightarrow{\partial u}}{\partial t}$ and $\vec{u}$ corresponds to one time instance.

Note that, for readability reasons, the symbol $\tilde{\bullet}$ will be dropped from now on and is implied when c_{ij} or C_{ij} appears in the context.

3 Heat Equation with Periodic Boundary Conditions

After deriving the HB time discretization in Sec. 2.2, it will now be applied to an initial test case based on the heat equation. As the outline states, this chapter aims to illustrate and test the HB method on a simple test case. Particularly, Sec. 3.2 is crucial as it contains the coupling of the HB with finite elements and introduces different approaches for solving the linear and nonlinear HB-coupled equations. These solution methods naturally extend to the more complex problems discussed in Chap. 4 and 5.

3.1 Problem Definition

Figure 3.1 depicts the domain of interest, which is an open set $\Omega \subset \mathbb{R}^2$ enclosed by its boundary $\partial\Omega = \Gamma_{\mathrm{HB}} \cup \Gamma_W$. These boundaries represent the harmonic balance and wall boundaries, Γ_{HB} and Γ_W, respectively. For simplicity in this first test case, a square with a side length of $L = 1\,\mathrm{m}$ is chosen for Ω and zero Dirichlet BCs, $g_W(t,x) = 0 : \mathbb{T} \times \Gamma_W \to \mathbb{R}$, are imposed on Γ_W. Furthermore, u, which is the temperature, adheres to a T-periodic function $g_{\mathrm{HB}}(t,x) = g_{\mathrm{HB}}(t+nT,x) : \mathbb{T} \times \Gamma_{\mathrm{HB}} \to \mathbb{R}$ with $n \in \mathbb{N}$ and the inspected time interval is $\mathbb{T} := [0,T]$.

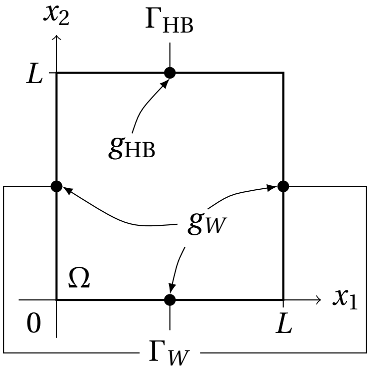

Figure 3.1: Investigated heat equation problem.

Finally, the heat equation problem in Ω reads

$$\begin{cases} \frac{\partial u}{\partial t} - \nabla \cdot (a \nabla u) = 0 & \text{in } \mathbb{T} \times \Omega, \\ u = g_{\text{HB}} & \text{on } \mathbb{T} \times \Gamma_{\text{HB}}, \\ u = 0 & \text{on } \mathbb{T} \times \Gamma_W, \end{cases} \tag{3.1}$$

where a is the thermal diffusivity.

Two approaches for defining a are considered. Initially, a is assumed constant, resulting in a linear PDE. Later, non-linearity is introduced by considering a to be a function of u:

$$a(u) = \left(a_1 + a_2 u^{a_3}\right), \quad a_1, a_2 \in \mathbb{R}^+, \quad a_3 \in \mathbb{N}^+, \tag{3.2}$$

where $a_{1/2/3}$ are constant model parameters. In addition, two different boundary value functions,

$$g_{\text{HB},1}(t,x) = \sin\left(\frac{\pi}{L}x\right)\sin(\omega t) \qquad \text{and} \qquad g_{\text{HB},2}(t,x) = \sin\left(\frac{\pi}{L}x\right)\sin(\omega t)^2, \tag{3.3}$$

are implemented. By choosing $\omega = 2\pi$, a period of $T = 1$ is obtained for all four cases, H1-H4, which are summarized in Tab. 3.1.

Table 3.1: Summary of the examined heat equation setups.

	H1	H2	H3	H4
a	const.	const.	$f(u)$	$f(u)$
BC	$g_{\text{HB},1}$	$g_{\text{HB},2}$	$g_{\text{HB},1}$	$g_{\text{HB},2}$

3.2 Discrete Weak Forms

With the initial test case formally introduced, this section explains how the HB and FEM can be combined to simultaneously account for both the spatial and temporal dependencies of the problem.

3.2.1 Weak and Matrix Form of the Linear Problem

Let

$$\Psi = \left\{\psi \in H^1(\Omega) : \psi|_{\Gamma_{\text{HB}} \cup \Gamma_W} = 0\right\} \tag{3.4}$$

be the space of temperature test functions ψ. Testing Eq. (3.1) with ψ and integrating by parts yields

$$\int_\Omega \psi \frac{\partial u}{\partial t} + \int_\Omega a \nabla\psi \cdot \nabla u - \int_{\partial\Omega} \left(a \psi \nabla u\right) \cdot \boldsymbol{n} = 0, \tag{3.5}$$

where $\boldsymbol{n}$ is the unit normal vector on $\partial\Omega$. Since $\psi = 0$ on $\partial\Omega$, Eq. (3.5) simplifies to

$$\int_\Omega \left(\psi \frac{\partial u}{\partial t} + a \nabla\psi \cdot \nabla u\right) = 0. \tag{3.6}$$

Recall from Sec. 2.2 that for each $t^{(j)}$ among the $2n+1$ collocation points in time, the examined PDE, in this case, Eq. (3.6), must be satisfied and further

$$\frac{\partial u}{\partial t}^{(j)} = \sum_{i=1}^{2n+1} u^{(i)} c_{ij}, \quad j = 1,2,...,2n+1. \tag{3.7}$$

When substituting (3.7) into the weak form from Eq. (3.6), a HB time discretized version is reached:

$$\sum_{i=1}^{2n+1} \left(\left(\int_\Omega \psi u^{(i)} \right) c_{ij} \right) + \int_\Omega a \nabla\psi \cdot \nabla u^{(j)} = 0, \quad j = 1,2,...,2n+1. \tag{3.8}$$

In order to obtain a spatially discretized approximation of $u^{(i)}$ for $i = 1,2,...,2n+1$, the FE approach introduced in Sec. 2.1 is employed. Therefore, a n_x-dimensional FEM subspace spanned by its basis functions ϕ_k is introduced. Confining $u^{(i)}$ as well as ψ to Ψ_h and expressing them using ϕ_k yields

$$\begin{gathered}\sum_{i=1}^{2n+1} \left(\sum_{k=1}^{n_x} \left(u_{h,k}^{(i)} \left(\int_\Omega \phi_l \phi_k \right) \right) c_{ij} \right) + \sum_{k=1}^{n_x} \left(u_{h,k}^{(j)} \int_\Omega a \nabla\phi_l \cdot \nabla\phi_k \right) = 0, \\ l = 1,2,\dots,n_x, \quad j = 1,2,\dots,2n+1.\end{gathered} \tag{3.9}$$

Moving forward, the expression of quantities using ϕ_k is no longer included for readability and is implied by the subscript $\bullet_h$.

Defining the vectors

$$\vec{u}^{(i)} = \left(u_1^{(i)}, u_2^{(i)}, ..., u_{n_x}^{(i)} \right)^T \text{ for } i = 1,2,...,2n+1$$

allows to write Eq. (3.9) as

$$\sum_{i=1}^{2n+1} \left(A_{lk} \vec{u}^{(i)} c_{ij} \right) + B_{lk} \vec{u}^{(j)} = f_l^{(j)}, \quad j = 1,2,...,2n+1, \tag{3.10}$$

where the entries of the two sparse FEM matrices, A_{lk} and $B_{lk} \in \mathbb{R}^{n_x \times n_x}$, are given by $\int_\Omega \phi_l \phi_k$ and $\int_\Omega \nabla\phi_l \cdot \nabla\phi_k$, respectively. The RHS vectors $f_l^{(j)} \in \mathbb{R}^{n_x}$ store the Dirichlet BC values at each time instance.

Instead of representing the solution as $2n+1$ vectors $\vec{u}^{(i)}$, which are coupled in time by the HB, a matrix $\mathcal{U}_{ki} \in \mathbb{R}^{(n_x)\times(2n+1)}$ can be defined. In this matrix, every row i corresponds to the i-th DOF in space and each column j contains the solution at the j-th time step. The same approach is used to define the RHS matrix $F_{lj} \in \mathbb{R}^{(n_x)\times(2n+1)}$ and finally, by introducing C_{ij}, this matrix form of the problem is obtained:

$$A_{lk} \mathcal{U}_{ki} C_{ij} + B_{lk} \mathcal{U}_{ki} \mathrm{I}_{ij} = F_{lj}, \tag{3.11}$$

where I_{ij} is the identity matrix.

Following [37], equations of similar type as (3.11) can be rewritten using the Kronecker product ⊗,

$$
\begin{aligned}
&(C_{ij}^T \otimes A_{lk})\,\mathrm{vec}(\mathcal{U}_{ki}) + (\mathrm{I}_{2n+1} \otimes B_{lk})\,\mathrm{vec}(\mathcal{U}_{ki}) = \\
&\qquad\underbrace{\Big((C_{ij}^T \otimes A_{lk}) + (\mathrm{I}_{2n+1} \otimes B_{lk})\Big)}_{K}\mathrm{vec}(\mathcal{U}_{ki}) = \mathrm{vec}(F_{lj}),
\end{aligned}
\tag{3.12}
$$

where I_{2n+1} is the identity matrix of size $2n+1$ and K is the system matrix. The linear vectorization operation, which transforms a matrix into a vector by stacking its columns on top of each other, is denoted with vec. Finally, solving this linear system of equations yields the harmonically balanced FE discretized approximate solution of Eq. (3.1).

3.2.2 Weak and Matrix Form of the Nonlinear Problem

After deriving the weak form of the linear problem, this subsection addresses how the varying thermal diffusivity introducing nonlinearity is accounted for in the weak and matrix forms.

Let

$$
\Psi = \left\{ \psi \in H^1(\Omega) : \psi\big|_{\Gamma_{\mathrm{HB}} \cup \Gamma_W} = 0 \right\} \tag{3.13}
$$

be the space of temperature test functions ψ. Testing Eq. (3.1) with ψ and integrating by parts leads to

$$
\int_\Omega \left(\psi \frac{\partial u}{\partial t} + a(u)\, \nabla\psi \cdot \nabla u \right) = 0. \tag{3.14}
$$

Eq. (3.7) is substituted into (3.14), where the thermal diffusivity is a function of the temperature. As a result, it becomes a varying quantity in time and space and is denoted with $a^{(i)}$ for $i = 1, 2, \ldots, 2n+1$:

$$
\sum_{i=1}^{2n+1} \left(\left(\int_\Omega \psi u^{(i)} \right) c_{ij} \right) + \int_\Omega a(u)^{(j)}\, \nabla\psi \cdot \nabla u^{(j)} = 0, \quad j = 1, 2, \ldots, 2n+1. \tag{3.15}
$$

For the spatial discretization of $u^{(i)}$ and $a^{(i)}$ for $i = 1, 2, \ldots, 2n+1$ a n_x-dimensional subspace spanned by the basis functions ϕ_k is constructed. Confining the quantities $u^{(i)}$, $a^{(i)}$ and ψ to the FEM subspace results in

$$
\sum_{i=1}^{2n+1} \left(\left(\int_\Omega \psi_h u_h^{(i)} \right) c_{ij} \right) + \int_\Omega a_h(u)^{(j)}\, \nabla\psi_h \cdot \nabla u_h^{(j)} = 0, \quad j = 1, 2, \ldots, 2n+1. \tag{3.16}
$$

While the matrices A_{lk} and C_{ij} remain similar to Subsec. 3.2.1, a different FEM matrix, $B_{lk}^{(i)}$, is obtained for every collocation point in time. Its entries are calculated using tensor contraction, transforming the local three-dimensional tensor into a matrix. Subsequently, a slightly modified version of Eq. (3.10) is reached:

$$
\sum_{i=1}^{2n+1} \left(A_{lk} \vec{u}^{(i)} c_{ij} \right) + B(u)_{lk}^{(j)}\, \vec{u}^{(j)} = f_l^{(j)}, \quad j = 1, 2, \ldots, 2n+1. \tag{3.17}
$$

Further, a modified Kronecker operator $\mathrm{I}_{n_x} \bar{\otimes} B(u)_{lk}^{(i)}$, which places the matrix $B(u)_{lk}^{(i)}$ on the i-th diagonal block of the system matrix K consisting of $(2n+1)^2$ subblocks, is introduced. Using this Kronecker operator, the matrix form of the problem now reads

$$\left((C_{ij}^T \otimes A_{lk}) + \left(\mathrm{I}_{n_x} \bar{\otimes} B(u)_{lk}^{(j)}\right)\right) \mathrm{vec}(\mathcal{U}_{ki}) = \mathrm{vec}(F_{lj}). \tag{3.18}$$

3.2.3 Linearised Nonlinear Problem

A Newton method with the update rule of

$$u_{(m+1)} = u_{(m)} + u'_{(m)} \tag{3.19}$$

is employed to solve the root finding problem stemming from Eq. (3.1)

$$f_0\left(u_{(m)}\right) = 0 = \frac{\partial u_{(m)}}{\partial t} - \nabla \cdot \left(a(u_{(m)}) \nabla u_{(m)}\right) \quad \text{in } \mathbb{T} \times \Omega. \tag{3.20}$$

Here, f_0 denotes the root finding problem function and $\bullet_{(m)}$ is the iteration number. The two RHS terms of Eq. (3.20) are identical with Eq. (3.1) and hence, the matrix, $F_{0,lj}$, containing the residuals for all times can be calculated using (3.12)

$$\left((C_{ij}^T \otimes A_{lk}) + \left(\mathrm{I}_{2n+1} \bar{\otimes} B(u_{(m)})_{lk}\right)\right) \mathrm{vec}(\mathcal{U}_{ki,(m)}) - \mathrm{vec}(F_{lj}) = \mathrm{vec}(F_{0,lj,(m)}). \tag{3.21}$$

The temperature perturbations, $u' \in H^1(\Omega) : \psi|_{\Gamma_{\mathrm{HB}} \cup \Gamma_W} = 0$, are given by

$$u'_{(m)} = \left(\nabla f_0(u_{(m)})\right)^{-1} f_0(u_{(m)}), \tag{3.22}$$

where $\nabla f_0(u_{(m)})$ is the Jacobian matrix of f_0 at $u_{(m)}$. Following the approach presented in [38], the evaluation of the inverse of $\nabla f_0(u_{(m)})$ is avoided by solving

$$\nabla f_0(u_{(m)}) u'_{(m)} = \lim_{\tau \to 0} \left(f_0(u_{(m)} + \tau u'_{(m)}) - f_0(u_{(m)})\right) = -f_0(u_{(m)}), \tag{3.23}$$

where τ is the step size. For the problem under consideration, this becomes

$$\begin{aligned} \lim_{\tau \to 0} \Bigg(\tau \frac{\partial u'_{(m)}}{\partial t} &- \nabla \cdot \left(a(u_{(m)} + \tau u'_{(m)}) \nabla \left(u_{(m)} + \tau u'_{(m)}\right)\right) \\ &+ \nabla \cdot \left(a(u_{(m)}) \nabla u_{(m)}\right) \Bigg) = -f_0(u_{(m)}). \end{aligned} \tag{3.24}$$

After applying the HB assumption on u', inserting (3.2) for a into (3.24) and since,

$$\lim_{\tau \to 0} \left(a_1 + a_2 \left(u_{(m)} + \tau u'_{(m)}\right)^{a_3}\right) = a_1 + a_2 \left(u_{(m)}^{a_3} + a_3 u_{(m)}^{a_3 - 1} u'_{(m)}\right), \tag{3.25}$$

the equation simplifies to

$$\begin{aligned} \sum_{i=1}^{2n+1} u'^{(i)}_{(m)} c_{ij} - \nabla \cdot \Bigg(a_2 a_3 \left(u_{(m)}^{a_3 - 1}\right)^{(j)} \nabla u_{(m)}^{(j)} u'^{(j)}_{(m)} & \\ + \left(a_1 + a_2 \left(u_{(m)}^{a_3}\right)^{(j)}\right) \nabla u'^{(j)}_{(m)} \Bigg) &= -f_0(u_{(m)}^{(j)}), \quad j = 1, 2, \ldots, 2n+1. \end{aligned} \tag{3.26}$$

3.2.4 Weak and Matrix Form of the Linearised Problem

Let

$$\Psi = \left\{\psi \in H^1(\Omega) : \psi\big|_{\Gamma_{\mathrm{HB}}\cup\Gamma_W} = 0\right\} \tag{3.27}$$

be the space of axial velocity perturbations test functions ψ. Testing Eq. (3.26) and integrating by parts results in

$$\begin{aligned}\sum_{i=1}^{2n+1}\left(\left(\int_\Omega \psi_h u'^{(i)}_{(m),h}\right)c_{ij}\right) + \int_\Omega \nabla\psi_h\cdot a_2a_3\left(u^{a_3-1}_{(m)}\right)^{(j)}\nabla u^{(j)}_{(m)}u'^{(j)}_{(m),h}\\ + \int_\Omega \nabla\psi_h\left(a_1+a_2\left(u^{a_3}_{(m)}\right)^{(j)}\right)\cdot\nabla u'^{(j)}_{(m),h} = -\int_\Omega \psi_h f_0(u^{(j)}_{(m)}), \quad j=1,2,\dots,2n+1,\end{aligned} \tag{3.28}$$

where u' and ψ are confined to a FEM subspace of Ψ. Lastly, the problem of finding $u'_{(m)}$ can be written as

$$\left((C^T_{ij} \otimes A_{lk}) + \left(\mathrm{I}_{2n+1}\bar{\otimes}B(u_{(m)})_{lk}\right)\right)\mathrm{vec}(\mathcal{U}'_{ki,(m)}) = \mathrm{vec}(F_0(u_{(m)})_{lj}), \tag{3.29}$$

where A_{lk} is similar to Subsec. 3.2.1, $\mathrm{vec}(F_0(u_{(m)})_{lj})$ contains the residuals of the (m)-th iteration and the entries of B_{lk} are given by

$$\begin{aligned}B(u_{(m)})^{(j)}_{lk} = \sum_{k=1}^{n_x}\int_\Omega \nabla\psi_l\cdot a_2a_3\left(u^{a_3-1}_{(m)}\right)^{(j)}\nabla u^{(j)}_{(m)}\psi_k + \nabla\psi_l\left(a_1+a_2\left(u^{a_3}_{(m)}\right)^{(j)}\right)\cdot\nabla\psi_k,\\ l=1,2,\dots,n_x, \quad j=1,2,\dots,2n+1.\end{aligned} \tag{3.30}$$

3.3 Implementation and Numerics

For the linear and nonlinear test problems, a custom FEM implementation in Matlab [39] is utilized. This code uses first-order elements and the resulting integrals are evaluated using Gaussian quadrature. A uniform Cartesian grid is employed and the maximum number of DOFs is limited so that the computation can be performed sequentially. Choosing Matlab offers the advantage of using the built-in "mldivide" [39] function for solving the obtained linear systems of equations. Consequently, no specification of the numerical solver or preconditioner is required and "mldivide" returns the solution of the linear problem in one solve call. The following algorithm based on the Newton method introduced in the previous subsections is used for the nonlinear case.

Algorithm 1 Newton Algorithm for Harmonic Balance Heat Equation.

```
Input: $\mathcal{U}_{ki,(0)}, n$                                        ▷ Initial guess $\mathcal{U}_{ki,(0)}$ and harmonics
Output: $\mathcal{U}_{ki,(k)}$                                           ▷ Temperature at last iteration $k$
procedure NewtonLoop($\mathcal{U}_{ki,(0)}, n$)
k ← 0
while convergence criterion is not met do
    $B(u)^{(j)}_{lk} \leftarrow$ Given $\mathcal{U}_{ki,(k)}$                         ▷ Update $B(u)^{(j)}_{lk}$, see (3.18)
    $F_0(u_{(k)})_{lj} \leftarrow$ Given $\mathcal{U}_{ki,(k)}, B(u)^{(j)}_{lk}, C_{ij}, A_{lk}$        ▷ Solve (3.21)
    $B(u_{(k)})_{lk} \leftarrow$ Given $\mathcal{U}_{ki,(k)}$                         ▷ Update $B(u_{(k)})$, see (3.30)
    $\mathcal{U}'_{ki,(k)} \leftarrow$ Given $F_0(u_{(k)})_{lj}, B(u_{(k)})_{lk}, C_{ij}, A_{lk}$     ▷ Solve (3.29)
    $\mathcal{U}_{ki,(k+1)} \leftarrow \mathcal{U}_{ki,(k)} + \tau\mathcal{U}'_{ki,(k)}$                       ▷ Update $\mathcal{U}_{ki}$
    $k \leftarrow k+1$
end while
end procedure
```

An initial guess of 0 was found to be adequate for the linear as well as the nonlinear case. The convergence criterion is defined as

$$\frac{\sqrt{\sum_{k=1}^{n_x(2n+1)}\left(u_{(m+1),k}-u_{(m),k}\right)^2}}{\sqrt{\sum_{k=1}^{n_x(2n+1)}u_{(m),k}^2}} < 10^{-10}. \tag{3.31}$$

The coupling of the Newton method with a different approach, such as a Fixed-Point-Iteration, is possible. This methodology is not required for the inspected problems as an initial guess of zero ensures good convergence of the Newton method.

3.4 Results and Discussion

3.4.1 Validation against Euler Method

The linear test cases H1 and H2 are used to validate the HB time discretization against an Euler method defined as [40]:

$$\frac{\partial u}{\partial t} = \frac{u(t+\tau_t)-u(t)}{\tau_t} \tag{3.32}$$

where τ_t is the time step size. For the inspected problem, the matrices from Eq. (3.12) can be used to solve

$$(A_{lk}+\tau_t B_{lk})\,\mathcal{U}(t+\tau_t)_{ki}\mathrm{I}_{ij} = \tau_t F_{lj} + A_{lk}\mathcal{U}(t)_{ki}\mathrm{I}_{ij}, \tag{3.33}$$

where only F_{lj} is changed and now contains the time derivative of the essential BCs. Two values for τ_t, 10^{-3} and 10^{-4} s, are tested and an initial guess of 0 K is used for $u(t=0)$.

The L^2 error, L^2_e, defined as the temporal integral of the integrated difference in L^2 norms,

$$L^2_e := \int_T \sqrt{\int_\Omega \left(u_{\mathrm{HB},k}-u_{\mathrm{Eu},k}\right)^{(i)2}} \tag{3.34}$$

and H^1 error, H_e^1,

$$H_e^1 := \int_T \sqrt{\int_\Omega \left(u_{\mathrm{HB},h} - u_{\mathrm{Eu},h}\right)^{(i)2} + \int_\Omega \left(\nabla u_{\mathrm{HB},h} - \nabla u_{\mathrm{Eu},h}\right)^{(i)2}} \tag{3.35}$$

are used to measure the discrepancy between the HB and Euler solution, u_{HB} and u_{Eu}, in the discrete space. Linear interpolation is used to map the Euler results onto the HB time instances. Table 3.2 summarizes the employed definitions of a.

Table 3.2: Summary of the thermal diffusivity definitions.

H1 and H2	H3 and H4
$a = 1\,\mathrm{m}^2/\mathrm{s}$	$a_1 = 2\,\mathrm{m}^2/\mathrm{s}$, $a_2 = 1\,\mathrm{K}^{-1}$, $a_3 = 3$

Figure 3.2 depicts the findings for the cases H1 (a) and H2 (b) with 1 (red), 5 (blue) and 10 (gold) harmonics plotted against the mesh size, h. Based on an initial assessment, low L_e^2 and H_e^1 are found regardless of the case, number of harmonics and time step, demonstrating a general agreement between the HB and Euler methods. For all settings except H2 with $n = 1$, a clear error reduction between the two approaches is found as τ_t decreases, indicating that the two methods converge to a similar result in time. When varying the mesh size, both errors remain constant after a minor initial variation, implying that the HB and Euler discretization converge similarly in space.

Upon closer examination, the order of the errors remains constant when switching from H1 to H2, with n set to 5 or 10. For H1, where the forced BC is a simple sine wave, one harmonic appears to capture the system's dynamics well. As a result, the temporally integrated errors grow when increasing n. On the contrary, $n = 1$ yields the highest error when the system is forced by $\sin(\omega t)^2$ in time. In this case, one harmonic cannot capture the system's behavior, leading to higher errors. Based on the results, five harmonics give the lowest total error and seem sufficient for describing the system.

It is also important to note that although choosing a lower number of harmonics results in less obtained time instances, a user-defined time resolution can be reached for any prescribed n. The values of the functions in the HB set, $\mathcal{S}$, are known for each time instance and every point in space adheres to this assumed prior knowledge. Consequently, the individual sine and cosine coefficients can easily be obtained in postprocessing. This is especially useful in cases such as H1, where a low number of harmonics gives accurate results.

Overall, the findings exhibit expected trends and the HB is successfully validated.

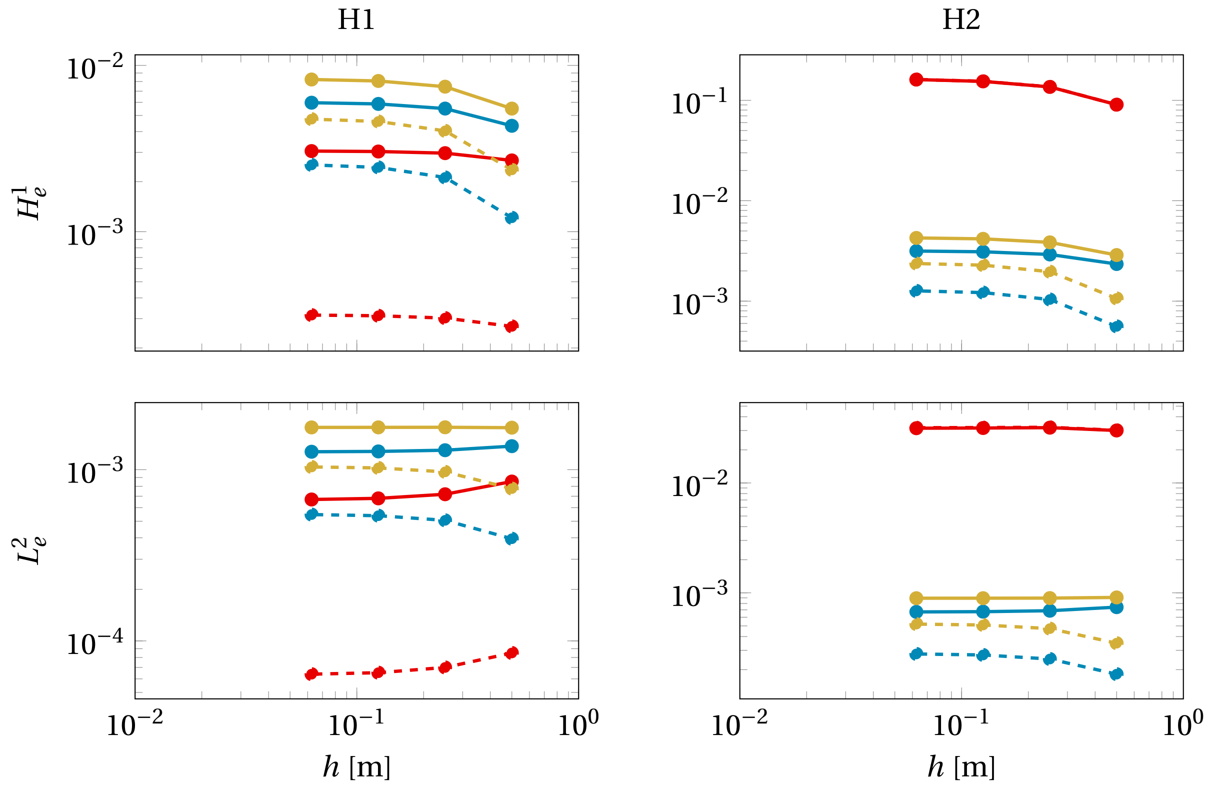

Figure 3.2: L_e^2 and H_e^1 between HB and Euler time discretization plotted against mesh size for cases H1 (left) and H2 (right). Number of harmonics set to 1 (red line), 5 (blue line) and 10 (golden line). Numerical time step set to 10^{-3} s (full line) and 10^{-4} s (dashed line).

3.4.2 Convergence Rates

After validating the HB method against the Euler method, the convergence rate reached with decreasing mesh size is tested for the different cases. Based on the lowest combined errors shown in Fig. 3.2, n is set to one for the linear and five for the nonlinear problems. Due to the lack of an analytical solution, the errors shown in Tab. 3.3 are calculated using the difference between the previous solution and the previous mesh size. As expected, both errors become smaller as the mesh size decreases. The optimum convergence rate of 2 in L_e^2 for linear elements appears to be matched in all cases. In H_e^1, the theoretical convergence rate is equal to 1. H1 and H2 show a rate of more than 1 for coarser resolutions, which decays to the expected value of approximately 1 as the mesh gets refined. This might be caused by the coarse starting mesh sizes, which potentially introduce high errors. Both nonlinear cases start at a lower rate before surpassing the optimum of 1. This could be attributed to the inspected mesh resolutions and is expected to decay at a lower rate with an increasing cell count. In summary, reasonable spatial convergence rates are found for all four settings.

Table 3.3: L^2_e and H^1_e for H1-H4 with varying h.

	H1		**H2**		**H3**		**H4**	
h [m]	L^2_e	H^1_e	L^2_e	H^1_e	L^2_e	H^1_e	L^2_e	H^1_e
0.5								
	3.4E-2	8.9E-2	3.8E-2	9.4E-2	2.7E-2	6.5E-2	2.5E-2	6.1E-2
0.25								
	8.7E-3	3.1E-2	9.7E-3	3.3E-2	1.1E-2	5.3E-2	1.1E-2	5.4E-2
0.125								
	2.2E-3	1.1E-2	2.5E-3	1.1E-2	2.8E-3	2.0E-2	2.7E-3	1.8E-2
0.0625								
	5.5E-4	3.9E-3	6.2E-4	4.0E-3	7.1E-4	7.7E-3	6.7E-4	5.9E-3
0.03125								

4 Diffuse Domain Pulsatile Pipe Flow

This chapter focuses on extending the HB approach towards a pipe flow. In addition, a new type of geometry representation, referred to as diffuse domain, will be introduced before moving on to solving the problem.

4.1 Diffuse Domain Approach

In the work of Kontogiannis et al. [7, 13], the domain is treated as an unknown in the inverse Navier-Stokes problem. As a result, the geometry is evolving during the solving process. So far, a cut-cell formulation addresses this issue, where the cells intersected by the boundary are cut to fit the domain. This method introduces complexity when dealing with higher order finite elements and requires a fine uniform cartesian grid leading to a high cell count [7, 13]. The diffuse domain approach, coupled with local refinement introduced in Subsec. 4.4.1, represents one possible pathway for dealing with these drawbacks. For the diffuse domain method a background tessellation, $\mathcal{T}$ with a boundary $\partial\mathcal{T}$, enclosing the physical domain of interest Ω is provided. The underlying idea is to multiply the problem with a phase field function Φ, which represents Ω by taking on a value of 1 in Ω and approaching 0 outside of Ω [41, 42].

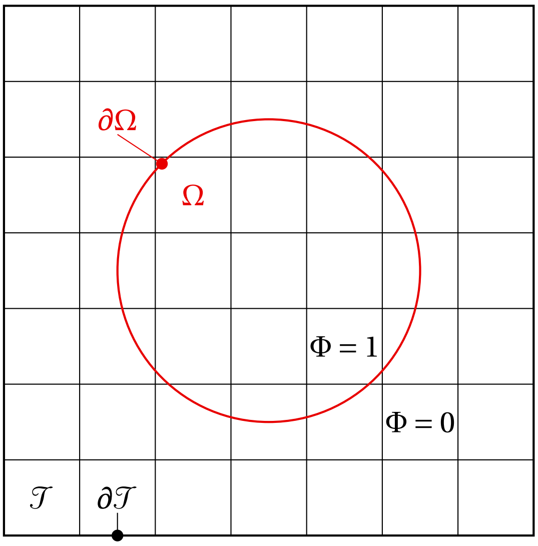

Figure 4.1: Diffuse Domain approach. Domain of interest (red) described by Φ and immersed into background tessellation (black).

By representing the physical domain using Φ, the governing equation can now be solved on $\mathcal{T}$ instead of Ω [41, 42]. An example depicting a circular domain (red) inside a background mesh (black) is shown in Fig. 4.1.

Similar to [41, 42], Φ is defined as

$$\Phi(x,t) := \frac{1}{2}\left(1 - \tanh\left(3\frac{\mathfrak{r}(x,t)}{\epsilon}\right)\right) \quad \text{in}\, \mathbb{T} \times \mathcal{T}, \tag{4.1}$$

where ϵ is a numerical parameter, which is based on the minimum mesh size h_{min}:

$$\epsilon = \alpha h_{\text{min}}, \tag{4.2}$$

where α is a user-defined parameter, which adjusts ϵ. Further, $\mathfrak{r}(x,t)$ is the signed distance function which measures the normal distance of a point to $\partial\Omega$ and is negative inside Ω. Consequently, the sharp boundary is replaced by a thin diffusive boundary region of width 2ϵ [41]. In the forward problem $\mathfrak{r}(x,t)$ is a known input parameter for Eq. (4.1).

Although all required quantities have been formally introduced, the task of actually applying them is still open. Since this work focuses on the HB method, the approach presented in [42] is employed and Φ is introduced into the already derived weak form by adjusting the integrals. While domain integrals become

$$\int_{\Omega} \psi u = \int_{\mathcal{T}} \Phi \psi u, \tag{4.3}$$

boundary type integrals now read

$$\int_{\partial\Omega} \phi u = \int_{\mathcal{T}} |\nabla\Phi| \phi u. \tag{4.4}$$

The interested reader is referred to [41–43] for a more thorough explanation of the diffuse domain method.

4.2 Problem Definition and Analytical Solution

Let $\mathcal{T}$ be a uniform mesh with boundary $\partial\mathcal{T}$ consisting of the shape regular simplices K, which cover the computational domain $[-3,3] \times [-3,3]$. Further, let Ω be an open set defined by

$$\Omega = \left\{(x_1, x_2, t) \in \mathbb{T} \times \mathbb{R}^2 \mid x_1^2 + x_2^2 < 1\right\}$$

and $\partial\Omega$ given by

$$\partial\Omega = \Gamma_{\text{W}} = \left\{(x_1, x_2, t) \in \mathbb{T} \times \mathbb{R} \mid x_1^2 + x_2^2 = 1\right\}$$

be the boundary of Ω. Zero Dirichlet BCs, $g_W(t,x) = 0 : \mathbb{T} \times \Gamma_W \to \mathbb{R}$, are imposed on Γ_W. Furthermore, u, which is the axial velocity, is forced by a spatially uniform T-periodic function $f_{\text{HB}}(t) = f_{\text{HB}}(t + nT) : \mathbb{T} \times \mathcal{T} \to \mathbb{R}$ with $n \in \mathbb{N}$ and the inspected time interval is $\mathbb{T} := [0, T]$.

Finally, the problem reads:

$$\begin{cases} \rho \frac{\partial u}{\partial t} - \nabla \cdot \left(\mu \nabla u \right) = f_{\mathrm{HB}} & \text{in } \mathbb{T} \times [\Omega \subset \mathcal{T}], \\ u = 0 & \text{on } \mathbb{T} \times [\Gamma_W \subset \mathcal{T}], \end{cases} \tag{4.5}$$

where ρ is the density and μ the viscosity, which are both set to unity. Figure 4.1 shows the examined geometry and $\imath$ can be calculated via

$$\imath(x_1, x_2, t) = \sqrt{x_1^2 + x_2^2} - 1 \quad \text{in} \, \mathbb{T} \times \mathcal{T}. \tag{4.6}$$

An analytical solution for (4.5) describing the flow in a circular pipe forced by a periodic pressure gradient was introduced in the work of Womersley [44]. Following the derivation shown in [45], the solution for the axial velocity can be written as

$$u(r,t) = \mathcal{R}\left(\sum_{n=0}^{\infty} \frac{i P_n'}{\rho n \omega} \left(1 - \frac{J_0\left(n^{1/2}\, i^{3/2} \mathrm{Wo}\, \frac{r}{R}\right)}{J_0\left(n^{1/2}\, i^{3/2} \mathrm{Wo}\right)} \right) e^{i n \omega t} \right), \quad n \in \mathbb{N}^+. \tag{4.7}$$

Here, $r = x_1^2 + x_2^2$ is the radial coordinate, R is the radius of the pipe and P_n' is the pressure gradient magnitude of each harmonic. Further, $\mathcal{R}(\bullet)$ is the real part of $\bullet$, i the imaginary number and J_0 the Bessel function of first kind and order zero.

Lastly, Wo denotes the Womersley number, which is defined as

$$\mathrm{Wo} := R \left(\frac{\omega \rho}{\mu} \right)^{1/2} = R \left(\frac{\omega}{\nu} \right)^{1/2}, \tag{4.8}$$

where ν is the kinematic viscosity. Wo describes the transient inertial to viscous forces ratio and provides insights into the flow behavior. The flow develops a parabolic velocity profile for low Wo, which is in phase with the forcing. In the case of high Wo, a phase lag between u and the forcing is present and the axial velocity profile is flattened [45, 46].

In the current work, the periodic forcing term, which represents the negative pressure gradient, is set to

$$f_{\mathrm{HB}}(t) = \sin(\omega t). \tag{4.9}$$

Subsequently, the pressure gradient sum in Eq. (4.7) collapses to a single harmonic:

$$\sum_{n=0}^{\infty} P_n' e^{i n \omega t} = -f_{\mathrm{HB}}(t) = i\, e^{i \omega t} \frac{\mathrm{Pa}}{\mathrm{m}}. \tag{4.10}$$

Moreover, R and ν are set to unity and thus will be dropped due to readability reasons. Consequently, Wo solely depends on ω via

$$\mathrm{Wo} = \omega^{1/2} \mathrm{s}^{1/2} \tag{4.11}$$

and by inserting Eq. (4.10) into (4.7) following analytical solution is found:

$$u(r,t) = \mathcal{R}\left(\frac{-1}{\omega} \left(1 - \frac{J_0\left(i^{3/2} \mathrm{Wo}\, r\, \mathrm{m}^{-1}\right)}{J_0\left(i^{3/2} \mathrm{Wo}\right)} \right) e^{i \omega t} \frac{\mathrm{m}}{\mathrm{s}^2} \right). \tag{4.12}$$

4.3 Discrete Weak Form

After defining the examined test case, the discrete weak form is now derived. Throughout the following section, ρ and μ are neglected and the inner product $\langle \bullet, \bullet \rangle$ is used to enhance the readability of the integrals. The section is based on the steady-state weak form shown in [42].

4.3.1 Diffuse Domain Weak Form

To find the diffuse domain weak form, Eq. (4.5) is tested with $\psi : \{\psi \in H^1(\mathcal{T})\}$ and integrated over Ω. Subsequently, integration by parts is performed to find:

$$\left\langle \psi, \frac{\partial u}{\partial t} \right\rangle_{\Omega} + \langle \nabla\psi, \nabla u \rangle_{\Omega} - \langle \psi, \nabla u \cdot \boldsymbol{n} \rangle_{\Gamma_{\mathrm{W}}} = \langle \psi, f \rangle_{\Omega}. \tag{4.13}$$

Similar to [41, 42], Φ is inserted and the integration domain is changed to $\mathcal{T}$ resulting in

$$\left\langle \Phi\psi, \frac{\partial u}{\partial t} \right\rangle_{\mathcal{T}} + \langle \Phi\nabla\psi, \nabla u \rangle_{\mathcal{T}} - \underbrace{\langle \Phi\psi, \nabla u \cdot \boldsymbol{n} \rangle_{\partial\mathcal{T}}}_{=0} - \langle |\nabla\Phi|\, \psi, \nabla u \cdot \boldsymbol{n} \rangle_{\mathcal{T}} + bc_{\mathrm{W}} = \langle \Phi\psi, f \rangle_{\mathcal{T}}. \tag{4.14}$$

Here, the boundary integral term is neglected since Φ tends to zero outside of Ω and the essential Dirichlet BC is accounted for by bc_W, which is explained in the next subsection.

4.3.2 Nitsche Boundary Condition

Following [7, 13, 42], the essential BCs on the wall are enforced using Nitsche's method [47], which introduces two additional terms to the weak form. The first one ensures symmetry in the bilinear form, while the second one accounts for the BC [13]:

$$bc_W := \langle \underbrace{- |\nabla\Phi|\, \nabla\psi \cdot \boldsymbol{n}}_{\text{symmetry}} + \underbrace{|\nabla\Phi|\, \gamma_{\mathrm{N}} \epsilon^{-3} \psi}_{\text{penalty}}, \underbrace{u - g_{\mathrm{W}} - z\nabla u \cdot \boldsymbol{n}}_{\text{BC expansion}} \rangle_{\mathcal{T}}, \tag{4.15}$$

where γ_{N} is a numerical parameter and a first-order Taylor expansion of the BC is used [41, 42].

4.3.3 Harmonic Balance and Matrix Form

Similar to the previous chapter, the final step now introduces the HB time discretization into Eq. (4.14). After confining u as well as ψ to a n_x-dimensional FEM subspace, the expression reads

$$\begin{aligned} \sum_{i=1}^{2n+1} \left\langle \Phi\psi_h, u_h^{(i)} \right\rangle_{\mathcal{T}} c_{ij} + \left\langle \Phi\nabla\psi_h, \nabla u_h^{(j)} \right\rangle_{\mathcal{T}} - \left\langle |\nabla\Phi|\, \psi_h, \nabla u_h^{(j)} \cdot \boldsymbol{n} \right\rangle_{\mathcal{T}} \\ + bc_{\mathrm{W},h}^{(j)} = \left\langle \Phi\psi_h, f^{(j)} \right\rangle_{\mathcal{T}}, \quad j = 1, 2, \ldots, 2n+1. \end{aligned} \tag{4.16}$$

As a result, the following FEM and HB matrices are obtained:

$$A_{lk}\mathcal{U}_{ki}C_{ij} = \sum_{i=1}^{2n+1} \left\langle \Phi, \psi_h u_h^{(i)} \right\rangle_{\mathcal{T}} c_{ij}, \quad k,l = 1,2,\dots,n_x, \quad j = 1,2,\dots,2n+1, \tag{4.17a}$$

$$\begin{aligned} B_{lk}\mathcal{U}_{ki}\mathrm{I}_{ij} = {} & \left\langle \Phi\nabla\psi_h, \nabla u_h^{(j)} \right\rangle_{\mathcal{T}} - \left\langle |\nabla\Phi|\,\psi_h, \nabla u_h^{(j)} \cdot \boldsymbol{n} \right\rangle_{\mathcal{T}} - \left\langle |\nabla\Phi|\nabla\psi_h \cdot \boldsymbol{n}, u_h^{(j)} \right\rangle_{\mathcal{T}} \\ & + \left\langle |\nabla\Phi| \frac{\gamma_\mathrm{N}}{\epsilon^3} \psi_h, u_h^{(j)} \right\rangle_{\mathcal{T}} + \left\langle |\nabla\Phi|\,\psi_h \cdot \boldsymbol{n}, \mathfrak{r}\nabla u_h^{(j)} \cdot \boldsymbol{n} \right\rangle_{\mathcal{T}} \\ & - \left\langle |\nabla\Phi| \frac{\gamma_\mathrm{N}}{\epsilon^3} \psi_h, \mathfrak{r}\nabla u_h^{(j)} \cdot \boldsymbol{n} \right\rangle_{\mathcal{T}}, \quad k,l = 1,2,\dots,n_x, \quad j = 1,2,\dots,2n+1, \end{aligned} \tag{4.17b}$$

$$\begin{aligned} F_{lj} = {} & \left\langle \Phi\psi_h, f^{(j)} \right\rangle_{\mathcal{T}} + \langle |\nabla\Phi| g_W, \nabla\psi_h \cdot \boldsymbol{n} \rangle_{\mathcal{T}} \\ & + \left\langle |\nabla\Phi| \frac{\gamma_\mathrm{N}}{\epsilon^3} \psi_h, g_W \right\rangle_{\mathcal{T}}, \quad l = 1,2,\dots,n_x, \quad j = 1,2,\dots,2n+1. \end{aligned} \tag{4.17c}$$

Finally, the linear system of equations can be rewritten as

$$\Big((C_{ij}^T \otimes A_{lk}) + (\mathrm{I}_{2n+1} \otimes B_{lk}) \Big) \mathrm{vec}(\mathcal{U}_{ki}) = \mathrm{vec}(F_{lj}). \tag{4.18}$$

To solve the equation efficiently, an approximation of the system matrix $\tilde{K}$, which is used for the preconditioning, is defined as:

$$\tilde{K} := \mathrm{I}_{2n+1} \otimes B_{lk}, \tag{4.19}$$

whose inverse is given by

$$\tilde{K}^{-1} = \mathrm{I}_{2n+1} \otimes B_{lk}^{-1}. \tag{4.20}$$

This preconditioning approach relies on the idea that most of the system's stiffness is introduced by $\mathrm{I}_{2n+1} \otimes B_{lk}$. Since the definition describes the steady state limit, this preconditioner naturally performs better in the regime of lower Wo numbers.

4.4 Implementation and Numerics

4.4.1 Adaptive Mesh Refinement

As stated in Sec. 4.1, the mesh is locally refined in regions close to $\partial\Omega$. In comparison with cut-cell formulations, such as [7, 13, 48], this method avoids the requirement of a fine background mesh. The refinement approach is briefly summarized in the following paragraph.

According to Fig. 4.2, the cell size is reduced by a factor of 2 in each dimension for every level of additional refinement. The level of refinements between the background mesh and the boundary regions are denoted with n_r and treated as a user-defined input parameter. Knowledge of h_max and n_r allows for the calculation of h_min as:

$$h_\mathrm{min} = \frac{h_\mathrm{max}}{2^{n_r}}. \tag{4.21}$$

Finally, for $i = 1,2,..,n_r$ times a check, if the distance between the cell center and the boundary is smaller than $\frac{1}{i}\epsilon\gamma_{\epsilon}$, is performed for all cells. After every loop, the simplices with a cell center inside the threshold are refined once. Figure 4.2 depicts the refinement process for $n_r = 2$.

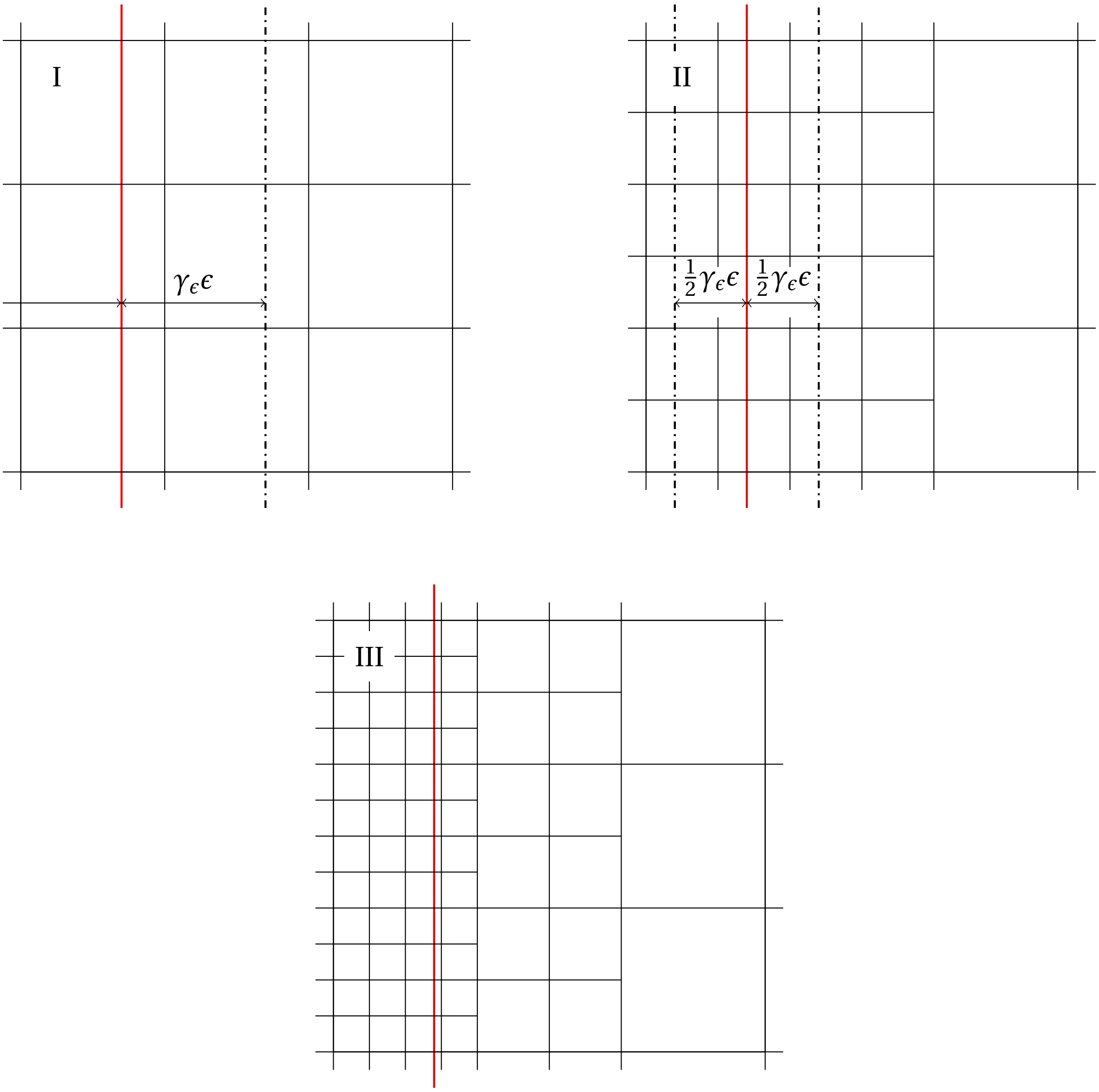
I
$\gamma_{\epsilon}\epsilon$
II
$\frac{1}{2}\gamma_{\epsilon}\epsilon$
$\frac{1}{2}\gamma_{\epsilon}\epsilon$
III

Figure 4.2: Employed adaptive mesh refinement strategy for two levels of additional refinement. Mesh (black) is refined close to $\partial\Omega$ (red). I: initial background mesh, II: mesh after first refinement circle and III: fully refined mesh.

4.4.2 Implementation and Numerical Setup

The HB diffuse domain solver is implemented using the C++ FEM library deal.II [49], PETSC [50–52] and p4est [53, 54], which allow for parallel computations. First-order elements are used for the axial velocity. The solver developed during this work is based on a steady-state diffuse domain Poisson solver by Matthias Henssler [42]. It is important to note that the unstable numerical behavior encountered in [42] was not observed and hence, the additional numerical stabilization is removed.

To minimize memory usage, only the individual FEM matrices are stored, and PETSc [50–52] matrix-vector operators are utilized to compute the Kronecker products in a matrix-free way. The system, see Eq. (4.18), is solved using FGMRES [55] and a custom preconditioner is employed to approximate the inverse of the system matrix. Due to the matrix-free approach, $2n+1$ solve calls are required to invert the custom preconditioner. Each of these inversions is computed using GMRES [56] and an additive Schwarz method (ASM) [57] preconditioner. The bandwidth of the Kronecker product matrices is improved by assigning the k-th DOF the global indices $k, k+1, k+2, ...k+2\,n$ instead of $k, n_x\,k, 2n_x\,k, ..., 2\,n\,n_x\,k$.

4.5 Results and Discussion

4.5.1 Velocity Profiles

The derived formulation and corresponding formulation are evaluated on a Womersley flow. The analysis includes three different cases characterized by a Wo of 10, 1, and 0.1, respectively. These specific numbers are reached by setting $\omega = 100$, 1 and 0.01 1/s according to Eq. (4.9). Based on the prior knowledge of the problem given by the analytical solution, the number of harmonics, n, is set to 1. Further, α is set to 8 and γ_N is set to $1/|\nabla\Phi|_\mathrm{max}$. This normalization by the maximum norm of the gradient follows the methodology in [42], where the penalty is prevented from dominating the system as the mesh is refined and $|\nabla\Phi|$ increases.

Figure 4.3 depicts one obtained velocity profile with $h_\mathrm{min} = 4.88281\cdot10^{-3}$ m and Wo = 1 at the first time instance, $t = T/3$ s. Additionally, the boundary of the domain is highlighted by the white line. The velocity profiles adhere to the prescribed domain and adjust to 0 m/s outside Ω. This indicates that the diffuse domain method from [42] still functions as intended. The chosen Wo of 1 lies within the quasi-static regime and the solution clearly shows a parabolic profile associated with a steady state solution with spatial uniform forcing [41, 45]. A peak velocity of around 0.23 m/s is observed, which corresponds to a maximum forcing amplitude of 1 Pa/m.

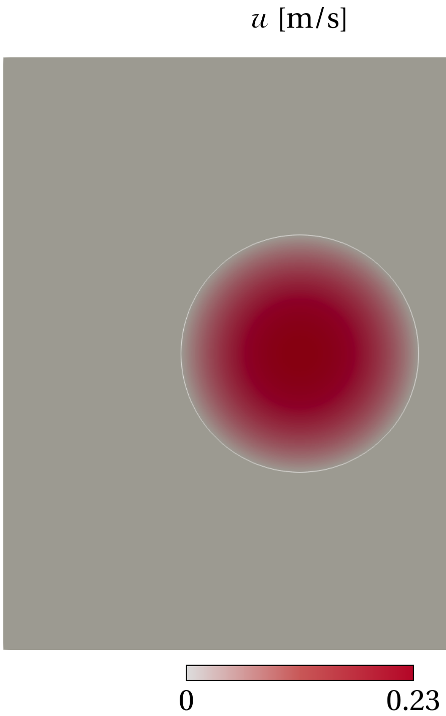

Figure 4.3: Axial velocity profile and domain boundary (white line) for Wo = 1 at $t = T/3$ s.

In order to compare the numerical findings with the analytical solution, line plots along the x_1-axis are used. Figure 4.4 shows the velocity profiles (red dashed line) for different values of Wo at $t = 2T/3$ s with $h_{\text{min}} = 4.88281 \cdot 10^{-3}$ m. A good agreement with the exact solution (black line) is found for all values of Wo, as almost no deviations are visible. This observation holds for all time instances and all Wo, which are shown in Fig. A.1, A.2 and A.3 in the appendix. There is a slight error for $t = 2T/3$ s at Wo = 10, which is expected to decrease with smaller mesh sizes.

The change of the flow behavior with varying Wo is also clearly visible in Fig. 4.4. A similar velocity profile is found for the two cases with Wo = 0.1 and 1, which is in phase with the forcing and thus, the steady-state solution. As previously stated, the custom preconditioner assumes a steady state. As a result, only 3 and 6 outer iterations are required for Wo = 0.1 and 1, while 54 iterations are performed for the third setup. The number of inner iterations corresponding to the $2n+1$ inversions of B_{lk} remains around 200-300 regardless of Wo. A second preconditioning method, where knowledge of the solution's structure is utilized to collapse the unsteady terms onto the diagonal of the system matrix, was tested. So far, no improvements have been achieved compared to the steady state preconditioning. Adjusting the preconditioner to reduce the iterations in the case of higher Wo numbers is an interesting starting point for further investigations. Despite this open issue, the diffuse domain solver in [42] has been successfully extended to incorporate the HB.

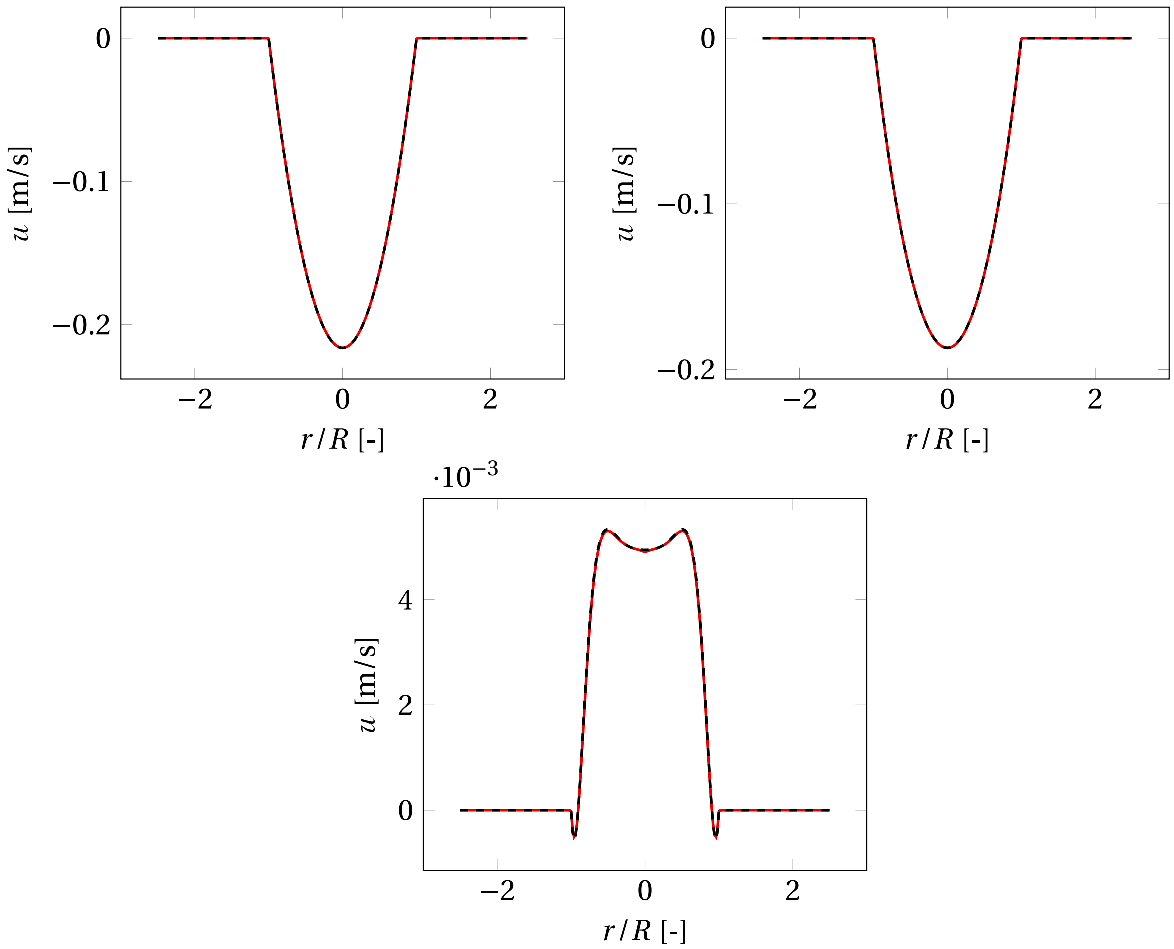

Figure 4.4: Axial velocity result (red line) and analytical solution (black dashed line) plotted along x_1 axis at $t = T/3$ s. Top left: Wo = 0.1, top right: Wo = 1 and bottom: Wo = 10.

4.5.2 Convergence Rates

In this subsection, a spatial convergence test is conducted for two different values of α, 4 and 8, which scale ϵ based on the minimum mesh size. Similar to the previous chapter, the temporal integral of the $2n+1$ errors between the numerical results and the analytical solution is used for the comparison. It is important to note that only the case of Wo = 1 is considered.

According to Fig. 4.5 and 4.6, a higher absolute value for both error types can be observed when setting α to 8. This finding is consistent with the results in [42], where the lower value of ϵ resulting in a smaller transition zone and a more accurate domain representation of the original problem is given as a possible explanation. Furthermore, there is a relation between the slope of the error sum for varying α. With $\alpha = 8$, a slope of approximately 1.5 in L_e^2 is achieved across all examined mesh sizes. On the contrary, for $\alpha = 4$, the convergence does not reach this level, approaching a slope of approximately 1 for the smallest considered mesh sizes. Such a distinct difference is not observed when comparing the H_e^1 results with different

α. The case with $\alpha = 4$ yields a slope of around 0.8, which is near the optimum for linear elements. In Fig. 4.6, a slope slightly higher than 1 is found, which is expected to diminish with further refinement.

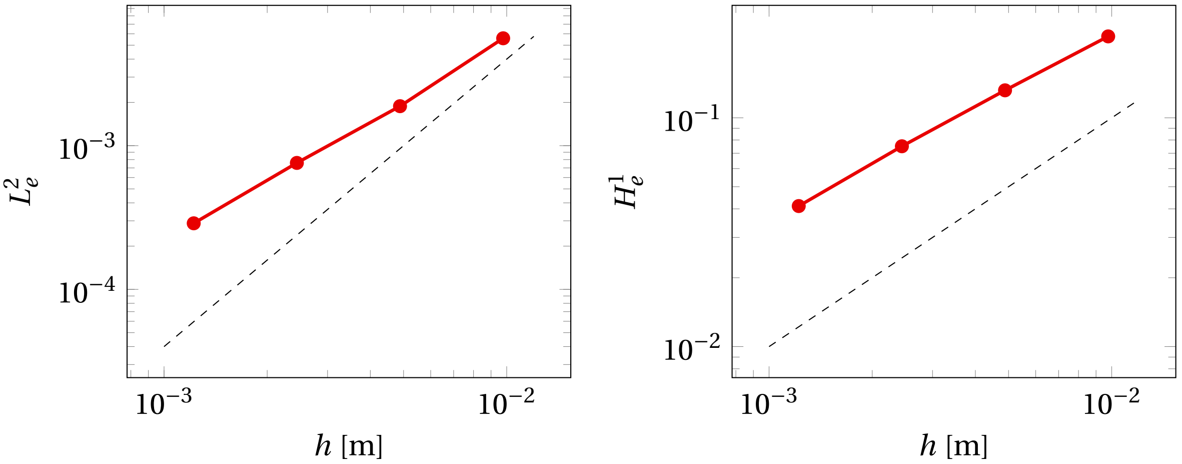

Figure 4.5: L_e^2 and H_e^1 between numerical and analytical solution plotted against mesh size. Left: reference slope of 2 (black dashed line) and right: reference slope of 1 (black dashed line). Diffuse domain parameter α set to 8.

The accuracy and convergence of the two-dimensional diffuse domain HB solver have been validated. Now, the view is shifted to the Stokes problem, which will be investigated in the following chapter.

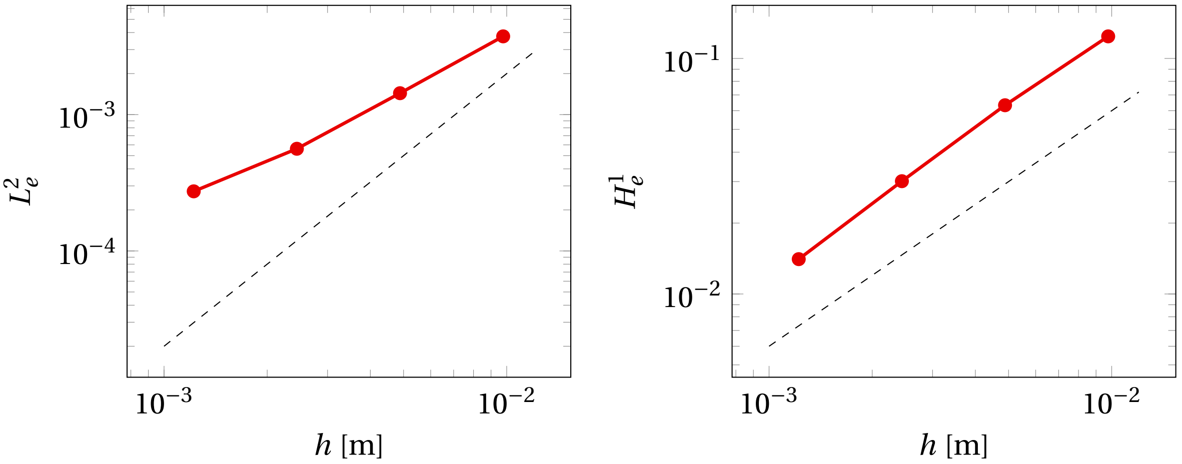

Figure 4.6: L_e^2 and H_e^1 between numerical and analytical solution plotted against mesh size. Left: reference slope of 2 (black dashed line) and right: reference slope of 1 (black dashed line). Diffuse domain parameter α set to 4.

5 Diffuse Domain Pulsatile Stokes Flow

This chapter extends the two-dimensional pipe flow to a three-dimensional problem, which is solved using the Stokes equation.

5.1 Problem Definition

Let $\mathcal{T}$ be a uniform mesh with boundary $\partial\mathcal{T}$ consisting of the cells K, which cover the computational domain $[-10, 10]^3$. Further, let Ω be an open set defined by

$$\Omega = \left\{(x_1, x_2, x_3) \in \mathbb{T} \times \mathbb{R}^3 \,\middle|\, x_1^2 + x_3^2 < 1 \cap |x_2| < 10\right\}.$$

Its boundary $\partial\Omega$ is given by $\partial\Omega = \Gamma_W \cap \Gamma_N$, where the Γ_W is defined as

$$\Gamma_W = \left\{(x_1, x_2, x_3) \in \mathbb{T} \times \mathbb{R}^2 \,\middle|\, x_1^2 + x_3^2 = 1 \cap |x_2| < 10\right\}$$

and the natural boundary part, Γ_N, is defined as

$$\Gamma_N = \left\{(x_1, x_2, x_3) \in \mathbb{T} \times \mathbb{R}^2 \,\middle|\, x_1^2 + x_3^2 < 1 \cap |x_2| = 10\right\}.$$

Zero Dirichlet BCs, $g_W(t, x) = 0 : \mathbb{T} \times \Gamma_W \to \mathbb{R}^2$, are imposed on Γ_W on the velocity $\vec{u} \in \mathbb{T} \times \mathbb{R}^3$ and a natural boundary condition, $g_N(t, x) = -2\nu\nabla^s\vec{u}\cdot\boldsymbol{n} + p\,\boldsymbol{n} = 0 : \mathbb{T} \times \Gamma_W \to \mathbb{R}^2$ is imposed on Γ_N. Here, $\nabla^s \equiv (\nabla^s\vec{u}_{ij} := \frac{1}{2}(\partial_j\vec{u}_i + \partial_i\vec{u}_j)$ is the strain-rate tensor and p denotes the pressure. A spatially uniform T-periodic function $\vec{f}_{\mathrm{HB}}(t) = \vec{f}_{\mathrm{HB}}(t + nT) : \mathbb{T} \times \mathcal{T} \to \mathbb{R}^3$ with $n \in \mathbb{N}$ is applied to the flow and the inspected time interval is $\mathbb{T} := [0, T]$. Consequently, the Stokes problem reads:

$$\begin{cases} \rho\frac{\partial u}{\partial t} - \nabla\cdot\left(2\mu\nabla^s\vec{u}\right) + \nabla p = \vec{f}_{\mathrm{HB}} & \text{in } \mathbb{T} \times [\Omega \subset \mathcal{T}], \\ -\nabla\cdot\vec{u} = \vec{0} & \text{in } \mathbb{T} \times [\Omega \subset \mathcal{T}], \\ \vec{u} = \chi_W g_W = \vec{0} & \text{on } \mathbb{T} \times [\Gamma_W \subset \mathcal{T}], \\ -2\nu\nabla^s\vec{u}\cdot\boldsymbol{n} + p\,\boldsymbol{n} = \chi_N g_N = \vec{0} & \text{on } \mathbb{T} \times [\Gamma_N \subset \mathcal{T}], \end{cases} \tag{5.1}$$

where $\chi_{W/N}$ are characteristic functions.

Remark that in [7, 13] the characteristic functions are strictly defined as unity on the associated boundary and zero elsewhere. When dealing with the diffuse domain method dealing with multiple BCs requires a split of the phase field gradient into $|\nabla\Phi_W|$ and $|\nabla\Phi_N|$, which represent the wall and natural phase field gradient, respectively. For the examined problem, a natural boundary condition with a value of $\vec{0}$ is chosen and the boundary integrals vanish. As a consequence, a slightly adapted version of Eq. (4.6) is used

$$\mathfrak{z}(x_1, x_2, x_3, t) = \sqrt{x_1^2 + x_3^2} - 1 \quad \text{in}\, \mathbb{T} \times \mathcal{T}, \tag{5.2}$$

and the corresponding gradient is sufficient for describing $|\nabla\Phi_W|$. Figure 5.1, which is not drawn to scale, depicts a bottom and side view of the inspected domain (red) within a background mesh (black).

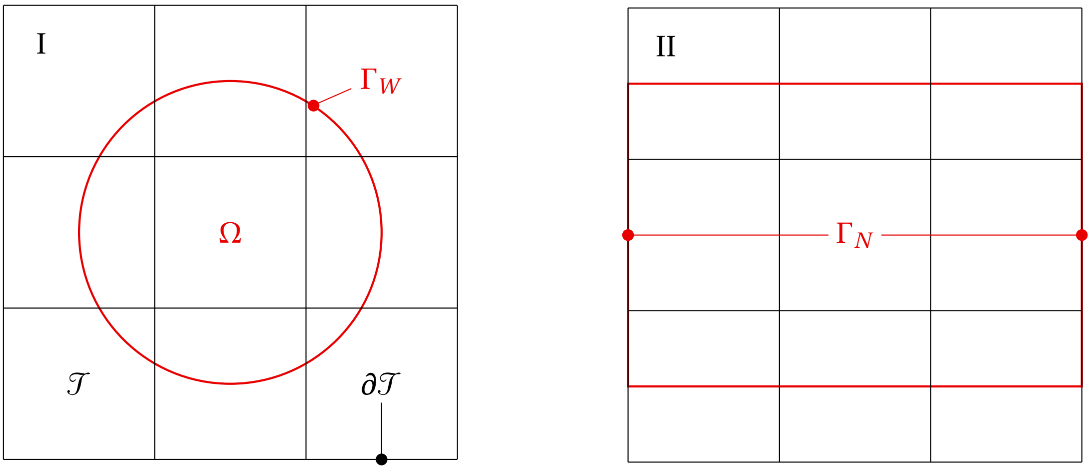

Figure 5.1: Inspected Stokes Problem domain (red) within a background mesh (black). Figure is not drawn to scale. Left: bottom / top view and right: side view.

An adapted forcing function $\vec{f}_{\mathrm{HB}}$, similar to the one of the two-dimensional case in the axial direction and with values of 0 in x_1 and x_3 direction, is applied. Consequently, the analytical solution for the velocity is similar to Eq. (4.7) for the axial velocity in $\vec{u}_a$ in x_2 direction. The radial velocity, $\vec{u}_r$, is given by $\vec{u}_r = \sqrt{\vec{u}_{x_1}^2 + \vec{u}_{x_3}^2} = 0$ in $\mathbb{T} \times \mathbb{R}$.

5.2 Discrete Weak and Matrix Forms

In this section, the matrix form of the problem is derived. Symbols are assigned to the individual integrals, which ensure comprehensibility. To highlight the generality of the approach, vanishing integrals over Γ_N and Γ_W will be included during the weak form derivation. Note that this section builds upon the steady state weak forms in [13, 42]. Throughout the following chapter, ρ and μ are assumed to be unity and will be disregarded for readability.

5.2.1 Diffuse Domain Weak Form

Let

$$\vec{\Psi} = \{\vec{\psi} \in \vec{H}^1(\Omega) : \vec{\psi}|_{\partial\mathcal{T}} = \vec{0}\} \tag{5.3}$$

be the space of velocity and

$$Q = \{q \in L^2(\Omega)\} \tag{5.4}$$

be the space of pressure test functions q.

Testing Eq. (5.1) with $\vec{\psi}$ and q and integrating over Ω gives

$$\begin{aligned}\left\langle \vec{\psi}, \frac{\partial \vec{u}}{\partial t} \right\rangle_\Omega + \langle 2\nabla^s\vec{\psi}, \nabla^s\vec{u}\rangle_\Omega - \langle 2\vec{\psi}, \nabla^s\vec{u}\cdot\boldsymbol{n}\rangle_{\partial\Omega} - \langle \nabla\cdot\vec{\psi}, p\rangle_\Omega + \langle \vec{\psi}\cdot\boldsymbol{n}, p\rangle_{\partial\Omega} \\ + \langle q, \nabla\cdot\vec{u}\rangle_\Omega = \left\langle \vec{\psi}, \vec{f}_{\text{HB}} \right\rangle_\Omega .\end{aligned} \tag{5.5}$$

Remark that as shown in Subsec. 4.3.1 the test functions are not required to equal zero on $\partial\mathcal{T}$. However, according to Fig. 5.1, the examined system is a section of an infinitely long pipe and the boundaries are chosen in a way that $\partial\mathcal{T} \cap \partial\Omega \neq \emptyset$, which implies that integrals over $\partial\mathcal{T}$ do not vanish naturally. Setting the velocity test function to $\vec{0}$ on $\partial\mathcal{T}$ presents a convenient way of removing these unwanted integrals over $\partial\mathcal{T}$, which alternatively could be avoided by simply extending the background mesh in x_2 direction.

The diffuse domain approach is introduced by changing the integration domain to $\mathcal{T}$ and term by term Eq. (5.5) becomes

$$a_I := \left\langle \Phi\vec{\psi}, \frac{\partial \vec{u}}{\partial t} \right\rangle_\mathcal{T} = \left\langle \vec{\psi}, \frac{\partial \vec{u}}{\partial t} \right\rangle_\Omega , \tag{5.6a}$$

$$a_{II} := \langle \Phi 2\nabla^s\vec{\psi}, \nabla^s\vec{u}\rangle_\mathcal{T} = \langle 2\nabla^s\vec{\psi}, \nabla^s\vec{u}\rangle_\Omega , \tag{5.6b}$$

$$b_{p,I} := \langle \Phi\left(\nabla\cdot\vec{\psi}\right), p\rangle_\mathcal{T} = \langle \nabla\cdot\vec{\psi}, p\rangle_\Omega , \tag{5.6c}$$

$$b_{\vec{u},I} := \langle \Phi q, \nabla\cdot\vec{u}\rangle_\mathcal{T} = \langle q, \nabla\cdot\vec{u}\rangle_\Omega , \tag{5.6d}$$

$$j_f := \left\langle \Phi\vec{\psi}, \vec{f}_{\text{HB}} \right\rangle_\mathcal{T} = \left\langle \vec{\psi}, \vec{f}_{\text{HB}} \right\rangle_\Omega , \tag{5.6e}$$

where a_I and a_{II} are velocity-velocity bilinear forms, $b_{p,I}$ and $b_{\vec{u},I}$ are velocity-pressure bilinear forms and j_f is a BC linear form.

After splitting the boundary integrals into the natural and essential parts of $\partial\Omega$, the integral over Γ_N can be combined into one term

$$-\left\langle 2\vec{\psi}, \nabla^s \vec{u} \cdot \boldsymbol{n} \right\rangle_{\partial\Gamma_N} + \left\langle \vec{\psi} \cdot \boldsymbol{n}, p \right\rangle_{\partial\Gamma_N} = \left\langle \vec{\psi}, -2\nabla^s \vec{u} \cdot \boldsymbol{n} + p\boldsymbol{n} \right\rangle_{\partial\Gamma_N} = \left\langle \vec{\psi}, g_N \right\rangle_{\partial\Gamma_N}. \tag{5.7}$$

The diffuse domain formulations of the boundary integrals now read

$$j_N := \left\langle |\nabla\Phi_N| \vec{\psi}, g_N \right\rangle_{\mathcal{T}} = \left\langle \vec{\psi}, g_N \right\rangle_{\partial\Gamma_N}, \tag{5.8a}$$

$$a_{III} := \left\langle |\nabla\Phi_W| 2\vec{\psi}, \nabla^s \vec{u} \cdot \boldsymbol{n} \right\rangle_{\mathcal{T}} = \left\langle 2\vec{\psi}, \nabla^s \vec{u} \cdot \boldsymbol{n} \right\rangle_{\partial\Gamma_W}, \tag{5.8b}$$

$$b_{p,II} := \left\langle |\nabla\Phi_W| \vec{\psi} \cdot \boldsymbol{n}, p \right\rangle_{\mathcal{T}} = \left\langle \vec{\psi} \cdot \boldsymbol{n}, p \right\rangle_{\partial\Gamma_W}, \tag{5.8c}$$

where j_N is a BC linear form.

Finally, following formulation of Eq. (5.5) is found

$$a_I + a_{II} - b_{p,I} + b_{\vec{u},I} + j_N - a_{III} + b_{p,II} + bc_W = j_f, \tag{5.9}$$

where bc_W incorporates the essential BC on the wall.

5.2.2 Nitsche Boundary Condition

Similar to Subsec. 4.3.2 Nitsche's method [47] is used to weakly enforce the essential BC on Γ_W:

$$bc_W := \langle \underbrace{|\nabla\Phi_W| \left(-2\nabla^s \vec{u} \cdot \boldsymbol{n} - q\boldsymbol{n}\right)}_{\text{symmetry}} + \underbrace{|\nabla\Phi_W| \gamma_{\mathrm{N}} \epsilon^{-3} \vec{\psi}}_{\text{penalty}}, \underbrace{\vec{u} - g_{\mathrm{W}}}_{\text{BC}} \rangle_{\mathcal{T}} \tag{5.10}$$

The terms induced by the symmetry part of Eq. (5.10) are accounted for by

$$a_{IV} := \left\langle |\nabla\Phi_W| 2\nabla^s \vec{u} \cdot \boldsymbol{n}, \vec{u} \right\rangle_{\mathcal{T}}, \tag{5.11a}$$

$$b_{\vec{u},II} := \left\langle |\nabla\Phi_W| q\boldsymbol{n}, \vec{u} \right\rangle_{\mathcal{T}}, \tag{5.11b}$$

$$j_{W,I} := \left\langle |\nabla\Phi_W| 2\nabla^s \vec{u} \cdot \boldsymbol{n}, g_W \right\rangle_{\mathcal{T}}, \tag{5.11c}$$

$$j_{W,II} := \left\langle |\nabla\Phi_W| q \cdot \boldsymbol{n}, g_W \right\rangle_{\mathcal{T}} \tag{5.11d}$$

and the terms stemming from the penalty read

$$a_V := \left\langle |\nabla\Phi_W| \frac{\gamma_{\mathrm{N}}}{\epsilon^3} \vec{\psi}, \vec{u} \right\rangle_{\mathcal{T}}, \tag{5.12a}$$

$$j_{W,III} := \left\langle |\nabla\Phi_W| \frac{\gamma_{\mathrm{N}}}{\epsilon^3} \vec{\psi}, g_W \right\rangle_{\mathcal{T}}, \tag{5.12b}$$

where $j_{W,I-III}$ are BC linear forms.

5.2.3 Numerical Stabilization

As reported in [42], unstable numerical behavior can be encountered in regions close to the boundary and outside of Ω, which is assumed to be caused by the diffuse domain method. To counteract this issue, ghost penalties based on the continuous interior penalty (CIP) stabilization, which penalizes jumps in the solution's gradient across neighboring cells, are applied as follows [13, 42].

Let every cell facet, $\mathcal{F}$, outside of the domain be part of the set F_g. Further, define a jump in gradient of a finite element discretized quantity, $||\nabla \bullet_h||$, as

$$||\nabla u_h|| = \nabla u_h|_K \cdot \boldsymbol{n}_{\mathcal{F}} - \nabla u_h|_{K'} \cdot \boldsymbol{n}_{\mathcal{F}}, \tag{5.13}$$

where $\mathcal{F} = K_1 \cup K_2$ is the intersection of two neighbouring cells K and K' and $\boldsymbol{n}_{\mathcal{F}}$ is the normal vector of the facet [42, 48]. A ghost penalty is applied to the pressure on all facets outside of Ω by the form

$$s_{g,p,h}\left(p_h, q_h\right) := \sum_{\mathcal{F} \in \mathrm{F}_g} \left\langle \gamma_{g,p}\, h^3\, ||\nabla p_h||, ||\nabla q_h|| \right\rangle_{\mathcal{F}}, \tag{5.14}$$

where $\gamma_{g,p}$ is the numerical pressure stabilization parameters [42, 48].

First-order elements are used for pressure and velocity and according to [58], using the same space for $\vec{u}$ and p gives an inf-sup unstable formulation [13]. To address this issue, numerical stabilization terms are employed following the work of Kontogiannis et al. [13]. The pressure stabilization is achieved using the CIP method.

Let every cell facet, $\mathcal{F}$, which is intersected by the boundary of the domain or inside of it, $\mathrm{K} \cap \Omega \neq \emptyset$, be part of the set $\mathrm{F}_{\mathrm{CIP}}$. The CIP stabilization terms, $s_{\mathrm{CIP},p}$, can be defined as:

$$s_{\mathrm{CIP},p,h}\left(p_h, q_h\right) := \sum_{\mathcal{F} \in \mathrm{F}_{\mathrm{CIP}}} \left\langle \gamma_{\mathrm{CIP},p}\, h^3\, ||\nabla p_h||, ||\nabla q_h|| \right\rangle_{\mathcal{F}}, \tag{5.15}$$

where $\gamma_{\mathrm{CIP},p}$ is the numerical pressure stabilization parameters [13, 42, 48].

In addition, v is set to unity and because of the ∇^s formulation, ∇-div stabilization [59–61] with a weight of 1 is incorporated by the term a_{II}.

After constraining $\vec{u}, \vec{\psi}, p$ and q to a FEM subspace and including the stabilization terms, the discretized form reads

$$\begin{aligned} a_{I,h} + a_{II,h} - b_{p,I,h} + b_{\vec{u},I,h} - a_{III,h} + b_{p,II,h} - a_{IV,h} - b_{\vec{u},II,h} + a_{V,h} + \\ + s_{g,p,h} + s_{\mathrm{CIP},p,h} = j_{f,h} - j_{W,I,h} - j_{W,II,h} + j_{W,III,h} - j_{N,h}. \end{aligned} \tag{5.16}$$

5.2.4 Harmonic Balance and Matrix Form

As a final step, the HB time discretization is applied to the velocity and Eq. (5.16) now reads:

$$\begin{aligned} \sum_{i=1}^{2n+1} a_{I,h}^{(i)} c_{ij} + a_{II,h}^{(j)} - b_{p,I,h}^{(j)} + b_{\vec{u},I,h}^{(j)} - a_{III,h}^{(j)} + b_{p,II,h}^{(j)} - a_{IV,h}^{(j)} - b_{\vec{u},II,h}^{(j)} + a_{V,h}^{(j)} + \\ + s_{g,p,h}^{(j)} + s_{\mathrm{CIP},p,h}^{(j)} = j_{f,h}^{(j)} - j_{W,I,h}^{(j)} - j_{W,II,h}^{(j)} + j_{W,III,h}^{(j)} - j_{N,h}^{(j)}, \quad j = 1, 2, ..., 2n+1. \end{aligned} \tag{5.17}$$

To write the system of equations in a matrix form following FEM matrices are defined:

$$A_{3l3k}\mathcal{U}_{3ki}C_{ij} = \sum_{i=1}^{2n+1} a^{(i)}_{I,h} c_{ij}, \quad k,l = 1,2,\dots,n_x, \quad j = 1,2,\dots,2n+1, \tag{5.18a}$$

$$B_{3l3k}\mathcal{U}_{3ki}\mathrm{I}_{ij} = a^{(j)}_{II,h} - a^{(j)}_{III,h} - a^{(j)}_{IV,h} + a^{(j)}_{V,h}, \quad k,l = 1,2,\dots,n_x, \quad j = 1,2,\dots,2n+1, \tag{5.18b}$$

$$D_{l3k}\mathcal{U}_{3ki}\mathrm{I}_{ij} = -b^{(j)}_{p,I,h} + b^{(j)}_{p,II,h} + b^{(j)}_{\vec{u},I,h} - b^{(j)}_{\vec{u},II,h}, \quad k,l = 1,2,\dots,n_x, \quad j = 1,2,\dots,2n+1, \tag{5.18c}$$

$$E_{lk}P_{ki}\mathrm{I}_{ij} = s^{(j)}_{\mathrm{CIP},p,h} + s^{(j)}_{g,p,h}, \quad k,l = 1,2,\dots,n_x, \quad j = 1,2,\dots,2n+1, \tag{5.18d}$$

$$F_{3lj} = j^{(j)}_{f,h} - j^{(j)}_{W,I,h} + j^{(j)}_{W,III,h} - j^{(j)}_{N,h}, \quad k,l = 1,2,\dots,n_x, \quad j = 1,2,\dots,2n+1, \tag{5.18e}$$

$$G_{lj} = -j^{(j)}_{W,II,h}, \quad k,l = 1,2,\dots,n_x, \quad j = 1,2,\dots,2n+1. \tag{5.18f}$$

where A_{3l3k} includes the unsteady velocity-velocity form, B_{3l3k} contains the steady velocity-velocity forms, D_{3lk} represents the velocity-pressure forms, E_{lk} contains the stabilization forms, F_{3lj} includes the velocity BC forms, G_{3lj} contains the pressure BC forms and P_{ki} is a matrix containing the pressure at all 2n+1 time instances.

By defining the Kronecker product FEM matrices, $\mathcal{A}$,$\mathcal{B}$ and $\mathcal{D}$, as

$$\mathcal{A} := C^T_{ij} \otimes A_{3l3k} + \mathrm{I}_{2n+1} \otimes B_{3l3k}, \tag{5.19a}$$

$$\mathcal{B} := \mathrm{I}_{2n+1} \otimes D_{3lk}, \tag{5.19b}$$

$$\mathcal{D} := \mathrm{I}_{2n+1} \otimes E_{lk}, \tag{5.19c}$$

and the vectors, $\vec{\mathcal{U}}$, $\vec{P}$, $\vec{F}$ and $\vec{G}$, as $\vec{\mathcal{U}} := \mathrm{vec}(\mathcal{U}_{3ki})$,$\vec{P} := \mathrm{vec}(P_{ki})$,$\vec{F} := \mathrm{vec}(F_{3lj})$ and $\vec{G} := \mathrm{vec}(G_{lj})$, the linear system can be written as:

$$\begin{pmatrix} \mathcal{A} & \mathcal{B} \\ \mathcal{B}^T & \mathcal{D} \end{pmatrix} \begin{pmatrix} \vec{\mathcal{U}} \\ \vec{P} \end{pmatrix} = \begin{pmatrix} \vec{F} \\ \vec{G} \end{pmatrix}. \tag{5.20}$$

5.2.5 Pressure Projection via Schur Complement

An expression for calculating $\vec{P}$ is given by multiplying the first line of Eq. (5.20) with $\mathcal{A}^{-1}$ and substituting it into the second line:

$$S\vec{P} = \mathcal{B}^T\mathcal{A}^{-1}\vec{F} - \vec{G}, \tag{5.21}$$

where $S \equiv \mathcal{B}^T\mathcal{A}^{-1}\mathcal{B} - \mathcal{D}$ is the Schur complement. After obtaining the pressure, $\vec{\mathcal{U}}$ is given by [13, 42]:

$$\mathcal{A}\vec{\mathcal{U}} = \vec{F} - \mathcal{B}\vec{P}. \tag{5.22}$$

For the preconditioning of $\mathcal{A}$ following formulation is used

$$\tilde{\mathcal{A}} = \mathrm{I}_{2n+1} \otimes B_{3l3k}. \tag{5.23}$$

To reduce the number of outer pressure iterations, the steady state approximation of the Schur complement introduced in [13] is adapted as follows:

$$\tilde{S} := \mathrm{I}_{2n+1} \otimes \left(D^T_{l3k} \left(\mathrm{diag}(B_{3l3k})\right)^{-1} D_{l3k} - E_{lk} \right). \tag{5.24}$$

5.3 Implementation and Numerics

5.3.1 Zero Function Space Extension

Figure 5.1 illustrates that the actual domain of interest is smaller than $\mathcal{T}$, leading to unnecessary computational loads created by solving the equations well outside of Ω. This issue can be effectively circumvented by leveraging deal.II's functionality of defining different function spaces for individual segments of the computational domain. This approach is integrated with the built-in option of assigning the function space $\Psi_{0,h} = 0$, whose members are zero everywhere. As a result, deal.ii autonomously reduces the computational effort because, in regions where $\Psi_{0,h} = 0$ is applied, no DOFs are created at all. This ensures that the obtained matrices and vectors do not possess any entries for these DOFs [49].

Furthermore, Fig. 5.2 demonstrates the employed implementation of identifying the active part of the mesh, $\mathcal{T}_{ac}$, (black) as a cuboid, which is surrounded by the inactive regions $\mathcal{T}_0$ (dashed grey). The criterion for applying $\Psi_{0,h} = 0$ to a cell is determined by a check if its center is located inside a cuboid given by

$$[-\left(R+\gamma_0\epsilon\right),\left(R+\gamma_0\epsilon\right)] \times [-10,10] \times [-\left(R+\gamma_0\epsilon\right),\left(R+\gamma_0\epsilon\right)].$$

Here, γ_0 is a user-defined numerical parameter.

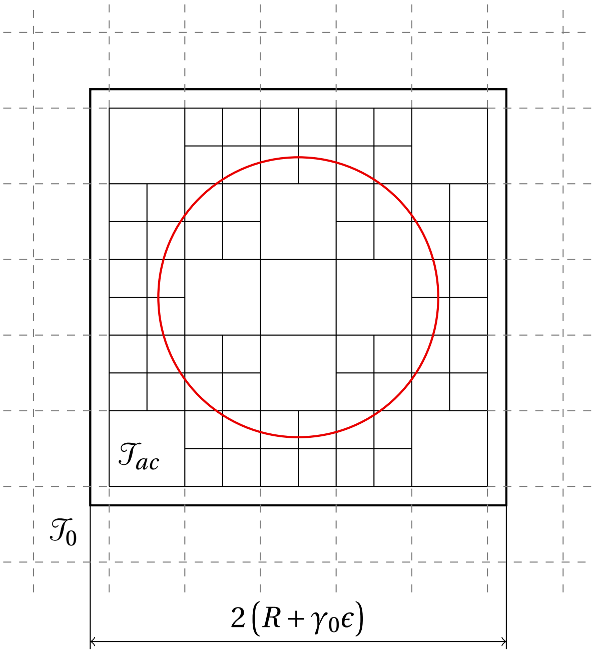

Figure 5.2: Zero function space extension minimizing the required number of active cells. Domain (red) immersed in active part of the mesh (black) surrounded by non active mesh with zero function space (grey dashed).

5.3.2 Implementation and Numerical Setup

The general setup explained in Subsec. 4.4.2 is reused for the Stokes solver. First-order elements for the pressure and the velocity are utilized. The solver developed during this work builds on a steady state diffuse domain Stokes solver by Matthias Henssler [42].

To minimize memory usage, only the individual FEM matrices are stored and PETSc [50–52] matrix-vector operators are utilized to compute the Kronecker products in a matrix-free way. The RHS of Eq. (5.21) is solved using GMRES, with a custom preconditioner employed to approximate the inverse of the system matrix. Each of the $2n+1$ inversions is performed with GMRES. Subsequently, the outer Schur complement is solved using FGMRES and the $2n+1$ individual inversions are solved with GMRES. The individual Schur complement preconditioner iterations are solved using FGMRES. Finally, a setup similar to the RHS one is utilized to find the velocity. Most individual $2n+1$ inversions are performed with an ASM preconditioner. It is worth noting that for a mesh size of $h_{\text{min}} = 3.90625 \cdot 10^{-2}$ m, the pressure solver returns with 0 iterations without actually solving for p. The root of this issue is not fully known at the moment. For this mesh size, a Jacobi preconditioner, which prevents the problem from occurring, is used. The bandwidth of the Kronecker product matrices is reduced by assigning the k-th DOF the global indices $k, k+1, k+2, \ldots k+2\,n$ instead of $k, n_x\,k, 2n_x\,k, \ldots, 2\,n\,n_x\,k$.

The decision if ghost penalties are applied on a facet is based on the signed distance of its center:

$$\mathfrak{z} < -\frac{\epsilon\sqrt{2}}{2}. \tag{5.25}$$

The same penalization value is used for CIP and ghost penalties and therefore, the check is not actively required.

5.4 Results and Discussion

5.4.1 Velocity and Pressure Profiles

The numerical parameter settings are summarized in Tab. 5.1. Naturally, three-dimensional cases lead to a significant increase in DOFs and as a result, slightly coarser meshes are used. Building on the results of the two-dimensional problem, where a smaller α resulted in lower errors for coarser grids, α is set to 4. In addition, when using a Nitsche penalty parameter normalized by the maximum phase field gradient norm, numerical instabilities constituted by a diverging p can occur. Similar problems were not encountered in the previous chapter. The source of these instabilities is currently unknown, but prescribing a constant value of 20 for γ_{N} prevents the issue. Implementing a Taylor expansion of the BC similar to the previous chapter could improve the numerical stability.

Table 5.1: Stokes problem user input parameters.

n_r	α	$\gamma_{\text{CIP},p}$	$\gamma_{g,p}$	γ_{N}	γ_0
2	4	0.05	0.05	20	$2(n_r + 0.1)$

Figure 5.3 contains an axial velocity profile alongside p, with $h_{\min} = 3.90625 \cdot 10^{-2}$ m and Wo = 1 at the time instance, $t = T$ s. As expected, the velocity profile is parabolic and adheres to the no-slip BC imposed on the walls. The peak velocity is 0.027 m/s, which is reasonable for a quasi-static flow under sinusoidal forcing with an amplitude of 1 Pa/m. Additionally, the velocity and pressure results depict the null function extension. The symmetric strain rate tensor, in combination with the natural BCs, causes the pressure fluctuations on the boundaries. These fluctuations induce higher velocities in regions near the pipe entrance and exit. The impairment of the velocity profiles shown in the following subsection is circumvented by setting the pipe length to 10 m, which ensures a decay of the induced error in velocity. Regardless of that, a symmetric pressure field is obtained, aligning with the expectations.

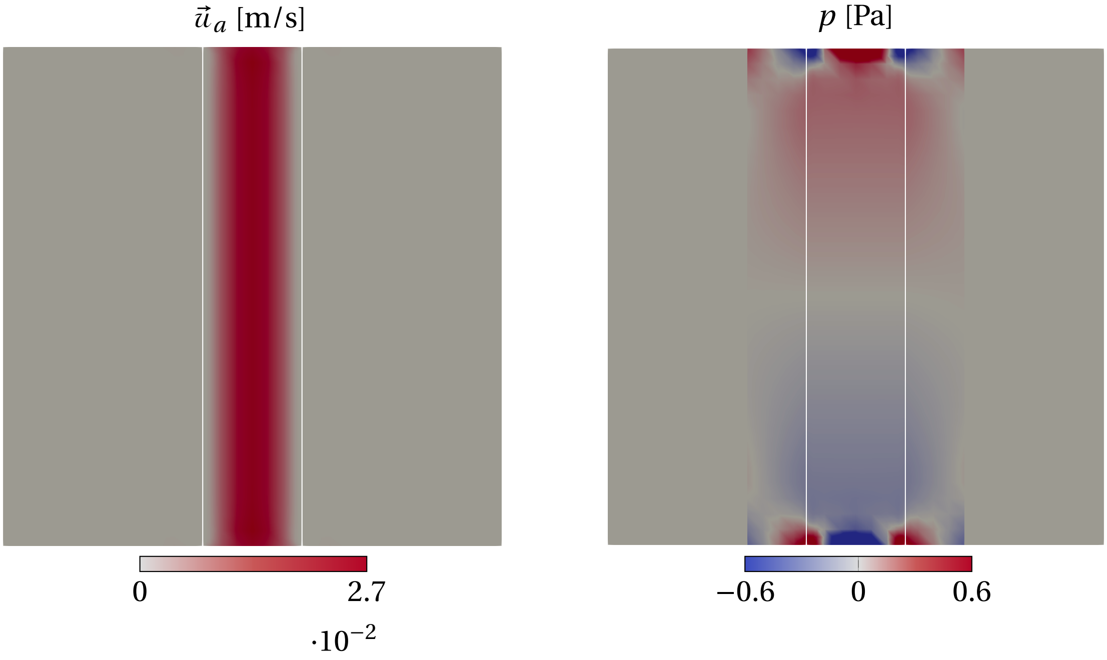

Figure 5.3: Axial velocity and pressure for Wo = 1 at $t = T$ s. Slice with x_1 normal passes through the pipe center and the domain boundary (white line) is shown.

Line plots are used to compare the computed velocity profiles with the analytical solution. During the analysis, the three different Wo numbers from the previous chapter are considered, with n, the number of harmonics, set to 1. Figure 5.4 presents the found velocity profiles (red dashed line) for the different Wo at $t = T/3$ s with $h_{\min} = 3.90625 \cdot 10^{-2}$ m. Across all Wo regimes, the general trend is reproduced, while the velocity is under-predicted in regions close to the walls. This discrepancy is attributed to the fact that the first-order BC expansion is not used, resulting in incorrect results near the wall. Furthermore, coarser meshes than the ones for the two-dimensional problem are tested due to the high cell count. As a result, big transition regions are obtained where the BC is active, leading to the flattening of the velocity profile near the wall. As shown in Fig. B.1, B.2 and B.3 in the appendix, this trend is consistently observed in all examined test cases, which share the same level of grid refinement.

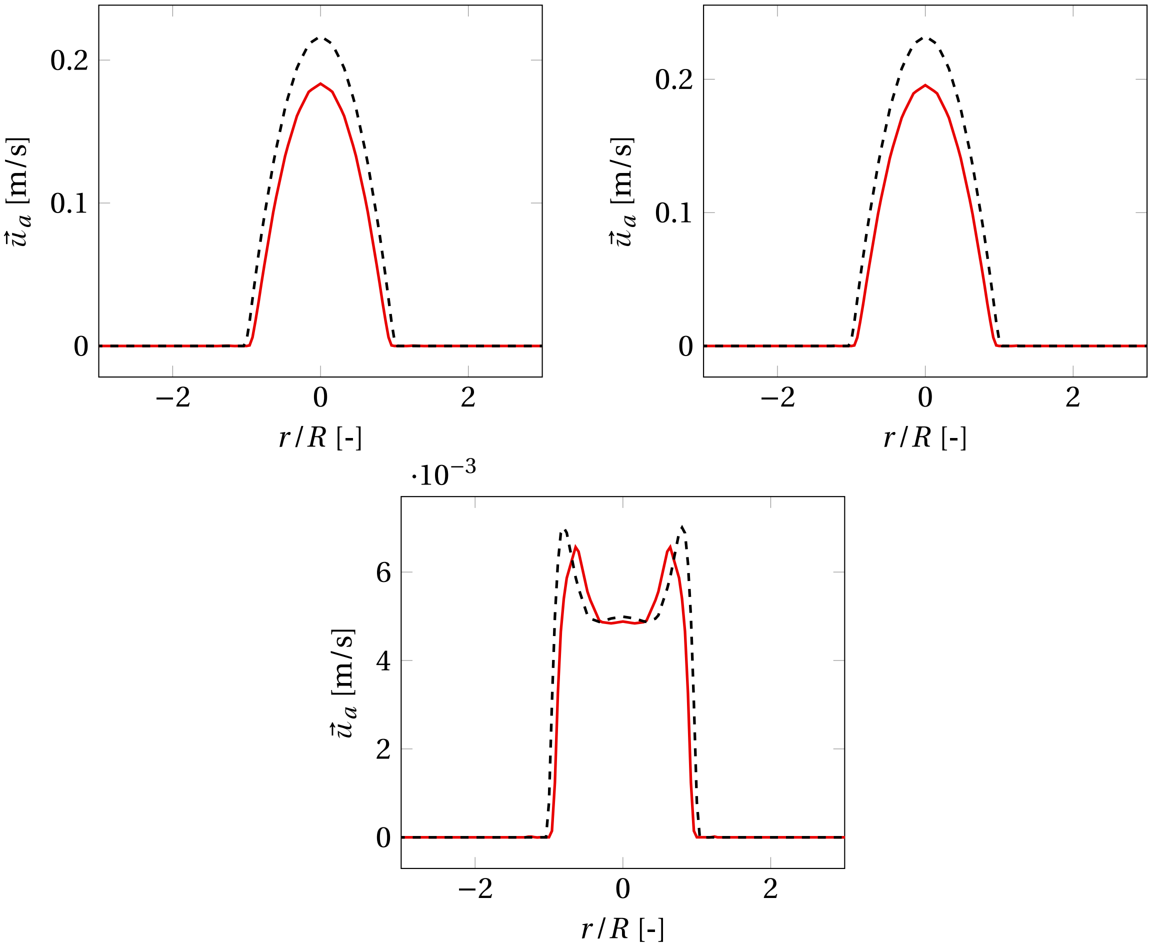

Figure 5.4: Axial velocity result (red line) and analytical solution (black dashed line) plotted along x_1-axis at $t = T/3$ s. Top left: Wo = 0.1, top right: Wo = 1 and bottom: Wo = 10.

Following the presented argument, the deviations are expected to decrease with further mesh refinement, resulting in thinner transition areas. Figure 5.5 contains the findings for Wo = 1 and $t = 2T/3$ s at two different mesh sizes. The grid corresponding to the RHS is refined by a factor of 2 and the deviation from the analytical solution decreases significantly, supporting this prognosis. Similar to the two-dimensional pipe flow problem, a clear correlation between Wo and the number of iterations is found. When comparing the iterations required for the $\mathcal{A}^{-1}$ solve for the RHS, the S^{-1} solve for the pressure and the $\mathcal{A}^{-1}$ solve for the velocity, following values are found for Wo = 0.1, 1 and 10: (3,12,3), (6,13,6) and (116,21,130). This highlights the need for a better preconditioner in the limit of higher Wo numbers.

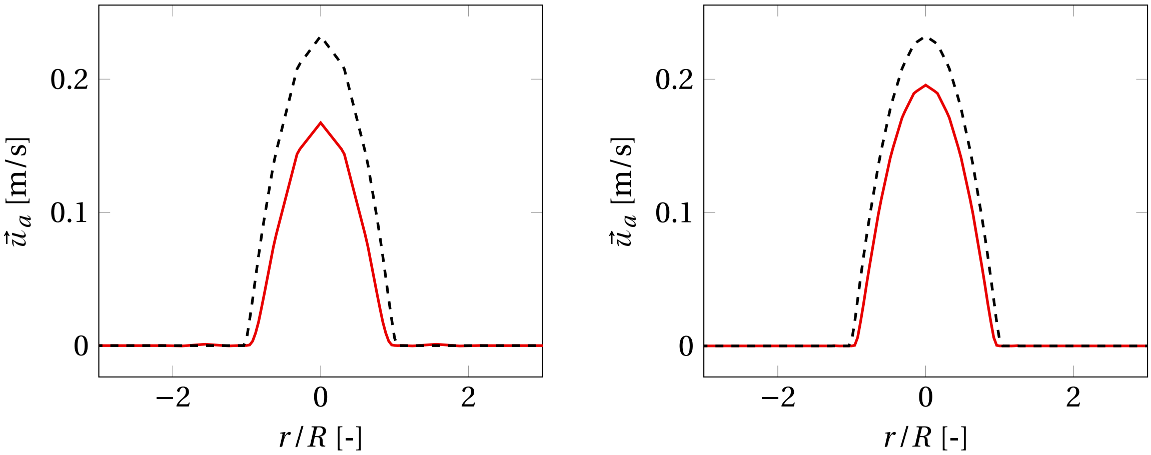

Figure 5.5: Axial velocity result (red line) and analytical solution (black dashed line) plotted along x_1-axis at $t = 2T/3$ s. Left: $h_{\text{min}} = 7.8125 \cdot 10^{-2}$ m and right: $h_{\text{min}} = 3.90625 \cdot 10^{-2}$ m.

5.4.2 Convergence Rates

Lastly, the spatial convergence is tested with Wo = 1 and varying mesh sizes. Figure 5.6 depicts the obtained errors, which are calculated using the temporal integral of all $2n+1$ time instances. Note that the impact of the boundary pressure fluctuations is avoided by only considering the x_2 normal plane at $x_2 = 0$ for the error calculation.

It is evident that the optimum rate of convergence is not achieved for either of the two errors as slopes of approximately 0.6 and 0.3 are found for L_e^2 and H_e^1, respectively. There are multiple potential reasons giving rise to this poor convergence. One possibility is the aforementioned flattening of the velocity profile introduced by the employed BC formulation. This results in high errors, which might impact the convergence rates. In addition, only low resolutions have been tested so far due to time constraints preventing further investigations. Since the extent of the flattening effect decreases with a higher level of refinement, the rates might improve as the mesh gets finer. Besides that, with α set to 4, the two-dimensional case exhibited slightly worse convergence than the theoretical optimum. These open questions are left for future research.

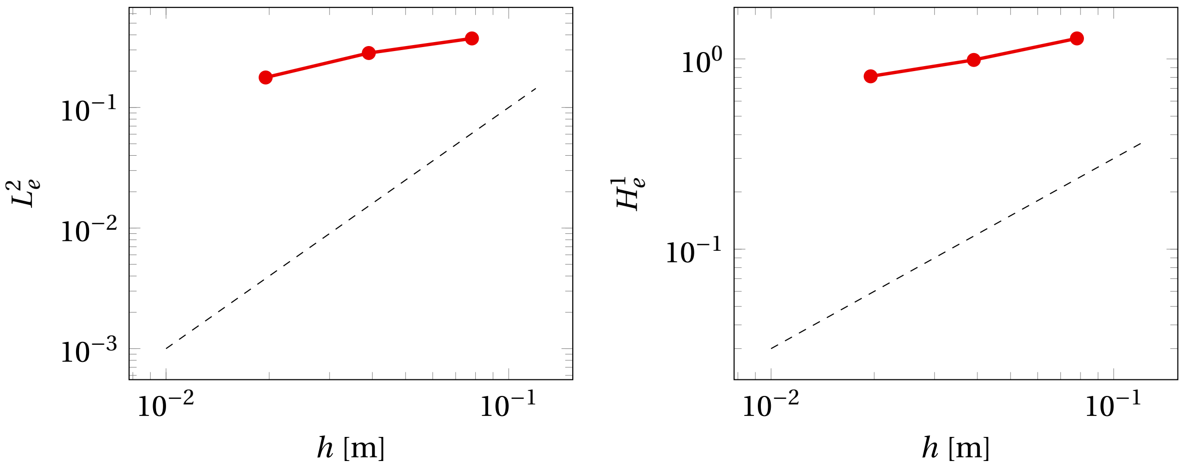

Figure 5.6: L_e^2 and H_e^1 between numerical and analytical solution plotted against mesh size. Left: reference slope of 2 (black dashed line) and right: reference slope of 1 (black dashed line).

6 Summary and Outlook

6.1 Summary

In this work, the HB time discretization coupled with the FEM for spatial resolution is applied to different PDEs. A two-dimensional heat equation with periodic boundary conditions serves as an initial test case. Four different variations, including linear and nonlinear formulations, are tested. In the linear variant, the problem is reformulated as a linear system of equations, while a Newton method is employed to solve the nonlinear system. The implemented HB solver for the linear formulations is successfully validated against an Euler method for time discretization and all setups show the theoretical optimal spatial convergence rates.

Furthermore, the HB formulation is incorporated into an existing two-dimensional steady-state Poisson solver, which relies on the diffuse domain approach to represent geometry. In this framework, a phase field function defines the geometry and the equation is solved on a background mesh. The essential BCs are weakly enforced using Nitsche's method. The obtained solver is tested on a pulsatile pipe flow with a known analytical solution. Three different flow regimes are considered and the results are in good agreement with the exact solution. A spatial convergence test is performed and good rates are obtained.

Lastly, a steady-state diffuse domain Stokes solver is coupled with the HB method. Similar to the two-dimensional solver, the BCs are weakly enforced using Nitsche's method. In order to stabilize the solution, penalty terms based on the CIP method are applied to the pressure inside and outside of the domain of interest. A three-dimensional pulsatile pipe flow with a known solution is investigated. The numerical velocity profiles show acceptable agreement with the analytical solution.

6.2 Outlook

In the context of the current work, multiple interesting possibilities for future research were identified. These include improving the stability of the Stokes implementation by preventing diverging pressure results. The incorporation of more sophisticated numerical stabilization terms might counteract this issue. Examples of this can be found in the work of Kontogiannis et al. [13]. Besides that, the lower-than-expected convergence rate and its dependency on α presents an open issue. Implementing a Taylor expansion of the BC could improve the stability and convergence rates of the Stokes solver.

A better preconditioner for high Wo constitutes the most important unresolved aspect encountered in the two- and three-dimensional pulsatile pipe flow problems. One possibility that potentially resolves this issue is shown in [62], where a Newton method is used to solve a nonlinear system. The almost circulant structure of the system matrix is leveraged by adjusting the diagonal block matrices so that a circulant structure is reached. Finally, this approximation is block diagonalized by applying a discrete Fourier transform. After performing the inversion in the Fourier space, an inverse discrete Fourier transform transforms the solution into the time domain [62]. Lastly, the solver extends to the Navier-Stokes equations.

Appendices

A Pulsatile Pipe Flow Results

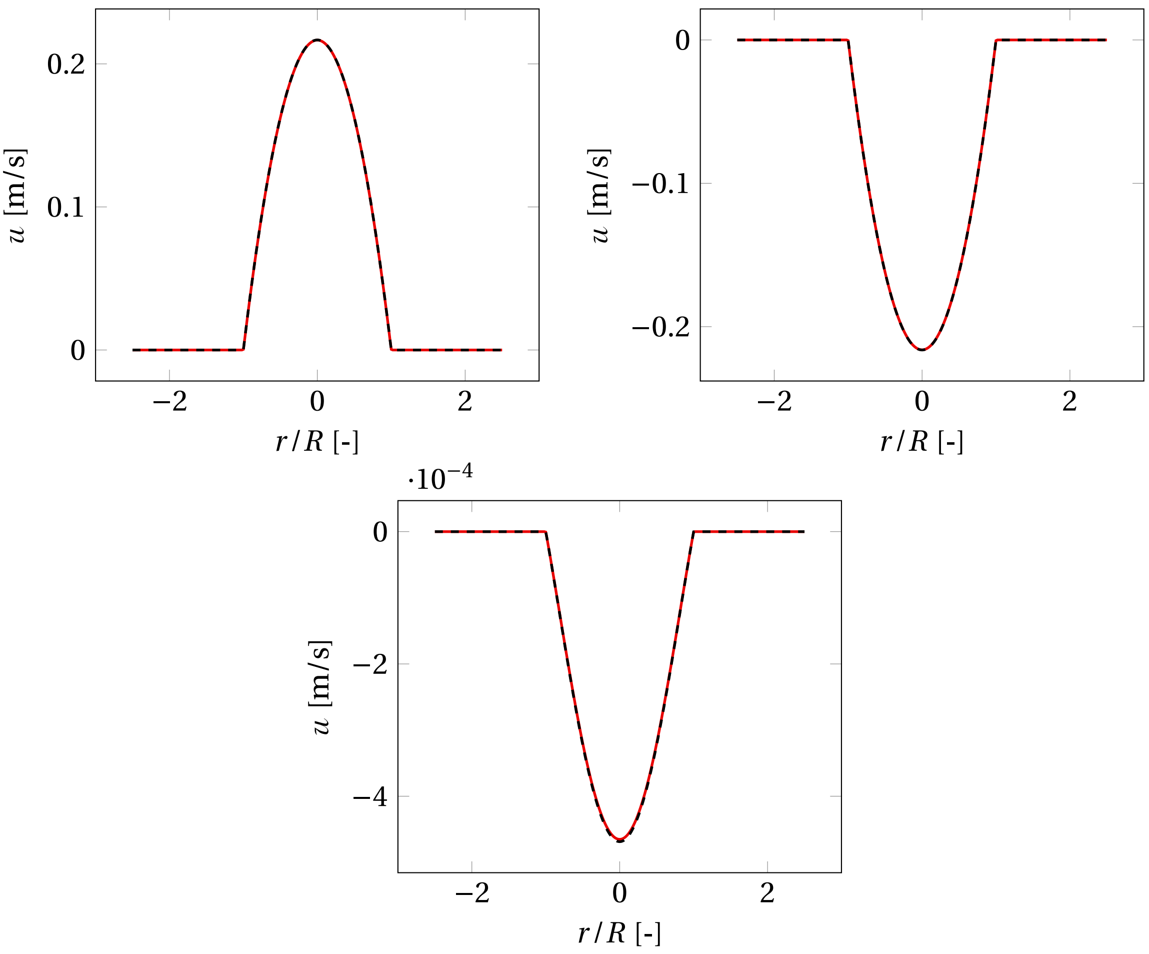

Figure A.1: Axial velocity result (red line) and analytical solution (black dashed line) plotted along x_1 axis for Wo = 0.1. Top left: $t = T/3$ s, top right: $t = 2T/3$ s and bottom: $t = T$ s.

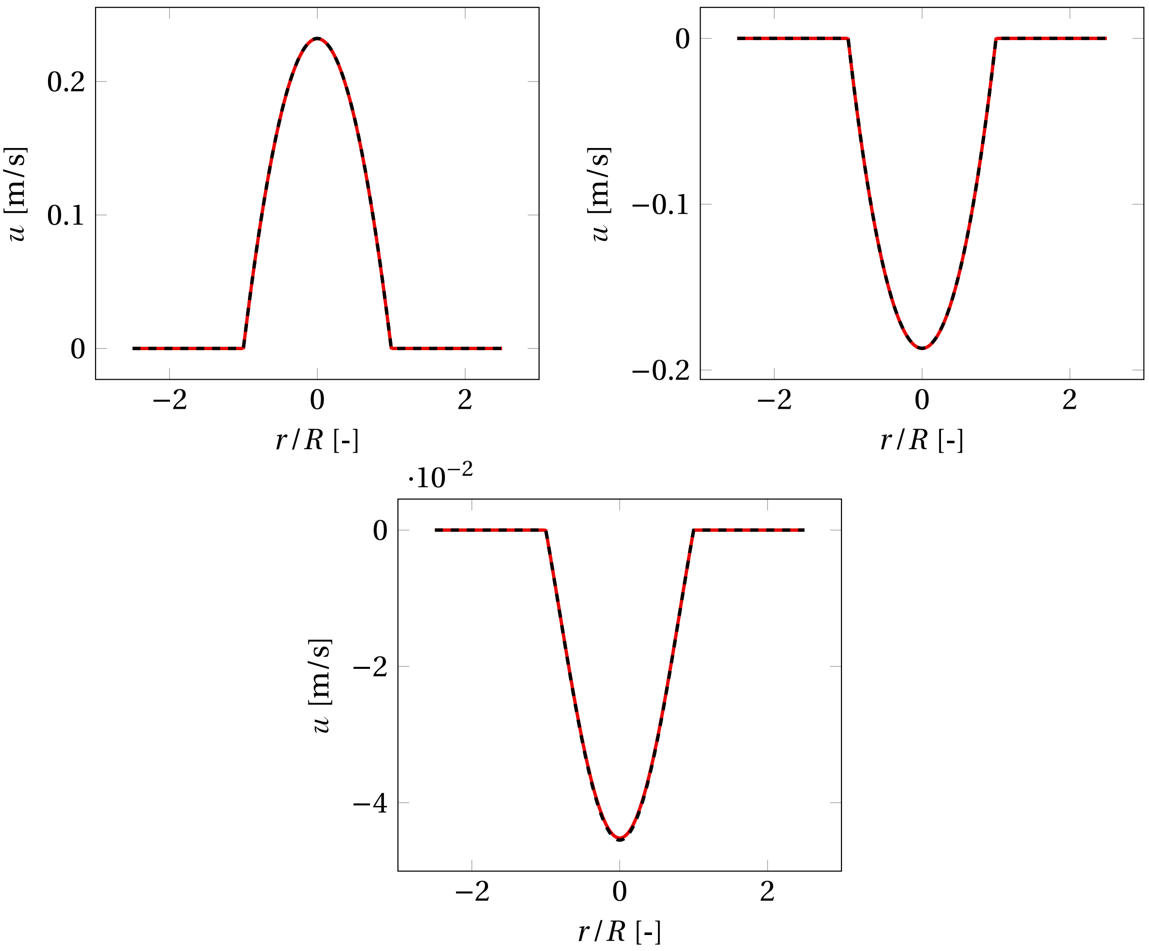

Figure A.2: Axial velocity result (red line) and analytical solution (black dashed line) plotted along x_1 axis for Wo = 1. Top left: $t = T/3$ s, top right: $t = 2T/3$ s and bottom: $t = T$ s.

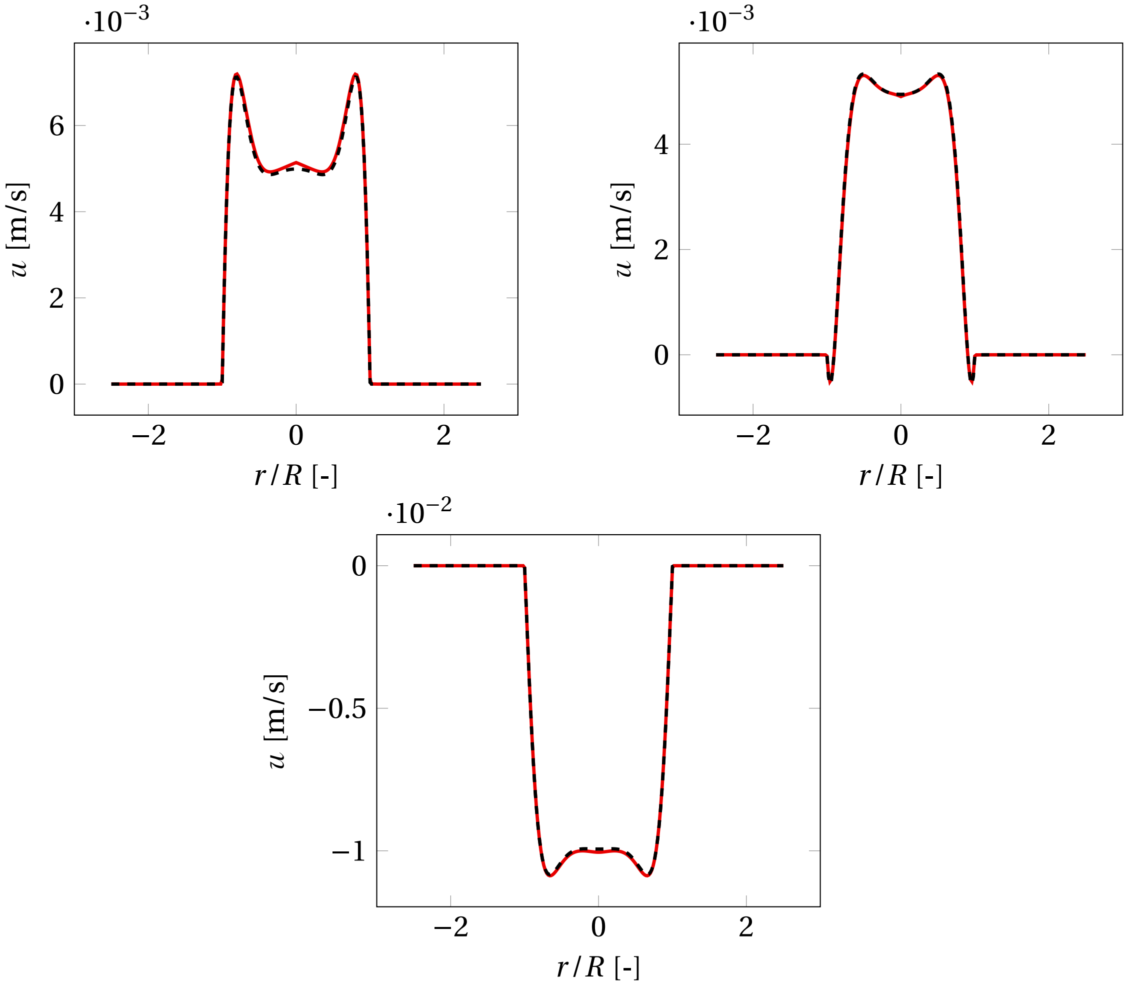

Figure A.3: Axial velocity result (red line) and analytical solution (black dashed line) plotted along x_1 axis for Wo = 10. Top left: $t = T/3$ s, top right: $t = 2T/3$ s and bottom: $t = T$ s.

B Pulsatile Stokes Flow Results

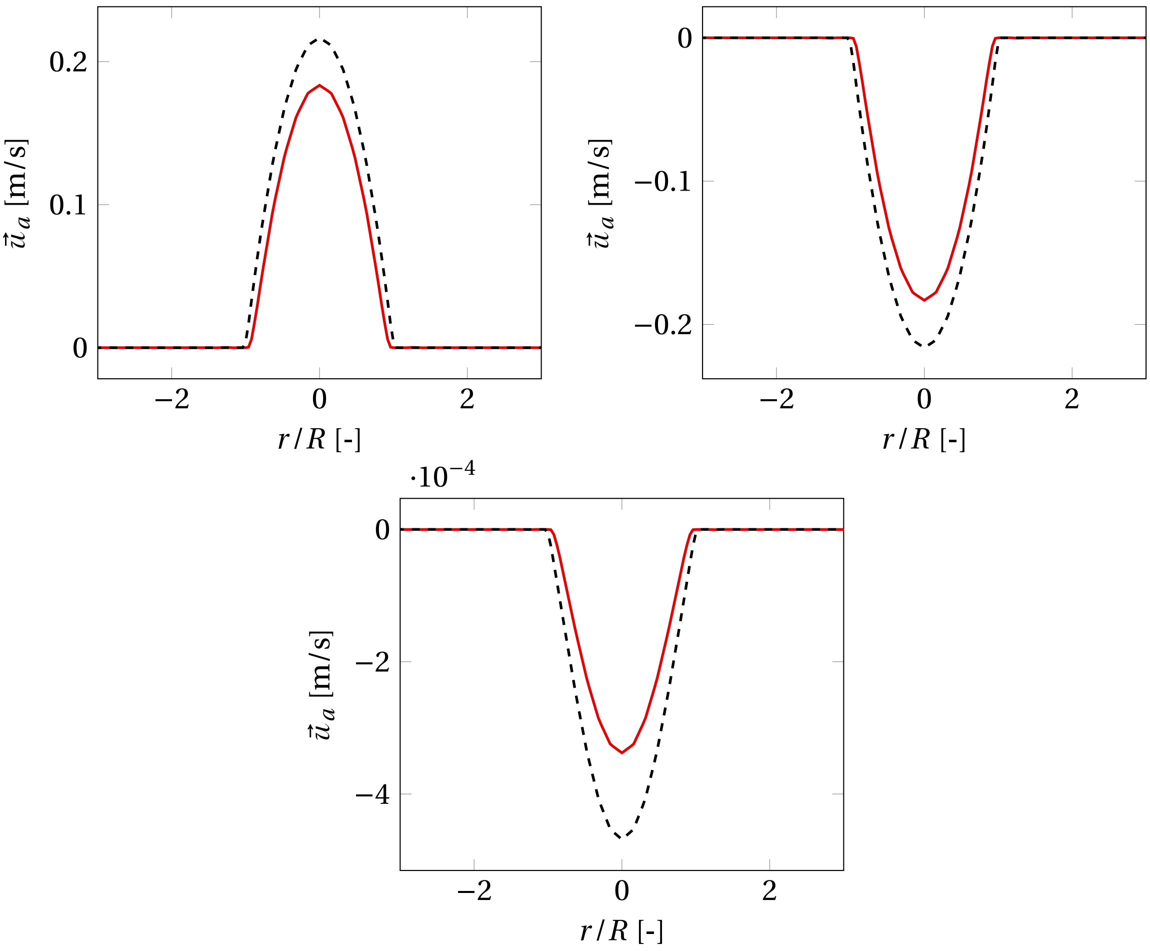

Figure B.1: Axial velocity result (red line) and analytical solution (black dashed line) plotted along x_1-axis for Wo = 0.1. Top left: $t = T/3$ s, top right: $t = 2T/3$ s and bottom: $t = T$ s.

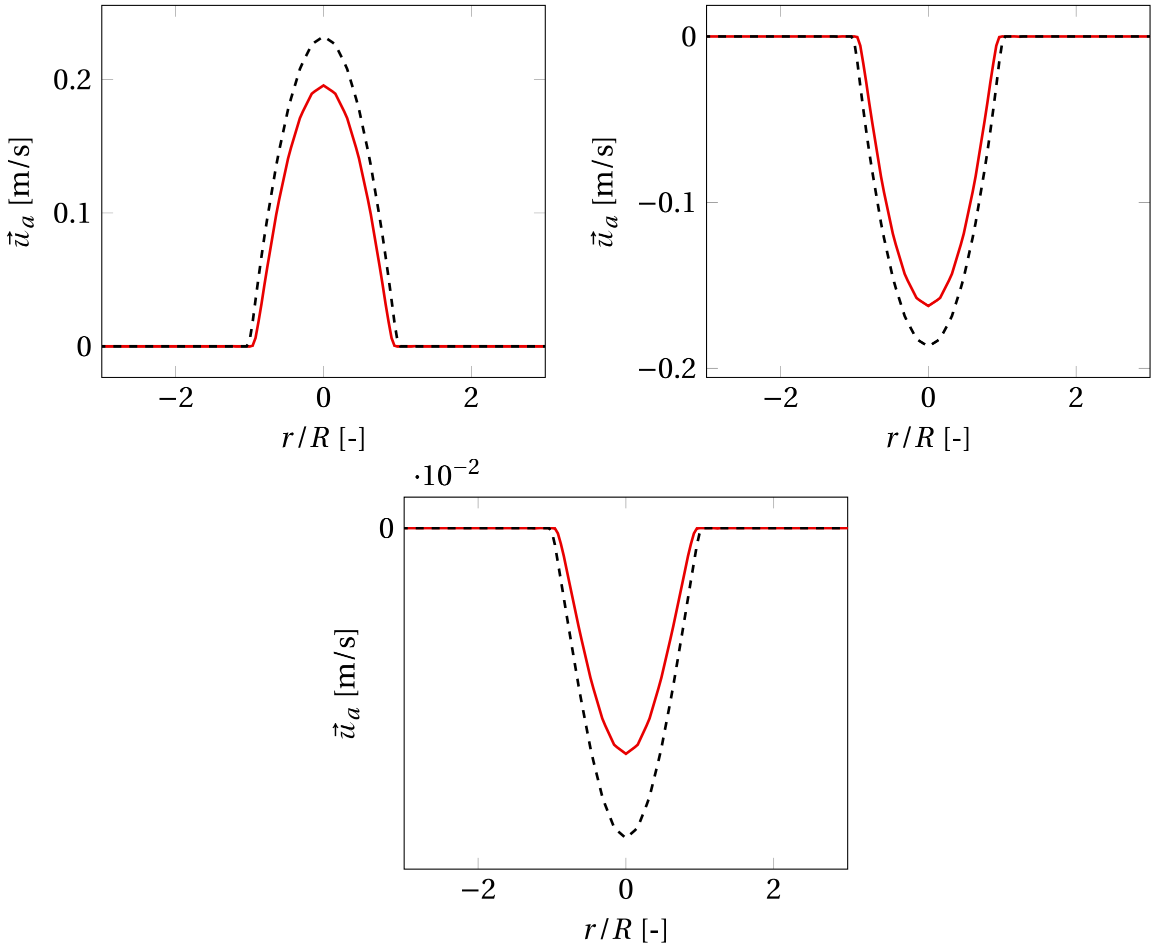

Figure B.2: Axial velocity result (red line) and analytical solution (black dashed line) plotted along x_1-axis for Wo = 1. Top left: $t = T/3$ s, top right: $t = 2T/3$ s and bottom: $t = T$ s.

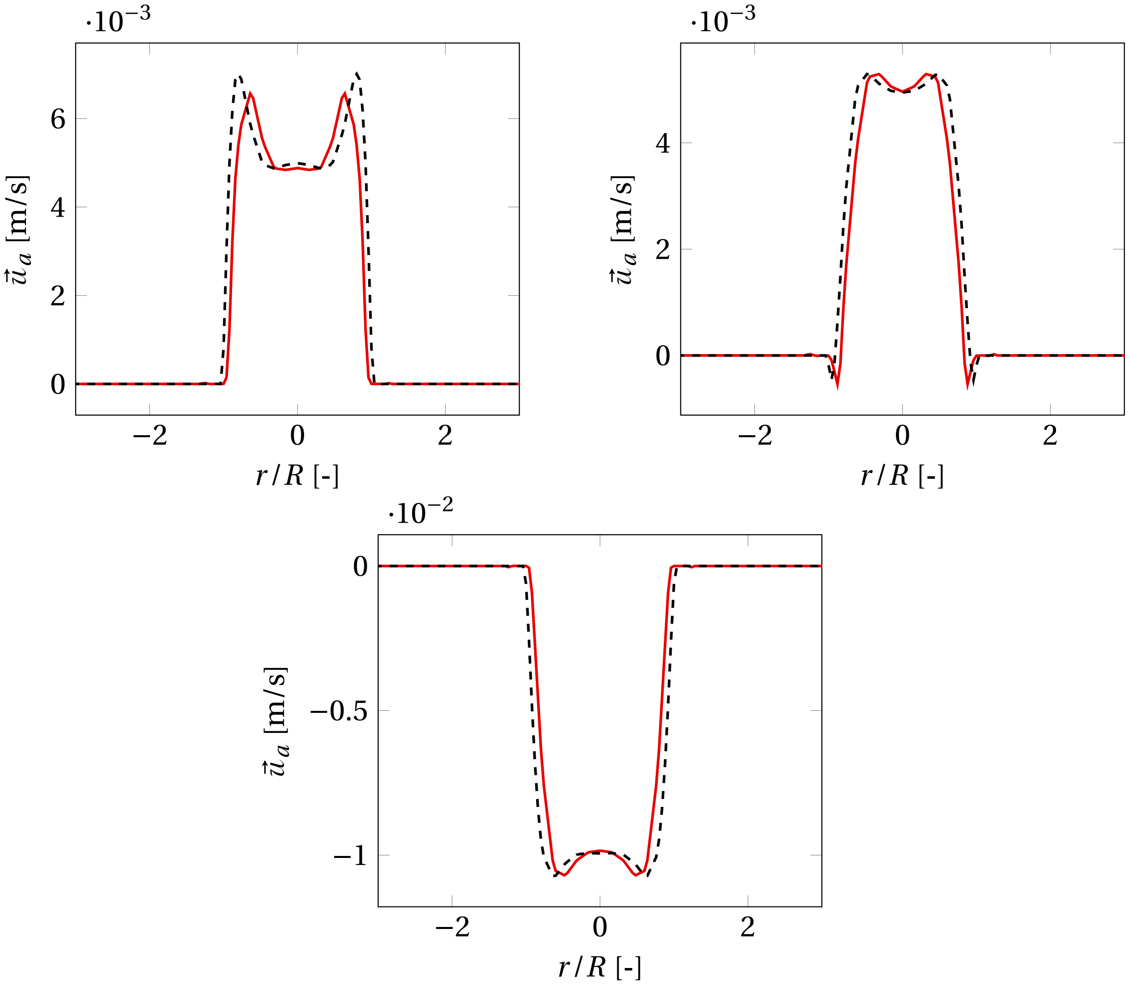

Figure B.3: Axial velocity result (red line) and analytical solution (black dashed line) plotted along x_1-axis for Wo = 10. Top left: $t = T/3$ s, top right: $t = 2T/3$ s and bottom: $t = T$ s.

List of Figures

List of Tables

Bibliography

[1] P. P. Sengupta, G. Pedrizzetti, P. J. Kilner, A. Kheradvar, T. Ebbers, G. Tonti, A. G. Fraser, and J. Narula. Emerging trends in CV flow visualization. *JACC Cardiovasc. Imaging*, 5(3): 305–316, 2012.

[2] K. Jacobs, L. Hahn, M. Horowitz, S. Kligerman, S. Vasanawala, and A. Hsiao. Hemodynamic assessment of structural heart disease using 4D flow MRI: How we do it. *AJR Am. J. Roentgenol.*, 217(6):1322–1332, 2021.

[3] C. J. Elkins, M. Markl, N. Pelc, and J. K. Eaton. 4D magnetic resonance velocimetry for mean velocity measurements in complex turbulent flows. *Exp. Fluids*, 34(4):494–503, 2003.

[4] C. J. Elkins and M. T. Alley. Magnetic resonance velocimetry: applications of magnetic resonance imaging in the measurement of fluid motion. *Exp. Fluids*, 43(6):823–858, 2007.

[5] G. C. Saliba, J. G. Korvink, and J. J. Brandner. Magnetic resonance velocimetry reveals secondary flow in falling films at the microscale. *Phys. Fluids (1994)*, 36(7), 2024.

[6] O. N. Jaspan, R. Fleysher, and M. L. Lipton. Compressed sensing MRI: a review of the clinical literature. *Br. J. Radiol.*, 88(1056):20150487, 2015.

[7] A. Kontogiannis, S. V. Elgersma, A. J. Sederman, and M. P. Juniper. Joint reconstruction and segmentation of noisy velocity images as an inverse Navier–Stokes problem. *J. Fluid Mech.*, 944(A40), 2022.

[8] D. L. Donoho. Compressed sensing. *IEEE Trans. Inf. Theory*, 52(4):1289–1306, 2006.

[9] C. M. Sandino, J. Y. Cheng, F. Chen, M. Mardani, J. M. Pauly, and S. S. Vasanawala. Compressed sensing: From research to clinical practice with deep neural networks: Shortening scan times for magnetic resonance imaging. *IEEE Signal Process. Mag.*, 37(1): 117–127, 2020.

[10] A. L. Pilastri and J. M. R. Tavares. Reconstruction algorithms in compressive sensing : An overview. In *Proceedings Doctoral Symposium in Informatics Engineering*, pages 1–6, 2016. URL `https://api.semanticscholar.org/CorpusID:5920985`.

[11] S. Qin. Simple algorithm for L 1norm regularisationbased compressed sensing and image restoration. *IET Image Process.*, 14(14):3405–3413, 2020.

[12] B. Cleary, B. Simonton, J. Bezney, E. Murray, S. Alam, A. Sinha, E. Habibi, J. Marshall, E. S. Lander, F. Chen, and A. Regev. Compressed sensing for highly efficient imaging transcriptomics. *Nat. Biotechnol.*, 39(8):936–942, 2021.

[13] A. Kontogiannis, S. V. Elgersma, A. J. Sederman, and M. P. Juniper. Bayesian inverse Navier-Stokes problems: joint flow field reconstruction and parameter learning. 2024. doi: 10.48550/arXiv.2406.18464.

[14] S. W. Funke, M. Nordaas, Ø. Evju, M. S. Alnæs, and K. A. Mardal. Variational data assimilation for transient blood flow simulations: Cerebral aneurysms as an illustrative example. *Int. J. Numer. Method. Biomed. Eng.*, 35(1), 2019.

[15] T. S. Koltukluoğlu and P. J. Blanco. Boundary control in computational haemodynamics. *J. Fluid Mech.*, 847:329–364, 2018.

[16] T. S. Koltukluoğlu. *Data Assimilation in Computational Hemodynamics*. Phd thesis, Eidgenössische Technische Hochschule Zürich, Zürich, 2019.

[17] T. S. Koltukluoğlu, G. Cvijetić, and R. Hiptmair. Harmonic balance techniques in cardiovascular fluid mechanics. In D. Shen, T. Liu, T. M. Peters, L. H. Staib, C. Essert, S. Zhou, P.-T. Yap, and A. Khan, editors, *Medical Image Computing and Computer Assisted Intervention – MICCAI 2019*, pages 486–494, Cham, 2019. Springer International Publishing.

[18] P. D. Morris, A. Narracott, H. von Tengg-Kobligk, D. A. Silva Soto, S. Hsiao, A. Lungu, P. Evans, N. W. Bressloff, P. V. Lawford, D. R. Hose, and J. P. Gunn. Computational fluid dynamics modelling in cardiovascular medicine. *Heart*, 102(1):18–28, 2016.

[19] S. Sankaran, H. J. Kim, G. Choi, and C. A. Taylor. Uncertainty quantification in coronary blood flow simulations: Impact of geometry, boundary conditions and blood viscosity. *J. Biomech.*, 49(12):2540–2547, 2016.

[20] D. Katritsis, L. Kaiktsis, A. Chaniotis, J. Pantos, E. P. Efstathopoulos, and V. Marmarelis. Wall shear stress: theoretical considerations and methods of measurement. *Prog. Cardiovasc. Dis.*, 49(5):307–329, 2007.

[21] J. Sotelo, J. Urbina, I. Valverde, C. Tejos, P. Irarrazaval, M. E. Andia, S. Uribe, and D. E. Hurtado. 3D quantification of wall shear stress and oscillatory shear index using a finite-element method in 3D CINE PC-MRI data of the thoracic aorta. *IEEE Trans. Med. Imaging*, 35(6):1475–1487, 2016.

[22] J. Aguayo, C. Bertoglio, and A. Osses. Distributed parameter identification for the Navier–Stokes equations for obstacle detection. *Inverse Probl.*, 40(1):015012, 2024.

[23] S. L. Cotter, M. Dashti, J. C. Robinson, and A. M. Stuart. Bayesian inverse problems for functions and applications to fluid mechanics. *Inverse Probl.*, 25(11):115008, 2009.

[24] M. Burger, Y. Korolev, and J. Rasch. Convergence rates and structure of solutions of inverse problems with imperfect forward models. *Inverse Probl.*, 35(2):024006, 2019.

[25] L. Bungert, M. Burger, Y. Korolev, and C.-B. Schönlieb. Variational regularisation for inverse problems with imperfect forward operators and general noise models. *Inverse Probl.*, 36(12):125014, 2020.

[26] M. Y. Hosseini and Y. Shiri. Flow field reconstruction from sparse sensor measurements with physics-informed neural networks. *Phys. Fluids (1994)*, 36(7), 2024.

[27] E.-Z. Rui, G.-Z. Zeng, Y.-Q. Ni, Z.-W. Chen, and S. Hao. Time-averaged flow field reconstruction based on a multifidelity model using physics-informed neural network (PINN) and nonlinear information fusion. *Int. J. Numer. Methods Heat Fluid Flow*, 34(1):131–149, 2024.

[28] V. Parfenyev, M. Blumenau, and I. Nikitin. Enhancing capabilities of particle image/tracking velocimetry with physics-informed neural networks. 2024. doi: 10.48550/arXiv.2404.01193.

[29] K. H. e. a. Huebner. *The finite element method for engineers.* J. Wiley, New York, 4th ed edition, 2001. ISBN 9780471370789.

[30] D. L. Logan. *A first course in the finite element method.* Thomson, United States, 4th ed edition, 2007. ISBN 9780534552985.

[31] O. C. Zienkiewicz, R. L. Taylor, and J. Z. Zhu. *The finite element method: its basis and fundamentals.* Elsevier, Butterworth-Heinemann, Amsterdam, seventh edition edition, 2013. ISBN 9781856176330. OCLC: ocn852808496.

[32] J.-L. Guermond. Fluid mechanics: Numerical methods. In J.-P. Francoise, G. L. Naber, and T. S. Tsun, editors, *Encyclopedia of mathematical physics: Five-volume set.* Academic Press, 2006.

[33] W. Bangerth. Finite element methods in scientific computing. Accessed at: 25.07.2024. URL `https://www.math.colostate.edu/~bangerth/videos/676/slides.21.5.pdf`.

[34] J. N. Reddy. *An Introduction to the finite element method.* Mc Graw-Hill Higher Eduation, New York, NY i, 3rd edition. edition, 2006. ISBN 9780070607415. OCLC: 1370638753.

[35] E. Sueli. *Lecture Notes on Finite Element Methods for Partial Differential Equations.* Mathematical Institute. University of Oxford, 2020, Accessed at: 17.09.2024. URL `https://people.maths.ox.ac.uk/suli/fem.pdf`.

[36] J. P. Boyd. *Chebyshev and Fourier spectral methods.* Dover Publications, 2001.

[37] C. F. Loan. The ubiquitous Kronecker product. *Journal of Computational and Applied Mathematics*, 123(1-2):85–100, 2000. doi: 10.1016/S0377-0427(00)00393-9.

[38] L. Zhao and T. Heister. The deal.II tutorial step-57: The incompressible, stationary navier stokes equations, 2017.

[39] The MathWorks Inc. MATLAB version: 9.14.0 (R2023a). 2023. URL `https://www.mathworks.com`.

[40] D. F. Griffiths and D. J. Higham. *Numerical methods for ordinary differential equations: Initial value problems.* Springer London, London, 2010.

[41] F. Yu, Z. Guo, and J. Lowengrub. Higher-order accurate diffuse-domain methods for partial differential equations with dirichlet boundary conditions in complex, evolving geometries. *J. Comput. Phys.*, 406(109174):109174, 2020.

[42] M. Henssler. Implicit domain solvers for poisson and stokes problems. Technical report, University of Cambridge, 2024.

[43] K. Y. Lervåg and J. Lowengrub. Analysis of the diffuse-domain method for solving PDEs in complex geometries. *Commun. Math. Sci.*, 13(6):1473–1500, 2015.

[44] J. R. Womersley. Method for the calculation of velocity, rate of flow and viscous drag in arteries when the pressure gradient is known. *J. Physiol.*, 127(3):553–563, 1955.

[45] J. Mestel. *Bio Fluid Mechanics: Lecture 14: Pulsatile flow in a long straight artery.* Imperial College London, 2009, Accessed at: 20.09.2024. URL `https://www.ma.ic.ac.uk/~ajm8/BioFluids/lec1114.pdf`.

[46] C. Loudon and A. Tordesillas. The use of the dimensionless womersley number to characterize the unsteady nature of internal flow. *J. Theor. Biol.*, 191(1):63–78, 1998.

[47] J. Nitsche. Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind. *Abh. Math. Semin. Univ. Hambg.*, 36(1):9–15, 1971.

[48] A. Massing, B. Schott, and W. A. Wall. A stabilized nitsche cut finite element method for the oseen problem. *Comput. Methods Appl. Mech. Eng.*, 328:262–300, 2018.

[49] D. Arndt, W. Bangerth, M. Bergbauer, M. Feder, M. Fehling, J. Heinz, T. Heister, L. Heltai, M. Kronbichler, M. Maier, P. Munch, J.-P. Pelteret, B. Turcksin, D. Wells, and S. Zampini. The deal.II library, version 9.5. *J. Numer. Math.*, 31(3):231–246, 2023.

[50] S. Balay, W. D. Gropp, L. C. McInnes, and B. F. Smith. Efficient management of parallelism in object oriented numerical software libraries. In E. Arge, A. M. Bruaset, and H. P. Langtangen, editors, *Modern Software Tools in Scientific Computing*, pages 163–202. Birkhäuser Press, 1997.

[51] S. Balay, S. Abhyankar, M. F. Adams, S. Benson, J. Brown, P. Brune, K. Buschelman, E. Constantinescu, L. Dalcin, A. Dener, V. Eijkhout, J. Faibussowitsch, W. D. Gropp, V. Hapla, T. Isaac, P. Jolivet, D. Karpeev, D. Kaushik, M. G. Knepley, F. Kong, S. Kruger, D. A. May, L. C. McInnes, R. T. Mills, L. Mitchell, T. Munson, J. E. Roman, K. Rupp, P. Sanan, J. Sarich, B. F. Smith, S. Zampini, H. Zhang, H. Zhang, and J. Zhang. PETSc/TAO users manual. Technical Report ANL-21/39 - Revision 3.21, Argonne National Laboratory, 2024.

[52] S. Balay, S. Abhyankar, M. F. Adams, S. Benson, J. Brown, P. Brune, K. Buschelman, E. M. Constantinescu, L. Dalcin, A. Dener, V. Eijkhout, J. Faibussowitsch, W. D. Gropp, V. Hapla, T. Isaac, P. Jolivet, D. Karpeev, D. Kaushik, M. G. Knepley, F. Kong, S. Kruger, D. A. May, L. C. McInnes, R. T. Mills, L. Mitchell, T. Munson, J. E. Roman, K. Rupp, P. Sanan, J. Sarich, B. F. Smith, S. Zampini, H. Zhang, H. Zhang, and J. Zhang. PETSc Web page. `https://petsc.org/`, 2024. URL `https://petsc.org/`.

[53] C. Burstedde, L. C. Wilcox, and O. Ghattas. P4est: Scalable algorithms for parallel adaptive mesh refinement on forests of octrees. *SIAM J. Sci. Comput.*, 33(3):1103–1133, 2011.

[54] C. Burstedde. Parallel tree algorithms for AMR and non-standard data access. *ACM Trans. Math. Softw.*, 46(4):1–31, 2020.

[55] Y. Saad. A flexible inner-outer preconditioned GMRES algorithm. *SIAM J. Sci. Comput.*, 14(2):461–469, 1993.

[56] Y. Saad and M. H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 7(3):856–869, 1986.

[57] O. Widlund and M. Dryja. *An additive variant of the Schwarz alternating method for the case of many subregions.* Technical Report 339, Ultracomputer Note 131. Department of Computer Science, Courant Institute, 1987.

[58] S. C. Brenner and L. R. Scott. *The mathematical theory of finite element methods.* Springer New York, New York, NY, 2008.

[59] M. Benzi and M. A. Olshanskii. An augmented lagrangianbased approach to the oseen problem. *SIAM J. Sci. Comput.*, 28(6):2095–2113, 2006.

[60] M. Olshanskii, G. Lube, T. Heister, and J. Löwe. Grad–div stabilization and subgrid pressure models for the incompressible Navier–Stokes equations. *Comput. Methods Appl. Mech. Eng.*, 198(49-52):3975–3988, 2009.

[61] T. Heister and G. Rapin. Efficient augmented lagrangiantype preconditioning for the oseen problem using GradDiv stabilization: AUGMENTED LAGRANGIAN-TYPE PRECONDITIONING USING GRAD-DIV STABILIZATION. *Int. J. Numer. Methods Fluids*, 71(1): 118–134, 2013.

[62] J. Moulin. *On the flutter bifurcation in laminar flows : linear and nonlinear modal methods*. Theses, Institut Polytechnique de Paris, 2020. URL `https://theses.hal.science/tel-03160958`.