

# Efficient numerical methods for the solution of coupled multiphysics problems



Liya Asner  
Linacre College  
Department of Computer Science  
University of Oxford

A thesis submitted for the degree of  
*Doctor of Philosophy*  
Trinity 2013

# Abstract

Multiphysics systems with interface coupling are used to model a variety of physical phenomena, such as arterial blood flow, air flow around aeroplane wings, or interactions between surface and ground water flows. Numerical methods enable the practical application of these models through computer simulations. Specifically a high level of detail and accuracy is achieved in finite element methods by discretisations which use extremely large numbers of degrees of freedom, rendering the solution process challenging from the computational perspective. In this thesis we address this challenge by developing a twofold strategy for improving the efficiency of standard finite element coupled solvers. First, we propose to solve a monolithic coupled problem using block-preconditioned GMRES with a new Schur complement approximation. This results in a modular and robust method which significantly reduces the computational cost of solving the system. In particular, numerical tests show mesh-independent convergence of the solver for all the considered problems, suggesting that the method is well-suited to solving large-scale coupled systems. Second, we derive an adjoint-based formula for goal-oriented *a posteriori* error estimation, which leads to a time-space mesh refinement strategy. The strategy produces a mesh tailored to a given problem and quantity of interest. The monolithic formulation of the coupled problem allows us to obtain expressions for the error in the Lagrange multiplier, which often represents a physically relevant quantity, such as the normal stress on the interface between the problem components. This adaptive refinement technique provides an effective tool for controlling the error in the quantity of interest and/or the size of the discrete system, which may be limited by the available computational resources. The solver and the mesh refinement strategy are both successfully employed to solve a coupled Stokes-Darcy-Stokes problem modelling flow through a cartridge filter.



## Acknowledgements

My first and biggest thanks go to my supervisors, who have been more helpful, encouraging, patient and understanding than I could have ever imagined possible. The support they gave me during my time at Oxford was overwhelming. Dr David Kay has been an inspiration: a constant source of ideas and ready assistance. Discussions with Professor David Gavaghan made me consider the broader context of my work, and provided the necessary push whenever I got too caught up in the technical details. Dr Simon Tavener gave me an immense amount of his time and attention despite living halfway across the globe.

I am extremely grateful to Dr David Nordsletten for giving me an opportunity to work with him at King's College London. His dedication to science and the breadth of his knowledge are incredibly inspiring.

I was only able to undertake this research thanks to the generosity of the Oxford University Scatcherd European Fund. My MSc supervisor, Professor Endre Süli, helped me with my applications for both the DPhil place and the scholarship, for which I am extremely grateful. He also opened my eyes to the beauty of numerical methods. The idea of going into research was planted in my head by Dr Christoph Ortner, who was the most inspiring class tutor I have ever encountered, and who encouraged my interest in finite element methods.

Jackie Cameron at Linacre and Julie Sheppard at Comlab are the very definitions of efficient. It is thanks to them that I have never had to worry about administrative issues, and could focus entirely on my work.

My friends and family kept me (relatively) sane through it all. The amount of support I received from everyone was, frankly, staggering. Thank you, Mum and Dad, Lydia and Anatole, Benji and Arielle, Yulia and the Adaskins, the Zaliapins, Svet and Martin, Sasha and FX, Denis and Masha, Tom and Hazel, Lui and Olya, Marina and Lesha, Marina, Katya, Sara-Jane, Raf, Gareth, Sara, Mikael and Anna.

# Contents

|          |                                                                    |           |
|----------|--------------------------------------------------------------------|-----------|
| <b>1</b> | <b>Introduction</b>                                                | <b>1</b>  |
| <b>2</b> | <b>Finite element methods</b>                                      | <b>9</b>  |
| 2.1      | Weak derivatives and Sobolev spaces . . . . .                      | 10        |
| 2.2      | Weak formulations of differential equations . . . . .              | 12        |
| 2.3      | Finite element discretisation . . . . .                            | 15        |
| 2.4      | Mixed methods . . . . .                                            | 22        |
| 2.5      | Time stepping . . . . .                                            | 31        |
| 2.5.1    | Implicit Euler method . . . . .                                    | 32        |
| 2.5.2    | Discontinuous Galerkin method . . . . .                            | 33        |
| <b>3</b> | <b>Coupled problems</b>                                            | <b>37</b> |
| 3.1      | Partitioned solvers . . . . .                                      | 38        |
| 3.2      | Monolithic solvers . . . . .                                       | 40        |
| 3.3      | Elliptic coupled problem . . . . .                                 | 41        |
| 3.3.1    | Weak formulation . . . . .                                         | 42        |
| 3.3.2    | Finite element method . . . . .                                    | 43        |
| 3.4      | Parabolic coupled problem . . . . .                                | 46        |
| 3.4.1    | Weak formulation . . . . .                                         | 46        |
| 3.4.2    | Finite element method . . . . .                                    | 47        |
| <b>4</b> | <b>Efficient solution of coupled systems using preconditioners</b> | <b>50</b> |
| 4.1      | Exact block preconditioner . . . . .                               | 51        |
| 4.2      | Schur complement approximations . . . . .                          | 54        |
| 4.3      | Numerical convergence study . . . . .                              | 61        |
| 4.4      | Practical solvers . . . . .                                        | 67        |
| 4.5      | Practical solver for fluid-structure interaction . . . . .         | 70        |
| 4.5.1    | Coupled fluid dynamics and solid mechanics equations . . . . .     | 71        |
| 4.5.1.1  | Strong form . . . . .                                              | 73        |
| 4.5.1.2  | Weak form . . . . .                                                | 74        |
| 4.5.1.3  | Time discretisation and finite element method . . . . .            | 77        |
| 4.5.1.4  | Preconditioner . . . . .                                           | 82        |
| 4.5.1.5  | Simulation results . . . . .                                       | 82        |
| 4.6      | Discussion . . . . .                                               | 87        |

|          |                                                                           |            |
|----------|---------------------------------------------------------------------------|------------|
| <b>5</b> | <b><i>A posteriori</i> error estimation and adaptive mesh refinement</b>  | <b>89</b>  |
| 5.1      | Steady coupled problems . . . . .                                         | 90         |
| 5.1.1    | The error representation formula . . . . .                                | 91         |
| 5.1.2    | The effect of adjoint approximation . . . . .                             | 94         |
| 5.1.3    | An adaptive strategy . . . . .                                            | 97         |
| 5.1.4    | Numerical results . . . . .                                               | 99         |
| 5.1.4.1  | Error in the domain . . . . .                                             | 99         |
| 5.1.4.2  | Error along the interface . . . . .                                       | 102        |
| 5.1.4.3  | Mixed error . . . . .                                                     | 104        |
| 5.1.4.4  | Accuracy of the approximate error representation formula . . . . .        | 105        |
| 5.1.5    | Non-matching finite element spaces in $\Omega_1$ and $\Omega_2$ . . . . . | 106        |
| 5.1.5.1  | Error-based vs uniform refinements . . . . .                              | 109        |
| 5.2      | Time-dependent problems . . . . .                                         | 112        |
| 5.2.1    | The error representation formula . . . . .                                | 114        |
| 5.2.2    | The effect of discretization error on the boundary . . . . .              | 118        |
| 5.2.3    | The effect of adjoint approximation . . . . .                             | 119        |
| 5.2.4    | An adaptive strategy . . . . .                                            | 123        |
| 5.2.5    | Numerical examples of the effectivity of the error estimate . . . . .     | 126        |
| 5.2.5.1  | Domain error . . . . .                                                    | 127        |
| 5.2.5.2  | Estimating the error on the interface . . . . .                           | 130        |
| 5.2.6    | Numerical examples of adaptive mesh refinement . . . . .                  | 132        |
| 5.2.6.1  | Equal diffusion coefficients . . . . .                                    | 132        |
| 5.2.6.2  | Different diffusion coefficients . . . . .                                | 138        |
| 5.3      | Discussion . . . . .                                                      | 140        |
| <b>6</b> | <b>Coupling free flow and flow through porous media</b>                   | <b>143</b> |
| 6.1      | Weak equations and the finite element method . . . . .                    | 145        |
| 6.2      | Finite element spaces . . . . .                                           | 153        |
| 6.3      | Solver . . . . .                                                          | 154        |
| 6.4      | Error estimation and mesh refinement . . . . .                            | 156        |
| 6.5      | Results . . . . .                                                         | 160        |
| 6.5.1    | Straight channel . . . . .                                                | 160        |
| 6.5.1.1  | Solver . . . . .                                                          | 161        |
| 6.5.1.2  | Error estimation and mesh refinement . . . . .                            | 162        |
| 6.5.2    | Channel with linear expansion and contraction . . . . .                   | 163        |
| 6.5.2.1  | Solver . . . . .                                                          | 164        |
| 6.5.2.2  | Error estimation and mesh refinement . . . . .                            | 165        |
| 6.5.3    | Channel with sudden symmetric expansion and contraction . . . . .         | 167        |
| 6.5.3.1  | Solver . . . . .                                                          | 167        |
| 6.5.3.2  | Error estimation and mesh refinement . . . . .                            | 169        |
| 6.6      | Discussion . . . . .                                                      | 171        |

|          |                                   |            |
|----------|-----------------------------------|------------|
| <b>7</b> | <b>Conclusion</b>                 | <b>173</b> |
| 7.1      | Summary and conclusions . . . . . | 173        |
| 7.2      | Future work . . . . .             | 176        |
|          | <b>References</b>                 | <b>179</b> |

# List of Figures

|     |                                                                                                                                                                                                                                                                                                                                                                            |    |
|-----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 1.1 | Schematic representation of two types of coupling: (a) processes $F_1(u_1, u_2)$ and $F_2(u_1, u_2)$ occurring in the common domain $\Omega$ and (b) processes $F_1(u_1)$ and $F_2(u_2)$ occurring in individual domains $\Omega_1$ and $\Omega_2$ respectively, and interacting through $C(u_1, u_2)$ on $\Gamma$ . . . . .                                               | 2  |
| 2.1 | Domain $\Omega$ with boundary $\partial\Omega$ in $\mathbb{R}^2$ . . . . .                                                                                                                                                                                                                                                                                                 | 10 |
| 2.2 | Examples of geometric features of finite element methods in 2D: triangular and quadrilateral partitions, a hanging node, approximation of curvilinear boundaries. . . . .                                                                                                                                                                                                  | 16 |
| 2.3 | Linear basis functions on a reference triangle. . . . .                                                                                                                                                                                                                                                                                                                    | 17 |
| 2.4 | Quadratic basis functions on a reference triangle. . . . .                                                                                                                                                                                                                                                                                                                 | 18 |
| 3.1 | Two domains $\Omega_1$ and $\Omega_2$ sharing a common interface $\Gamma$ . . . . .                                                                                                                                                                                                                                                                                        | 37 |
| 4.1 | Uniform vs interface refinement of meshes that conform/do not conform on the interface $\Gamma$ . . . . .                                                                                                                                                                                                                                                                  | 62 |
| 4.2 | GMRES relative residual convergence for the practical method using $\mathbf{S}_4$ with AMG. The parabolic-elliptic coupled problem is solved on refinements of the nonconforming mesh ( $41 \times 41$ points in $\Omega_1$ and $23 \times 23$ points in $\Omega_2$ ) with GMRES $TOL = 10^{-10}$ , $\Delta t = 0.1$ . . . . .                                             | 69 |
| 4.3 | GMRES relative residual convergence for the practical method using $\mathbf{S}_5$ with AMG. The parabolic-elliptic coupled problem is solved on refinements of the nonconforming mesh ( $41 \times 41$ points in $\Omega_1$ and $23 \times 23$ points in $\Omega_2$ ) with GMRES $TOL = 10^{-10}$ , $\Delta t = 0.1$ . . . . .                                             | 70 |
| 4.4 | Schematic representation of the geometry in fluid-structure interaction. $\Omega_f$ is the fluid domain, $\Omega_s$ the solid domain, and $\Gamma$ the interface. . . .                                                                                                                                                                                                    | 71 |
| 4.5 | Sample meshes for the fluid-structure interaction problem. The triangulation is shown in the fluid domain $\Omega_f$ to distinguish it from the solid domain $\Omega_s$ . . . . .                                                                                                                                                                                          | 84 |
| 4.6 | Velocity profile and domain deformations for fluid-structure interaction in 2D. Inflow velocity at the bottom of $\Omega_f$ is $v_2 = 10X_1(1 - X_1)\sin^2(2\pi t)$ . Parameters: $\mu = 1$ , $\rho = 1$ , $\nu = 0.4$ , $E = 2900$ . Colour corresponds to velocity magnitude, and is scaled from blue (lowest) to red (highest) for each time step individually. . . . . | 85 |

|      |                                                                                                                                                                                                                                                                                                                                                                               |     |
|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 4.7  | Velocity profile and domain deformations for fluid-structure interaction in 2D. Inflow velocity at the bottom of $\Omega_f$ is $v_2 = 10X_1(1 - X_1)\sin^2(2\pi t)$ . Parameters: $\mu = 0.01$ , $\rho = 1$ , $\nu = 0.4$ , $E = 2900$ . Colour corresponds to velocity magnitude, and is scaled from blue (lowest) to red (highest) for each time step individually. . . . . | 85  |
| 5.1  | Original and adjoint solutions for the problems with the quantities of interest measuring the solution in the domain. . . . .                                                                                                                                                                                                                                                 | 100 |
| 5.2  | Refined mesh for the problems with quantities of interest measuring the solution in the domain. . . . .                                                                                                                                                                                                                                                                       | 101 |
| 5.3  | Adjoint solutions for a problem with the Lagrange multiplier as the quantity of interest. . . . .                                                                                                                                                                                                                                                                             | 103 |
| 5.4  | Adapted meshes for problems with error on the interface $\Gamma$ as the quantity of interest. . . . .                                                                                                                                                                                                                                                                         | 104 |
| 5.5  | Refinements of the initial meshes that do not match on the interface $\Gamma$ . . . . .                                                                                                                                                                                                                                                                                       | 107 |
| 5.6  | Refinements of the meshes for piecewise-quadratic finite element functions in $\Omega_1$ and piecewise-linear finite element functions in $\Omega_2$ . . . . .                                                                                                                                                                                                                | 110 |
| 5.7  | Refined meshes obtained using the error-representation formula for the error on the interface $\Gamma$ . . . . .                                                                                                                                                                                                                                                              | 111 |
| 5.8  | Snapshots of a Gaussian travelling along an elliptic trajectory. . . . .                                                                                                                                                                                                                                                                                                      | 133 |
| 5.9  | Example 5.2.6.1. Meshes obtained after a time block-wise refinement for the quantity of interest in the whole domain $\Omega_1 \cup \Omega_2$ (at most 350 elements allowed to be refined in a block based on local errors). . . . .                                                                                                                                          | 134 |
| 5.10 | Example 5.2.6.1. Meshes in sample blocks obtained after a time block-wise refinement for the quantity of interest in the whole domain $\Omega_1 \cup \Omega_2$ (at most $NMAX$ elements allowed to be refined in a block based on local errors). . . . .                                                                                                                      | 136 |
| 5.11 | Example 5.2.6.1. Meshes obtained after a time block-wise refinement for the quantity of interest on the interface $\Gamma$ (at most 700 elements allowed to be refined in a block based on local errors). . . . .                                                                                                                                                             | 137 |
| 5.12 | Snapshots of a Gaussian travelling along a trajectory constructed of two semi-ellipses. . . . .                                                                                                                                                                                                                                                                               | 139 |
| 5.13 | Example 5.2.6.2. Meshes obtained after a time block-wise refinement for the quantity of interest in the whole domain $\Omega_1 \cup \Omega_2$ with $k_1 = 0.2$ , $k_2 = 1$ . . . . .                                                                                                                                                                                          | 141 |
| 6.1  | Schematic representation of the subdomains for the problem modelling coupling of free fluid flow and flow through a porous medium. . . . .                                                                                                                                                                                                                                    | 145 |
| 6.2  | Straight channel with parabolic inflow, $\mu = 1$ , $\mathbf{K} = \mathbf{I}$ . . . . .                                                                                                                                                                                                                                                                                       | 161 |
| 6.3  | Channel with linear expansion and contraction, parabolic inflow, $\mu = 1$ , $\mathbf{K} = \mathbf{I}$ . . . . .                                                                                                                                                                                                                                                              | 164 |
| 6.4  | Channel with linear expansion and contraction, parabolic inflow, $\mu = 0.01$ , $K = 0.01$ . . . . .                                                                                                                                                                                                                                                                          | 166 |
| 6.5  | Channel with sudden symmetric expansion and contraction, parabolic inflow, $\mu = 1$ , $\mathbf{K} = \mathbf{I}$ . . . . .                                                                                                                                                                                                                                                    | 168 |

|     |                                                                                                                                                                       |     |
|-----|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 6.6 | Channel with sudden symmetric expansion and contraction, parabolic inflow, $\mu = 0.01$ , $K = 0.01$ . . . . .                                                        | 169 |
| 6.7 | Channel with sudden symmetric expansion and contraction, parabolic inflow. Quantity of interest approximates point values of fluid pressure near the corners. . . . . | 171 |

# Notation

## *Indices*

- $D$  – dimension of the spatial variable
- $d$  – index of a spatial variable component
- $m, M$  – number of spatial degrees of freedom in a finite-dimensional system
- $i, j$  – index of a spatial degree of freedom in a finite-dimensional system
- $N$  – number of time steps
- $n$  – index of a time step

## *Domains, boundaries, interfaces*

- $\mathbb{R}^D$  – Euclidean space of dimension  $D$
- $\Omega$  – spatial domain
- $\partial\Omega$  – boundary of domain  $\Omega$
- $\Gamma$  – interface between domains/part of domain boundary
- $\mathbf{n}$  – outward-facing unit normal vector
- $\boldsymbol{\tau}$  – tangential unit vector
- $\tau(\Omega)$  – triangulation of domain  $\Omega$
- $DOF$  – number of degrees of freedom in a discretised system

## *Variables and quantities of interest*

- $\mathbf{x}$  – space/coordinate on the deformed mesh in ALE
- $\mathbf{X}$  – coordinate on the reference mesh in ALE
- $t$  – time
- $u$  – solution function in elliptic and parabolic problems
- $\phi$  – basis function of a finite element space
- $\lambda$  – Lagrange multiplier
- $\mathbf{u}$  – fluid velocity/solid deformation
- $p$  – pressure
- $\mathbf{v}$  – fluid velocity
- $\mathbf{w}$  – domain velocity
- $\mathbf{d}$  – mesh deformation
- $\boldsymbol{\sigma}$  – Cauchy stress tensor
- $\mathbf{J}$  – domain transform
- $J$  – determinant of the domain transform
- $(QoI)$  – quantity of interest in error estimation
- $\mathcal{E}$  – error in the quantity of interest



- $\epsilon$  – error in the *a posteriori* estimate
- $\mathcal{R}$  – effectivity ratio:  $\mathcal{R} = 1 - \epsilon/\mathcal{E}$

*Functions, operators and forms*

- $\mathcal{F}$  – abstract operator
- $\delta_{i,j}$  – Kronecker delta, i.e.  $\delta_{i,j} = 1$  if  $i = j$ , 0 otherwise
- $\nabla$  – gradient:  $\nabla u = \left( \frac{\partial u}{\partial x_1}, \dots, \frac{\partial u}{\partial x_d} \right)^T$
- $(\cdot, \cdot)$  – scalar product in  $L^2(\Omega)$ , i.e.  $(w, v) = \int_{\Omega} wv \, d\mathbf{x} \, \forall w, v \in L^2(\Omega)$
- $a(\cdot, \cdot)$  – bilinear form
- $[u]_n$  – jump in  $u$  at time  $t_n$ , i.e.  $[u]_n = \lim_{t \rightarrow t_n+0} u(t) - \lim_{t \rightarrow t_n-0} u(t)$
- $\pi$  – spatial projection operator
- $\theta$  – temporal projection operator

*Matrices and vectors*

- $\mathbf{A}$  – single system matrix/fluid velocity block
- $\mathbf{B}$  – off-diagonal block in a saddle-point problem/fluid pressure block
- $\mathbf{C}$  – coupling matrix
- $\mathbf{F}$  – source term vector
- $\mathbf{K}$  – stiffness matrix
- $\mathbf{M}$  – mass matrix
- $\mathbf{P}$  – preconditioner/discrete fluid pressure vector
- $\mathbf{S}$  – Schur complement
- $\mathbf{U}$  – discrete solution vector/discrete fluid velocity vector
- $\mathbf{\Lambda}$  – discrete Lagrange multiplier vector

*Function spaces*

- $L^2(\Omega)$  – square-integrable functions on  $\Omega$ ,  
i.e.  $L^2(\Omega) = \left\{ v : \Omega \rightarrow \mathbb{R} \mid \left( \int_{\Omega} |v|^2 \, d\mathbf{x} \right)^{1/2} < \infty \right\}$
- $C(\Omega)$  – continuous functions on  $\Omega$
- $C^1(\Omega)$  – continuously differentiable functions on  $\Omega$
- $C^2(\Omega)$  – twice continuously differentiable functions on  $\Omega$
- $C^\infty(\Omega)$  – infinitely continuously differentiable functions on  $\Omega$
- $H^1(\Omega)$  – functions that are square-integrable and have square-integrable first weak derivatives
- $H_D^1(\Omega)$  – functions in  $H^1(\Omega)$  satisfying a given Dirichlet boundary condition
- $H_0^1(\Omega)$  – functions in  $H^1(\Omega)$  satisfying the zero Dirichlet boundary condition on a given boundary of  $\Omega$
- $H(\text{div}, \Omega)$  – square-integrable 2D functions with square-integrable divergence
- $X, Y$  – continuous function spaces
- $X^h, Y^h$  – discrete (finite element) spaces
- $P^k(\Omega)$  – continuous piecewise-polynomials of degree  $\leq k$
- $P_d^k(\Omega)$  – discontinuous piecewise-polynomials of degree  $\leq k$

*Parameters*

|              |   |                                                           |
|--------------|---|-----------------------------------------------------------|
| $k$          | – | stiffness coefficient                                     |
| $\mu$        | – | fluid viscosity                                           |
| $\rho$       | – | fluid density                                             |
| $E$          | – | Young’s modulus                                           |
| $\nu$        | – | Poisson ratio                                             |
| $\mathbf{D}$ | – | stress-strain relationship tensor for linear elasticity   |
| $\mathbf{K}$ | – | permeability tensor for flow through a porous medium      |
| $T$          | – | final time                                                |
| $TOL$        | – | global tolerance                                          |
| $LATOL$      | – | local tolerance                                           |
| $NMAX$       | – | highest number of elements allowed in a single time block |

# Chapter 1

## Introduction

Mathematical modelling has become an essential tool in building our understanding of physical processes, benefitting from the constantly increasing availability of computational resources. Numerical methods enable the practical application of mathematical models in science, research and engineering through the use of computer simulations. The enhancement of computational techniques has made it possible to address problems in a variety of fields from meteorology to cell biology. In this work we aim to develop efficient numerical methods for solving coupled multiphysics problems.

The term ‘coupled systems’ may cover a wide range of mathematical problems which involve two or more distinct components interacting with each other. We consider coupling in the context of partial differential equations (PDEs), which are used to model a multitude of physical processes through the mathematical representation of the fundamental conservation laws. From the modelling perspective we can distinguish coupled processes sharing a common environment and those interacting only through a shared interface. A well-known example of the first type is the bidomain model in cardiac electrophysiology [71, 83], where homogenised equations for conservation of ion current and microscopic equations for ion channel behaviour are solved

in the same domain for intra- and extracellular potential. A popular problem of the second type is fluid-structure interaction, where the interaction occurs solely through the fluid-structure interface. Schematically the difference between these types of coupling can be represented as shown in Figure 1.1.

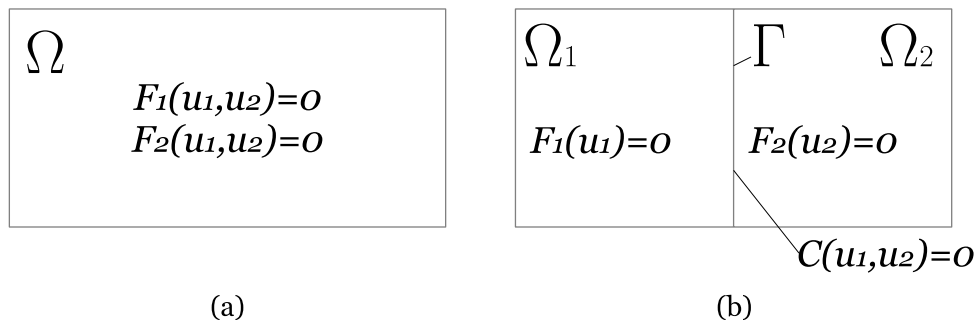


Figure 1.1: Schematic representation of two types of coupling: (a) processes  $F_1(u_1, u_2)$  and  $F_2(u_1, u_2)$  occurring in the common domain  $\Omega$  and (b) processes  $F_1(u_1)$  and  $F_2(u_2)$  occurring in individual domains  $\Omega_1$  and  $\Omega_2$  respectively, and interacting through  $C(u_1, u_2)$  on  $\Gamma$ .

In this work we consider the systems where coupling occurs through an interface. The interest in such problems is driven by applications in a variety of engineering fields. Fluid-structure interaction in particular is used to model air flow around airplane wings, boat sails, tall buildings, bridges, as well as blood flow in human arteries or the heart, and air flow in the airways or the lungs [43, 55, 79, 85, 94, 117, 122, 126]. Coupled models of free flow and flow through porous media are used in the study of interactions between surface and groundwater flows, as well as air and oil filters [40, 69, 76, 84]. Computer simulations of these and other coupled systems are of immediate practical interest.

The complexity of coupled multiphysics models usually means that an analytic solution is not available. Numerical methods provide an incredibly powerful alternative way of solving PDEs using computer software. The main idea behind such methods lies in discretisation of the operators or the solution space. The approximation of the

original continuous problem by its finite-dimensional counterpart is a fundamental step in reducing the underlying PDEs to a system of algebraic equations, for which many solution methods exist [67, 104]. However, the need for accurate solutions computed with a high level of detail means that the size of the discrete problems can be exceptionally high, and the computational cost of producing a solution in such a problem is extreme. It can be argued that the increasing computational complexity is more than matched by the increasing availability and reducing cost of resources, however, additional challenges arise in high-performance systems, which can be both software and hardware-related [36, 42, 109]. Optimisation of numerical methods used to solve a particular problem is a way to make sure that the best possible results are achieved given available resources. The ability to reduce computation time by algorithmic means can also lead to more extensive and more effective use of models, as the demand for real-time simulation exists in fields like surgery [35, 107].

The task of reducing the computational complexity can be approached from two perspectives: the discretisation, and the linear solver. The generation of a computational mesh for a given problem geometry is usually directly related to the accuracy of the computed solution and the operational cost of the solution method. Specifically in finite element methods the formulation allows the derivation of *a priori* error estimates for the approximate solution of sufficiently regular problems [21, 78]. While providing important information on convergence behaviour, *a priori* estimates can not be used for practical error control and mesh refinement, since they rely on our knowledge of the exact solution. At the same time, *a posteriori* error estimation is based entirely on our ability to compute the approximate solution, and can be used to assess solution accuracy directly.

There are several ways to perform *a posteriori* error estimation. Gradient re-

covery methods [129, 128, 127] are based on the idea of replacing the true solution gradient in the *a priori* estimate with an approximation arising from the numerical solution. This class of methods is widely used due to the relative simplicity of the idea. However, the accuracy of the estimator is not guaranteed by rigorous proofs in most cases, even if it is sometimes observed in practice. Local error estimators use the Galerkin orthogonality property to impose a bound on the error by solving local residual problems [9, 17, 74]. A detailed review of these methods, as well as further references can be found in [2, 3, 120].

An alternative approach to *a posteriori* error estimation relies on the solution of a global adjoint problem [47, 64, 99]. In particular, [47, 49, 50, 123] address the goal-oriented technique, which is based on solving an adjoint problem to estimate the error in a quantity of interest given by a functional of the solution, as opposed to the error in a norm of the solution. The functional often represents a value of practical interest, e.g. a weighted average of a variable in the domain.

*A posteriori* error estimation has an additional advantage in that it can be used for adaptive mesh refinement [9, 12, 30, 37, 101, 120]. While *a priori* error bounds normally indicate that refining the mesh leads to reducing the error, if we can compute local *a posteriori* estimates then specific areas or time steps can be refined – those where the errors are the highest. Unless the errors are distributed perfectly uniformly, an adaptive refinement is likely to produce a significantly smaller mesh than a uniform refinement, allowing us to recompute a more accurate solution faster or using fewer computational resources than would be required for a computation on a uniformly refined mesh. Of course, the *a posteriori* procedures themselves can incur a significant computational cost. It is therefore necessary to consider the suitability of such algorithms for a specific practical problem. In large-scale problems performing

*a posteriori* error estimation on a relatively coarse mesh, and producing a computational mesh which is best suited to the solution can be advantageous [97, 113, 124]. In this work we do not consider an alternative finite element refinement strategy, *p*-refinement, where better local accuracy is achieved through increasing the order of the finite elements instead of (or alongside) refining the mesh. For details on this approach we refer to [8, 10, 80, 108].

We focus on goal-oriented *a posteriori* error estimation in the context of coupled problems. In particular, we are interested in estimating quantities of interest on the coupling interface, which can model, for instance, wall traction in fluid-structure interaction – an important metric for practical considerations.

A redistribution of the system’s degrees of freedom is aimed at achieving higher solution accuracy provided there are limited computational resources, or using the least computational resources given a certain accuracy that the solution needs to achieve. When using adaptive refinement we assume that there is a fixed relationship between the amount of memory and processing power available for computations, and the size of the discrete system that we can solve. This relationship, however, can change dramatically depending on the linear solver that one chooses to employ.

A straightforward way of addressing a matrix problem is to use a direct solver, e.g. the *LU* factorisation [67]. The implementation is independent of the form of the system matrix, and is readily available in most software libraries for matrix computations\*. Moreover, the method does not introduce additional errors into the finite element solution, since it solves the discretised problem up to machine precision. However, when applied to large-scale sparse problems, direct methods become extremely computationally demanding, both in terms of processing power and required memory.

---

\*For a study comparing different implementations of direct methods we refer to [68].

The standard implementation of the  $LU$  factorisation requires  $\mathcal{O}(N^3)$  operations for an  $N \times N$  matrix, and while the power of  $N$  in the operation count can be reduced for sparse matrices, it is often prohibitive for large matrices that arise in practical applications.

Even if the resources are available factorisation time can become exceedingly long. It is therefore common to use iterative methods, whereby the discrete solution is not obtained exactly, but up to a certain prescribed tolerance on a norm of the residual. Specifically, Krylov subspace methods, such as GMRES [105] or BiCGSTAB [119] can provide a reliable and efficient alternative to direct methods [25]. The convergence of these methods, and therefore the feasibility of their use, is highly dependent on the matrix of the system, which can become very ill-conditioned for a refined finite element problem. It is therefore essential to use a good preconditioner in such problems [82, 91, 95, 111]. The choice of such a preconditioner is therefore determined by the two primary factors: the effect on the iterative method's convergence, and the cost of building it and using it in computations. In this work our aim is to find a preconditioner that would be suited to solving coupled problems specifically.

The overall structure of the thesis is as follows. First we give an overview of the mathematical tools which are necessary for the development of the methods that constitute the body of this work. Chapter 2 outlines the basic concepts of the standard continuous Galerkin finite element methods including the weak form of partial differential equations, finite element discretisation and error bounds. Two aspects are addressed separately: formulations for mixed problems, such as incompressible flow, and time-dependence.

Chapter 3 considers the merits and drawbacks of the partitioned and the monolithic approaches to solving generic coupled problems. We propose that the main



benefit of the partitioned approach, namely its modularity, can be gained in monolithic coupled solvers through the use of block preconditioners. In order to study this option in detail we formulate the weak and the finite element problems for equations coupled using Lagrange multipliers.

Block preconditioners for the monolithic coupled systems formulated in Chapter 3 are investigated in Chapter 4. Following a standard procedure of block Gaussian elimination we arrive at a modified block system where individual solution procedures for component blocks can be used. However, we note that the construction and use of one of the blocks of this modified system – the Schur complement – is excessively costly from the computational perspective, and not scalable for large problems. It is therefore argued that an approximation to the Schur complement should be used to make the solver both modular and efficient. Approximations which do not rely on the underlying physics of the components are put forward alongside an approximation by Elman [45]. We discuss the computational complexity of the associated solution methods, and study their performance numerically. A practical solver for coupled problems based on Schur complement approximation and the algebraic multigrid method [24] is described and tested, particularly for a multiphysics problem modelling fluid-structure interaction.

Chapter 5 looks at *a posteriori* error estimation as means of assessing the accuracy of solutions in coupled problems, and using this error to drive adaptive spatial and temporal refinement of the discrete system. The aim of this procedure is to identify the regions and time intervals where the errors are highest, and reduce them through local refinement. First we consider steady coupled problems, for which we derive error estimates based on solutions of adjoint problems, assess the accuracy of these estimates and formulate a refinement algorithm based on local error estimates. The

convergence of the error estimate is verified in a number of examples. Further, the adaptive strategy is tested for a succession of problems with various quantities of interest, including those supported on the coupling interface, with specific attention given to problems with finite element spaces that do not match along the interface. We also highlight the advantage of using an adaptive refinement strategy in the problems where the errors are known to be highly localised. The second part of Chapter 5 develops the theory for time-dependent coupled problems. The appropriate error estimate is derived, and the accuracy of this estimate is established. An adaptive strategy is obtained where it is possible to distinguish spatial and temporal errors, and therefore refine both the time intervals and the spatial mesh. Space-time blocks are utilised to improve the method's efficiency. The estimator and the refinement strategy are then tested in a number of numerical examples with different quantities of interest.

The methods developed in Chapters 4 and 5 are re-examined in a different multi-physics setting in Chapter 6. We consider a two-dimensional model of flow through a filter represented by a channel with a porous section in the middle. Taking into account the different nature of the governing equations, and the presence of three components, we reformulate the expression for the solver and the *a posteriori* error estimate. Numerical tests are then carried out to assess the effect of the different physics and geometry on the performance of the methods including the refinement strategy.

Finally, the results are summarised and discussed in Chapter 7. We also outline the areas where further work would be necessary to either evaluate the methods' applicability, develop them further, or facilitate their practical use.

# Chapter 2

## Finite element methods

Numerical methods are an established means of solving differential equations that are of practical interest in a variety of applied problems. Finite difference, finite volume and finite element methods are the most widely used types of such methods. Their basic idea is replacing the original infinite-dimensional problem by a finite-dimensional approximation, which is, generally speaking, easier to solve.

The main idea of finite difference methods is to approximate the derivatives appearing in the equations with finite differences at a set of discrete points in the domain of interest. Their advantages include relative simplicity of derivation and implementation. However, finite difference methods are best suited for simple domain shapes and regular meshes, while using them for problems on complex geometries requires non-standard treatment even for standard differential equations. For an introduction to finite differences in partial differential equations we refer to [90, 116, 114].

Finite volume methods [87, 121] rely on the application of the divergence theorem within small subdomains of the original domain – finite volumes – and approximating fluxes through volume boundaries. These methods are traditionally used in the field of computational fluid dynamics due to their conservative properties, although

formulating problems on complex geometries is again far from trivial.

Finite element methods are based on weakening the restrictions on the solution space in the continuous setting, and searching for the approximate solution in the subspace which spans basis functions supported on small regions inside the domain. These methods are well-suited to solving problems on complex domains, and are therefore widely used in practical applications.

In this work we consider only finite element methods (FEMs) for solving partial differential equations (PDEs), even though some ideas, namely those relating to preconditioning of block systems, are also applicable to problems discretised otherwise. This chapter comprises an overview of several theoretical and practical aspects of the classical FEMs. The theory and notation presented here are essential in developing the techniques that form the core of this thesis.

## 2.1 Weak derivatives and Sobolev spaces

Let us consider an open bounded domain  $\Omega \subset \mathbb{R}^D$  with a smooth boundary  $\partial\Omega$ . While the general theory applies to any dimension  $D$ , we restrict the problems we consider to the physically relevant two- and three-dimensional Euclidean space.

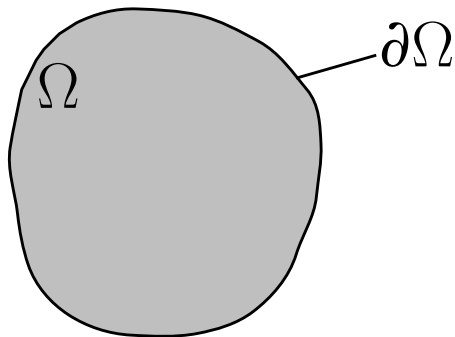


Figure 2.1: Domain  $\Omega$  with boundary  $\partial\Omega$  in  $\mathbb{R}^2$ .

We define the standard function spaces:  $L^2(\Omega)$  containing all Lebesgue square in-

tegrable functions on  $\Omega$ ,  $C(\Omega)$  containing all continuous functions on  $\bar{\Omega}$ ,  $C^1(\Omega)$ ,  $C^2(\Omega)$  and  $C^\infty(\Omega)$  containing all once-, twice- and infinitely continuously differentiable functions on  $\bar{\Omega}$  respectively. The classical, or, in the context of FEMs, *strong* formulations of second-order differential equations usually imply that the solutions lie in  $C^2(\Omega)$ , which in practice is often too restrictive. FEMs are derived from what is known as *weak* formulations of differential equations, which only assume weak differentiability of solutions.

Let  $\Omega \subset \mathbb{R}$ , then a function  $u \in L^2(\Omega)$  is said to have a weak derivative  $Du \in L^2(\Omega)$  if

$$\int_{\Omega} u \frac{dv}{dx} dx = - \int_{\Omega} Du v dx \quad \forall v \in C^\infty(\Omega) \text{ such that } v = 0 \text{ on } \partial\Omega.$$

If  $u \in C^1(\Omega)$  this definition is consistent with the integration by parts formula, and the definition of the classical derivative of  $u$ . It is therefore common to use the same notation for the weak and classical derivatives:

$$\frac{du}{dx} = Du.$$

The definition is extended naturally to multiple dimensions. If  $\alpha$  is a  $D$ -dimensional multi-index, i.e.  $\alpha = (\alpha_1, \dots, \alpha_D)$  with integer  $\alpha_d$ ,  $d = 1, \dots, D$ , then  $u$  is said to have a weak derivative  $D^\alpha u$  of order  $\alpha$  if

$$\int_{\Omega} u \frac{\partial^{|\alpha|} v}{\partial x_1^{\alpha_1} \dots \partial x_D^{\alpha_D}} d\mathbf{x} = (-1)^{|\alpha|} \int_{\Omega} D^\alpha u v d\mathbf{x} \quad \forall v \in C^\infty(\Omega) \text{ such that } v = 0 \text{ on } \partial\Omega.$$

This definition of weak derivatives allows us to introduce a *Sobolev space* that will be

used throughout the thesis:

$$H^1(\Omega) = \{u \in L^2(\Omega) \mid Du \in L^2(\Omega)\},$$

the space of square-integrable functions that have square-integrable weak derivatives of the first order. For our purposes it is sufficient to consider this and a few other specific  $L^2$ -based Sobolev spaces. The vast theory behind more general spaces is in the purview of the field of functional analysis, and beyond the scope of this work. Formal definitions, and a detailed study of properties of Sobolev spaces can be found, for example, in [1, 21].

## 2.2 Weak formulations of differential equations

As an example, let us consider an elliptic boundary value problem in  $\Omega$ :

$$-\nabla \cdot (k \nabla u) + cu = f \quad \text{in } \Omega, \tag{2.1a}$$

$$u = u_D \quad \text{on } \Gamma_D, \tag{2.1b}$$

$$k \nabla u \cdot \mathbf{n} = g_N \quad \text{on } \Gamma_N. \tag{2.1c}$$

Here  $u : \Omega \rightarrow \mathbb{R}$ ,  $\nabla$  is the classical gradient operator,  $k$ ,  $c$ ,  $f$ ,  $u_D$  and  $g_N$  are given functions of appropriate regularity,  $\mathbf{n}$  is the outward pointing unit normal vector to the appropriate boundary. The boundary of  $\Omega$  is divided into two parts which correspond to Dirichlet and Neumann boundary conditions:  $\partial\Omega = \Gamma_D \cup \Gamma_N$ , and  $\Gamma_D \cap \Gamma_N = \emptyset$ . The strong form of (2.1a) requires  $u$  to be at least twice differentiable. To weaken the regularity restrictions we multiply the equation by an arbitrary function  $v$ , called

a *test function*, and integrate over  $\Omega$ :

$$\int_{\Omega} (-\nabla \cdot (k \nabla u) + cu) v \, d\mathbf{x} = \int_{\Omega} f v \, d\mathbf{x}.$$

Assuming the conditions of the divergence theorem are satisfied, the left-hand side of this equation can be rewritten:

$$\begin{aligned} \int_{\Omega} (-\nabla \cdot (k \nabla u) + cu) v \, d\mathbf{x} &= - \int_{\partial\Omega} (k \nabla u \cdot \mathbf{n}) v \, ds + \int_{\Omega} (k \nabla u \cdot \nabla v + cuv) \, d\mathbf{x} \\ &= - \int_{\Gamma_D} (k \nabla u \cdot \mathbf{n}) v \, ds - \int_{\Gamma_N} g_N v \, ds + \int_{\Omega} (k \nabla u \cdot \nabla v + cuv) v \, d\mathbf{x}. \end{aligned}$$

Taking note of the Dirichlet condition (2.1b) for  $u$ , and letting  $v = 0$  on  $\Gamma_D$ , we arrive at the following equation:

$$- \int_{\Gamma_N} g_N v \, ds + \int_{\Omega} (k \nabla u \cdot \nabla v + cuv) \, d\mathbf{x} = \int_{\Omega} f v \, d\mathbf{x}.$$

In this equation the second derivatives of  $u$  need not exist, moreover, the gradients only need to exist in the weak sense. With that in mind, both the solution and the test functions can come from  $H^1(\Omega)$ , as long as they satisfy the appropriate Dirichlet boundary conditions. The following notation will be used for convenience and consistency:

$$X = H^1(\Omega),$$

$$X_D = H_D^1(\Omega) = \{v \in X \mid v = u_D \text{ on } \Gamma_D\},$$

$$X_0 = H_0^1(\Omega) = \{v \in X \mid v = 0 \text{ on } \Gamma_D\}.$$

The Neumann condition (2.1c) is included naturally in the weak equation. The weak formulation of (2.1) is as follows:

*Find  $u \in X_D$  such that*

$$\int_{\Omega} (k \nabla u \cdot \nabla v + cuv) \, d\mathbf{x} = \int_{\Omega} f v \, d\mathbf{x} + \int_{\Gamma_N} g_N v \, ds \quad \forall v \in X_0. \quad (2.2)$$

Introducing a symmetric bilinear form  $a(\cdot, \cdot) : X \times X \rightarrow \mathbb{R}$  for the left-hand side

$$a(w, v) = \int_{\Omega} (k \nabla w \cdot \nabla v + cuv) \, d\mathbf{x}, \quad (2.3)$$

and using the standard notation  $(\cdot, \cdot)$  for the scalar product in  $L^2(\Omega)$ , and  $\langle \cdot, \cdot \rangle_{\Gamma_N}$  for the dual product on  $\Gamma_N$  we write the same weak equation in shorthand:

*Find  $u \in X_D$  such that*

$$a(u, v) = (f, v) + \langle g_N, v \rangle_{\Gamma_N} \quad \forall v \in X_0. \quad (2.4)$$

Existence and uniqueness of such a  $u$  follows from the Lax-Milgram theorem (theorem 2.7.7 in [21]) formulated below.

**Theorem 1.** (*Lax-Milgram*) *Let  $X$  be a Hilbert space,  $F$  be a continuous linear functional on  $X$ , and  $a(\cdot, \cdot)$  be a continuous and coercive bilinear form on this space, i.e.  $\exists C > 0, \alpha > 0$  such that*

$$a(v, w) \leq C \|v\|_X \|w\|_X \quad \forall v, w \in X \quad (\text{continuity}),$$

$$a(v, v) \geq \alpha \|v\|_X^2 \quad \forall v \in X \quad (\text{coercivity}).$$

*Then there exists a unique  $u \in X$  such that  $a(u, v) = F(v) \quad \forall v \in X$ .*



In terms of the weak problem (2.4) the continuity is satisfied by the Cauchy-Schwartz inequality, and coercivity holds when  $k, c > 0$ . The linear functional  $F(v) = (f, v) + \langle g_N, v \rangle$  is continuous on  $X_0$  if the data are sufficiently regular, e.g.  $f, g_N \in L^2(\Omega)$ . However, the test and trial spaces are not the same  $X_D \neq X_0$ . However, this can be remedied by rewriting the problem in terms of  $\hat{u} = u - \hat{u}_D$ , where  $\hat{u}_D \in X_D$  is an extension of  $u_D$  onto the whole domain  $\Omega$  such that  $\|\hat{u}_D\|_X$  is minimised. In this case (2.4) can be equivalently rewritten:

*Find  $\hat{u} \in X_0$  such that*

$$a(\hat{u}, v) = (f, v) - a(\hat{u}_D, v) + \langle g_N, v \rangle_{\Gamma_N} \quad \forall v \in X_0. \quad (2.5)$$

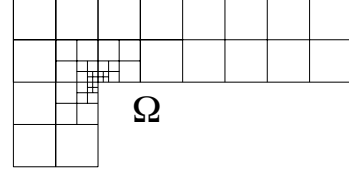
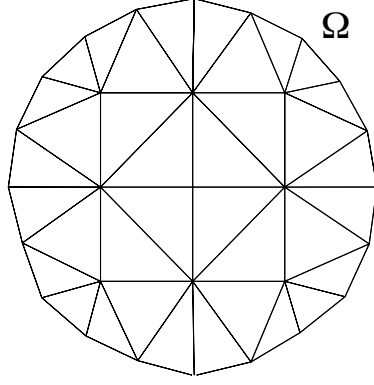
The conditions of Lax-Milgram theorem are satisfied for problem (2.5), and therefore a unique solution  $\hat{u} \in X_0$  exists for it. This implies the existence and uniqueness of  $u$  satisfying (2.4).

## 2.3 Finite element discretisation

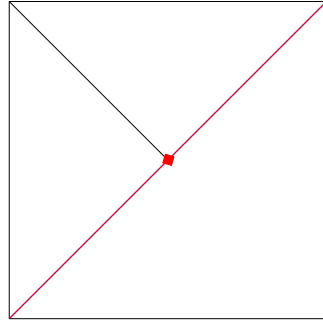
The transition from an infinite-dimensional to a finite-dimensional problem is what distinguishes different numerical methods for solving differential equations. The FE approximation space is constructed as follows: first, the problem domain is partitioned into small *element domains*, and second, the *element* is defined by prescribing for each element domain a set of nodes and nodal values, effectively defining basis functions of the approximation space given their general form (for example, as piecewise-linears).

Element domains are normally shaped as triangles or squares in  $\mathbb{R}^2$ , tetrahedra or hexahedra in  $\mathbb{R}^3$  (Fig. 2.2a). They must have non-overlapping interiors, and their boundaries can only intersect along common edges or at common vertices, i.e. there

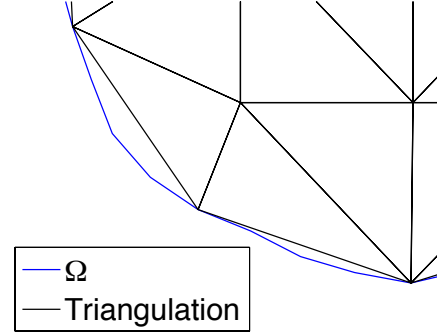
must be no hanging nodes (Fig. 2.2b).



(a) Partition



(b) Hanging node



(c) Curvilinear boundary

Figure 2.2: Examples of geometric features of finite element methods in 2D: triangular and quadrilateral partitions, a hanging node, approximation of curvilinear boundaries.

For the case of triangles or tetrahedra, element domains with extremely acute or obtuse angles may cause the discretised system to become numerically singular, and it is therefore important that the use of such elements is avoided. In addition, if  $\Omega$  has curvilinear boundaries (Fig. 2.2c) one has to be aware of the additional error introduced into the solution as a result of domain approximation by a polygon or a polyhedron. In this work we assume that the domain can be triangulated exactly, unless stated otherwise.

The partition of  $\Omega$  into admissible element domains is known as domain *triangu-*

*lation* or *tesselation*. All the nodes, edges and faces of element domains constitute the problem *mesh*.

Defining local basis functions and nodal values completes the definition of a FE space (see e.g. Figures 2.3-2.4 for Lagrangian linear and quadratic local bases on triangles). For a rigorous definition of finite elements, and a description of different types of elements we refer to [21]. We note that in the classical FE framework the solution is approximated using continuous piecewise-polynomial functions. To preserve a certain level of clarity, we consider low-order Lagrangian elements, although this is non-essential for the methods developed in the thesis.

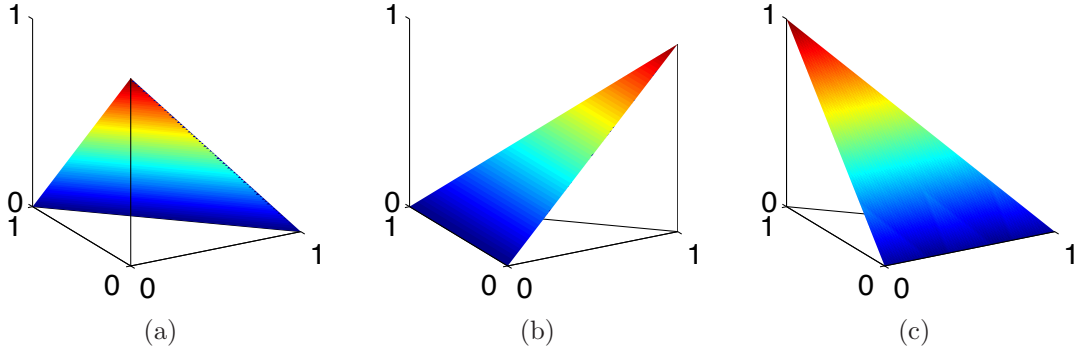


Figure 2.3: Linear basis functions on a reference triangle.

With all necessary components in place, we can write the FE discretisation of the weak problem (2.4). Let  $X^h$  be the approximation space on a triangulation  $\tau$  of  $\Omega$ , and let  $X_D^h = X^h \cap X_D$  and  $X_0^h = X^h \cap X_0$ . For simplicity we will only consider boundary data which is in the discrete space, i.e.  $u_D \in X^h$ , unless otherwise stated. The FEM for (2.4) is obtained by restricting the original function space to its finite-dimensional counterpart:

*Find  $u^h \in X_D^h$  such that*

$$a(u^h, v^h) = (f, v^h) + \langle g_N, v^h \rangle_{\Gamma_N} \quad \forall v^h \in X_0^h. \quad (2.6)$$

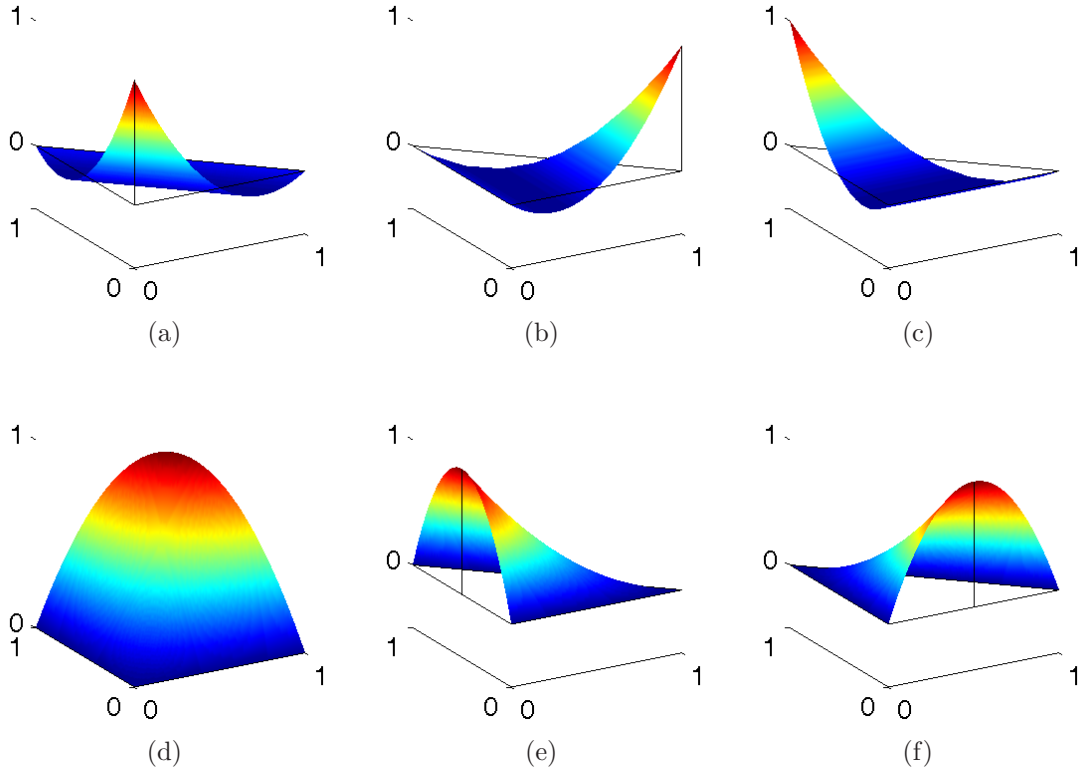


Figure 2.4: Quadratic basis functions on a reference triangle.

As in the continuous setting, the Lax-Milgram theorem guarantees existence and uniqueness of solutions for problems with coercive  $a(\cdot, \cdot)$  (provided the data are sufficiently regular, and by considering an associated problem with homogeneous Dirichlet conditions). The following result, known as Céa's theorem (theorem 2.8.1 in [21]), also provides a theoretical error bound for an approximate solution  $u^h$ .

**Theorem 2.** *Let  $u \in X_D$  solve (2.4) and  $u^h \in X_D^h$  solve (2.6), and let the bilinear form  $a(\cdot, \cdot)$  be continuous with constant  $C > 0$  and coercive with constant  $\alpha > 0$ . Then*

$$\|u - u^h\|_X \leq \frac{C}{\alpha} \min_{v^h \in X_0^h} \|u - v^h\|_X.$$

The theorem shows that the approximate solution  $u^h$  is quasi-optimal in  $X^h$ , being proportional to the best approximation from this subspace of  $X$ . Given a triangulation

$\tau$  of the domain  $\Omega$ , let  $X^h$  be the space of continuous piecewise-polynomial functions of degree no greater than  $k$ , i.e.  $X^h = P^k(\Omega)^*$ . We choose the Lagrangian basis  $\{\phi_1, \phi_2, \dots, \phi_m\}$  of  $X^h$  defined by the nodal values at the nodes  $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m\}$ , namely

$$\phi_i(\mathbf{x}_j) = \delta_{i,j} = \begin{cases} 1, & i = j, \\ 0, & i \neq j. \end{cases}$$

We observe that a basis of  $X_0^h$  can be constructed by removing  $\phi_i$  with  $\mathbf{x}_i \in \Gamma_D$  from the basis of  $X^h$ . Let us assume that the indices of such basis functions are  $M+1, \dots, m$ , and therefore  $X_0^h = \text{span}\{\phi_1, \dots, \phi_M\}$ . The finite-dimensional weak problem (2.6) is equivalent to:

*Find  $u^h \in X_D^h$  such that*

$$a(u^h, \phi_i) = (f, \phi_i) + \langle g_N, \phi_i \rangle_{\Gamma_N} \quad \forall i = 1, \dots, M. \quad (2.7)$$

Any function from  $X^h$  can be presented in the form of a basis expansion. Let this basis expansion for  $u^h$  be

$$u^h = \sum_{j=1}^m U_j \phi_j$$

with  $U_j = u^h(\mathbf{x}_j)$ . Substituting this expression into (2.7), and using the fact that  $a(\cdot, \cdot)$  is bilinear, we obtain

$$\sum_{i=1}^m a(\phi_i, \phi_j) U_i = (f, \phi_j) + \langle g_N, \phi_j \rangle_{\Gamma_N} \quad \forall j = 1, \dots, M. \quad (2.8)$$

Moreover, the restriction  $u^h \in X_D^h$  imposes additional conditions on the basis coefficients of  $u^h$ , namely  $U_i = u_D(\mathbf{x}_i)$  for  $i = M+1, \dots, m$ .

---

\*Functions in  $P^k(\Omega)$  are polynomials on each element domain in  $\tau$ .

We introduce the following matrix notation:  $\mathbf{A} = (a_{i,j}) \in \mathbb{R}^{m \times m}$  is the system matrix,  $\mathbf{F} = (F_1, \dots, F_m)^T$  is the load vector,  $\mathbf{U} = (U_1, \dots, U_m)^T$  is the solution vector, where

$$a_{i,j} = \begin{cases} a(\phi_i, \phi_j) & \forall i = 1, \dots, M, j = 1, \dots, m, \\ \delta_{i,j} & \forall i = M+1, \dots, m, j = 1, \dots, m, \end{cases}$$

$$F_i = \begin{cases} (f, \phi_i) + \langle g_N, \phi_i \rangle_{\Gamma_N} & \forall i = 1, \dots, M, \\ u_D(\mathbf{x}_i) & \forall i = M+1, \dots, m. \end{cases}$$

Using this notation, the finite-dimensional weak problem (2.8) can be written as a matrix equation:

*Find  $\mathbf{U} \in \mathbb{R}^m$  such that*

$$\mathbf{A}\mathbf{U} = \mathbf{F}. \quad (2.9)$$

The matrix of the system for an elliptic problem in the form (2.1a) is composed of two parts that often occur in FE calculations. Namely, if  $k \equiv \text{const}$  and  $c \equiv \text{const}$ , then

$$a(w, v) = \int_{\Omega} (k \nabla w \cdot \nabla v + c w v) \, d\mathbf{x} = k \int_{\Omega} \nabla w \cdot \nabla v \, d\mathbf{x} + c \int_{\Omega} w v \, d\mathbf{x}.$$

The linear system (2.9) can be rewritten as

$$(k\mathbf{K} + c\mathbf{M})\mathbf{U} = \mathbf{F},$$

where  $\mathbf{K} = (k_{i,j})$ ,  $\mathbf{M} = (m_{i,j}) \in \mathbb{R}^{m \times m}$  with the following (nontrivial) elements:

$$\begin{aligned} k_{i,j} &= \int_{\Omega} \nabla \phi_i \cdot \nabla \phi_j \, d\mathbf{x}, \quad \forall i = 1, \dots, M, \quad \forall j = 1, \dots, m, \\ m_{i,j} &= \int_{\Omega} \phi_i \phi_j \, d\mathbf{x}, \quad \forall i = 1, \dots, M, \quad \forall j = 1, \dots, m. \end{aligned}$$

The remaining elements are defined to preserve the Dirichlet boundary conditions in the matrix system, e.g.  $k_{i,j} = 1/k \, \delta_{i,j}$  and  $m_{i,j} = 0$  for  $i = M + 1, \dots, m$  and  $j = 1, \dots, m$ .  $\mathbf{K}$  is known as the *stiffness matrix*, and  $\mathbf{M}$  as the *mass matrix*.

The FE space is constructed in such a way that each basis function is supported on a few element domains only. If the elements are sufficiently small, most entries of the matrix  $\mathbf{A}$  are equal to zero, since most basis functions have nonintersecting supports. The system is therefore highly sparse. From the point of view of software used to solve the numerical problem this is beneficial in two ways: firstly, the memory required for matrix storage is significantly reduced as compared to full matrices, and secondly, many linear solvers can take advantage of the sparsity and produce the solution faster. The precise sparsity pattern of the matrix  $\mathbf{A}$  depends on the order of elements (i.e. the order of polynomials used as shape functions), and the dimension of the problem space  $\mathbb{R}^D$ .

The system (2.9) is a classical linear algebra problem, and a vast number of methods have been developed to solve the problems of this kind [67]. At the same time, the choice of a method that is robust and efficient is extremely important for applications, and is highly dependent on the nature of the matrix  $\mathbf{A}$ . The two classes of methods that can be used to solve (2.9) are direct and iterative methods. The former produce an exact solution of the linear system (up to machine accuracy), and include such methods as *LU*, Cholesky and *QR*. However, these methods become too

computationally and memory intensive as the problem dimension  $N$  grows, especially for three-dimensional problems. This is where iterative methods like BiCG [119] or GMRES [105] provide an alternative. While they only provide a solution which is accurate up to a given tolerance (or, more precisely, solutions with the norms of associated residuals  $\mathbf{R} = \mathbf{F} - \mathbf{A}\mathbf{U}$  under a given tolerance), both computational and memory requirements are usually lower than those for direct methods. For an overview of iterative linear solvers for large generic FEM systems we refer to survey papers [14, 18]. While the matrices obtained above are non-symmetric, they can be easily transformed into symmetric form by simple algebraic operations. If this is the case, then symmetric matrix solvers, such as conjugate gradients [72] or MINRES [96] can be used. In this work we address the question of choosing an efficient solver for coupled systems in Chapter 4. For the purpose of this chapter the final step of deriving a FEM is writing it in matrix form similar to (2.9).

The FE derivation follows the same pattern as above for any PDE. However, there are certain problems that require special treatment. Among these are systems of equations with incompressibility-type conditions, which we address separately in the following section.

## 2.4 Mixed methods

In a number of classical problems, such as in solid mechanics and fluid dynamics, the governing equation based on force balance can be supplemented by a condition representing conservation of mass, and stemming from assumed incompressibility of the material. One way of simultaneously treating these equations is by using mixed methods, sometimes also referred to as Lagrange multiplier methods. This section



follows the approach used in Chapter 12 of [21], where detailed proofs of results cited may be found.

We consider a two-dimensional Stokes problem for steady fluid flow [13]. Assuming that the inertial forces in the system are negligible, and that the fluid density and viscosity are constant and equal to 1 in the domain of interest, the problem can be written as follows:

$$-\Delta \mathbf{u} + \nabla p = \mathbf{f} \quad \text{in } \Omega, \quad (2.10a)$$

$$\nabla \cdot \mathbf{u} = 0 \quad \text{in } \Omega, \quad (2.10b)$$

$$\mathbf{u} = \mathbf{u}_D \quad \text{on } \Gamma_D, \quad (2.10c)$$

Here  $\mathbf{u}$  represents the two-dimensional velocity,  $p$  the (scalar) pressure,  $\mathbf{f}$  the external forces. Equation (2.10a) ensures the equilibrium of forces in the system, and equation (2.10b) fluid incompressibility. The given velocity profile  $u_D$  on the boundary  $\Gamma_D = \partial\Omega$  is, generally speaking, nonzero. However, the problem (2.10) transforms into a homogeneous one by considering a (unique minimum-norm) extension of  $u_D$  onto  $\Omega$ , and rewriting the equations in terms of  $\hat{u} = u - u_D$ . We will therefore consider  $u_D = 0$  on  $\Gamma_D$  for simplicity. We will also assume  $\mathbf{f} \neq 0$ , so that the problem admits non-zero velocity solutions.

Furthermore, the problem with all Dirichlet boundary conditions, as described above, does not allow for unique identification of  $p$  on  $\Omega$ , and therefore needs to be supplemented by an additional condition. We will choose  $p \in L_0^2(\Omega)$ , where

$$L_0^2(\Omega) = \left\{ q \in L^2(\Omega) \mid \int_{\Omega} q \, d\mathbf{x} = 0 \right\}. \quad (2.11)$$

As in previous sections, we convert the above problem into its weak form by multiplying the governing equations by arbitrary functions, integrating over  $\Omega$ , and applying the divergence theorem together with the boundary conditions. This leads to the following equations:

*Find  $(\mathbf{u}, p) \in X_0 \times Y$  such that*

$$\begin{aligned} \int_{\Omega} \left( \sum_{i=1,2} \nabla \mathbf{u}_i \cdot \nabla \mathbf{v}_i - (\nabla \cdot \mathbf{v}) p \right) d\mathbf{x} &= \int_{\Omega} \mathbf{f} \cdot \mathbf{v} d\mathbf{x} \quad \forall \mathbf{v} \in X_0, \\ \int_{\Omega} (\nabla \cdot \mathbf{u}) q d\mathbf{x} &= 0 \quad \forall q \in Y, \end{aligned}$$

with  $X = (H^1(\Omega))^2$ ,  $X_0 = \{\mathbf{v} \in X \mid \mathbf{v} = 0 \text{ on } \Gamma_D\}$  and  $Y = L_0^2(\Omega)$ . Introducing the operator notation as before, namely

$$\begin{aligned} a(\mathbf{v}, \mathbf{w}) &= \int_{\Omega} \left( \sum_{i=1,2} \nabla \mathbf{v}_i \cdot \nabla \mathbf{w}_i \right) d\mathbf{x}, \\ b(\mathbf{v}, q) &= - \int_{\Omega} (\nabla \cdot \mathbf{v}) q d\mathbf{x}, \end{aligned}$$

we may rewrite the weak form again:

*Find  $(\mathbf{u}, p) \in X_0 \times Y$  such that*

$$a(\mathbf{u}, \mathbf{v}) + b(\mathbf{v}, p) = (\mathbf{f}, \mathbf{v}) \quad \forall \mathbf{v} \in X_0, \tag{2.12a}$$

$$b(\mathbf{u}, q) = 0 \quad \forall q \in Y. \tag{2.12b}$$

In general, the following weak problems are referred to as mixed:

*Find  $(u, p) \in V \times Q$  such that*

$$a(u, v) + b(v, p) = F(v) \quad \forall v \in V, \tag{2.13a}$$

$$b(u, q) = 0 \quad \forall q \in Q, \quad (2.13b)$$

where  $F$  is a linear functional on  $V$ . The question of well-posedness of this abstract problem can be addressed in the following way. Defining a closed subspace of  $V$ ,  $Z = \{v \in V \mid b(v, q) = 0 \quad \forall q \in Q\}$  leads to a more familiar formulation:

*Find  $u \in Z$  such that*

$$a(u, v) = F(v) \quad \forall v \in Z.$$

If  $a(\cdot, \cdot)$  is coercive on  $Z$ , the Lax-Milgram theorem guarantees that the above problem has a unique solution. For the Stokes problem this operator is the two-dimensional version of the  $a(\cdot, \cdot)$  in the elliptic problem considered in section 2.2 with  $c = 0$  and  $k = 1$ , and is therefore coercive.

In order to find  $p$  in (2.13) we need to solve

$$b(v, p) = \hat{F}(v) \quad \forall v \in V, \quad (2.14)$$

where  $\hat{F}(v) = F(v) - a(u, v)$ . The following lemma (Lemma 12.2.12 in [21]) establishes the conditions necessary for the existence of a unique solution in this problem.

**Lemma 1.** *Let the linear form  $\hat{F}(\cdot) : V \rightarrow \mathbb{R}$  be continuous, and the bilinear forms  $a(\cdot, \cdot) : V \times V \rightarrow \mathbb{R}$  and  $b(\cdot, \cdot) : V \times Q \rightarrow \mathbb{R}$  be continuous, i.e.  $\exists C > 0$  such that*

$$a(v, w) \leq C \|v\|_V \|w\|_V \quad \forall v, w \in V,$$

$$b(v, q) \leq C \|v\|_V \|q\|_Q \quad \forall v \in V, q \in Q,$$

and  $b(\cdot, \cdot)$  satisfies the Babuška-Brezzi condition [6, 22, 23], i.e.  $\exists \beta > 0$  such that

$$\beta \|q\|_Q \leq \sup_{v \in V} \frac{b(v, q)}{\|v\|_V} \quad \forall q \in Q.$$

Then equation (2.14) has a unique solution.

The bilinear forms  $a$  and  $b$  as defined above for the Stokes problem satisfy the conditions of this lemma [4], and thus admit a unique solution.

By considering discrete subspaces  $V^h \subset V$  and  $Q^h \subset Q$ , we arrive at the following discrete formulation of the problem:

Find  $(u^h, p^h) \in V^h \times Q^h$  such that

$$a(u^h, v^h) + b(v^h, p^h) = F(v^h) \quad \forall v^h \in V^h, \quad (2.15a)$$

$$b(u^h, q^h) = 0 \quad \forall q^h \in Q^h. \quad (2.15b)$$

As in the continuous case, we can define  $Z^h = \{v^h \in V^h \mid b(v^h, q^h) = 0 \quad \forall q^h \in Q^h\}$ , and rewrite the mixed problem as follows:

Find  $u^h \in Z^h$  such that

$$a(u^h, v^h) = F(v^h) \quad \forall v^h \in Z^h.$$

The following result (Theorem 12.3.7 in [21]) provides an error bound for the solution  $u^h$  of the above problem. Generally speaking  $Z^h \subset Z$  is not guaranteed, and therefore we cannot apply Céa's theorem directly.

**Theorem 3.** Let  $a(\cdot, \cdot)$  be coercive on  $Z \cup Z^h$ ,  $(u, p) \in V \times Q$  and  $(u^h, p^h) \in V^h \times Q^h$

be the solutions of (2.13) and (2.15) respectively,  $V^h \subset V$  and  $Q^h \subset Q$ . Then

$$\|u - u^h\|_V \leq \left(1 + \frac{C}{\alpha}\right) \inf_{v^h \in Z^h} \|u - v^h\|_V + \frac{C}{\alpha} \inf_{q^h \in Q^h} \|p - q^h\|_Q,$$

where  $C$  is the continuity constant for  $a(\cdot, \cdot)$  and  $b(\cdot, \cdot)$ , and  $\alpha$  is the coercivity constant for  $a(\cdot, \cdot)$  on  $Z \cup Z^h$ .

A similar result providing an error bound on the  $p^h$  component of the solution is desirable. In the continuous case, the Babuška-Brezzi condition was necessary to ensure well-posedness of the problem for  $p$ . The discrete counterpart of this condition is known as the *inf-sup condition* (Lemma 12.5.1 in [21]).

**Lemma 2.** *Let  $F(\cdot)$  be continuous on  $V$ ,  $a(\cdot, \cdot)$  be coercive and continuous on  $V \times V$ , and  $b$  be continuous on  $V \times Q$ ,  $V^h \subset V$ ,  $Q^h \subset Q$ . The problem*

$$b(v^h, p^h) = F(v^h) - a(u^h, v^h) \quad \forall v^h \in V^h,$$

*has a unique solution  $u^h \in V^h$  if and only if*

$$\beta^h = \inf_{q^h \in Q^h} \sup_{v^h \in V^h} \frac{|b(v^h, q^h)|}{\|v^h\|_V \|q^h\|_Q} > 0. \quad (2.16)$$

This condition is crucial in establishing whether  $p$  can be obtained stably using given approximation spaces  $V^h$  and  $Q^h$ . If it holds, the following error bound is valid (corollary 12.5.13 in [21]).

**Theorem 4.** *Let all of the conditions of Lemma 2 hold. Then*

$$\|p - p^h\|_Q \leq c \inf_{v^h \in Z^h} \|u - v^h\|_V + (1 + c) \inf_{q^h \in Q^h} \|p - q^h\|_Q, \quad \text{where } c = \frac{C}{\beta^h} \left(1 + \frac{C}{\alpha}\right),$$

$\alpha$  the coercivity constant of  $a(\cdot, \cdot)$ , and  $\beta^h$  the inf-sup constant in Lemma 2.

Thus, the quality of the approximate solution to the mixed problem depends both on the approximability of the solution spaces, and their interrelation in the form of the inf-sup condition.

Now we return to the Stokes equations, and consider a specific class of approximation spaces which adheres to the inf-sup condition, and is therefore widely used for the solution of flow problems.

Let  $P^k$ , as before, be the space of continuous piecewise polynomials of degree  $k$  on  $\Omega$  defined on a non-degenerate triangulation  $\tau$  of maximum diameter  $h$ . The Taylor-Hood approximation spaces are formed as follows:

$$X^h = (P^k)^2, \text{ with } X_0^h = X^h \cap X_0, \quad (2.17a)$$

$$Y^h = \left\{ q^h \in P^{k-1} \mid \int_{\Omega} q^h \, d\mathbf{x} = 0 \right\}. \quad (2.17b)$$

The finite element formulation of the Stokes problem using these spaces is as follows:

*Find  $(\mathbf{u}^h, p^h) \in X_0^h \times Y^h$  such that*

$$a(\mathbf{u}^h, \mathbf{v}^h) + b(\mathbf{v}^h, p) = (\mathbf{f}, \mathbf{v}^h) \quad \forall \mathbf{v}^h \in X_0^h, \quad (2.18a)$$

$$b(\mathbf{u}^h, q^h) = 0 \quad \forall q^h \in Y^h. \quad (2.18b)$$

The following results showing inf-sup stability of Taylor-Hood elements (theorems 12.6.6 and 12.6.7 in [21]) hold true for all  $k \geq 2$ .

**Theorem 5.** *Suppose none of the triangles in  $\tau$  have two edges on  $\partial\Omega$ . Then the Taylor-Hood spaces (2.17) satisfy the inf-sup condition (2.16) for (2.18) with  $\beta$  independent of  $h$ .*

**Theorem 6.** *Suppose none of the triangles in  $\tau$  have two edges on  $\partial\Omega$ , and let  $0 \leq s \leq 2$ . Suppose  $(\mathbf{u}, p) \in (H^{s+1}(\Omega))^2 \times H^s(\Omega)$  solves (2.12), and  $(\mathbf{u}^h, p^h)$  solves (2.18). Then*

$$\|\mathbf{u} - \mathbf{u}^h\|_{(H^1(\Omega))^2} + \|p - p^h\|_{L^2(\Omega)} \leq Ch^s \left( \|\mathbf{u}\|_{(H^{s+1}(\Omega))^2} + \|p\|_{H^s(\Omega)} \right).$$

Assuming an inf-sup stable approximation space can be chosen for a given mixed problem, the finite element formulation does not need additional stabilisation (see e.g. [60, 59]), and the method can be written directly in matrix form. Let the basis expansions of the solution components be

$$u_1^h = \sum_{j=1}^m U_{1,j} \phi_j, \quad u_2^h = \sum_{j=1}^m U_{2,j} \phi_j, \quad p^h = \sum_{k=1}^l P_k \psi_k,$$

where  $m$  is the dimension of  $P^k(\tau)$ , and  $l$  is the dimension of  $P^{k-1}$ . Let the solution vector have the following components:

$$\mathbf{U} = (U_{1,1}, \dots, U_{1,m}, U_{2,1}, \dots, U_{2,m})^T \quad \text{and} \quad \mathbf{P} = (P_1, \dots, P_l)^T.$$

The matrix  $\mathbf{A}$  and the vector  $\mathbf{F}$  are similar to their counterparts from section 2.3, apart from the fact that  $\mathbf{A}$  is block-diagonal and  $\mathbf{F}$  is a block vector with blocks corresponding to velocity components. The mixed term matrix  $\mathbf{B}$  has characteristic nonzero components as  $b_{i,k} = b(\phi_i, \psi_k)$ ,  $i = 1, \dots, 2m$ ,  $k = 1, \dots, l$ . Then the matrix system arising from the finite element discretisation of the Stokes problem is as follows:

Find  $(\mathbf{U}, \mathbf{P})^T \in \mathbb{R}^{2m+l}$  such that

$$\begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & 0 \end{pmatrix} \begin{pmatrix} \mathbf{U} \\ \mathbf{P} \end{pmatrix} = \begin{pmatrix} \mathbf{F} \\ 0 \end{pmatrix}. \quad (2.19)$$

An additional equation restricting the choice of  $p^h \in Y^h$  needs to be added to this system, then existence and uniqueness of the solution, guaranteed by the inf-sup condition, is equivalent to the non-singularity of the matrix in equation (2.19).

This matrix has what is known as saddle point structure, which comes from the zero block on the diagonal. In practical terms this block causes additional difficulties in solving the matrix equations. These will be considered in detail in Chapter 4.

The need to solve saddle point problems when modelling incompressible materials can be circumvented by using penalty methods [73, 100, 110]. Their general idea is to eliminate the pressure variable (which performs the role of the Lagrange multiplier in the optimisation context) in the governing equation, and replace it with the constraint scaled by large numbers, thus forcing the constraint to be satisfied simultaneously with the main equation. While these methods present certain practical advantages, their scope is limited in application to multicompartment and multiphysics problems, where coupling between different components may introduce multiple Lagrange multipliers, which cannot necessarily be eliminated, or scaled together.

Having considered mixed formulations, which are used for modelling a number of important types of applied problems, we move on to a different aspect of numerical solution of PDEs: time dependence.



## 2.5 Time stepping

The problems (2.1) and (2.10) considered in the previous sections describe the respective physical systems in their equilibrium states. However, it is often important to characterise the evolution of such systems through time. From the physics perspective, time-dependent systems include a non-zero acceleration term in Newton's second law. Mathematically this means that temporal derivatives need to be included in the equations. In this section we describe the treatment of such terms in the context of finite element methods.

Introducing time dependence into equations (2.1) results in the following problem:

$$\frac{\partial u}{\partial t} - \nabla \cdot (k \nabla u) + cu = f \quad \text{in } \Omega \times (0, T], \quad (2.20a)$$

$$u = u_D \quad \text{on } \partial\Gamma_D \times (0, T], \quad (2.20b)$$

$$k \nabla u \cdot \mathbf{n} = g_N \quad \text{on } \partial\Gamma_N \times (0, T], \quad (2.20c)$$

$$u = u_0 \quad \text{in } \Omega \times \{0\}. \quad (2.20d)$$

Here  $u, k, c, f, u_D, g_N : \Omega \times (0, T] \rightarrow \mathbb{R}$ ,  $T > 0$ , and  $u_0 : \Omega \rightarrow \mathbb{R}$  is a prescribed initial condition.

The spatial FE discretisation of this parabolic problem can be carried out exactly as for elliptic problems in Section 2.3, leading to a semi-discrete problem, which is in essence a finite set of ordinary differential equations. This, in turn, can be solved using either finite difference (FD), or discontinuous Galerkin (dG) methods.

We use the following partition of the time interval  $(0, T]$  in both time stepping approaches:

$$0 = t_0 < t_1 < \dots < t_N = T, \quad N \geq 1, \quad \text{with } \Delta t_n = t_n - t_{n-1}, \quad I_n = (t_{n-1}, t_n], \quad n = 1, \dots, N.$$

### 2.5.1 Implicit Euler method

Let  $u_n$  be the semidiscrete approximation to the solution at time  $t_n$ ,  $k_n = k(\mathbf{x}, t_n)$ ,  $c_n = c(\mathbf{x}, t_n)$  and  $f_n = f(\mathbf{x}, t_n)$ ,  $n = 1, \dots, N$ . The implicit, or backward, approximation of the time derivative in (2.20a) leads to the following semidiscrete problem:

For  $n = 1, \dots, N$

$$\frac{u_n - u_{n-1}}{\Delta t_n} - \nabla \cdot (k_n \nabla u_n) + c_n u_n = f_n \quad \text{in } \Omega, \quad (2.21a)$$

$$u_n = u_D(\mathbf{x}, t_n) \quad \text{on } \partial\Gamma_D, \quad (2.21b)$$

$$k_n \nabla u_n \cdot \mathbf{n} = g_N(\mathbf{x}, t_n) \quad \text{on } \partial\Gamma_N, \quad (2.21c)$$

with  $u_0$  a given function. Essentially, for each time step we need to solve a problem similar to that considered in Section 2.3.

Using the same notation for function spaces as before, namely  $X = H^1(\Omega)$ ,  $X_D = H_D^1(\Omega)$ ,  $X_0 = H_0^1(\Omega)$ , we may write the (semidiscrete) weak form of the problem (2.21):

For  $n = 1, \dots, N$  find  $u_n \in X_D$  such that

$$\left( \frac{u_n - u_{n-1}}{\Delta t_n}, v \right) + a_n(u_n, v) = (f_n, v) + \langle g_N(\cdot, t_n), v \rangle_{\Gamma_N} \quad \forall v \in X_0.$$

The spatial discretisation of this problem is identical to the process described in Section 2.3. Denoting the discrete solution vector at time  $t_n$  as  $\mathbf{U}^n$ , and reusing matrix notation introduced above (with appropriate Dirichlet boundary conditions) we may write the fully discrete system as follows:

For  $n = 1, \dots, N$  find  $\mathbf{U}^n \in \mathbb{R}^m$  such that

$$(\mathbf{M} + \Delta t_n \mathbf{A}^n) \mathbf{U}^n = \Delta t_n \mathbf{F}^n + \mathbf{M} \mathbf{U}^{n-1}.$$

Note that in this section and throughout this work notation  $\mathbf{A}^n$  is used to denote the matrix computed for time step  $n$ , not the  $n$ th power of matrix  $\mathbf{A}$ .

The discrete initial condition is obtained by projecting  $u_0$  onto the finite element space. The time stepping process is therefore straightforward: at each time step we need to invert the matrix of the system  $\mathbf{M} + \Delta t_n \mathbf{A}^n$ . If the time step size  $\Delta t_n = \Delta t$  and the PDE matrix  $\mathbf{A}^n = \mathbf{A}$  do not change in time, the inversion can be performed once at the beginning of the procedure, and reused subsequently.

The implicit Euler method allows us to compute the approximate solution of the system at discrete points in time. Alternatively, one can utilise the finite element framework to compute the coefficients of a basis expansion in a finite-dimensional space in the whole of the time interval  $[0, T]$ .

### 2.5.2 Discontinuous Galerkin method

The use of the discontinuous Galerkin approach in time-dependent problems combines the convenience of derivation and analysis coming with finite element formulations with the efficiency of time stepping coming from the discontinuity in time.

In order to write the weak form of the parabolic equation we introduce the following function spaces:

$$X = L^2(0, T; H^1(\Omega)),$$

$$X_D = \{v \in X \mid v = u_D \text{ on } \Gamma_D \times (0, T]\},$$

$$X_0 = \{v \in X \mid v = 0 \text{ on } \Gamma_D \times (0, T]\},$$

where  $L^2(0, T; H^1(\Omega))$  contains all functions that are square-integrable in time over  $(0, T]$  and square-integrable with their first weak derivatives over  $\Omega$ .

Multiplying (2.20a) by an arbitrary test function  $v$ , integrating over  $\Omega \times (0, T]$ , and applying the divergence theorem results in the following weak problem:

*Find  $u \in X_D$  such that*

$$\begin{aligned} \int_0^T \int_{\Omega} \left( \frac{\partial u}{\partial t} v + k \nabla u \cdot \nabla v + cuv \right) d\mathbf{x} = \\ \int_0^T \int_{\Omega} f v d\mathbf{x} + \int_0^T \int_{\Gamma_N} g_N v ds \quad \forall v \in X_0, \\ u = u_0 \quad \text{in } \Omega \times \{0\}. \end{aligned}$$

We can rewrite this problem using shorthand notation, namely for the bilinear form  $a(\cdot, \cdot)$  introduced in (2.3), scalar product in  $L^2(\Omega)$ , and dual product on  $\Gamma_N$ :

*Find  $u \in X_D$  such that*

$$\begin{aligned} \int_0^T \left\{ \left( \frac{\partial u}{\partial t}, v \right) + a(u, v) \right\} dt = \int_0^T \{ (f, v) + \langle g_N, v \rangle_{\Gamma_N} \} dt \quad \forall v \in X_0, \\ u = u_0 \quad \text{in } \Omega \times \{0\}. \end{aligned}$$

Following [48], we may consider space-time slabs  $S_n = \Omega \times I_n$ . In each of the  $S_n$ ,  $n = 1, \dots, N$  the FE discretisation can be based on the following function space:

$$X_n^h = \left\{ v_n^h \left| v_n^h = \sum_{r=0}^q t^r \hat{v}, \hat{v} \in P^k(\Omega) \right. \right\}. \quad (2.22)$$

The global function space is

$$X^h = \{ v^h \in X | v^h = v_n^h \in X_n^h \text{ for } t \in (t_{n-1}, t_n], n = 1, \dots, N \},$$

and the corresponding trial and test spaces are  $X_D^h = X^h \cap X_D$  and  $X_0^h = X^h \cap X_0$  respectively.

The difference between a standard finite element, or continuous Galerkin method, and a discontinuous one is that no continuity restrictions are imposed at the temporal element boundaries  $t_n$ ,  $n = 1, \dots, N$ . The highest polynomial power  $q$  in (2.22) determines the order of the method  $dG(q)$ :

*Find  $u^h \in X_D^h$  with  $u_{0-}^h = u_0$  such that*

$$\begin{aligned} \sum_{n=1}^N \left( ([u^h]_{n-1}, v_{(n-1)+}^h) + \int_{I_n} \left\{ \left( \frac{\partial u^h}{\partial t}, v^h \right) + a(u^h, v^h) \right\} dt \right) \\ = \sum_{n=1}^N \int_{I_n} \{ (f, v^h)_+ + \langle g_N, v^h \rangle_{\Gamma_N} \} dt \quad \forall v^h \in X_0^h, \end{aligned} \quad (2.23)$$

Here we use the standard notation for the left and right limits of a function, and its jump at a point:

$$v_{n+} = \lim_{t \rightarrow t_n+0} v(t), \quad v_{n-} = \lim_{t \rightarrow t_n-0} v(t), \quad [v]_n = v_{n+} - v_{n-}.$$

The system decouples in time, since  $u^h$  is assumed discontinuous between time steps. Using the basis expansions as before, we can write this equation as a matrix system for  $q = 0$ :

*For  $n = 1, \dots, N$  find  $U^n \in \mathbb{R}^m$  such that*

$$(\mathbf{M} + \Delta t_n \mathbf{A}^n) \mathbf{U}^n = \Delta t F^n + \mathbf{M} U^{n-1}.$$

We note that increasing the order of temporal approximation  $q$  by one means that the dimension of the system matrix increases by  $m$ , the dimension of  $X^h$ . Unless

stated otherwise,  $dG(0)$  is used for time stepping, allowing us to keep the matrix size the same as in the steady case.

The treatment of time dependence concludes our review of problem-specific aspects of FEMs relevant to the current work.

The finite element methods described in this chapter are widely used to solve PDE problems numerically. We have defined the necessary function spaces, derived weak and finite element formulations for a generic elliptic problem, and formulated the basic error bound for it. Separately, we considered mixed problems and the inf-sup condition for existence and uniqueness of their solutions, parabolic problems and time stepping methods. All these aspects of the numerics were described in the context of a single PDE problem in one domain in order to introduce the ideas required for the development of the method.

Extending the theory to multiple domains or physics will be made possible by addressing the interaction between the components. In the following chapter we discuss the possible ways of modelling the coupling, and perform the full derivation of a finite element method for coupled problems.

# Chapter 3

## Coupled problems

As described in the Introduction, we will focus our attention on coupling through interfaces between domains, as illustrated by Figure 3.1, where the system comprises two components residing in subdomains  $\Omega_1$  and  $\Omega_2$  intersecting via a common interface  $\Gamma$ . The motivation for this work comes from multiphysics systems, with different

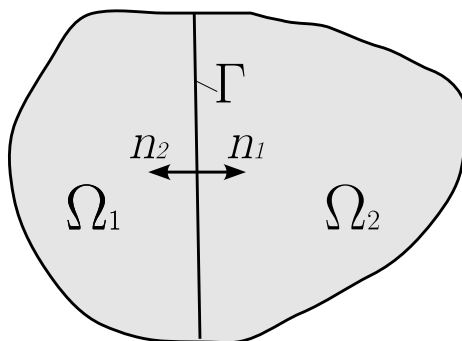


Figure 3.1: Two domains  $\Omega_1$  and  $\Omega_2$  sharing a common interface  $\Gamma$ .

equations to be solved in  $\Omega_1$  and  $\Omega_2$ , say, unobstructed flow in  $\Omega_1$  and flow through porous medium in  $\Omega_2$ . However, the question of coupling is also a crucial aspect in domain decomposition. Domain decomposition has arisen as a way of solving large-scale systems by splitting them into smaller parts when the computational resources available are insufficient to solve the system as a whole, or simply when it is more

efficient to solve multiple small systems than a single large one. We refer to [16, 20, 56, 66, 98, 112] for discussion of domain decomposition methods, and the closely related additive Schwartz method for solving linear systems of equations.

To keep the notation as clear as possible all theoretical results are written out for two-component models. The extension to multiple domains is natural, and the methods have been successfully applied to a three-domain problem in Chapter 6.

In this chapter we discuss the existing approaches to coupling, namely partitioned and monolithic solvers. We propose that by using a block preconditioner in a monolithic setting one can effectively combine the best features of the two approaches. In order to explore this solution strategy in the following chapters, we derive the finite element formulations for generic coupled problems using Lagrange multipliers.

Let us consider an abstract coupled system comprising two components:

$$\mathcal{F}_1[u_1, u_2, t] = 0,$$

$$\mathcal{F}_2[u_1, u_2, t] = 0.$$

Here  $u_1$  and  $u_2$  represent the states of the two components, while  $\mathcal{F}_1$  and  $\mathcal{F}_2$  are the operators modelling the physical laws governing the behaviour of these components, and  $t$  is the time.

### 3.1 Partitioned solvers

The basic idea of the partitioned (staggered, segregated, or operator decomposition) approach is to decouple the components, and feed shared information from one sub-problem to the other.



Weakly coupled algorithms [54, 57] assume slow variation of the solution, and use historic states in the coupling terms. The implementations of this method may vary, but the idea can be described as follows:

*At time step  $n$*

*find  $u_1^n$  such that  $\mathcal{F}_1(u_1^n, u_2^{n-1}, t^n) = 0$ ,  
then find  $u_2^n$  such that  $\mathcal{F}_2(u_1^n, u_2^n, t^n) = 0$ .*

Strongly coupled methods [52, 53, 89] rely on iterating between the components at each time step to capture the coupling better:

*At time step  $n$*

*for  $k = 1, 2, \dots$   
find  $u_1^{n,k}$  such that  $\mathcal{F}_1(u_1^{n,k}, u_2^{n,k-1}, t^n) = 0$ ,  
then find  $u_2^{n,k}$  such that  $\mathcal{F}_2(u_1^{n,k}, u_2^{n,k}, t^n) = 0$ .*

The popularity of both types of partitioned algorithms can be explained by the relative ease of their implementation, and the ability to reuse existing efficient component solvers, or modularity. Some of the methods can be easily parallelised if the information is passed between the sub-problems simultaneously, not sequentially.

At the same time, operator decomposition methods are often unstable. While for certain problems it is possible to obtain an explicit stability condition [65, 102], to our knowledge it has not been done for multiphysics problems. The choice of the initial guess, the starting domain and the time step have a significant effect on the performance of these methods. Strongly coupled methods generally have better stability than their weakly coupled counterparts, but their associated computational load is much higher because of the iteration. Furthermore, accuracy is a concern even for stable methods. It has been shown that if second-order methods are used

in the sub-systems, the overall method is only first-order. All of these issues can be addressed individually for certain types of problems, and they certainly do not prevent extensive use of the method in practice. However, if using partitioned solvers one has to be aware of their limitations, and of the fact that generic methods are not guaranteed to produce a reliable solution. For a further discussion of operator decomposition we refer to review papers [53, 57].

## 3.2 Monolithic solvers

The monolithic approach [70, 94, 88] simply states that all equations in the system are solved simultaneously:

*At each time step  $n$*

*find a pair  $(u_1^n, u_2^n)$  such that*

$$\mathcal{F}_1(u_1^n, u_2^n, t^n) = 0,$$

$$\mathcal{F}_2(u_1^n, u_2^n, t^n) = 0.$$

There is no weakening of the coupling between the two components, and therefore the method's properties depend entirely on the approximation, e.g. the choice of the finite element bases and the time-stepping scheme.

The difficulty in using this approach is predominantly computational, since the system has non-standard complex structure. There is no natural modularity in this case, as existing solvers for  $\mathcal{F}_1$  and  $\mathcal{F}_2$  cannot be used directly. The resulting discretised system in any practical problem is likely to be large, indefinite and poorly conditioned, making the use of both standard direct and iterative methods unus-

tainable.

The best solver would combine the advantages of partitioned and monolithic schemes by retaining the coupling while making use of existing efficient component solvers. This is achieved by the use of block preconditioners in an iterative method for the full monolithic system. Block preconditioners decouple the action of the system components without weakening the coupling in the system itself, and therefore present a perfect candidate for a coupled solver, as we show in Chapter 4. In order to write down the mathematics behind this method we need to derive the matrix equations of a finite element method for a monolithic coupled solver.

### 3.3 Elliptic coupled problem

Let  $\Omega_1$  and  $\Omega_2$  be bounded open domains in  $\mathbb{R}^D$  with boundaries  $\partial\Omega_1$  and  $\partial\Omega_2$  respectively. Let these boundaries intersect along a common interface  $\Gamma = \partial\Omega_1 \cap \partial\Omega_2 \neq \emptyset$ , each of  $\partial\Omega_i$  comprising three non-intersecting parts:  $\partial\Omega_i = \Gamma_i^D \cup \Gamma_i^N \cup \bar{\Gamma}$ ,  $i = 1, 2$ . We consider the following elliptic coupled problem:

$$-\nabla \cdot (k_i \nabla u_i) + c_i u_i = f_i \quad \text{in } \Omega_i, \ i = 1, 2, \quad (3.1a)$$

$$u_i = u_{i,D} \quad \text{on } \Gamma_i^D, \ i = 1, 2, \quad (3.1b)$$

$$k_i \nabla u_i \cdot \mathbf{n}_i = g_{i,N} \quad \text{on } \Gamma_i^N, \ i = 1, 2, \quad (3.1c)$$

$$u_1 - u_2 = 0 \quad \text{on } \Gamma, \quad (3.1d)$$

$$k_1 \nabla u_1 \cdot \mathbf{n}_1 + k_2 \nabla u_2 \cdot \mathbf{n}_2 = 0 \quad \text{on } \Gamma, \quad (3.1e)$$

where  $k_i \in C^1(\Omega_i)$ ,  $\exists k_{i,0} \in \mathbb{R}^+$  s.t.  $k_i \geq k_{i,0}$ ,  $c_i$  and  $f_i \in L^2(\Omega_i)$ , and  $\mathbf{n}_i$  are the outward normal vectors to the respective boundaries of  $\Omega_i$ ,  $i = 1, 2$ . Note that

$\mathbf{n}_1 = -\mathbf{n}_2$  on  $\Gamma$ .

Equations (3.1d) and (3.1e) model the interactions between the two components of the system. The former represents continuity of the solutions across the interface  $\Gamma$ , the latter continuity of the fluxes.

### 3.3.1 Weak formulation

Following the derivation in Section 2.2, we introduce function spaces

$$\begin{aligned} X_i &= H^1(\Omega_i), \quad i = 1, 2, \\ X_{i,D} &= \{v \in X_i \mid v = u_{i,D} \text{ on } \Gamma_i^D\}, \quad i = 1, 2, \\ X_{i,0} &= \{v \in X_i \mid v = 0 \text{ on } \Gamma_i^D\}, \quad i = 1, 2, \\ Y &= L^2(\Gamma). \end{aligned}$$

As before, multiplying equations (3.1a) by arbitrary functions from  $X_{i,0}$ , integrating over  $\Omega_i$  and applying the divergence theorem, we obtain

$$\begin{aligned} - \int_{\Gamma} (k_i \nabla u_i \cdot \mathbf{n}_i) v_i \, ds - \int_{\Gamma_i^D} (k_i \nabla u_i \cdot \mathbf{n}_i) v_i \, ds - \int_{\Gamma_i^N} (k_i \nabla u_i \cdot \mathbf{n}_i) v_i \, ds \\ + \int_{\Omega_i} (k_i \nabla u_i \cdot \nabla v_i + c_i u_i v_i) \, d\mathbf{x} = \int_{\Omega_i} f_i v_i \, d\mathbf{x}, \quad i = 1, 2. \end{aligned}$$

The integrals over  $\Gamma_i^D$  vanish since  $v_i = 0$  on  $\Gamma_i^D$ , and the Neumann conditions allow us to rewrite the integrals over  $\Gamma_i^N$ . Finally, we wish to incorporate the coupling into the weak formulation. For this purpose we introduce a Lagrange multiplier

$$\lambda = k_1 \nabla u_1 \cdot \mathbf{n}_1 = -k_2 \nabla u_2 \cdot \mathbf{n}_2 \text{ on } \Gamma.$$

Using shorthand notation, we rewrite the above weak form as follows:

$$\mp \langle \lambda, v_i \rangle_\Gamma + a_i(u_i, v_i) = (f_i, v_i) + \langle g_{i,N}, v \rangle_{\Gamma_i^N}, \quad i = 1, 2,$$

where the ‘-’ sign corresponds to  $i = 1$  and the ‘+’ sign to  $i = 2$ . To complete the weak formulation of the coupled problem, we multiply (3.1d) by an arbitrary function  $\nu \in Y$ , and integrate over  $\Gamma$  to get

$$\langle \nu, u_1 - u_2 \rangle_\Gamma = 0.$$

The full weak problem is therefore as follows:

*Find  $(u_1, u_2, \lambda) \in X_{1,D} \times X_{2,D} \times Y$  such that*

$$\mp \langle \lambda, v_i \rangle_\Gamma + a_i(u_i, v_i) = (f_i, v_i) + \langle g_{i,N}, v_i \rangle_{\Gamma_i^N} \quad \forall v_i \in X_{i,0}, \quad i = 1, 2, \quad (3.2a)$$

$$\langle \nu, u_1 - u_2 \rangle_\Gamma = 0 \quad \forall \nu \in Y. \quad (3.2b)$$

Note that the continuity equation (3.1d) is incorporated into the weak form explicitly, whilst the flux equation (3.1e) is imposed via the Lagrange multiplier.

### 3.3.2 Finite element method

We construct regular partitions of the domains  $\tau_i(\Omega_i)$ ,  $i = 1, 2$ , and a regular partition of the interface  $\tau(\Gamma)$ , and define the finite dimensional function spaces

$$X_i^h = P^k(\Omega_i), \quad X_{i,D}^h = X_i^h \cap X_{i,D}, \quad X_{i,0}^h = X_i^h \cap X_{i,0}, \quad Y^h = P^k(\Gamma).$$

The choice of the interface partition is natural when  $\tau_1(\Omega_1)$  and  $\tau_2(\Omega_2)$  conform on  $\Gamma$ . Otherwise the finite element space for the Lagrange multiplier is restricted by the discrete inf-sup condition, similarly to the Lagrange multiplier in mixed problems. We always choose the partitioning of the interface  $\Gamma$  to coincide with the finer projection from  $\tau_1(\Omega_1)$  or  $\tau_2(\Omega_2)$ . This choice guarantees stability of the coupled finite element method [94]. The finite element discretisation of (3.2) is as follows:

*Find  $(u_1^h, u_2^h, \lambda^h) \in X_{1,D}^h \times X_{2,D}^h \times Y^h$  such that*

$$\mp \langle \lambda^h, v_i^h \rangle_\Gamma + a_i(u_i^h, v_i^h) = (f_i, v_i^h) + \langle g_{i,N}, v_i^h \rangle_{\Gamma_i^N} \quad \forall v_i^h \in X_{i,0}^h, \quad i = 1, 2, \quad (3.3a)$$

$$\langle \nu^h, u_1^h - u_2^h \rangle_\Gamma = 0 \quad \forall \nu^h \in Y^h. \quad (3.3b)$$

Expanding the components of the discrete solution in terms of the respective bases

$$u_i^h = \sum_{j=1}^{m_i} U_{i,j} \phi_{i,j}, \quad i = 1, 2, \quad \text{and} \quad \lambda^h = \sum_{k=1}^m \Lambda_k \nu_k,$$

and assuming the ordering of basis functions to be such that the indices corresponding to Dirichlet nodes are  $(M_i + 1, \dots, m_i)$ ,  $i = 1, 2$ , we may rewrite (3.3):

*Find  $(\mathbf{U}_1, \mathbf{U}_2, \mathbf{\Lambda}) \in \mathbb{R}^{m_1} \times \mathbb{R}^{m_2} \times \mathbb{R}^m$  such that*

$$\mp \sum_{k=1}^m \Lambda_k \langle \nu_k, \phi_{i,l} \rangle_\Gamma + \sum_{j=1}^{m_i} U_{i,j} a_i(\phi_{i,j}, \phi_{i,l}) = (f_i, \phi_{i,l}) + \langle g_{i,N}, \phi_{i,l} \rangle_{\Gamma_i^N}, \quad l = 1, \dots, M_i, \\ i = 1, 2,$$

$$U_{i,j} = u_{i,D}(\mathbf{x}_j), \quad i = 1, 2, j = M_i + 1, \dots, m_i,$$

$$\sum_{j=1}^{m_1} U_{1,j} \langle \nu_k, \phi_{1,j} \rangle_\Gamma - \sum_{j=1}^{m_2} U_{2,j} \langle \nu_k, \phi_{2,j} \rangle_\Gamma = 0, \quad k = 1, \dots, m.$$

We define the following matrices and vectors: component matrices  $\mathbf{A}_i = (a_{j,l}^i) \in \mathbb{R}^{m_i \times m_i}$  for  $j, l = 1, \dots, m_i$ , coupling matrices  $\mathbf{C}_i = (c_{j,k}^i) \in \mathbb{R}^{m_i \times m}$  for  $j = 1, \dots, m_i$  and  $k = 1, \dots, m$ , load vectors  $\mathbf{F}_i = (f_j^i) \in \mathbb{R}^{m_i}$  for  $j = 1, \dots, m_i$ , where

$$\begin{aligned} a_{j,l}^i &= \begin{cases} a_i(\phi_{i,j}, \phi_{i,l}) & \forall i = 1, 2, j = 1, \dots, M_i, l = 1, \dots, m_i, \\ \delta_{j,l} & \forall i = 1, 2, j = M_i + 1, \dots, m_i, l = 1, \dots, m_i, \end{cases} \\ c_{j,k}^i &= \begin{cases} \langle \nu_k, \phi_{i,j} \rangle_\Gamma & \forall i = 1, 2, j = 1, \dots, M_i, k = 1, \dots, m, \\ 0 & \forall i = 1, 2, j = M_i + 1, \dots, m_i, k = 1, \dots, m, \end{cases} \\ f_j^i &= \begin{cases} (f_i, \phi_{i,j}) + \langle g_{i,N}, \phi_{i,j} \rangle_{\Gamma_i^N} & \forall i = 1, 2, j = 1, \dots, M_i, \\ u_{i,D}(\mathbf{x}_j) & \forall i = 1, 2, j = M_i + 1, \dots, m_i. \end{cases} \end{aligned}$$

Rewriting (3.3) in matrix form we obtain the following:

$$\begin{pmatrix} \mathbf{A}_1 & 0 & -\mathbf{C}_1 \\ 0 & \mathbf{A}_2 & \mathbf{C}_2 \\ -\mathbf{C}_1^T & \mathbf{C}_2^T & 0 \end{pmatrix} \begin{pmatrix} \mathbf{U}_1 \\ \mathbf{U}_2 \\ \boldsymbol{\Lambda} \end{pmatrix} = \begin{pmatrix} \mathbf{F}_1 \\ \mathbf{F}_2 \\ 0 \end{pmatrix}. \quad (3.4)$$

Equation (3.4) represents a basic discretised abstract monolithic coupled problem. The matrix of this system has clear block structure with terms corresponding to individual components, and those related to coupling being entirely separate thanks to the Lagrange multiplier formulation. This structure leads to effective use of block preconditioning, as will be seen in Chapter 4.

## 3.4 Parabolic coupled problem

The derivation of the finite element method for a parabolic coupled problem follows directly from the above and Section 2.5. For the sake of completeness we present this derivation here in full, and later refer to the relevant equations. Domains, functions, and time intervals are defined as above.

The strong formulation of a coupled parabolic problem is as follows:

$$\frac{\partial u_i}{\partial t} - \nabla \cdot (k_i \nabla u_i) + c_i u_i = f_i \quad \text{in } \Omega_i \times (0, T], \quad i = 1, 2, \quad (3.5a)$$

$$u_i = u_i^D \quad \text{on } \Gamma_i^D \times (0, T], \quad i = 1, 2, \quad (3.5b)$$

$$k_i \nabla u_i \cdot \mathbf{n}_i = g_{i,N} \quad \text{on } \Gamma_i^N \times (0, T], \quad i = 1, 2, \quad (3.5c)$$

$$u_1 - u_2 = 0 \quad \text{on } \Gamma \times (0, T], \quad (3.5d)$$

$$k_1 \nabla u_1 \cdot \mathbf{n}_1 + k_2 \nabla u_2 \cdot \mathbf{n}_2 = 0 \quad \text{on } \Gamma \times (0, T], \quad (3.5e)$$

$$u_i = u_{i,0} \quad \text{in } \Omega_i \times \{0\}, \quad i = 1, 2. \quad (3.5f)$$

Here  $u_{i,0}$  is a prescribed initial condition. This system can model, for example, reaction-diffusion between two environments with different rates of reaction-diffusion  $c_i$ ,  $k_i$ , or heat transfer if  $c_1 = c_2 = 0$ .

### 3.4.1 Weak formulation

The function spaces required for the weak formulation are defined as follows:

$$X_i = L^2(0, T; H^1(\Omega_i)),$$

$$X_{i,D} = \{v \in X \mid v = u_i^D \text{ on } \Gamma_i^D \times (0, T]\},$$



$$X_{i,0} = \{v \in X \mid v = 0 \text{ on } \Gamma_i^D \times (0, T]\},$$

$$Y = L^2(0, T; L^2(\Gamma)).$$

Multiplying (3.5a) and (3.5d) by arbitrary functions in the respective test spaces, integrating over the respective domains, and applying the divergence theorem results in the following weak formulation:

*Find  $(u_1, u_2, \lambda) \in X_{1,D} \times X_{2,D} \times Y$  such that*

$$\begin{aligned} \int_0^T \left\{ \mp \langle \lambda, v_i \rangle_\Gamma + \left( \frac{\partial u_i}{\partial t}, v_i \right) + a_i(u_i, v_i) \right\} dt \\ = \int_0^T \left\{ (f_i, v_i) + \langle g_{i,N}, v_i \rangle_{\Gamma_i^N} \right\} dt \quad \forall v_i \in X_{i,0}, \quad i = 1, 2, \end{aligned} \quad (3.6a)$$

$$\int_0^T \langle \nu, u_1 - u_2 \rangle_\Gamma dt = 0 \quad \forall \nu \in Y, \quad (3.6b)$$

$$u_i|_{t=0} = u_{i,0}, \quad i = 1, 2. \quad (3.6c)$$

### 3.4.2 Finite element method

Let  $\tau_i(\Omega_i)$ ,  $i = 1, 2$  and  $\tau(\Gamma)$  be domain and interface partitions as before. The following finite element spaces will be used\*:

$$X_i^h = \{v \in X_i \mid v(\cdot, t) \in P^k(\Omega_i) \quad \forall t \in (0, T], \quad v(x, \cdot) \in P^0(0, T) \quad \forall x \in \Omega_i\}, \quad i = 1, 2,$$

$$Y^h = \{\nu \in Y \mid \nu(\cdot, t) \in P^k(\Gamma) \quad \forall t \in (0, T], \quad \nu(s, \cdot) \in P^0(0, T) \quad \forall s \in \Gamma\},$$

and let the Dirichlet restrictions be  $X_{i,D}^h = X_i^h \cap X_{i,D}$  and  $X_{i,0}^h = X_i^h \cap X_{i,0}$ ,  $i = 1, 2$ .

---

\*  $P^0(0, T)$  is used to denote the space of discontinuous piecewise-constant functions in time.

The finite element method for (3.6) can then be written as:

Find  $(u_1^h, u_2^h, \lambda^h) \in X_{1,D}^h \times X_{2,D}^h \times Y^h$  with  $u_{i,0-}^h = u_{i,0}$ ,  $i = 1, 2$ , such that

$$\sum_{n=1}^N \left\{ \left( [u_i^h]_{n-1}, v_{i,(n-1)+}^h \right) - \int_{I_n} (R_i, v_i^h) dt \right\} = 0 \quad \forall v_i^h \in X_{i,0}^h, \quad i = 1, 2, \quad (3.7a)$$

$$\sum_{n=1}^N \int_{I_n} \langle \nu^h, u_1^h - u_2^h \rangle_{\Gamma} dt = 0 \quad \forall \nu^h \in Y^h, \quad (3.7b)$$

Here  $R_i$  denote the residuals:

$$\begin{aligned} (R_i, v_i) = & \pm \langle \lambda^h, v_i \rangle_{\Gamma} + (f_i, v_i) + \langle g_{i,N}, v_i \rangle_{\Gamma_i^N} \\ & - \left( \frac{\partial u_i^h}{\partial t}, v_i \right) - a_i(u_i^h, v_i) \quad \forall v_i \in X_i^h, \quad i = 1, 2. \end{aligned} \quad (3.8)$$

Using basis function expansions with coefficient vectors  $\mathbf{U}_{1,n}$ ,  $\mathbf{U}_{2,n}$  and  $\mathbf{\Lambda}_n$  at time step  $t_n$ , the component matrices  $\mathbf{A}_i$ , the coupling matrices  $\mathbf{C}_i$  and the load vectors  $\mathbf{F}_i(t)$ , and defining mass matrices  $\mathbf{M}_i = (m_{j,l}^i) \in \mathbb{R}^{m_i \times m_i}$ ,  $j, l = 1, \dots, m_i$ ,  $i = 1, 2$ :

$$m_{j,l}^i = \begin{cases} (\phi_{i,j}, \phi_{i,l}) & \forall j = 1, \dots, M_i, \quad l = 1, \dots, m_i, \\ \delta_{j,l} & \forall j = M_i + 1, \dots, m_i, \quad l = 1, \dots, m_i, \end{cases} \quad (3.9)$$

we can write the finite element method in matrix form:

Start with  $\mathbf{U}_{i,0}$ , the vector of coefficients in the basis expansion of  $u_{i,0}$  in the finite-dimensional space,  $i = 1, 2$ .

For  $n = 1, 2, \dots, N$

$$\begin{pmatrix} \mathbf{M}_1 + \Delta t_n \mathbf{A}_1 & 0 & -\Delta t_n \mathbf{C}_1 \\ 0 & \mathbf{M}_2 + \Delta t_n \mathbf{A}_2 & \Delta t_n \mathbf{C}_2 \\ -\Delta t_n \mathbf{C}_1^T & \Delta t_n \mathbf{C}_2^T & 0 \end{pmatrix} \begin{pmatrix} \mathbf{U}_{1,n} \\ \mathbf{U}_{2,n} \\ \boldsymbol{\Lambda}_n \end{pmatrix} = \begin{pmatrix} \mathbf{M}_1 \mathbf{U}_{1,n-1} + \Delta t_n \int_{I_n} \mathbf{F}_1(t) \, dt \\ \mathbf{M}_2 \mathbf{U}_{2,n-1} + \Delta t_n \int_{I_n} \mathbf{F}_2(t) \, dt \\ 0 \end{pmatrix}. \quad (3.10)$$

The structure of this matrix equation is similar to (3.4), but the assembly procedure is more time-consuming, since the right-hand sides have to be re-evaluated at every time step.

In this chapter we have discussed the partitioned and monolithic approaches to solving coupled problems. We propose to use a block preconditioner on the monolithic coupled problem with the aim to explore the advantages of each of the approaches, namely modularity offered by the partitioned schemes, as well as robustness and reliability of the monolithic solvers. Finally, we derive the weak and finite element formulations of elliptic and parabolic problems coupled using a Lagrange multiplier. These formulations are used in the following chapters, where we investigate solver choices for the coupled system and derive the *a posteriori* error estimates for the finite element solutions of coupled problems.

# Chapter 4

## Efficient solution of coupled systems using preconditioners

In Chapter 3 we discussed the partitioned and monolithic approaches to solving coupled systems. Specifically we noted that monolithic methods, whereby the system equations are solved simultaneously, are generally more robust and reliable, but lack the modularity of the partitioned schemes. We proposed that by solving the monolithic system using an iterative method with a block preconditioner, we gain the advantages of both types of techniques.

In this chapter we revisit this argument by considering the matrix form of the finite element method for a coupled problem. Next, we investigate several approximations to the exact block preconditioner, which are computationally efficient for large-scale applications. A series of numerical tests is carried out in order to establish the effect of approximation in the preconditioner on the iterative solver convergence, and an efficient solver for coupled systems is proposed based on the test results and efficiency considerations.

## 4.1 Exact block preconditioner

Irrespective of the nature of the monolithic coupled problem considered, the system arising from a finite element formulation with Lagrange multipliers has the following block structure, see Equations (3.4) and (3.10):

$$\underbrace{\begin{pmatrix} \mathbf{A}_1 & 0 & -\mathbf{C}_1 \\ 0 & \mathbf{A}_2 & \mathbf{C}_2 \\ -\mathbf{C}_1^T & \mathbf{C}_2^T & 0 \end{pmatrix}}_{\mathbf{A}} \underbrace{\begin{pmatrix} \mathbf{U}_1 \\ \mathbf{U}_2 \\ \boldsymbol{\Lambda} \end{pmatrix}}_{\mathbf{U}} = \underbrace{\begin{pmatrix} \mathbf{F}_1 \\ \mathbf{F}_2 \\ 0 \end{pmatrix}}_{\mathbf{F}}. \quad (4.1)$$

We refer to (4.1) as an abstract coupled problem. Generally speaking, the matrix  $\mathbf{A}$  need not be symmetric or positive definite, permitting the use of any non-singular blocks  $\mathbf{A}_1$  and  $\mathbf{A}_2$  (for example, such as those in the discrete version of the Navier-Stokes equations). The only assumption we make regarding  $\mathbf{A}$  is that it is non-singular, i.e. that the discrete problem we are solving is well-posed.

As mentioned earlier, it is not practical to solve (4.1) with direct methods when the dimensions of the system components  $\mathbf{A}_1$  and  $\mathbf{A}_2$  are high. Moreover, straightforward application of the appropriate iterative methods, namely GMRES for non-symmetric, or MINRES for symmetric problems, is neither fast, nor efficient, since the condition number of the system matrix  $\mathbf{A}$ ,  $\text{cond}(\mathbf{A}) = \|\mathbf{A}\| \|\mathbf{A}^{-1}\|$ , can be extremely high [19]. Consequently, a preconditioner for  $\mathbf{A}$  should be sought to improve convergence of the iterative method [82, 91, 95, 111].

More specifically, we consider a right preconditioner  $\mathbf{P}$  leading to the following system:

$$\mathbf{A}\mathbf{P}^{-1}\mathbf{V} = \mathbf{F}, \quad (4.2a)$$

$$\mathbf{P}\mathbf{U} = \mathbf{V}. \quad (4.2b)$$

The choice of  $\mathbf{P}$  should be such that both equations (4.2a) and (4.2b) are easier to solve than the original equation (4.1). Namely, the product  $\mathbf{A}\mathbf{P}^{-1}$  should have a significantly lower condition number than the original matrix  $\mathbf{A}$ , and  $\mathbf{P}^{-1}\mathbf{V}$  should be cheap to evaluate for a given vector  $\mathbf{V}$ . Moreover, we wish to take advantage of the block structure of the system, since there may already exist efficient methods for solving the individual subproblems (specifically, for evaluating the product  $\mathbf{A}_i^{-1}\mathbf{V}_i$  for given vectors  $\mathbf{V}_i$ ,  $i = 1, 2$ ).

Direct application of forward Gaussian elimination by columns [67, 91] to (4.1) leads to the following block lower-triangular matrix with identity blocks  $\mathbf{I}$  on the diagonal:

$$\mathbf{A}\mathbf{P}^{-1} = \begin{pmatrix} \mathbf{I} & 0 & 0 \\ 0 & \mathbf{I} & 0 \\ -\mathbf{C}_1^T\mathbf{A}_1^{-1} & \mathbf{C}_2^T\mathbf{A}_2^{-1} & \mathbf{I} \end{pmatrix}. \quad (4.3)$$

The corresponding right preconditioner is

$$\mathbf{P} = \begin{pmatrix} \mathbf{A}_1 & 0 & -\mathbf{C}_1 \\ 0 & \mathbf{A}_2 & \mathbf{C}_2 \\ 0 & 0 & -\mathbf{S} \end{pmatrix}, \quad \mathbf{P}^{-1} = \begin{pmatrix} \mathbf{A}_1^{-1} & 0 & -\mathbf{A}_1^{-1}\mathbf{C}_1\mathbf{S}^{-1} \\ 0 & \mathbf{A}_2^{-1} & \mathbf{A}_2^{-1}\mathbf{C}_2\mathbf{S}^{-1} \\ 0 & 0 & -\mathbf{S}^{-1} \end{pmatrix}, \quad (4.4)$$

where the matrix  $\mathbf{S}$  is known as the Schur complement, and is computed as follows:

$$\mathbf{S} = \mathbf{C}_1^T \mathbf{A}_1^{-1} \mathbf{C}_1 + \mathbf{C}_2^T \mathbf{A}_2^{-1} \mathbf{C}_2. \quad (4.5)$$

All eigenvalues of  $\mathbf{A}\mathbf{P}^{-1}$  are equal to 1, and (4.2a) can therefore be solved by GMRES in 2 iterations [91]. At the same time, we have an explicit solution to (4.2b) in (4.4). Both matrix equations can be fully resolved assuming we have a way of computing the action of  $\mathbf{A}_i^{-1}$ ,  $i = 1, 2$ , and  $\mathbf{S}^{-1}$  on vectors of appropriate dimensions. We have therefore shown that using a block preconditioner naturally leads to modularity, while allowing the solver to retain its monolithic nature.

However, in a practical application using (4.3) and (4.4) is likely to be insufficient for full modularity. This will be the case if the existing component solvers do not perform full factorisations of the respective matrices  $\mathbf{A}_i$ ,  $i = 1, 2$ , but instead can only compute the approximate action of  $\mathbf{A}_i^{-1}$  on vectors using, for example, an iterative solver. In order to solve the system (4.2a)-(4.2b) we need to be able to compute the action of  $\mathbf{S}^{-1} = (\mathbf{C}_1^T \mathbf{A}_1^{-1} \mathbf{C}_1 + \mathbf{C}_2^T \mathbf{A}_2^{-1} \mathbf{C}_2)^{-1}$ . If the exact formula for the inverse of the Schur complement is used then full factorisations of  $\mathbf{A}_i$ ,  $i = 1, 2$ , are required. Assuming that these are not available, the next logical step would be to approximate the action of  $\mathbf{S}^{-1}$  on a vector using the actions of  $\mathbf{A}_i^{-1}$  on vectors. A preconditioner using such an approximation  $\hat{\mathbf{S}}^{-1}$  instead of  $\mathbf{S}^{-1}$  would lead to the following matrix in (4.2a):

$$\mathbf{A}\mathbf{P}^{-1} = \begin{pmatrix} \mathbf{I} & 0 & 0 \\ 0 & \mathbf{I} & 0 \\ -\mathbf{C}_1^T \mathbf{A}_1^{-1} & \mathbf{C}_2^T \mathbf{A}_2^{-1} & \mathbf{S}\hat{\mathbf{S}}^{-1} \end{pmatrix}.$$

The majority of the eigenvalues of this matrix are still equal to 1, apart from those

coming from the  $\mathbf{S}\hat{\mathbf{S}}^{-1}$  block. Convergence of the preconditioned GMRES will depend solely on these eigenvalues. If  $N_\lambda$  is the number of distinct eigenvalues of  $\mathbf{S}$  then the algorithm constructs all of Krylov subspaces for  $\mathbf{A}\mathbf{P}^{-1}$  in  $N_\lambda + 1$  steps (see [67, Theorem 10.2.5]). It is therefore guaranteed to converge in  $N_\lambda + 1 \leq \text{rank}(\mathbf{S}) + 1$  iterations. Since the dimension of  $\mathbf{S}$  corresponds to the boundary degrees of freedom, while the dimensions of  $\mathbf{A}_i$ ,  $i = 1, 2$ , correspond to the domain degrees of freedom,  $\text{rank}(\mathbf{S})$  is likely to be much smaller than  $\text{rank}(\mathbf{A})$ , and the number of iterations will therefore be much lower than in the unpreconditioned GMRES. Of course, convergence in  $\text{rank}(\mathbf{S}) + 1$  iterations can still be too slow, as  $\text{rank}(\mathbf{S})$  grows with mesh size. Moreover, if the mesh is locally refined around the interface, then the growth of  $\text{rank}(\mathbf{S})$  can be asymptotically as fast as  $\mathcal{O}(h^{-D})$ , where  $h$  is the smallest element radius, and  $D$  is the dimension of the spatial variable. It is therefore essential that  $\mathbf{S}^{-1}$  is approximated well. Different options for this approximation are considered below.

## 4.2 Schur complement approximations

We revisit the expression for the Schur complement (4.5):

$$\mathbf{S} = \mathbf{C}_1^T \mathbf{A}_1^{-1} \mathbf{C}_1 + \mathbf{C}_2^T \mathbf{A}_2^{-1} \mathbf{C}_2.$$

We may consider each of the coupling matrices  $\mathbf{C}_i$ ,  $i = 1, 2$ , as being composed of two blocks:  $\mathbf{C}_{i,I}$ , which projects the degrees of freedom on the interface to the Lagrange multiplier space, and an all-zero block for the remaining degrees of freedom. There is an ordering of the system degrees of freedom such that coupling and component matrices can be represented in terms of interface and domain degrees of freedom ( $I$



and  $D$  respectively):

$$\mathbf{C}_i = \begin{pmatrix} 0 \\ \mathbf{C}_{i,I} \end{pmatrix}, \quad \mathbf{A}_i = \begin{pmatrix} \mathbf{A}_{i,DD} & \mathbf{A}_{i,DI} \\ \mathbf{A}_{i,ID} & \mathbf{A}_{i,II} \end{pmatrix}, \quad i = 1, 2. \quad (4.6)$$

We can use block Gaussian elimination to invert  $\mathbf{A}_i$ ,  $i = 1, 2$ . The following block system results from one step of forward row elimination:

$$\begin{pmatrix} \mathbf{I} & -\mathbf{A}_{i,DI}\mathbf{A}_{i,II}^{-1} \\ 0 & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{A}_{i,DD} & \mathbf{A}_{i,DI} \\ \mathbf{A}_{i,ID} & \mathbf{A}_{i,II} \end{pmatrix} = \underbrace{\begin{pmatrix} \mathbf{A}_{i,DD} - \mathbf{A}_{i,DI}\mathbf{A}_{i,II}^{-1}\mathbf{A}_{i,ID} & 0 \\ \mathbf{A}_{i,ID} & \mathbf{A}_{i,II} \end{pmatrix}}_{\mathbf{RHS}}. \quad (4.7)$$

The invertibility of  $\mathbf{A}_{i,II}$  blocks for  $i = 1, 2$  follows from the invertibility of the finite element matrix for the problem resulting in  $\mathbf{A}_i$  solved on the domain containing only the strip of elements adjacent to the interface with Dirichlet boundary conditions applied on all boundaries of this strip, apart from the interface itself. Since we assume that the original component problems are well-posed on a polygonal domain  $\Omega_i$  with appropriate boundary conditions, they will also be well-posed on the polygonal domain made up of elements with an edge or a node on  $\Gamma$  with these boundary conditions. The invertibility of  $\mathbf{A}_{i,DD}$ ,  $i = 1, 2$  (required further in the text) follows similarly from the invertibility of the finite element matrix for the problem resulting in  $\mathbf{A}_i$  solved on the whole domain  $\Omega_i$  with a Dirichlet boundary condition applied on the interface.

We define component Schur complements  $\mathbf{S}_{i,A} = \mathbf{A}_{i,DD} - \mathbf{A}_{i,DI}\mathbf{A}_{i,II}^{-1}\mathbf{A}_{i,ID}$ ,  $i = 1, 2$ . Multiplying (4.7) by  $\mathbf{A}_i^{-1}$  on the right, and then by  $\mathbf{RHS}^{-1}$  on the left yields the

following expressions for the inverses of the component matrices:

$$\begin{aligned}
\mathbf{A}_i^{-1} &= \begin{pmatrix} \mathbf{S}_{i,A} & 0 \\ \mathbf{A}_{i,ID} & \mathbf{A}_{i,II} \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{I} & -\mathbf{A}_{i,DI}\mathbf{A}_{i,II}^{-1} \\ 0 & \mathbf{I} \end{pmatrix} \\
&= \begin{pmatrix} \mathbf{S}_{i,A}^{-1} & 0 \\ -\mathbf{A}_{i,II}^{-1}\mathbf{A}_{i,ID}\mathbf{S}_{i,A}^{-1} & \mathbf{A}_{i,II}^{-1} \end{pmatrix} \begin{pmatrix} \mathbf{I} & -\mathbf{A}_{i,DI}\mathbf{A}_{i,II}^{-1} \\ 0 & \mathbf{I} \end{pmatrix} \\
&= \begin{pmatrix} \mathbf{S}_{i,A}^{-1} & -\mathbf{S}_{i,A}^{-1}\mathbf{A}_{i,DI}\mathbf{A}_{i,II}^{-1} \\ -\mathbf{A}_{i,II}^{-1}\mathbf{A}_{i,ID}\mathbf{S}_{i,A}^{-1} & \mathbf{A}_{i,II}^{-1} + \mathbf{R}_{i,A} \end{pmatrix}, \quad i = 1, 2,
\end{aligned}$$

where  $\mathbf{R}_{i,A} = \mathbf{A}_{i,II}^{-1}\mathbf{A}_{i,ID}\mathbf{S}_{i,A}^{-1}\mathbf{A}_{i,DI}\mathbf{A}_{i,II}^{-1}$ ,  $i = 1, 2$ .

We also note that

$$\mathbf{C}_i^T \mathbf{A}_i^{-1} \mathbf{C}_i = \begin{pmatrix} 0 & \mathbf{C}_{i,I}^T \end{pmatrix} \mathbf{A}_i^{-1} \begin{pmatrix} 0 \\ \mathbf{C}_{i,I} \end{pmatrix} = \mathbf{C}_{i,I}^T (\mathbf{A}_{i,II}^{-1} + \mathbf{R}_{i,A}) \mathbf{C}_{i,I},$$

yielding

$$\mathbf{S} = \sum_{i=1,2} \mathbf{C}_{i,I}^T (\mathbf{A}_{i,II}^{-1} + \mathbf{R}_{i,A}) \mathbf{C}_{i,I}. \quad (4.8)$$

Similarly to (4.6), there exists an ordering of the system degrees of freedom (with those corresponding to the interface coming first) such that

$$\mathbf{C}_i = \begin{pmatrix} \mathbf{C}_{i,I} \\ 0 \end{pmatrix}, \quad \mathbf{A}_i = \begin{pmatrix} \mathbf{A}_{i,II} & \mathbf{A}_{i,ID} \\ \mathbf{A}_{i,DI} & \mathbf{A}_{i,DD} \end{pmatrix}. \quad (4.9)$$

As before, with appropriate invertibility assumptions we can use block Gaussian elim-

ination to arrive at the following expression for the component inverses:

$$\mathbf{A}_i^{-1} = \begin{pmatrix} \mathbf{S}_{i,A}^{-1} & -\mathbf{S}_{i,A}^{-1} \mathbf{A}_{i,ID} \mathbf{A}_{i,DD}^{-1} \\ -\mathbf{A}_{i,DD}^{-1} \mathbf{A}_{i,DI} \mathbf{S}_{i,A}^{-1} & \mathbf{A}_{i,DD}^{-1} + \mathbf{R}_{i,A} \end{pmatrix},$$

where  $\mathbf{S}_{i,A} = \mathbf{A}_{i,II} - \mathbf{A}_{i,ID} \mathbf{A}_{i,DD}^{-1} \mathbf{A}_{i,DI}$  and  $\mathbf{R}_{i,A} = \mathbf{A}_{i,DD}^{-1} \mathbf{A}_{i,DI} \mathbf{S}_{i,A}^{-1} \mathbf{A}_{i,ID} \mathbf{A}_{i,DD}^{-1}$ . Consequently, another expression for the Schur complement is

$$\mathbf{S} = \sum_{i=1,2} \mathbf{C}_{i,I}^T (\mathbf{A}_{i,II} - \mathbf{A}_{i,ID} \mathbf{A}_{i,DD}^{-1} \mathbf{A}_{i,DI})^{-1} \mathbf{C}_{i,I}. \quad (4.10)$$

Both expressions (4.8) and (4.10) are exact. As discussed before, the use of preconditioned GMRES is reliant on approximating the action of  $\mathbf{S}^{-1}$  well. If we use the exact formula for  $\mathbf{S}^{-1}$  then it is insufficient to be able to compute the action of  $\mathbf{A}_{i,II}$  on a vector of appropriate size – the full inverse of  $\mathbf{A}_{i,II}$  is required. Moreover, both expressions include inverses of blocks associated with domain degrees of freedom, which are by no means feasible to compute. The respective expressions for  $\mathbf{S}^{-1}$ ,

$$\begin{aligned} \mathbf{S}^{-1} &= \left( \sum_{i=1,2} \mathbf{C}_{i,I}^T (\mathbf{A}_{i,II}^{-1} + \mathbf{R}_{i,A}) \mathbf{C}_{i,I} \right)^{-1}, \text{ or} \\ \mathbf{S}^{-1} &= \left( \sum_{i=1,2} \mathbf{C}_{i,I}^T (\mathbf{A}_{i,II} - \mathbf{A}_{i,ID} \mathbf{A}_{i,DD}^{-1} \mathbf{A}_{i,DI})^{-1} \mathbf{C}_{i,I} \right)^{-1} \end{aligned}$$

involve inverses of sums of matrices, which in the general case cannot be expanded. We write two naive approximations to such inverses, and assess their applicability along with an alternative approximation to  $\mathbf{S}^{-1}$ . The different approximations are denoted by  $\mathbf{S}_k$ ,  $k = 1, 2, \dots$

If we assume that the interaction between the degrees of freedom on the interface and those in the domains is negligible then the off-diagonal blocks of  $\mathbf{A}_i$  are close

to zero, and therefore  $\mathbf{A}_i$  is block-diagonal. In this case  $\mathbf{R}_{i,A} = 0$  in (4.8) and  $\mathbf{A}_{ID}\mathbf{A}_{DD}^{-1}\mathbf{A}_{DI} = 0$  in (4.10), and either expression for  $\mathbf{S}$  can be reduced to the following approximation:

$$\mathbf{S}_1 = \sum_{i=1,2} \mathbf{C}_{i,I}^T \mathbf{A}_{i,II}^{-1} \mathbf{C}_{i,I}. \quad (4.11)$$

Using this approximation is computationally cheaper than the exact expressions, since it only involves computing the inverse of the  $II$ -block corresponding to interface nodes. For regular meshes the size of this block, equal to  $\text{rank}(\mathbf{S})$ , is of the order of the square root of the dimension of the domain block. However, in a problem where the mesh has to be refined around the interface, the size of the interface block can be of the same order as the domain block, and the computational cost of fully inverting  $\mathbf{A}_{i,II}$  would be too high.

We can weaken the assumptions by letting  $\mathbf{R}_{i,A}$  in (4.8) be nonnegligible, but supposing that  $\mathbf{A}_{i,DD}$  is the dominating term in the expression for  $\mathbf{S}_{i,A}$ ,  $i = 1, 2$ . This leads to the second approximation:

$$\mathbf{S}_2 = \sum_{i=1,2} \mathbf{C}_{i,I}^T \left( \mathbf{A}_{i,II}^{-1} + \mathbf{A}_{i,II}^{-1} \mathbf{A}_{i,ID} \mathbf{A}_{i,DD}^{-1} \mathbf{A}_{i,DI} \mathbf{A}_{i,II}^{-1} \right) \mathbf{C}_{i,I}. \quad (4.12)$$

Both  $\mathbf{S}_1$  and  $\mathbf{S}_2$  provide approximations to the actual Schur complement, which is a dense matrix. Even though its size is normally significantly smaller than that of the full system, computing its inverse is a costly operation. If we were able to approximate  $\mathbf{S}^{-1}$  directly, the computational cost of the method would be significantly reduced.

Doing this turns out to be problematic, since  $\mathbf{S}$  is a sum of two matrices. However, if we make a crude approximation by operating in terms of orders of magnitude, a subsequent simplification can be achieved. Namely, whether the terms in the sum are similar, i.e. of the same order, or one of them dominates the sum, we may write

$\mathbf{S} \sim \mathbf{C}_I^T \mathbf{A}_{II}^{-1} \mathbf{C}_I$ , where  $\mathbf{C}_I$  and  $\mathbf{A}_{II}$  refer to the blocks corresponding to either  $i = 1$  or  $i = 2$  if the terms are similar, or the dominating  $i$  if one term dominates. In both cases  $\mathbf{S}^{-1} \sim \mathbf{C}_I^{-T} \mathbf{A}_{II} \mathbf{C}_I^{-1}$ . This approximation is a poor one if both terms are significant, but not similar. However, it can be close enough for the purpose of preconditioning in certain problems of interest. This will be tested further, along with the rest of approximations.

Generally speaking,  $\mathbf{C}_I$  can be a rectangular matrix, which maps finite element functions on the interface  $\Gamma$  to Lagrange multiplier functions, and these need not be in the same space. However, we always choose the Lagrange multiplier space to coincide with the richest interface space out of the two, i.e. that containing the most degrees of freedom. The respective matrices are used to compute the approximation to  $\mathbf{S}^{-1}$ . In this case,  $\mathbf{C}_I$  is a square symmetric invertible matrix. Moreover, a practical choice of the component  $i = 1$  or  $i = 2$  used to build the inverse of the Schur complement should be informed by our knowledge of the physical systems involved. This question is further discussed in Sections 4.3, 4.5.1.4, 6.3.

Making use of the approximation based on the dominant component, we invert the significant term(s) in (4.11) and (4.12) to obtain

$$\mathbf{S}_3^{-1} \sim \mathbf{C}_I^{-T} \mathbf{A}_{II} \mathbf{C}_I^{-1}, \quad (4.13)$$

$$\mathbf{S}_4^{-1} \sim \mathbf{C}_I^{-T} (\mathbf{A}_{II} - \mathbf{A}_{ID} \mathbf{A}_{DD}^{-1} \mathbf{A}_{DI}) \mathbf{C}_I^{-1}. \quad (4.14)$$

Using the approximation  $\mathbf{S}_3^{-1}$  has a very low associated computational cost, since, aside from sparse multiplication, it only requires applying the action of the inverse of what is normally a banded matrix  $\mathbf{C}_I$ .

The cost of using  $\mathbf{S}_4$  is higher – the additional expense comes from computing the

action of  $\mathbf{A}_{DD}^{-1}$  on a vector. However, this normally requires an order of operations less than computing the full inverse, as in the case of  $\mathbf{S}_2$ . In terms of computational cost, both  $\mathbf{S}_3$  and  $\mathbf{S}_4$  have a significant advantage over  $\mathbf{S}_1$  and  $\mathbf{S}_2$ .

An alternative approximation for the inverse of the Schur complement can be obtained using the idea behind the *BFBt* preconditioner, or approximate commutators developed and analysed for Navier-Stokes equations in [44, 45]. We denote

$$\mathbf{A}_F = \begin{pmatrix} \mathbf{A}_1 & 0 \\ 0 & \mathbf{A}_2 \end{pmatrix}, \quad \mathbf{C}_F = \begin{pmatrix} -\mathbf{C}_1 \\ \mathbf{C}_2 \end{pmatrix},$$

so that the Schur complement can be written as  $\mathbf{S} = \mathbf{C}_F^T \mathbf{A}_F^{-1} \mathbf{C}_F$ . This expression is approximated by commuting the factors in the product and replacing  $\mathbf{A}_F^{-1}$  with a lower-dimensional counterpart, which is obtained by solving a least-squares problem  $\mathbf{C}_F \tilde{\mathbf{A}}_F = \mathbf{A}_F \mathbf{C}_F$ . We denote this approximation by  $\mathbf{S}_5$ , and refer to [45] for the detailed derivation.

$$\begin{aligned} \mathbf{S}_5^{-1} &= (\mathbf{C}_F^T \mathbf{C}_F)^{-1} \mathbf{C}_F^T \mathbf{A}_F \mathbf{C}_F (\mathbf{C}_F^T \mathbf{C}_F)^{-1} \\ &= (\mathbf{C}_1^T \mathbf{C}_1 + \mathbf{C}_2^T \mathbf{C}_2)^{-1} (\mathbf{C}_1^T \mathbf{A}_1 \mathbf{C}_1 + \mathbf{C}_2^T \mathbf{A}_2 \mathbf{C}_2) (\mathbf{C}_1^T \mathbf{C}_1 + \mathbf{C}_2^T \mathbf{C}_2)^{-1}. \end{aligned}$$

We will investigate the performance of this preconditioner along with the former four, noting that its use does not require us to compute full inverses of  $\mathbf{A}$  blocks, only actions of inverses of banded matrices  $(\mathbf{C}_F^T \mathbf{C}_F)$ .

In order to select an approximation for the Schur complement best suited to preconditioning of coupled problems, we test the methods' performance numerically in Section 4.3.

### 4.3 Numerical convergence study

In this section we perform numerical tests to investigate the performance of preconditioned GMRES using the Schur complement approximations described above.

Initially we consider two types of problems: one with elliptic-elliptic coupling, and another with elliptic-parabolic coupling. In both scenarios we test the approximations obtained on two-dimensional meshes refined uniformly or only along the interface (see Figure 4.1a).

In the former case the ratio of the dimension of  $\mathbf{A}$  to the dimension of  $\mathbf{S}$  grows as the mesh is refined. This should lead to relative improvement of convergence with the size of the problem, since even the worst-case number of iterations  $\text{rank}(\mathbf{S}) + 1$  grows slower than the problem size. However, when the mesh is refined specifically around the coupling interface, the ratio of dimensions of  $\mathbf{A}$  and  $\mathbf{S}$  decreases, leading to unsatisfactory worst-case convergence. If any of the preconditioners perform poorly, this should be most obvious in the case of interface refinement.

In addition we perform tests on meshes that do not conform on the interface. In this case the terms  $\mathbf{C}_1^T \mathbf{A}_1^{-1} \mathbf{C}_1$  and  $\mathbf{C}_2^T \mathbf{A}_2^{-1} \mathbf{C}_2$  will be qualitatively different, and the test results will provide an indication as to whether the crude approximation of the inverse of a sum of matrices can be used successfully.

Let  $\Omega_1 = (0, 1) \times (0, 1)$  and  $\Omega_2 = (1, 2) \times (0, 1)$ , so that  $\Gamma = \{(1, y) \mid y \in (0, 1)\}$ , and let the domain boundaries be as follows:

$$\begin{aligned}\Gamma_1^D &= \{(0, y) \mid y \in (0, 1)\}, \quad \Gamma_1^N = \{(x, y) \mid x \in (0, 1), y = 0, 1\}, \\ \Gamma_2^D &= \{(2, y) \mid y \in (0, 1)\}, \quad \Gamma_2^N = \{(x, y) \mid x \in (1, 2), y = 0, 1\}.\end{aligned}$$

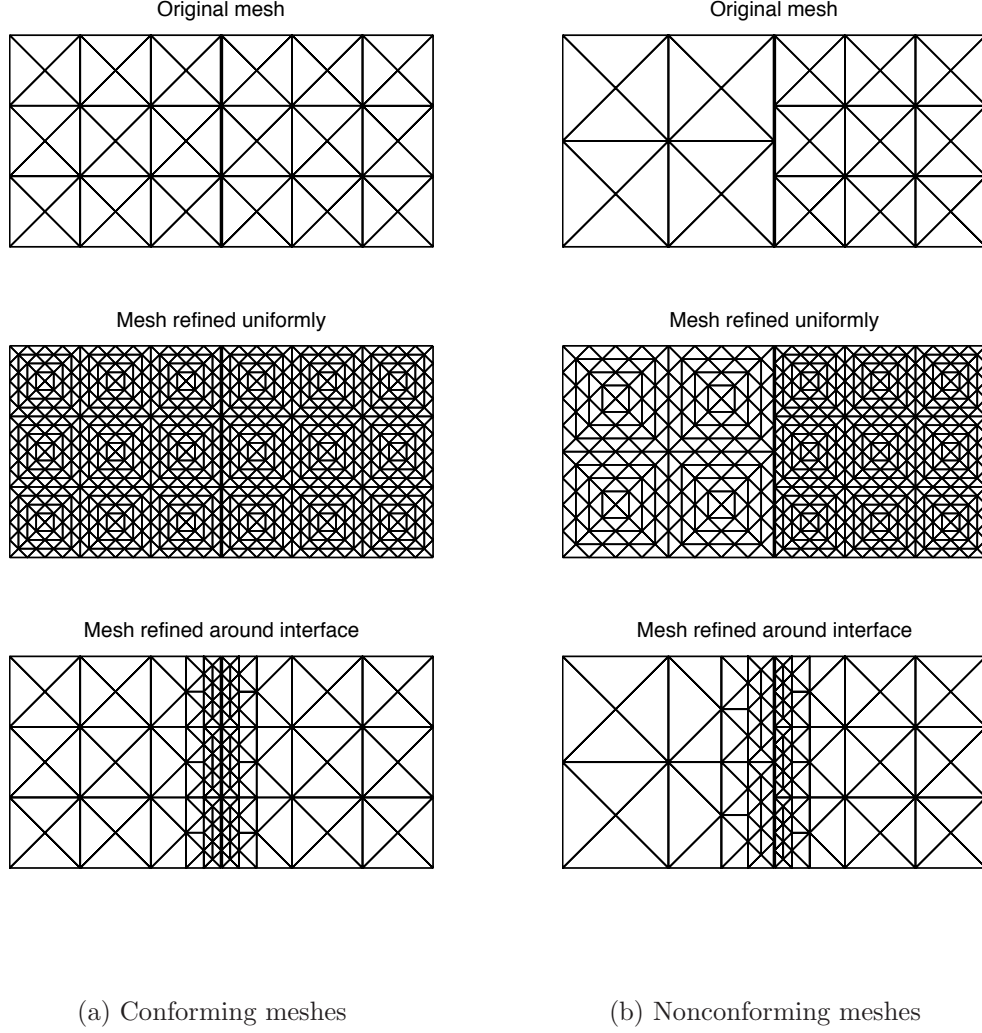


Figure 4.1: Uniform vs interface refinement of meshes that conform/do not conform on the interface  $\Gamma$ .

We consider the elliptic-elliptic coupled problem (3.1) with  $c_1 = c_2 = 0$ :

$$-\nabla \cdot (k_i \nabla u_i) = f_i \quad \text{in } \Omega_i, \quad i = 1, 2,$$

$$u_i = u_{i,D} \quad \text{on } \Gamma_i^D, \quad i = 1, 2,$$

$$k_i \nabla u_i \cdot \mathbf{n}_i = g_{i,N} \quad \text{on } \Gamma_i^N, \quad i = 1, 2,$$

$$u_1 - u_2 = 0 \quad \text{on } \Gamma,$$



$$k_1 \nabla u_1 \cdot \mathbf{n}_1 + k_2 \nabla u_2 \cdot \mathbf{n}_2 = 0 \quad \text{on } \Gamma.$$

The right-hand side functions  $f_i \equiv k_i$  and the boundary conditions  $u_{i,D} \equiv 0$ ,  $g_{i,N} \equiv 0$  are chosen so that the functions  $u_i = 0.5x(2 - x)$  are the exact solutions for this problem with a given choice of constants  $k_i$ ,  $i = 1, 2$ .

The test elliptic-parabolic problem was:

$$\begin{aligned} \frac{\partial u_1}{\partial t} - \nabla \cdot (k_1 \nabla u_1) &= f_1 \text{ in } \Omega_1 \times (0, T], \\ -\nabla \cdot (k_2 \nabla u_2) &= f_2 \text{ in } \Omega_2 \times (0, T], \end{aligned}$$

with coupling and boundary conditions (3.1b)-(3.1e) as above for  $t \in (0, T]$ . Here the data are chosen so that  $u_i = 0.5tx(2 - x)$ , i.e.  $f_1 = 0.5x(2 - x) + k_1t$ ,  $f_2 = k_2t$ . The initial condition is  $u_1 = u_2 = 0$  at  $t = 0$ .

In all examples we used  $k_1 = 1$ ,  $k_2 = 0.1$ , unless stated otherwise.

The finite element spaces in both problems in this section comprised piecewise-linear functions. Time-stepping was performed using the implicit Euler scheme with  $\Delta t = 10^{-1}$ .

GMRES was only used to solve Equation (4.2a), and wherever we needed to invert one of the blocks a direct solver was employed. In practice, one would use an existing efficient solver for each subproblem, depending on its form.

$\mathbf{S}_3^{-1}$  and  $\mathbf{S}_4^{-1}$  were computed based on the matrices arising from the first of the identical components in the case of elliptic-elliptic coupling on conforming meshes, the parabolic component in the case of elliptic-parabolic coupling on conforming meshes, and the component defined on the finer mesh in the case of elliptic-elliptic and elliptic-parabolic coupling on non-conforming meshes.

GMRES iteration counts for both problems on conforming and nonconforming meshes with uniform and interface refinements are shown in Tables 4.1-4.3. The stopping criterion was for the relative residual error to be less than  $10^{-10}$ . Throughout this section  $DOF$  denotes the total number of degrees of freedom in the system, and  $DOF_{\Gamma}$  denotes the number of degrees of freedom corresponding to the interface, equal to the dimension of the Schur complement. The results for the time-dependent problem are given for the first time step, since the iteration counts did not change significantly throughout the simulation.

|        |                | Elliptic-Elliptic |                |                |                |                | Elliptic-Parabolic |                |                |                |                |
|--------|----------------|-------------------|----------------|----------------|----------------|----------------|--------------------|----------------|----------------|----------------|----------------|
| $DOF$  | $DOF_{\Gamma}$ | $\mathbf{S}_1$    | $\mathbf{S}_2$ | $\mathbf{S}_3$ | $\mathbf{S}_4$ | $\mathbf{S}_5$ | $\mathbf{S}_1$     | $\mathbf{S}_2$ | $\mathbf{S}_3$ | $\mathbf{S}_4$ | $\mathbf{S}_5$ |
| 3403   | 41             | 14                | 11             | 29             | 21             | 12             | 14                 | 11             | 29             | 22             | 12             |
| 13203  | 81             | 17                | 14             | 36             | 20             | 15             | 17                 | 14             | 35             | 21             | 14             |
| 52003  | 161            | 21                | 17             | 43             | 19             | 19             | 21                 | 17             | 42             | 20             | 17             |
| 206403 | 321            | 24                | 20             | 50             | 17             | 22             | 24                 | 20             | 50             | 18             | 19             |

Table 4.1: GMRES iteration counts for 4 uniform refinements of the original mesh ( $41 \times 41$  points in each sub-domain) for the elliptic-elliptic and the elliptic-parabolic coupled problems.

|       |                | Elliptic-Elliptic |                |                |                |                | Elliptic-Parabolic |                |                |                |                |
|-------|----------------|-------------------|----------------|----------------|----------------|----------------|--------------------|----------------|----------------|----------------|----------------|
| $DOF$ | $DOF_{\Gamma}$ | $\mathbf{S}_1$    | $\mathbf{S}_2$ | $\mathbf{S}_3$ | $\mathbf{S}_4$ | $\mathbf{S}_5$ | $\mathbf{S}_1$     | $\mathbf{S}_2$ | $\mathbf{S}_3$ | $\mathbf{S}_4$ | $\mathbf{S}_5$ |
| 3403  | 41             | 16                | 12             | 29             | 21             | 12             | 14                 | 11             | 28             | 22             | 12             |
| 3659  | 81             | 18                | 14             | 34             | 19             | 15             | 19                 | 16             | 38             | 21             | 16             |
| 4170  | 161            | 26                | 18             | 40             | 18             | 17             | 24                 | 20             | 46             | 20             | 19             |
| 5185  | 321            | 34                | 21             | 47             | 16             | 20             | 30                 | 25             | 55             | 19             | 21             |

Table 4.2: GMRES iteration counts for 4 interface refinements of the original mesh ( $41 \times 41$  points in each sub-domain) for the elliptic-elliptic and the elliptic-parabolic coupled problems.

|        |              | Elliptic-Elliptic |                |                |                |                | Elliptic-Parabolic |                |                |                |                |
|--------|--------------|-------------------|----------------|----------------|----------------|----------------|--------------------|----------------|----------------|----------------|----------------|
| $DOF$  | $DOF_\Gamma$ | $\mathbf{S}_1$    | $\mathbf{S}_2$ | $\mathbf{S}_3$ | $\mathbf{S}_4$ | $\mathbf{S}_5$ | $\mathbf{S}_1$     | $\mathbf{S}_2$ | $\mathbf{S}_3$ | $\mathbf{S}_4$ | $\mathbf{S}_5$ |
| 2251   | 41           | 13                | 11             | 32             | 28             | 11             | 13                 | 11             | 32             | 28             | 13             |
| 8667   | 81           | 16                | 13             | 41             | 27             | 14             | 16                 | 13             | 41             | 28             | 16             |
| 34003  | 161          | 19                | 16             | 51             | 25             | 16             | 19                 | 16             | 51             | 26             | 20             |
| 134691 | 321          | 22                | 18             | 61             | 23             | 20             | 22                 | 18             | 61             | 24             | 25             |

Table 4.3: GMRES iteration counts for 4 uniform refinements of the original non-conforming mesh ( $41 \times 41$  points in  $\Omega_1$  and  $23 \times 23$  points in  $\Omega_2$ ) for the elliptic-elliptic and the elliptic-parabolic coupled problems.

In the elliptic-parabolic coupled problems the effect of the decreasing time step size is only significant when  $\mathbf{S}_4$  is used. All other approximations, notably  $\mathbf{S}_5$  appear unaffected by the decay in  $\Delta t$ . Representative convergence data for  $\Delta t = 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}$  is shown in Tables 4.4 (Schur complement based on the parabolic problem on the finer mesh) and 4.5 (Schur complement based on the elliptic problem on the finer mesh).

| $DOF$                 | $DOF_\Gamma$ | $\mathbf{S}_3$ |    |    |    | $\mathbf{S}_4$ |    |    |    | $\mathbf{S}_5$ |    |    |    |
|-----------------------|--------------|----------------|----|----|----|----------------|----|----|----|----------------|----|----|----|
| $-\log_{10} \Delta t$ |              | 1              | 2  | 3  | 4  | 1              | 2  | 3  | 4  | 1              | 2  | 3  | 4  |
| 2251                  | 41           | 27             | 29 | 30 | 29 | 22             | 23 | 23 | 21 | 13             | 13 | 13 | 20 |
| 2477                  | 81           | 38             | 40 | 41 | 38 | 24             | 26 | 25 | 23 | 16             | 16 | 15 | 18 |
| 2931                  | 161          | 43             | 46 | 48 | 46 | 21             | 22 | 24 | 22 | 19             | 19 | 18 | 18 |
| 3837                  | 321          | 48             | 51 | 54 | 53 | 18             | 20 | 21 | 21 | 22             | 23 | 22 | 20 |

Table 4.4: GMRES iteration counts for 4 interface refinements of the original mesh ( $41 \times 41$  points in  $\Omega_1$  and  $23 \times 23$  points in  $\Omega_2$ ) for the parabolic-elliptic coupled problem.

| $DOF$                 | $DOF_\Gamma$ | $\mathbf{S}_3$ |    |    |    | $\mathbf{S}_4$ |    |    |    | $\mathbf{S}_5$ |    |    |    |
|-----------------------|--------------|----------------|----|----|----|----------------|----|----|----|----------------|----|----|----|
| $-\log_{10} \Delta t$ |              | 1              | 2  | 3  | 4  | 1              | 2  | 3  | 4  | 1              | 2  | 3  | 4  |
| 2251                  | 41           | 28             | 30 | 31 | 32 | 23             | 26 | 29 | 31 | 13             | 13 | 12 | 13 |
| 2477                  | 81           | 38             | 40 | 41 | 42 | 25             | 28 | 34 | 41 | 16             | 16 | 15 | 15 |
| 2931                  | 161          | 43             | 46 | 48 | 47 | 22             | 24 | 31 | 40 | 19             | 19 | 18 | 17 |
| 3837                  | 321          | 48             | 51 | 53 | 54 | 18             | 21 | 27 | 36 | 22             | 23 | 21 | 20 |

Table 4.5: GMRES iteration counts for 4 interface refinements of the original mesh ( $23 \times 23$  points in  $\Omega_1$  and  $41 \times 41$  points in  $\Omega_2$ ) for the parabolic-elliptic coupled problem.

Note that although the iteration counts are higher when  $\mathbf{S}_4$  is based on the elliptic block as compared to  $\mathbf{S}_4$  based on the parabolic block, GMRES convergence is mesh-independent in both cases.  $\mathbf{S}_5$ , on the other hand, results in convergence which is independent of the time step size.

The five methods show consistent performance in all tests with uniform and interface refinements of conforming and non-conforming meshes for elliptic-elliptic and elliptic-parabolic coupling.

The iteration counts for approximations  $\mathbf{S}_1$  and  $\mathbf{S}_2$  are never significantly lower than those for approximations  $\mathbf{S}_4$  and  $\mathbf{S}_5$ , and grow with mesh size. Combined with their computational inefficiency (namely, the need to invert a dense matrix at every GMRES iteration) this suggests no advantage to their further use.

Approximation  $\mathbf{S}_3$ , while possessing a very low computational overhead, shows the highest iteration counts by a wide margin. This suggests that the inherent dependence of the interface and domain degrees of freedom should not be neglected in the approximation. Moreover, the iteration counts grow along with the number of degrees of freedom on the interface, meaning that the solver using  $\mathbf{S}_3$  is not scalable.

For these three methods, as well as  $\mathbf{S}_5$ , iteration counts grow along with the number of degrees of freedom on the interface, not the total number of degrees of freedom. This can be seen by comparing Tables 4.1 and 4.2.

Finally, while iteration counts for  $\mathbf{S}_5$  grow with mesh size, the use of  $\mathbf{S}_4$  results in mesh-independent convergence of GMRES. The iteration counts indicate that the method's convergence properties actually improve as the mesh size decreases. In the problems considered here  $\mathbf{S}_4$  proves to be a good approximation, is relatively cheap to compute, and is scalable to large problems. However, the choice of the

problem component used in the construction of  $\mathbf{S}_4$  is somewhat limited by mesh-related considerations. The physics producing a ‘richer’ space trace on the interface has to be used in the current setup.

We note that in the case of approximation  $\mathbf{S}_4$  the iteration counts are higher for nonconforming meshes. This is as expected, since the two components of the Schur complement are no longer the same, and hence the approximation of an inverse of a sum as an inverse of the finer component deteriorates. It is important, however, that despite the approximation, the iteration counts are still bounded.

## 4.4 Practical solvers

Based on the assessment of computational costs of the methods and the numerical tests in Section 4.3, approximating the inverse of the Schur complement using  $\mathbf{S}_4$  and  $\mathbf{S}_5$  provide the best results of those considered. These choices are computationally more efficient than the rest. Moreover, the use of  $\mathbf{S}_4$  results in mesh-independent convergence of GMRES.

The next step in establishing the practical value of these preconditioners is to replace the direct solvers for blocks with scalable counterparts. If we consider  $\mathbf{S}_4$  computing  $\mathbf{A}_{DD}^{-1}$  exactly for a large  $\mathbf{A}$  is not practical, since the computation time for direct solvers grows at least quadratically with the size of the matrix, and furthermore, requires memory to store dense triangular matrices obtained in the course of the factorisation. It is therefore necessary to use an approximation to the action of  $\mathbf{A}_{DD}^{-1}$  on a vector of the appropriate size.  $\mathbf{S}_5$  is reliant on the computation of the action of  $(\mathbf{C}_F^T \mathbf{C}_F)^{-1}$ . This matrix normally has simpler structure than  $\mathbf{A}_{DD}$ , and may therefore be easier to solve iteratively.

The best choice of such iterative solvers is dependent on the structure of the blocks, and ultimately on the nature of the problem and its discretisation. In the examples considered here we use AMG-preconditioned GMRES to approximate the action of  $\mathbf{A}_1^{-1}$  and  $\mathbf{A}_2^{-1}$  on a vector. Specifically, the use of a fixed number of algebraic multigrid (AMG) [24, 115] cycles with a fixed number of smoothing iterations allows us to control the number of floating point operations involved in each GMRES iteration. In the implementation of  $\mathbf{S}_4$  we also use GMRES preconditioned with AMG to approximate the action of  $\mathbf{A}_{DD}^{-1}$ . Overall, we expect this solver to retain mesh-independent convergence. For  $\mathbf{S}_5$ , where the only additional approximation is required for the action of a symmetric positive definite matrix  $(\mathbf{C}_F^T \mathbf{C}_F)^{-1}$  on a vector, we can use a simpler approximation, e.g. a fixed number of iterations of conjugate gradients [67, 72].

A representative convergence test was performed for both methods: we solved the elliptic-parabolic coupled problem with uniform and interface refinements of non-conforming meshes. For  $\mathbf{S}_4$  the action of  $\mathbf{A}_{DD}^{-1}$  in the solver was approximated with two AMG  $V$ -cycles using 2 Jacobi iterations for smoothing. For  $\mathbf{S}_5$  the action of  $(\mathbf{C}_F^T \mathbf{C}_F)^{-1}$  was approximated using 5 conjugate gradients iterations. Increasing the number of  $V$ -cycles, smoother iterations or conjugate gradients iterations results in a lower overall number of GMRES iterations for the full system, but comes at the cost of higher per-iteration times. The values chosen here resulted in sufficiently fast convergence.

As evident from Table 4.6, the practical methods exhibit similar behaviour to their exact counterparts considered in the previous section. For a fixed time step size and as the mesh size decays iteration counts remain low and bounded for  $\mathbf{S}_4$ , and low, but growing, for  $\mathbf{S}_5$ . Figures 4.2 and 4.3 further illustrate the methods' convergence:

all exhibit an initial sharp drop in the residual corresponding to identification of unit eigenvalues of  $\mathbf{A}\mathbf{P}^{-1}$ , and then a steadier decay to the prescribed tolerance, corresponding to identification of the eigenvalues of  $\mathbf{S}\hat{\mathbf{S}}^{-1}$ .

|              | $\mathbf{S}_4$ |       |           |      | $\mathbf{S}_5$ |       |           |      |
|--------------|----------------|-------|-----------|------|----------------|-------|-----------|------|
|              | Uniform        |       | Interface |      | Uniform        |       | Interface |      |
| $DOF_\Gamma$ | Iter           | T, s  | Iter      | T, s | Iter           | T, s  | Iter      | T, s |
| 41           | 27             | 1.14  | 27        | 1.12 | 20             | 0.85  | 20        | 0.86 |
| 81           | 28             | 3.48  | 28        | 1.20 | 24             | 2.31  | 24        | 1.04 |
| 161          | 26             | 9.59  | 27        | 1.35 | 27             | 7.13  | 27        | 1.20 |
| 321          | 27             | 39.07 | 26        | 1.34 | 34             | 32.47 | 30        | 1.47 |

Table 4.6: GMRES iteration counts and CPU times for the practical method using  $\mathbf{S}_4$  with AMG. The parabolic-elliptic coupled problem is solved on refinements of the nonconforming mesh ( $41 \times 41$  points in  $\Omega_1$  and  $23 \times 23$  points in  $\Omega_2$ ) with GMRES  $TOL = 10^{-10}$ ,  $\Delta t = 0.1$ .

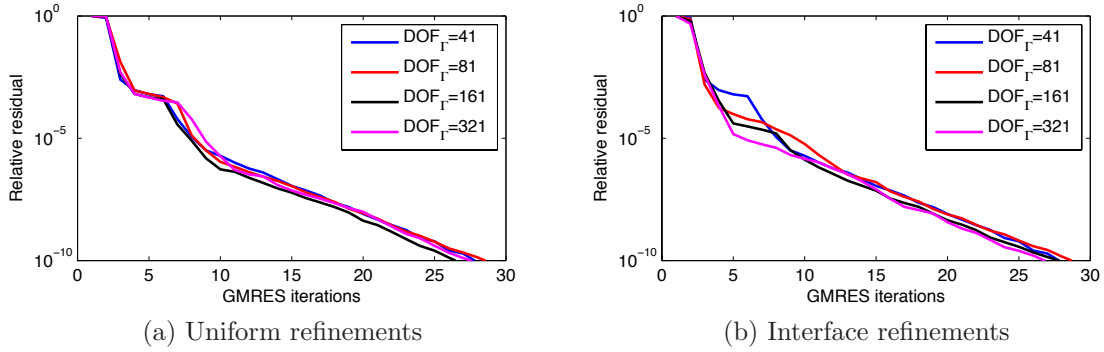


Figure 4.2: GMRES relative residual convergence for the practical method using  $\mathbf{S}_4$  with AMG. The parabolic-elliptic coupled problem is solved on refinements of the nonconforming mesh ( $41 \times 41$  points in  $\Omega_1$  and  $23 \times 23$  points in  $\Omega_2$ ) with GMRES  $TOL = 10^{-10}$ ,  $\Delta t = 0.1$ .

In addition we show CPU times taken by the solvers to run on a 1.8 GHz Intel Core i7 processor (excluding setup of AMG matrices, which is performed using the IFISS library\* developed at the University of Manchester [46]). The results are shown for a sequential Matlab implementation, however it should be noted that parallelisation of a block preconditioner can be based directly on parallelisation of block solvers, and

\*<http://www.cs.umd.edu/~elman/ifiss.html>

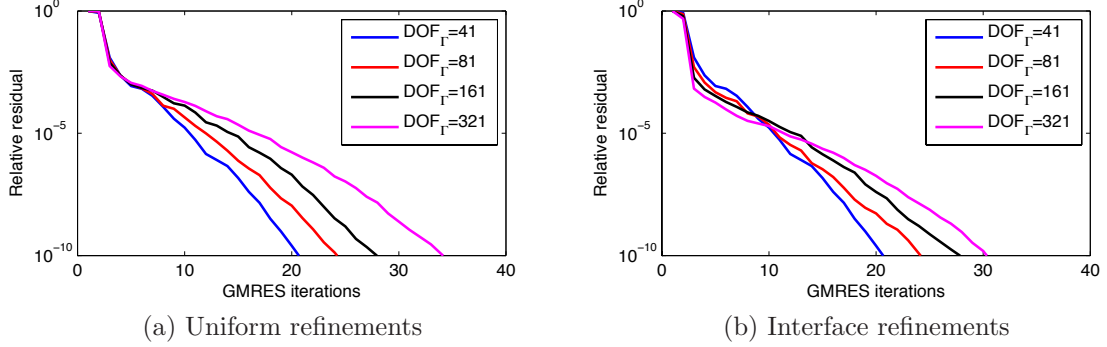


Figure 4.3: GMRES relative residual convergence for the practical method using  $\mathbf{S}_5$  with AMG. The parabolic-elliptic coupled problem is solved on refinements of the nonconforming mesh ( $41 \times 41$  points in  $\Omega_1$  and  $23 \times 23$  points in  $\Omega_2$ ) with GMRES  $TOL = 10^{-10}$ ,  $\Delta t = 0.1$ .

therefore incorporates another advantage of segregated solvers over unpreconditioned monolithic solvers. As seen in Table 4.6, solve times scale linearly with the total number of degrees of freedom in the domain.

## 4.5 Practical solver for fluid-structure interaction

The preconditioners developed above performed well for coupled elliptic-elliptic, and parabolic-elliptic problems. We made sure there was a qualitative difference between the  $\mathbf{A}_1$  and  $\mathbf{A}_2$  blocks of the system matrix by considering different diffusion coefficients  $k_1$  and  $k_2$ , by mixing time-dependent and quasistatic behaviour, and by considering nonconforming meshes in  $\Omega_1$  and  $\Omega_2$ . A more fundamental difference arises in a multiphysics setting, such as fluid-structure interaction (FSI), where the underlying governing equations for the subproblems have dissimilar nature. FSI problems are further complicated by the fact that the fluid domain changes over time in response to stresses from the deformed solid. While it is common to solve structural equations in the Lagrangian framework, where all equations are written on the reference domain, fluid equations require special treatment, e.g. the use of arbitrary



Lagrangian-Eulerian (ALE) coordinates [41, 92, 93, 95].

Before we can present the results for the block preconditioner in this setting, we describe the specific FSI formulation that was used, the weak form of the equations, and the finite element method employed for discretisation.

#### 4.5.1 Coupled fluid dynamics and solid mechanics equations

Let  $\Omega_f(0)$  be the reference fluid domain, transforming over time under stress from the solid into  $\Omega_f(t)$ . We denote the coordinate on the reference domain by  $\mathbf{X}$ , and the coordinate on the deformed domain by  $\mathbf{x}(\mathbf{X}, t)$  (see Figure 4.4a). We write the

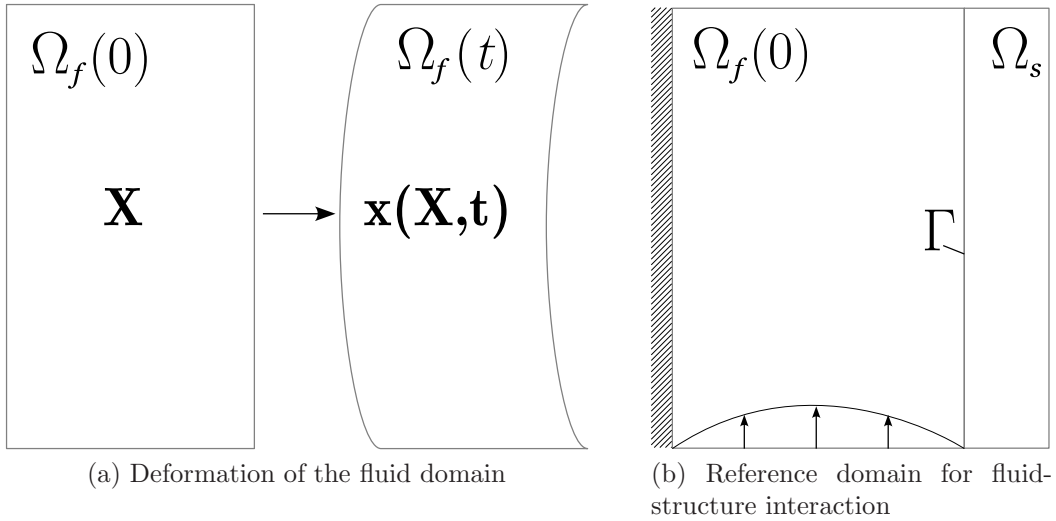


Figure 4.4: Schematic representation of the geometry in fluid-structure interaction.  $\Omega_f$  is the fluid domain,  $\Omega_s$  the solid domain, and  $\Gamma$  the interface.

incompressible Navier-Stokes equations [13] in the deformed coordinates:

$$\rho \left( \frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} - \mathbf{w}) \cdot \nabla \mathbf{v} \right) = \nabla \cdot (\mu \nabla \mathbf{v} - p \mathbf{I}), \quad (\mathbf{x}, t) \in \Omega_f(t) \times (0, T],$$

$$\nabla \cdot \mathbf{v} = 0, \quad (\mathbf{x}, t) \in \Omega_f(t) \times (0, T].$$

Here  $\mathbf{v} : \Omega_f(t) \times [0, T] \rightarrow \mathbb{R}^2$  is the fluid velocity,  $\mathbf{w} = \frac{\partial \mathbf{x}}{\partial t}$  is the domain velocity,  $p : \Omega_f(t) \times [0, T] \rightarrow \mathbb{R}$  is the pressure, and  $\rho, \mu \in \mathbb{R}^+$  are respectively density and viscosity, assumed constant.

The solid problem is formulated on the reference domain  $\Omega_s$ , i.e. in the Lagrangian coordinates. We consider the simplest linear elastic constitutive law for the stress-strain relationship. Physically this law is only applicable to small deformations [75]. Nevertheless, it serves the purpose in providing a governing equation that is qualitatively different to Navier-Stokes. Furthermore, we do not impose incompressibility in the solid domain, resulting in additional divergence between the problem components: the fluid block of the matrix will have saddle-point structure, while the solid block will not.

We write the strong form of the governing equation following [58], as

$$\nabla_S \cdot \boldsymbol{\sigma} = 0, \quad (\mathbf{X}, t) \in \Omega_s \times (0, T],$$

where  $\boldsymbol{\sigma} = \mathbf{D} \nabla_S$  is the linear Cauchy stress tensor written as  $\boldsymbol{\sigma} = (\sigma_{11}, \sigma_{22}, \sigma_{12})^T$ ,  $\mathbf{u} : \Omega_s \times [0, T] \rightarrow \mathbb{R}^2$  is the solid deformation,  $\nabla_S$  the symmetric gradient operator, and  $\mathbf{D}$  the matrix in the linear stress-strain relationship:

$$\nabla_S = \begin{pmatrix} \frac{\partial}{\partial X_1} & 0 \\ 0 & \frac{\partial}{\partial X_2} \\ \frac{\partial}{\partial X_2} & \frac{\partial}{\partial X_1} \end{pmatrix}, \quad \mathbf{D} = \frac{E}{(1+\nu)(1-2\nu)} \begin{pmatrix} 1-\nu & \nu & 0 \\ \nu & 1-\nu & 0 \\ 0 & 0 & (1-2\nu)/2 \end{pmatrix},$$

$E$  being the Young's modulus and  $\nu$  the Poisson ratio. As  $\nu$  approaches the value of 0.5, the material approaches the limiting incompressible case. In certain cases we

need to write the stress as a  $2 \times 2$  matrix, which is denoted by

$$\boldsymbol{\tau} = \begin{pmatrix} \sigma_{11} & \frac{1}{2}\sigma_{12} \\ \frac{1}{2}\sigma_{12} & \sigma_{22} \end{pmatrix}.$$

Let the interface between the reference fluid and solid domains be denoted by  $\Gamma(0) = \partial\Omega_f(0) \cap \partial\Omega_s$ , and let the corresponding boundary of the physical fluid domain at time  $t$  be denoted by  $\Gamma(t) = \mathbf{x}(\Gamma(0), t)$ . The schematic representation of the problem geometry at time  $t = 0$  is shown in Figure 4.4b.

The coupling between the fluid and the solid problems is based on continuity of velocities and stresses on the interface  $\Gamma(0)$  at all times  $t \in (0, T]$ :

$$\begin{aligned} \hat{\mathbf{v}} &= \frac{\partial \mathbf{u}}{\partial t}, \\ (\mu \nabla_{\mathbf{x}} \hat{\mathbf{v}} - \hat{p} \mathbf{I}) \mathbf{n} &= \boldsymbol{\tau} \mathbf{n}. \end{aligned}$$

Here  $\hat{\mathbf{v}}(\mathbf{X}, t) = \mathbf{v}(\mathbf{x}(\mathbf{X}, t), t)$  and  $\hat{p}(\mathbf{X}, t) = p(\mathbf{x}(\mathbf{X}, t), t)$ .

#### 4.5.1.1 Strong form

Augmenting the system with appropriate initial and boundary conditions, we obtain the following strong form of the coupled FSI problem:

$$\rho \left( \frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} - \mathbf{w}) \cdot \nabla \mathbf{v} \right) = \nabla \cdot (\mu \nabla \mathbf{v} - p \mathbf{I}), \quad (\mathbf{x}, t) \in \Omega_f(t) \times (0, T], \quad (4.16a)$$

$$\nabla \cdot \mathbf{v} = 0, \quad (\mathbf{x}, t) \in \Omega_f(t) \times (0, T], \quad (4.16b)$$

$$\mathbf{v} = \mathbf{v}_D, \quad (\mathbf{x}, t) \in \Gamma_{f,D}(t) \times (0, T] \quad (4.16c)$$

$$(\mu \nabla \mathbf{v} - p \mathbf{I}) \mathbf{n} = 0, \quad (\mathbf{x}, t) \in \Gamma_{f,N}(t) \times (0, T], \quad (4.16d)$$

$$\hat{\mathbf{v}}(\mathbf{X}, 0) = \mathbf{v}^0(\mathbf{X}), \quad \mathbf{X} \in \Omega_f(0), \quad (4.16e)$$

$$\nabla_S \cdot \boldsymbol{\sigma} = 0, \quad (\mathbf{X}, t) \in \Omega_s \times (0, T], \quad (4.16f)$$

$$\mathbf{u} = \mathbf{u}_D, \quad (\mathbf{X}, t) \in \Gamma_{s,D} \times (0, T], \quad (4.16g)$$

$$\boldsymbol{\tau} \mathbf{n} = 0, \quad (\mathbf{X}, t) \in \Gamma_{s,N} \times (0, T], \quad (4.16h)$$

$$\hat{\mathbf{v}} = \frac{\partial \mathbf{u}}{\partial t}, \quad (\mathbf{X}, t) \in \Gamma(0) \times (0, T], \quad (4.16i)$$

$$(\mu \nabla_{\mathbf{x}} \hat{\mathbf{v}} - \hat{p} \mathbf{I}) \mathbf{n} = \boldsymbol{\tau} \mathbf{n}, \quad (\mathbf{X}, t) \in \Gamma(0) \times (0, T]. \quad (4.16j)$$

Each of the boundaries of the physical fluid domain at time  $t$ , and the reference solid domain are composed of non-intersecting Dirichlet, Neumann and interface parts:  $\partial\Omega_f(t) = \Gamma_{f,D}(t) \cup \Gamma_{f,N}(t) \cup \Gamma(t)$  and  $\partial\Omega_s = \Gamma_{s,D} \cup \Gamma_{s,N} \cup \Gamma(0)$ .

#### 4.5.1.2 Weak form

Let  $\Omega_f(t)$ ,  $\Omega_s \subset \mathbb{R}^2$ . We introduce the following function spaces:

$$\begin{aligned} X_f^t &= [L^2(0, T; H^1(\Omega_f(t)))]^2, \\ X_{f,D}^t &= \{\boldsymbol{\varphi} \in X_f^t \mid \boldsymbol{\varphi} = \mathbf{v}_D \text{ on } \Gamma_{f,D}(t) \times (0, T]\}, \\ X_{f,0}^t &= \{\boldsymbol{\varphi} \in X_f^t \mid \boldsymbol{\varphi} = \mathbf{0} \text{ on } \Gamma_{f,D}(t) \times (0, T]\}, \\ Y_f^t &= L^2(0, T; L^2(\Omega_f(t))), \\ X_s &= [L^2(0, T; H^1(\Omega_s))]^2, \\ X_{s,D} &= \{\boldsymbol{\psi} \in X_s \mid \boldsymbol{\psi} = \mathbf{u}_D \text{ on } \Gamma_{s,D} \times (0, T]\}, \\ X_{s,0} &= \{\boldsymbol{\psi} \in X_s \mid \boldsymbol{\psi} = \mathbf{0} \text{ on } \Gamma_{s,D} \times (0, T]\}, \\ Y &= [L^2(0, T; L^2(\Gamma(0)))]^2. \end{aligned}$$

For any  $t \in (0, T]$  we can multiply the momentum equation (4.16a) by an arbitrary

function  $\boldsymbol{\varphi}$  from  $X_{f,0}^t$  and integrate the product over  $\Omega_f(t)$ , leading to

$$\int_{\Omega_f(t)} \left( \rho \frac{\partial \mathbf{v}}{\partial t} \cdot \boldsymbol{\varphi} + \rho ((\mathbf{v} - \mathbf{w}) \cdot \nabla \mathbf{v}) \cdot \boldsymbol{\varphi} - (\nabla \cdot (\mu \nabla \mathbf{v} - p \mathbf{I})) \cdot \boldsymbol{\varphi} \right) d\mathbf{x} = 0. \quad (4.17)$$

In order to transform the integral of the time derivative, we introduce the Jacobian matrix of the domain transform:

$$\mathbf{J}(\mathbf{X}, t) = \frac{\partial \mathbf{x}(\mathbf{X}, t)}{\partial \mathbf{X}},$$

and let  $J = \det(\mathbf{J})$ . Direct evaluation shows (see e.g. [118]) that

$$\frac{\partial J}{\partial t} = J \nabla \cdot \mathbf{w},$$

hence we can rewrite the time derivative term as follows:

$$\begin{aligned} \int_{\Omega_f(t)} \rho \frac{\partial \mathbf{v}(\mathbf{x}, t)}{\partial t} \cdot \boldsymbol{\varphi}(\mathbf{x}, t) d\mathbf{x} &= \int_{\Omega_f(0)} \rho \frac{\partial \mathbf{v}(\mathbf{x}(\mathbf{X}, t), t)}{\partial t} \cdot \boldsymbol{\varphi}(\mathbf{x}(\mathbf{X}, t), t) J d\mathbf{X} \\ &= \int_{\Omega_f(0)} \rho \frac{\partial \hat{\mathbf{v}}(\mathbf{X}, t)}{\partial t} \cdot \hat{\boldsymbol{\varphi}}(\mathbf{X}, t) J d\mathbf{X} = \int_{\Omega_f(0)} \rho \frac{\partial}{\partial t} (\hat{\mathbf{v}} \cdot \hat{\boldsymbol{\varphi}} J) d\mathbf{X} - \int_{\Omega_f(0)} \rho \hat{\mathbf{v}} \cdot \hat{\boldsymbol{\varphi}} \frac{\partial J}{\partial t} d\mathbf{X} \\ &= \frac{d}{dt} \int_{\Omega_f(0)} \rho \hat{\mathbf{v}} \cdot \hat{\boldsymbol{\varphi}} J d\mathbf{X} - \int_{\Omega_f(0)} \rho \hat{\mathbf{v}} \cdot \hat{\boldsymbol{\varphi}} (\nabla_{\mathbf{x}(\mathbf{X}, t)} \cdot \hat{\mathbf{w}}(\mathbf{X}, t)) J d\mathbf{X} \\ &= \frac{d}{dt} \int_{\Omega_f(t)} \rho \mathbf{v} \cdot \boldsymbol{\varphi} d\mathbf{x} - \int_{\Omega_f(t)} \rho \mathbf{v} \cdot \boldsymbol{\varphi} (\nabla \cdot \mathbf{w}) d\mathbf{x}. \end{aligned}$$

Returning to (4.17) and using the divergence theorem, we get

$$\begin{aligned} \frac{d}{dt} \int_{\Omega_f(t)} \rho \mathbf{v} \cdot \boldsymbol{\varphi} d\mathbf{x} &+ \int_{\Omega_f(t)} \rho (\mathbf{v} \cdot \nabla \mathbf{v}) \cdot \boldsymbol{\varphi} d\mathbf{x} - \int_{\Omega_f(t)} \rho \nabla (\mathbf{v} \cdot \mathbf{w}) \cdot \boldsymbol{\varphi} d\mathbf{x} \\ &+ \int_{\Omega_f(t)} (\mu \nabla \mathbf{v} : \nabla \boldsymbol{\varphi} - p \nabla \cdot \boldsymbol{\varphi}) d\mathbf{x} - \int_{\Gamma(t)} (\mu \nabla \mathbf{v} \cdot \mathbf{n} - p \mathbf{n}) \cdot \boldsymbol{\varphi} ds = 0 \quad \forall \boldsymbol{\varphi} \in X_{f,0}^t. \end{aligned}$$

The only remaining boundary term is on  $\Gamma(t)$ , since the terms on  $\Gamma_{f,D}$  vanish by the choice of  $\boldsymbol{\varphi}$ , and the terms on  $\Gamma_{f,N}$  by the corresponding Neumann condition.

The weak form of the incompressibility equation is:

$$\int_{\Omega_f(t)} (\nabla \cdot \mathbf{v}) q \, d\mathbf{x} = 0, \quad \forall q \in Y_f^t.$$

Next we write the weak form of the solid equation:

$$-\int_{\Omega_s} \nabla_S \cdot (\mathbf{D} \nabla_S \mathbf{u}) \cdot \boldsymbol{\psi} \, d\mathbf{x} = \int_{\Omega_s} (\mathbf{D} \nabla_S \mathbf{u}) \cdot \nabla_S \boldsymbol{\psi} \, d\mathbf{x} - \int_{\Gamma(0)} \boldsymbol{\tau} \mathbf{n} \cdot \boldsymbol{\psi} \, ds = 0, \quad \forall \boldsymbol{\psi} \in X_{s,0},$$

As before, boundary terms on  $\Gamma_{s,D}$  vanish due to the choice of  $\boldsymbol{\psi}$ , and on  $\Gamma_{s,N}$  due to the Neumann boundary condition. We introduce the Lagrange multiplier  $\boldsymbol{\lambda}$ , such that

$$\boldsymbol{\lambda} = \mu \nabla_{\mathbf{x}} \hat{\mathbf{v}} \cdot \mathbf{n} - \hat{p} \mathbf{n} = \boldsymbol{\tau} \mathbf{n} \quad \text{on } \Gamma(0).$$

This multiplier can be substituted into interface terms in both the main fluid and solid equations. The term in the fluid equation needs to be fully transformed to the reference coordinates:

$$\boldsymbol{\lambda} = \mu \mathbf{J}^{-1} \nabla_{\mathbf{x}} \hat{\mathbf{v}} \cdot \mathbf{n} - \hat{p} \mathbf{n} \quad \text{on } \Gamma(0).$$

Finally, we write down the weak form of the interface condition:

$$\int_{\Gamma(0)} \left( \hat{\mathbf{v}} - \frac{\partial \mathbf{u}}{\partial t} \right) \cdot \boldsymbol{\nu} \, ds = 0, \quad \forall \boldsymbol{\nu} \in Y.$$

The full weak system can now be written as:

Find  $(\mathbf{v}, p, \mathbf{u}, \boldsymbol{\lambda}) \in X_{f,D}^t \times Y_f^t \times X_{s,D} \times Y$  such that

$$\begin{aligned} & \frac{d}{dt} \int_{\Omega_f(t)} \rho \mathbf{v} \cdot \boldsymbol{\varphi} \, d\mathbf{x} + \int_{\Omega_f(t)} \rho (\mathbf{v} \cdot \nabla \mathbf{v}) \cdot \boldsymbol{\varphi} \, d\mathbf{x} - \int_{\Omega_f(t)} \rho \nabla (\mathbf{v} \cdot \mathbf{w}) \cdot \boldsymbol{\varphi} \, d\mathbf{x} \\ & + \int_{\Omega_f(t)} (\mu \nabla \mathbf{v} : \nabla \boldsymbol{\varphi} - p \nabla \cdot \boldsymbol{\varphi}) \, d\mathbf{x} - \int_{\Gamma(0)} \boldsymbol{\lambda} \cdot \hat{\boldsymbol{\varphi}} \, ds = 0 \quad \forall \boldsymbol{\varphi} \in X_{f,0}^t, \end{aligned} \quad (4.18a)$$

$$\int_{\Omega_f(t)} (\nabla \cdot \mathbf{v}) q \, d\mathbf{x} = 0, \quad \forall q \in Y_f^t, \quad (4.18b)$$

$$\int_{\Omega_s} (\mathbf{D} \nabla_S \mathbf{u}) \cdot \nabla_S \boldsymbol{\psi} \, d\mathbf{x} - \int_{\Gamma(0)} \boldsymbol{\lambda} \cdot \boldsymbol{\psi} \, ds = 0, \quad \forall \boldsymbol{\psi} \in X_{s,0}, \quad (4.18c)$$

$$\int_{\Gamma(0)} \left( \hat{\mathbf{v}} - \frac{\partial \mathbf{u}}{\partial t} \right) \cdot \boldsymbol{\nu} \, ds = 0, \quad \forall \boldsymbol{\nu} \in Y. \quad (4.18d)$$

#### 4.5.1.3 Time discretisation and finite element method

First we subdivide the time interval: let  $0 = t_0 < t_1 < \dots < t_N = T$  with  $\Delta t_n = t_n - t_{n-1}$ ,  $n = 1, 2, \dots, N_t$ . We denote  $\Omega_f(t_n) = \Omega_f^n$ ,  $\Gamma^n = \Gamma(t_n)$  and

$$\begin{aligned} \text{RHS}^n &= \int_{\Omega_f^n} \rho (\mathbf{v} \cdot \nabla \mathbf{v}) \boldsymbol{\varphi} \, d\mathbf{x} - \int_{\Omega_f^n} \rho \nabla (\mathbf{v} \cdot \mathbf{w}) \boldsymbol{\varphi} \, d\mathbf{x} + \int_{\Omega_f^n} (\mu \nabla \mathbf{v} : \nabla \boldsymbol{\varphi} - p \nabla \cdot \boldsymbol{\varphi}) \, d\mathbf{x} \\ &- \int_{\Gamma^0} \boldsymbol{\lambda}^n \cdot \hat{\boldsymbol{\varphi}} \, ds. \end{aligned}$$

We use the trapezium rule to integrate equation (4.18a) in time on each of the sub-intervals  $(t_{n-1}, t_n)$ ,  $n = 1, 2, \dots, N_t$ :

$$\int_{\Omega_f^n} \rho \mathbf{v} \cdot \boldsymbol{\varphi} \, d\mathbf{x} - \int_{\Omega_f^{n-1}} \rho \mathbf{v} \cdot \boldsymbol{\varphi} \, d\mathbf{x} + \frac{\Delta t}{2} (\text{RHS}^n + \text{RHS}^{n-1}) = 0, \quad \forall \boldsymbol{\varphi} \in \mathbf{X}_{f,0}^t. \quad (4.19)$$

Similarly, for (4.18d)

$$\frac{\Delta t}{2} \int_{\Gamma^0} (\mathbf{v}^n + \mathbf{v}^{n-1}) \cdot \boldsymbol{\nu} \, ds - \int_{\Gamma^0} (\mathbf{u}^n - \mathbf{u}^{n-1}) \cdot \boldsymbol{\nu} \, ds = 0, \quad \forall \boldsymbol{\nu} \in Y. \quad (4.20)$$

Equations (4.18b) and (4.18c) do not contain time derivatives, so we simply require for them to hold at each time step  $t_n$ . The initial condition is applied at  $n = 0$ .

To discretise the problem in space we assume  $\Omega_f^n$ ,  $n = 1, 2, \dots, N$  and  $\Omega_s$  can be triangulated, with meshes conforming on  $\Gamma^n$  for  $n = 1, \dots, N$ . We define the following finite-dimensional function spaces:

$$\begin{aligned} X_f^{h,n} &= [P^2(\Omega_f^n)]^2, \\ X_{f,D}^{h,n} &= X_{f,D}^{t_n} \cap X_f^{h,n}, \\ X_{f,0}^{h,n} &= X_{f,0}^{t_n} \cap X_f^{h,n}, \\ Y_f^{h,n} &= P^1(\Omega_f^n), \\ X_s^h &= [P^2(\Omega_s)]^2, \\ X_{s,D}^h &= X_{s,D} \cap X_s^h, \\ X_{s,0}^h &= X_{s,0} \cap X_s^h, \\ Y^h &= [P^1(\Gamma)]^2. \end{aligned}$$

Let also the bases of these spaces be as follows:

$$\boldsymbol{\varphi}^n = \begin{pmatrix} \varphi_1^n \\ \varphi_2^n \\ \vdots \\ \varphi_{N_f}^n \end{pmatrix}, \quad \boldsymbol{\psi} = \begin{pmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_{N_s} \end{pmatrix}, \quad \mathbf{q}^n = \begin{pmatrix} q_1^n \\ q_2^n \\ \vdots \\ q_{M_f}^n \end{pmatrix}, \quad \boldsymbol{\nu} = \begin{pmatrix} \nu_1 \\ \nu_2 \\ \vdots \\ \nu_M \end{pmatrix},$$



so that  $X_f^{h,n} = \text{span}\{\boldsymbol{\varphi}_1^n, \dots, \boldsymbol{\varphi}_{N_f}^n\}$ ,  $X_s^h = \text{span}\{\boldsymbol{\psi}_1, \dots, \boldsymbol{\psi}_{N_s}\}$ ,  $Y_f^{h,n} = \text{span}\{q_1^n, \dots, q_{M_f}^n\}$ , and  $Y^h = \text{span}\{\boldsymbol{\nu}_1, \dots, \boldsymbol{\nu}_M\}$ . In this notation  $\boldsymbol{\varphi}_i^n = (\varphi_{1,i}^n, \varphi_{2,i}^n) \in \mathbb{R}^2$  for  $i = 1, \dots, N_f$ ,  $\boldsymbol{\psi}_k = (\psi_{1,k}, \psi_{2,k}) \in \mathbb{R}^2$  for  $k = 1, \dots, N_s$ , and  $\boldsymbol{\nu}_r = (\nu_{1,r}, \nu_{2,r}) \in \mathbb{R}^2$  for  $r = 1, \dots, M$ .

We can write the finite element method by approximating the test and the trial spaces in (4.18) with their finite-dimensional counterparts. We express the approximate solutions in terms of the respective bases, specifically

$$v_d^{h,n} = \mathbf{V}_d^n \cdot \boldsymbol{\varphi}_d^n, \quad d = 1, 2$$

$$p^{h,n} = \mathbf{P}^n \cdot \mathbf{q}^n,$$

$$u_d^{h,n} = \mathbf{U}_d^n \cdot \boldsymbol{\psi}_d, \quad d = 1, 2$$

$$\lambda_d^{h,n} = \boldsymbol{\Lambda}_d^n \cdot \boldsymbol{\nu}_d, \quad d = 1, 2$$

$$w_d^{h,n} = \mathbf{W}_d^n \cdot \boldsymbol{\varphi}_d^n, \quad d = 1, 2.$$

The coefficient vectors are  $\mathbf{V}_d^n, \mathbf{W}_d^n \in \mathbb{R}^{N_f}$ ,  $\mathbf{P}^n \in \mathbb{R}^{M_f}$ ,  $\mathbf{U}_d^n \in \mathbb{R}^{N_s}$  and  $\boldsymbol{\Lambda}_d^n \in \mathbb{R}^M$ .

We introduce the following matrices:

$$\mathbf{M}_f^n \in \mathbb{R}^{N_f \times N_f} : \quad (m_f)_{i,j}^n = \int_{\Omega_f^n} \rho \boldsymbol{\varphi}_i^n \cdot \boldsymbol{\varphi}_j^n \, d\mathbf{x},$$

$$\mathbf{K}_f^n \in \mathbb{R}^{N_f \times N_f} : \quad (k_f)_{i,j}^n = \int_{\Omega_f^n} \mu \nabla \boldsymbol{\varphi}_i^n : \nabla \boldsymbol{\varphi}_j^n \, d\mathbf{x},$$

$$\mathbf{K}_s \in \mathbb{R}^{N_s \times N_s} : \quad (k_s)_{k,l}^n = \int_{\Omega_s^n} (\mathbf{D} \nabla_S \boldsymbol{\psi}_k) : \nabla_S \boldsymbol{\psi}_l \, d\mathbf{x},$$

$$\mathbf{C}\mathbf{V}^n(\mathbf{V}^m) \in \mathbb{R}^{N_f \times N_f} : (cv(\mathbf{V}^m))_{i,j}^n = \sum_{d=1,2} \sum_h V_h^m \int_{\Omega_f^n} \rho \varphi_{d,i}^n \frac{\partial \varphi_{d,j}^n}{\partial x_d} \varphi_{d,h}^m \, d\mathbf{x},$$

$$\mathbf{D}^n(\mathbf{W}^m) \in \mathbb{R}^{N_f \times N_f} : (d(\mathbf{W}^m))_{i,j}^n = \sum_{d=1,2} \sum_h W_h^m \int_{\Omega_f^n} \rho \frac{\partial}{\partial x_d} (\varphi_{d,i}^n \varphi_{d,j}^n) \varphi_{d,h}^m \, d\mathbf{x},$$

$$\mathbf{B}^n \in \mathbb{R}^{N_f \times M_f} : \quad (b)_{i,l}^n = \int_{\Omega_f^n} (\nabla \cdot \boldsymbol{\varphi}_i^n) q_l \, d\mathbf{x},$$

$$\mathbf{C}_f^n \in \mathbb{R}^{N_f \times M} : \quad (c_f)_{i,r}^n = \frac{\Delta t}{2} \int_{\Gamma^0} \hat{\boldsymbol{\varphi}}_i^n \nu_r \, ds,$$

$$\mathbf{C}_s \in \mathbb{R}^{N_s \times M} : \quad (c_s)_{k,r}^n = \int_{\Gamma^0} \psi_k \nu_r \, ds.$$

These matrices do not take into account the Dirichlet boundary conditions, which can be applied directly by replacing the appropriate rows of the full matrix and right-hand side vector with the discrete versions of these conditions.

Let us also have

$$\begin{aligned} \mathbf{A}_f^n &= \mathbf{M}_f^n + \frac{\Delta t}{2} [\mathbf{K}_f^n + \mathbf{C}\mathbf{V}^n(\mathbf{U}^n) - \mathbf{D}^n(\mathbf{W}^n)] \\ \mathbf{A}\mathbf{A}^{n-1} &= \mathbf{M}_f^{n-1} - \frac{\Delta t}{2} [\mathbf{K}_f^{n-1} + \mathbf{C}\mathbf{V}^{n-1}(\mathbf{U}^{n-1}) - \mathbf{D}^{n-1}(\mathbf{W}^{n-1})], \end{aligned}$$

and additionally

$$\mathbf{V}^n = \begin{pmatrix} \mathbf{V}_1^n \\ \mathbf{V}_2^n \end{pmatrix}, \quad \mathbf{U}^n = \begin{pmatrix} \mathbf{U}_1^n \\ \mathbf{U}_2^n \end{pmatrix}, \quad \mathbf{\Lambda}^n = \begin{pmatrix} \mathbf{\Lambda}_1^n \\ \mathbf{\Lambda}_2^n \end{pmatrix}.$$

The finite element method can be written in pseudo-matrix form: for  $n = 1, 2, \dots, N_t$  find  $(\mathbf{V}^n, P^n, \mathbf{U}^n, \mathbf{\Lambda}^n)$  such that

$$\begin{pmatrix} \mathbf{A}_f^n & -\mathbf{B}^n & 0 & -\mathbf{C}_f^n \\ -(\mathbf{B}^n)^T & 0 & 0 & 0 \\ 0 & 0 & \mathbf{K}_s & \mathbf{C}_s \\ -(\mathbf{C}_f^n)^T & 0 & \mathbf{C}_s^T & 0 \end{pmatrix} \begin{pmatrix} \mathbf{V}^n \\ P^n \\ \mathbf{U}^n \\ \mathbf{\Lambda}^n \end{pmatrix} = \begin{pmatrix} \mathbf{A}\mathbf{A}_f^{n-1} & -\mathbf{B}^{n-1} & 0 & -\mathbf{C}_f^{n-1} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -(\mathbf{C}_f^{n-1})^T & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{V}^{n-1} \\ P^{n-1} \\ \mathbf{U}^{n-1} \\ \mathbf{\Lambda}^{n-1} \end{pmatrix}$$

What is written as matrices  $\mathbf{A}_f^n$  and  $\mathbf{A}\mathbf{A}_f^{n-1}$  in the above equations is actually dependent on the velocities  $\mathbf{V}^n$  and  $\mathbf{V}^{n-1}$  respectively, as well as the domain velocities  $\mathbf{W}^n$  and  $\mathbf{W}^{n-1}$  respectively. The system needs further linearisation, as described below. At the same time, the matrix representing the solid component of the problem,  $\mathbf{K}_s$ , is

independent of time and current solution, and has relatively simple banded structure under the linear elasticity assumption.

The fluid component has the following two sources of nonlinearity: mesh movement and convection. It is common to assume that mesh movement over a single time step is small, and the problem of computing the ALE coordinates at every time step is therefore normally decoupled from the governing equations. The mesh displacements in  $\Omega_f(t)$  can be obtained by assuming elastic or Laplacian deformation, which ensures smooth node movement [41, 92, 95]. We use the latter technique: at each time step the mesh deformation  $\mathbf{d}$  (relative to the mesh on  $\Omega_f(0)$ ) is such that

$$-\Delta \mathbf{d} = 0 \text{ in } \Omega_f(0), \quad (4.21a)$$

$$\mathbf{d} = \mathbf{u} \text{ on } \Gamma(0), \quad (4.21b)$$

$$\mathbf{d} = 0 \text{ on } \partial\Omega_f(0) \setminus \Gamma(0). \quad (4.21c)$$

A separate finite element procedure is setup to solve this boundary value problem on the same mesh as the fluid problem. If the finite element solution to (4.21) at time step  $n$  is denoted by  $\mathbf{d}^n$ , then the domain velocity to be used in the ALE Navier-Stokes equations is calculated as  $\mathbf{w} = (\mathbf{d}^n - \mathbf{d}^{n-1})/(t_n - t_{n-1})$ . Under the assumption of small mesh deformations, it is sufficient to perform a few fixed point iterations between solving (4.21) and (4.16) to obtain a good approximation of the current position of mesh nodes.

The convective nonlinearity is addressed by Newton iteration in standard form: an approximate solution to  $\mathbf{F}(\mathbf{x}) = 0$  is given by

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \left( \frac{\partial \mathbf{F}}{\partial \mathbf{x}} \right)^{-1} \mathbf{F}(\mathbf{x}^k), \quad k = 0, 1, \dots \quad (4.22)$$

We take the solution at the previous time-step as the initial guess for the iteration, which, given sufficiently small time steps, is sufficiently close to the solution at the current time step for the Newton iteration to converge. The analysis of the properties of the Jacobian matrix for the FSI system was beyond the scope of this work, but its non-singularity and convergence of the method were verified algorithmically.

#### 4.5.1.4 Preconditioner

The linear system arising at each Newton iteration of every time step was solved using right-preconditioned GMRES as described in Section 4.4, i.e. using two AMG  $V$ -cycles to approximate the action of  $\mathbf{A}_{DD}^{-1}$  in the  $\mathbf{S}_4^{-1}$  approximation and five conjugate gradients iterations in the  $\mathbf{S}_5^{-1}$  approximation. The complexity of the fluid problem usually means that the fluid mesh needs to be significantly finer than the solid mesh. The nonlinearity, time dependence and a larger mesh size of the fluid component all suggest that  $\mathbf{S}_4^{-1}$  is based on the fluid block. Moreover, since the coupling block only has non-zero elements in the rows corresponding to velocity degrees of freedom, the ‘interface’ block  $\mathbf{A}_{II}$  in the expression for  $\mathbf{S}_4^{-1}$  is extracted from the matrices  $\mathbf{A}_{i,f}^n$ ,  $i = 1, 2$  only.

#### 4.5.1.5 Simulation results

The problem was solved on the following domains:

$$\Omega_f(0) = \{(X_1, X_2) \in \mathbb{R}^2 \mid X_1 \in (0, 1), X_2 \in (0, 2)\},$$

$$\Omega_s = \{(X_1, X_2) \in \mathbb{R}^2 \mid X_1 \in (1, 1.2), X_2 \in (0, 2)\},$$

$$\Gamma = \{(1, X_2) \mid X_2 \in (0, 2)\}.$$

A unidirectional pulsating parabolic inflow condition was used to drive flow and deformations:

$$\hat{\mathbf{v}}(\mathbf{X}, t) = \begin{pmatrix} 0 \\ 10 \sin^2(2\pi t) X_1 (1 - X_1) \end{pmatrix}, \quad \mathbf{X} \in \Gamma_{in}, \quad t \in [0, T].$$

In the fluid domain we imposed a no-slip boundary condition  $\mathbf{v} = \mathbf{0}$  at  $X_1 = 0$ , and a no-flux boundary condition  $(\mu \nabla \mathbf{v} - p \mathbf{I}) \mathbf{n} = \mathbf{0}$  at  $X_2 = 2$ . In the solid domain the top and bottom boundaries were fixed:  $\mathbf{u} = \mathbf{0}$  at  $X_2 = 0$  and  $X_2 = 2$ , and the outer boundary was allowed to move freely:  $\boldsymbol{\tau} \mathbf{n} = \mathbf{0}$  at  $X_1 = 1.2$ .

The following parameter values were used in all tests:  $\rho = 1$ ,  $E = 2900$ ,  $\nu = 0.4$ .

Computational meshes were obtained by refining the original uniform mesh around the interface. The original grid in  $\Omega_f$  had 21 nodes in each of the  $X_1$  and  $X_2$  directions, and in  $\Omega_s$  6 nodes in the  $X_1$  direction and 21 nodes in the  $X_2$  direction. A coarse analogue of such an original mesh is displayed in Figure 4.5a, and the mesh once refined around the interface in Figure 4.5b. Table 4.7 lists dimensions for the test meshes, namely the total number of degrees of freedom in the problem and the number of degrees of freedom on the interface.

| Mesh         | 1    | 2    | 3     | 4     |
|--------------|------|------|-------|-------|
| $DOF$        | 5604 | 7317 | 10811 | 17803 |
| $DOF_\Gamma$ | 41   | 81   | 161   | 321   |

Table 4.7: Mesh sizes for the FSI problem: total number of degrees of freedom and the number of degrees of freedom on the interface. Meshes are refined around the interface.

inf – sup stable  $P2 - P1$  elements (continuous piecewise-quadratic for velocity, and continuous piecewise-linear for pressure) were used in the fluid domain, while  $P2$  elements were used in the solid domain. Time stepping was performed using the

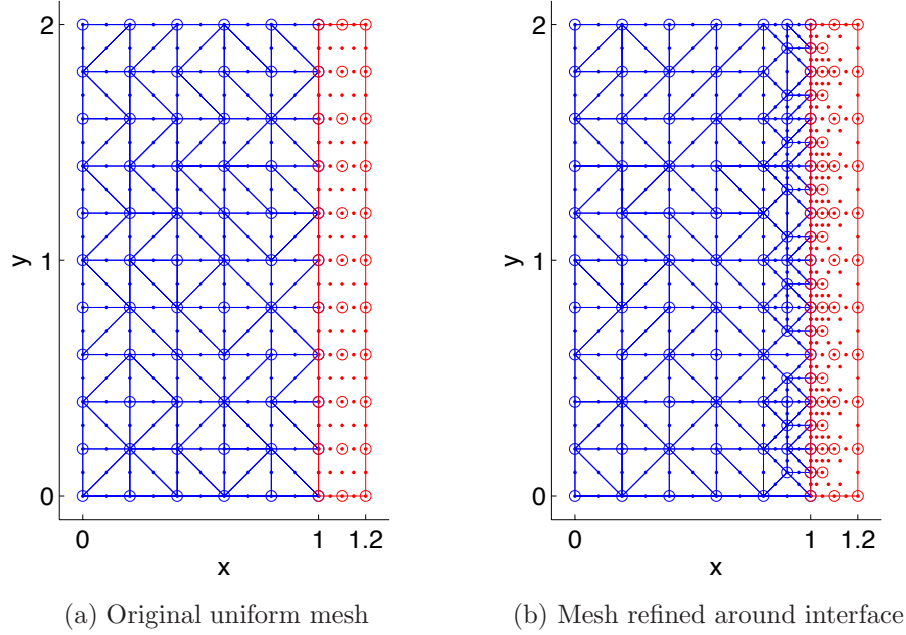


Figure 4.5: Sample meshes for the fluid-structure interaction problem. The triangulation is shown in the fluid domain  $\Omega_f$  to distinguish it from the solid domain  $\Omega_s$ .

implicit Euler scheme.

Sample solution profiles and solid configurations for this problem are shown in Figure 4.6. As the inflow velocity grows, pressure exerted on the elastic wall increases, and the wall gets pushed out. When the pressure drops, the wall gets pulled back in. The effect of lowering the viscosity can be observed in Figure 4.7: the changes in inflow velocity propagate faster, and hence the response of the solid occurs sooner than in the case of higher viscosity.

The maximum iteration counts for GMRES required to converge to relative tolerance  $TOL = 10^{-10}$  on meshes 1-4 are given in Table 4.8. We also show the maximum number of iterations required for convergence of the Newton iteration, and the maximum time in seconds that GMRES took to converge (per nonlinear iteration per time step).

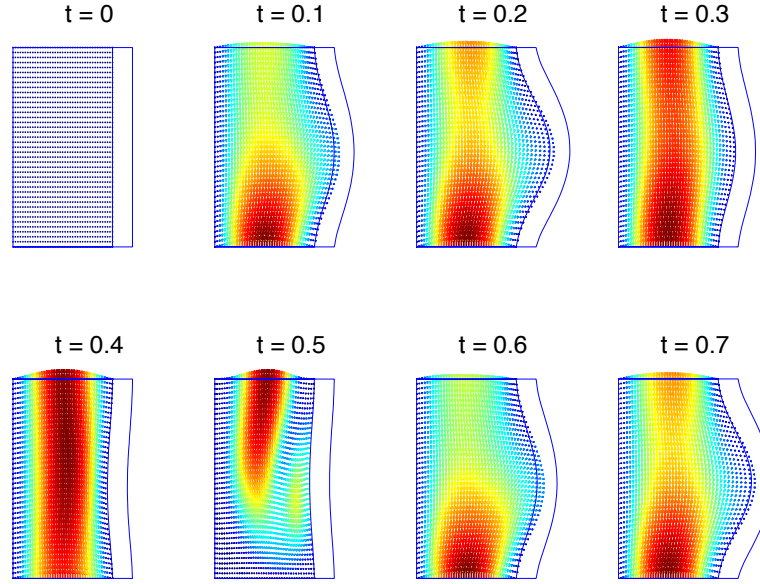


Figure 4.6: Velocity profile and domain deformations for fluid-structure interaction in 2D. Inflow velocity at the bottom of  $\Omega_f$  is  $v_2 = 10X_1(1 - X_1)\sin^2(2\pi t)$ . Parameters:  $\mu = 1$ ,  $\rho = 1$ ,  $\nu = 0.4$ ,  $E = 2900$ . Colour corresponds to velocity magnitude, and is scaled from blue (lowest) to red (highest) for each time step individually.

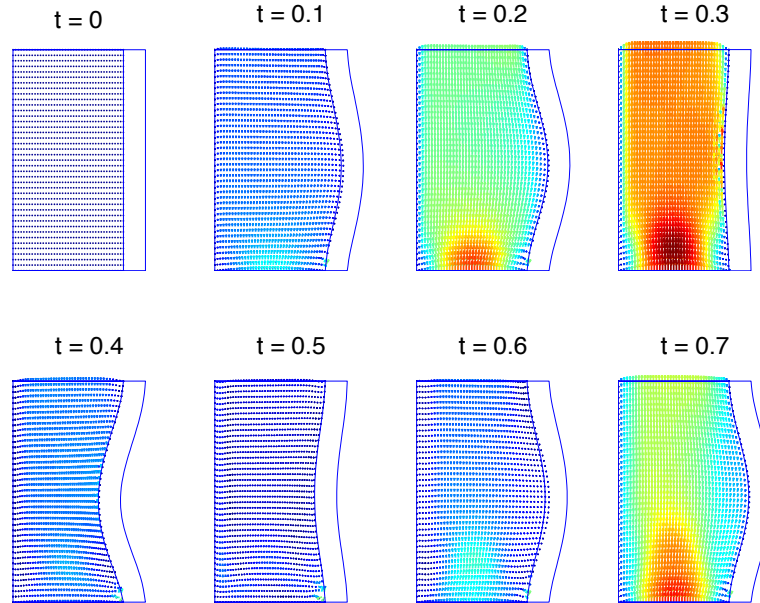


Figure 4.7: Velocity profile and domain deformations for fluid-structure interaction in 2D. Inflow velocity at the bottom of  $\Omega_f$  is  $v_2 = 10X_1(1 - X_1)\sin^2(2\pi t)$ . Parameters:  $\mu = 0.01$ ,  $\rho = 1$ ,  $\nu = 0.4$ ,  $E = 2900$ . Colour corresponds to velocity magnitude, and is scaled from blue (lowest) to red (highest) for each time step individually.

The iteration counts once again indicate that convergence of preconditioned GMRES using  $\mathbf{S}_4^{-1}$  is mesh-independent, while  $\mathbf{S}_5^{-1}$  gives iteration counts which grow as the mesh is refined. However, we note that there is the opposite dependence on the viscosity parameter: as  $\mu$  decreases the iteration counts for the  $\mathbf{S}_4^{-1}$  approximation grow, while those for the  $\mathbf{S}_5^{-1}$  approximation decrease. This is due to the fact that  $\mathbf{S}_4$  is only based on the fluid component of the problem, which becomes a worse representation of the Schur complement as  $\mu \rightarrow 0$ . Both components are equally represented within  $\mathbf{S}_5$ , and in this case smaller  $\mu$  results in a better approximation.

We also observe that due to the simplicity of computation of  $\mathbf{S}_5^{-1}$  (the only multiplication by an inverse involves a banded matrix of size  $DOF_T$ ), the GMRES times did not grow significantly as  $\mu$  decreased. Computing  $\mathbf{S}_4^{-1}$  involves multiplication by the inverse of the  $\mathbf{A}_{DD}$  matrix, whose conditioning worsens as  $\mu \rightarrow 0$ , which affects AMG solve times.

CPU times, being implementation dependent, should in this case be taken as indicative rather than definitive. However, the fact that times for  $\mathbf{S}_5$  appear to be far less affected by decreasing  $\mu$  than the times for  $\mathbf{S}_4$  highlight the importance of optimising all aspects of implementation.

The dependence of GMRES iteration counts on  $\mu$ , however, suggests that the nature of the components of a coupled problem is an important factor in the performance of the monolithic method with block preconditioning.

We also verify that the choice of time step size does not have a significant effect on GMRES convergence. Table 4.9 contains iteration counts for different values of  $\Delta t$  and  $\mu$  for a problem on the original uniform mesh with  $DOF_T = 21$ . It can be observed that while GMRES convergence is largely unaffected by varying  $\Delta t$ , the (outer) Newton iteration counts grow with  $\Delta t$  for small  $\mu$ . The flow is generally more



|   | $S_4$              |                    |                    | $S_5$              |                    |                    |
|---|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|
| M | $\mu = 1$          | $\mu = 0.1$        | $\mu = 0.01$       | $\mu = 1$          | $\mu = 0.1$        | $\mu = 0.01$       |
| 1 | <b>32</b> /4/9.04  | <b>40</b> /4/10.72 | <b>60</b> /5/17.43 | <b>44</b> /4/8.64  | <b>35</b> /4/7.00  | <b>42</b> /5/8.26  |
| 2 | <b>34</b> /4/11.24 | <b>41</b> /4/14.49 | <b>71</b> /5/24.99 | <b>56</b> /4/10.51 | <b>48</b> /4/10.35 | <b>42</b> /5/12.69 |
| 3 | <b>34</b> /4/16.17 | <b>42</b> /4/19.96 | <b>73</b> /5/35.48 | <b>69</b> /4/15.62 | <b>62</b> /4/14.41 | <b>48</b> /5/18.57 |
| 4 | <b>33</b> /4/22.78 | <b>40</b> /4/28.28 | <b>73</b> /5/57.24 | <b>87</b> /4/21.56 | <b>82</b> /4/20.48 | <b>62</b> /5/26.71 |

Table 4.8: Maximum number of GMRES iterations/number of Newton iterations/CPU time in seconds per one GMRES solve for the FSI problem.  $\Delta t = 0.01$ ,  $TOL = 10^{-10}$ .

difficult to resolve for low viscosity (high Reynold's number). Small  $\Delta t$  provides a better initial guess for the solution, based on the previous time step, and leads to the discretised fluid equations being dominated by the mass matrix, therefore reducing the effect of the nonlinear convective term. Larger values of  $\Delta t$  require more iterations for the Newton's method to converge. This behaviour is unrelated to the block preconditioner used, but can be addressed separately by specific solvers for the fluid subproblem.

|            | $S_4$             |                    |                     | $S_5$             |                   |                    |
|------------|-------------------|--------------------|---------------------|-------------------|-------------------|--------------------|
| $\Delta t$ | $\mu = 1$         | $\mu = 0.1$        | $\mu = 0.01$        | $\mu = 1$         | $\mu = 0.1$       | $\mu = 0.01$       |
| 0.1        | <b>19</b> /6/81.4 | <b>26</b> /6/116.2 | <b>43</b> /12/196.9 | <b>27</b> /6/77.8 | <b>26</b> /6/70.7 | <b>35</b> /12/99.2 |
| 0.05       | <b>18</b> /5/94.9 | <b>25</b> /5/98.7  | <b>44</b> /10/221.6 | <b>27</b> /5/82.1 | <b>26</b> /5/75.2 | <b>37</b> /10/93.2 |
| 0.01       | <b>18</b> /3/71.4 | <b>25</b> /3/102.0 | <b>33</b> /4/130.4  | <b>25</b> /3/70.9 | <b>26</b> /3/71.8 | <b>30</b> /4/93.4  |

Table 4.9: Maximum number of GMRES iterations/number of Newton iterations/GMRES CPU time in seconds for fluid-structure interaction in 2D: different values of  $\Delta t$ ,  $\mu$  ( $DOF_F = 21$ ,  $TOL = 10^{-6}$ ).

## 4.6 Discussion

The aim of this chapter was to show that the monolithic approach can be used efficiently to solve coupled problems. On the one hand, the interaction between the components is not downplayed by decomposing the system: the system is always

treated as a whole. On the other hand, existing solvers for the sub-problems can be used to achieve efficiency of implementation. Having proposed and compared several preconditioning strategies based on approximating the Schur complement block, we identified approximations which are computationally efficient. One of the methods is shown to provide mesh-independent convergence of preconditioned GMRES in the tests considered, therefore guaranteeing scalability of the solver to large meshes. Another method, based on the so-called BFBt preconditioner, exhibits competitive, although not mesh-independent, iteration counts, and performs particularly well for FSI problems with low viscosity (high Reynolds number).

Having a robust way of solving large-scale coupled problems allows us to consider *a posteriori* error estimation and adaptive mesh refinement in such problems. While the goal of adaptive mesh refinement is to use the smallest possible number of degrees of freedom to solve a problem to desired accuracy, the method developed in the following chapter requires multiple solves of block systems through time, and moreover, multiple solves of more refined dual systems through time. The ability to perform these solves quickly and using limited computational resources is extremely important.

The tests performed in this chapter provide a basis for further investigation in more complex settings of multiphysics problems. A further example of coupling unobstructed fluid flow and flow through a porous medium is studied in Chapter 6, where we revisit the question of performance of block preconditioners developed here.

## Chapter 5

# *A posteriori* error estimation and adaptive mesh refinement

An important aspect of numerical methods is the assessment of the solutions' accuracy. Since in most practical problems the analytic solution cannot be obtained and its regularity cannot be established, it is impossible to use *a priori* error estimates effectively. In contrast, *a posteriori* methods only require knowledge of the approximate solution and can therefore be applied to a broad range of problems.

The presence of the Lagrange multiplier in the formulation of monolithic coupled problems used in this work allows us to derive natural expressions for the error in a quantity of interest that is supported along the interface. This is particularly useful in applications where the Lagrange multiplier represents a physical quantity, such as the normal stress along arterial walls when assessing the risk of aneurysm rupture [39, 38], and the flux through the boundary between a fluid and a porous medium when modelling air flow in the lungs [86]. This is made possible by the application of the monolithic approach to coupling, as described in Chapter 3, and the availability of efficient solvers for the monolithic problems, such as those developed in Chapter 4.

Error estimation for coupled problems solved using operator decomposition techniques was considered in [31, 32, 52, 53, 125].

In this chapter we introduce the *a posteriori* error estimation technique for coupled problems by considering a simple steady case in Section 5.1. Namely, we discuss the idea of a quantity of interest, and derive an *a posteriori* representation formula for the error in this quantity. We assess the accuracy of this formula, and present an adaptive mesh refinement technique which is based on local error estimates. In order to illustrate the methodology we perform a number of numerical tests to study the performance of the formula and the refinement strategy. The theory is extended to time-dependent problems in Section 5.2, including a new representation formula and its accuracy estimate, and a block space-time adaptive mesh refinement strategy. This is followed by numerical examples of error estimation and adaptive mesh refinement in coupled time-dependent problems.

Most of the material presented in Section 5.2 was published as [5].

## 5.1 Steady coupled problems

We begin by recalling the weak formulation of an elliptic coupled problem (3.2) from Section 3.3.1:

*Find  $(u_1, u_2, \lambda) \in X_{1,D} \times X_{2,D} \times Y$  such that*

$$\mp \langle \lambda, v_i \rangle_\Gamma + a_i(u_i, v_i) = (f_i, v_i) + \langle g_{i,N}, v_i \rangle_{\Gamma_i^N} \quad \forall v_i \in X_{i,0}, \quad i = 1, 2, \quad (5.1a)$$

$$\langle \nu, u_1 - u_2 \rangle_\Gamma = 0 \quad \forall \nu \in Y. \quad (5.1b)$$

The finite element formulation is given by (3.3):

Find  $(u_1^h, u_2^h, \lambda^h) \in X_{1,D}^h \times X_{2,D}^h \times Y^h$  such that

$$\mp \langle \lambda^h, v_i^h \rangle_\Gamma + a_i(u_i^h, v_i^h) = (f_i, v_i^h) + \langle g_{i,N}, v_i^h \rangle_{\Gamma_i^N} \quad \forall v_i^h \in X_{i,0}^h, \quad i = 1, 2, \quad (5.2a)$$

$$\langle \nu^h, u_1^h - u_2^h \rangle_\Gamma = 0 \quad \forall \nu^h \in Y^h. \quad (5.2b)$$

For simplicity we assume that  $g_{i,N} = 0$  for  $i = 1, 2$ . The following derivations can be carried out in a similar manner for non-zero Neumann conditions, but require adjustments in the definitions of residuals, and the error bounds.

### 5.1.1 The error representation formula

Let the errors in the solution of (5.2) be  $e_i = u_i - u_i^h$ ,  $i = 1, 2$ , and  $e_\lambda = \lambda - \lambda^h$ . The quantity of interest ( $QoI$ ) is defined as a linear functional of the solution with given coefficient functions  $\psi_1$ ,  $\psi_2$  and  $\psi$ , with the corresponding error  $\mathcal{E}$ , i.e.

$$\begin{aligned} (QoI) &= - \langle \psi, \lambda \rangle_\Gamma + (\psi_1, u_1)_{\Omega_1} + (\psi_2, u_2)_{\Omega_2}, \\ \mathcal{E} &= - \langle \psi, e_\lambda \rangle_\Gamma + (\psi_1, e_1)_{\Omega_1} + (\psi_2, e_2)_{\Omega_2}, \end{aligned} \quad (5.3)$$

where  $(\cdot, \cdot)_{\Omega_i}$  for  $i = 1, 2$  is the inner product in  $L^2(\Omega_i)$ :

$$(v_i, w_i)_{\Omega_i} = \int_{\Omega_i} v_i w_i \, d\mathbf{x}, \quad i = 1, 2.$$

The choice of the coefficients enables us to specify both localized and globally supported errors. For example, to calculate the average error in  $\Omega_1$  we choose  $\psi_1 = 1$  and  $\psi_2 = \psi = 0$ , and to calculate the average error along  $\Gamma$  we choose  $\psi_1 = \psi_2 = 0$  and  $\psi = 1$ .

A representation formula will provide a practical tool for estimating  $\mathcal{E}$ . We derive

it using the solution of an adjoint problem, which in the weak form can be written as follows:

*Find  $(\phi_1, \phi_2, \mu) \in X_{1,0} \times X_{2,0} \times Y$  such that*

$$\mp \langle \mu, v_i \rangle_\Gamma + a_i(\phi_i, v_i) = (\psi_i, v_i)_{\Omega_i} \quad \forall v_i \in X_{i,0}, \quad i = 1, 2, \quad (5.4a)$$

$$\langle \rho, \phi_1 - \phi_2 \rangle_\Gamma = \langle \rho, \psi \rangle_\Gamma \quad \forall \rho \in Y. \quad (5.4b)$$

The following theorem formalises the derivation of the error representation formula for the coupled problem (5.2).

**Theorem 7.** *The error in the quantity of interest  $\mathcal{E}$  defined by (5.3) can be computed as*

$$\mathcal{E} = R_\Gamma(\mu) + R_1(\phi_1) + R_2(\phi_2), \quad (5.5)$$

where  $(\phi_1, \phi_2, \mu)$  is the solution of the adjoint problem (5.4),  $R_i(\phi_i)$ ,  $i = 1, 2$  and  $R_\Gamma$  are the generalised residuals in the original problem (5.2) for the domains or the interface respectively, and are given by

$$R_i(\phi_i) = \pm \langle \lambda^h, \phi_i \rangle_\Gamma + (f_i, \phi_i)_{\Omega_i} - a_i(u_i^h, \phi_i), \quad i = 1, 2,$$

and

$$R_\Gamma(\mu) = \langle \mu, u_1^h - u_2^h \rangle_\Gamma.$$

*Proof.* The errors  $e_i = u_i - u_i^h$ ,  $i = 1, 2$ , are permissible as test functions in the weak adjoint equation (5.4a), hence

$$\mp \langle \mu, u_i - u_i^h \rangle_\Gamma + a_i(\phi_i, u_i - u_i^h) = (\psi_i, e_i)_{\Omega_i}, \quad i = 1, 2. \quad (5.6)$$

Similarly,  $\phi_i$ ,  $i = 1, 2$  are permissible as test functions in the weak formulation of the original problem (5.1), giving

$$a_i(u_i, \phi_i) = \pm \langle \lambda, \phi_i \rangle_\Gamma + (f_i, \phi_i)_{\Omega_i}, \quad i = 1, 2. \quad (5.7)$$

Since  $a_i(v, w)$  is symmetric and linear, we can substitute (5.7) into (5.6) yielding

$$\mp \langle \mu, u_i - u_i^h \rangle_\Gamma \pm \langle \lambda, \phi_i \rangle_\Gamma + (f_i, \phi_i)_{\Omega_i} - a_i(\phi_i, u_i^h) - (\psi_i, e_i)_{\Omega_i} = 0, \quad i = 1, 2,$$

or, equivalently,

$$\mp \langle \mu, u_i - u_i^h \rangle_\Gamma \pm \langle e_\lambda, \phi_i \rangle_\Gamma + R_i(\phi_i) - (\psi_i, e_i)_{\Omega_i} = 0, \quad i = 1, 2.$$

Adding the equations for  $i = 1$  and  $i = 2$  and rearranging the terms,

$$\begin{aligned} & - \langle e_\lambda, \phi_1 - \phi_2 \rangle_\Gamma + (\psi_1, e_1)_{\Omega_1} + (\psi_2, e_2)_{\Omega_2} \\ & = - \langle \mu, u_1 - u_2 \rangle_\Gamma + \langle \mu, u_1^h - u_2^h \rangle_\Gamma + R_1(\phi_1) + R_2(\phi_2). \end{aligned}$$

It follows from the weak continuity conditions (5.1b) and (5.4b) that

$$\langle \mu, u_1 - u_2 \rangle_\Gamma = 0 \quad \text{and} \quad \langle e_\lambda, \phi_1 - \phi_2 \rangle_\Gamma = \langle e_\lambda, \psi \rangle_\Gamma.$$

Hence the error can be written as  $\mathcal{E} = R_\Gamma(\mu) + R_1(\phi_1) + R_2(\phi_2)$ . □

**Remark 1.** *The strong form of the adjoint problem is*

$$-\nabla \cdot (k_i \nabla \phi_i) + c_i \phi_i = \psi_i \quad \text{in } \Omega_i, \quad i = 1, 2,$$

$$\begin{aligned}
\phi_i &= 0 && \text{on } \Gamma_i^D, \ i = 1, 2, \\
k_i \nabla \phi_i \cdot \mathbf{n}_i &= 0 && \text{on } \Gamma_i^N, \ i = 1, 2, \\
\phi_1 - \phi_2 &= \psi && \text{on } \Gamma, \\
k_1 \nabla \phi_1 \cdot \mathbf{n}_1 + k_2 \nabla \phi_2 \cdot \mathbf{n}_2 &= 0 && \text{on } \Gamma,
\end{aligned}$$

**Remark 2.** *By Galerkin orthogonality, the error representation formula produces  $\mathcal{E} = 0$  in the special case when the exact adjoint solution lies in the finite element space of the original problem.*

*Indeed, let  $\phi_1 \in X_{1,0}^h$ ,  $\phi_2 \in X_{2,0}^h$  and  $\mu \in Y^h$  be used as test functions in (5.2).*

*Then*

$$\begin{aligned}
R_i(\phi_i) &= \pm \langle \lambda^h, \phi_i \rangle_\Gamma - a_i(u_i^h, \phi_i) + (f_i, \phi_i)_{\Omega_i} = 0, \ i = 1, 2, \\
\langle \mu, u_1^h - u_2^h \rangle_\Gamma &= 0.
\end{aligned}$$

*It follows immediately from the representation formula (5.5) that  $\mathcal{E} = 0$ .*

### 5.1.2 The effect of adjoint approximation

Expression (5.5) is exact, since no approximations were made in the derivation. In practice we can rarely obtain the exact adjoint solution. Functions  $(\phi_1, \phi_2, \mu)$  are usually computed using what is deemed to be a more accurate finite element method than (5.2). The error in the adjoint problem is then assumed negligible. The higher accuracy can be achieved by using either higher order basis functions, or a finer mesh. We consider the error in the representation formula under such circumstances.

Let the adjoint solutions  $\phi_i^h \in \hat{X}_{i,0}^h$ ,  $i = 1, 2$ , and  $\mu \in \hat{Y}^h$ , such that  $X_{i,0}^h \subset \hat{X}_{i,0}^h \subset X_{i,0}$  and  $Y^h \subset \hat{Y}^h \subset Y$ . The finite element formulation for the adjoint problem is:



Find  $(\phi_1^h, \phi_2^h, \mu^h) \in \hat{X}_{1,0}^h \times \hat{X}_{2,0}^h \times \hat{Y}^h$  such that

$$\mp \langle \mu^h, v_i^h \rangle_\Gamma + a_i(\phi_i^h, v_i^h) = (\psi_i, v_i^h)_{\Omega_i} \quad \forall v_i^h \in \hat{X}_{i,0}^h, \quad i = 1, 2, \quad (5.9a)$$

$$\langle \rho^h, \phi_1^h - \phi_2^h \rangle_\Gamma = \langle \rho^h, \psi \rangle_\Gamma \quad \forall \rho^h \in \hat{Y}^h. \quad (5.9b)$$

Subtracting (5.9a) from (5.4a) and (5.9b) from (5.4b), we have

$$\mp \langle \mu - \mu^h, v_i^h \rangle_\Gamma + a_i(\phi_i - \phi_i^h, v_i^h) = 0 \quad \forall v_i^h \in \hat{X}_{i,0}^h, \quad i = 1, 2, \quad (5.10a)$$

$$\langle \rho^h, \phi_1 - \phi_2 - \phi_1^h + \phi_2^h \rangle_\Gamma = 0 \quad \forall \rho^h \in \hat{Y}^h. \quad (5.10b)$$

Since  $u_i^h \in X_{i,0}^h \subset \hat{X}_{i,0}^h$ , we substitute  $u_i^h$  for  $v_i^h$  in (5.10a) and add the two equations for  $i = 1$  and 2 yielding

$$- \langle \mu - \mu^h, u_1^h - u_2^h \rangle_\Gamma + a_1(\phi_1 - \phi_1^h, u_1^h) + a_2(\phi_2 - \phi_2^h, u_2^h) = 0. \quad (5.11)$$

Similarly,  $\lambda^h \in Y^h \subset \hat{Y}^h$ , and (5.10b) holds for  $\rho^h = \lambda^h$ , so that

$$\langle \lambda^h, \phi_1 - \phi_2 - \phi_1^h + \phi_2^h \rangle_\Gamma = 0. \quad (5.12)$$

Recalling that

$$\mathcal{E} = \langle \mu, u_1^h - u_2^h \rangle_\Gamma + \langle \lambda^h, \phi_1 - \phi_2 \rangle_\Gamma + \sum_{i=1,2} \{ (f_i, \phi_i)_{\Omega_i} - a_i(\phi_i, u_i^h) \},$$

we add (5.11) and subtract (5.12) to give

$$\mathcal{E} = \langle \mu, u_1^h - u_2^h \rangle_\Gamma - \langle \mu - \mu^h, u_1^h - u_2^h \rangle_\Gamma + \langle \lambda^h, \phi_1 - \phi_2 \rangle_\Gamma$$

$$\begin{aligned}
& - \langle \lambda^h, \phi_1 - \phi_2 - \phi_1^h + \phi_2^h \rangle_\Gamma + \sum_{i=1,2} \{ (f_i, \phi_i)_{\Omega_i} - a_i(\phi_i, u_i^h) + a_i(\phi_i - \phi_i^h, u_i^h) \} \\
& = \langle \mu^h, u_1^h - u_2^h \rangle_\Gamma + \langle \lambda^h, \phi_1^h - \phi_2^h \rangle_\Gamma + \sum_{i=1,2} \{ (f_i, \phi_i)_{\Omega_i} - a_i(\phi_i^h, u_i^h) \}.
\end{aligned}$$

Only the  $(f_i, \phi_i)_{\Omega_i}$  terms in the resulting formula depend on the exact adjoint solution.

We add and subtract  $(f_i, \phi_i^h)_{\Omega_i}$  to obtain

$$\mathcal{E} = R_\Gamma(\mu^h) + R_1(\phi_1^h) + R_2(\phi_2^h) + \sum_{i=1,2} (f_i, \phi_i - \phi_i^h)_{\Omega_i} = \hat{\mathcal{E}} + \epsilon,$$

where  $\hat{\mathcal{E}}$  is the approximate error representation formula,

$$\hat{\mathcal{E}} = R_\Gamma(\mu^h) + R_1(\phi_1^h) + R_2(\phi_2^h), \quad (5.13)$$

and  $\epsilon$  is the error in the representation formula due to errors in the adjoint solution,

$$\epsilon = (f_1, \phi_1 - \phi_1^h)_{\Omega_1} + (f_2, \phi_2 - \phi_2^h)_{\Omega_2}.$$

Cauchy-Schwartz and triangle inequalities imply that

$$|\epsilon| \leq \|f_1\|_{L^2(\Omega_1)} \|\phi_1 - \phi_1^h\|_{L^2(\Omega_1)} + \|f_2\|_{L^2(\Omega_2)} \|\phi_2 - \phi_2^h\|_{L^2(\Omega_2)}.$$

If the exact adjoint solutions are sufficiently regular, e.g.  $\phi_i \in H^p(\Omega_i)$ ,  $i = 1, 2$ , and  $\phi_i^h$  are piecewise polynomials of degree  $(p-1)$ , then the error in the adjoint solution can be bounded:

$$\|\phi_i - \phi_i^h\|_{L^2(\Omega_i)} \leq \hat{C} h^p |\phi_i|_{H^p(\Omega_i)}, \quad \text{where } \hat{C} = \text{const.}$$

This, in turn, indicates that the error in the representation formula is  $\epsilon = \mathcal{O}(h^p)$ .

We introduce the effectivity ratio  $\mathcal{R}$ ,

$$\mathcal{R} = \frac{\hat{\mathcal{E}}}{\mathcal{E}} = 1 - \frac{\epsilon}{\mathcal{E}}. \quad (5.14)$$

which provides a measure of the accuracy of the approximate error  $\hat{\mathcal{E}}$ . In Section (5.1.4) we calculate the approximate error measure  $\hat{\mathcal{E}}$  for the numerical tests using the adjoint finite element solutions in the representation formula. Where possible, we also calculate the exact error measure  $\mathcal{E}$  by substituting the exact and the finite element solutions into the definition (5.3). The effectivity ratio  $\mathcal{R}$  quantifies the error in the approximate error representation formula.

### 5.1.3 An adaptive strategy

Assuming that the error  $\epsilon$  in the representation formula is negligible compared to the approximate error measure  $\hat{\mathcal{E}}$ , and the exact error  $\mathcal{E}$  is well-represented by  $\hat{\mathcal{E}}$ , we can use the approximate measure (5.13) as the basis for adaptive mesh refinement. Let  $\pi^h$  denote orthogonal projection operators from the adjoint finite element spaces to the finite element test spaces,

$$\pi^h : \begin{cases} \hat{X}_i^h \rightarrow X_i^h, \text{ such that } \forall v^h \in \hat{X}_i^h, \forall w^h \in X_i^h : (v^h - \pi^h v^h, w^h)_{\Omega_i} = 0, \quad i = 1, 2, \\ \hat{Y}^h \rightarrow Y^h, \text{ such that } \forall \nu^h \in \hat{Y}^h, \forall w^h \in X_j^h : \langle \nu^h - \pi^h \nu^h, w^h \rangle_{\Gamma} = 0, \end{cases}$$

where  $j$  is the index of the domain that is chosen to define the finite element space on  $\Gamma$  (specifically where the mesh in the domain conforms with the mesh on the interface).

By Galerkin orthogonality  $R_\Gamma(\pi^h \mu^h) + R_1(\pi^h \phi_1^h) + R_2(\pi^h \phi_2^h) = 0$ , and

$$\mathcal{E} = R_\Gamma(\mu^h - \pi^h \mu^h) + R_1(\phi_1^h - \pi^h \phi_1^h) + R_2(\phi_2^h - \pi^h \phi_2^h).$$

The error can be decomposed into local contributions from individual elements  $\Delta$ , as

$$\begin{aligned} \hat{\mathcal{E}} &= \sum_{i=1,2} \sum_{\Delta \in \tau_i} \hat{\mathcal{E}}_\Delta = \sum_{i=1,2} \sum_{\Delta \in \tau_i} \{R_\Gamma^\Delta(\mu^h - \pi^h \mu^h) + R_i^\Delta(\phi_i^h - \pi^h \phi_i^h)\}, \text{ where} \\ R_\Gamma^\Delta(\mu^h - \pi^h \mu^h) &= \int_{\Gamma \cap \partial \Delta} (\mu^h - \pi^h \mu^h)(u_1^h - u_2^h) \, ds, \\ R_i^\Delta(\phi_i^h - \pi^h \phi_i^h) &= \pm \int_{\Gamma \cap \partial \Delta} \lambda^h(\phi_i^h - \pi^h \phi_i^h) \, ds \\ &\quad + \int_{\Delta} \{f_i(\phi_i^h - \pi^h \phi_i^h) - a_i(u_i^h, \phi_i^h - \pi^h \phi_i^h)\} \, d\mathbf{x}, \quad i = 1, 2. \end{aligned}$$

Local contributions exceeding a local tolerance  $|\hat{\mathcal{E}}_\Delta| > LATOL$  indicate the elements that should be refined. The local tolerance for the original uniform mesh can be computed as the ratio of a given global tolerance  $TOL$  to the total number of elements.

**Remark 3.** *Since we compute a linear functional of the error, rather than a norm of the error, the local refinement strategy might appear disadvantageous, simply because*

$$\hat{\mathcal{E}} \leq \sum_{i=1,2} \sum_{\Delta \in \tau_i} |\hat{\mathcal{E}}_\Delta|$$

*is an extremely crude estimate. In particular, for problems with symmetric solutions local errors can be significant, and yet the global error can be zero due to cancellation. The refinement algorithm used in the simulations does not impose the global condition  $|\hat{\mathcal{E}}| < TOL$ , only the local condition  $|\hat{\mathcal{E}}_\Delta| < LATOL$ .*

### 5.1.4 Numerical results

Below we present numerical results illustrating the use of the error representation formula (5.13). All of the problems in this section were solved on a rectangular domain with  $\Omega_1 = (0, 1) \times (0, 1)$ ,  $\Omega_2 = (0, 1) \times (1, 2)$ , and  $\Gamma = \{(1, s) \mid s \in (0, 1)\}$ . Unless stated otherwise, the initial meshes are uniform and conform on the interface  $\Gamma$ , the finite element spaces for all of  $u_1^h$ ,  $u_2^h$ ,  $\lambda^h$ ,  $\phi_1^h$ ,  $\phi_2^h$  and  $\mu^h$  consist of piecewise-linear functions, and higher accuracy of the adjoint solution is achieved by solving the adjoint on the original mesh twice refined uniformly.

#### 5.1.4.1 Error in the domain

First, we consider a quantity of interest which is the average in the sub-domain  $\Omega_2$ . The choice of  $\psi_1 = \psi = 0$  and  $\psi_2 = 1$  leads to the following error in the quantity of interest:

$$\mathcal{E} = \int_{\Omega_2} e_2 \, d\mathbf{x}.$$

We choose  $k_1 = k_2 = 1$ , the right-hand sides such that  $u_1 = u_2 = \sin(\pi x) \sin(\pi y)$ , and use zero Dirichlet boundary conditions everywhere on the outside boundaries. Starting with a uniform mesh with a total of 453 degrees of freedom, we refine triangular elements in the domain based on local error estimates  $\hat{\mathcal{E}}_\Delta$ , as described above. The global tolerance value is  $TOL = 0.02^*$ . We allow at most three refinements in a single element of the initial mesh so as to have a control over the mesh size.

The finite element solutions of the original and adjoint problems are shown in Figures 5.1a and 5.1b respectively, and the refined mesh in Figure 5.2a. As expected,

---

\*In all numerical tests the global tolerance  $TOL$  is chosen for illustrative purposes, i.e. so that the global error is above the tolerance, and therefore produces a refinement, but at the same time it is not so high as to require uniform refinement. Depending on the goals of mesh refinement, in a practical problem this value can be determined in a similar manner, by the maximum error allowed, or by the maximum mesh size allowed.

the mesh in  $\Omega_2$  is significantly more refined than the mesh in  $\Omega_1$  since the quantity of interest lies in  $\Omega_2$ , and the adjoint solution is concentrated in this sub-domain. The estimated values of the error and the effectivity ratios are shown in Table 5.1. The effectivity ratios remain close to one, meaning that the error estimates represent the actual error well.

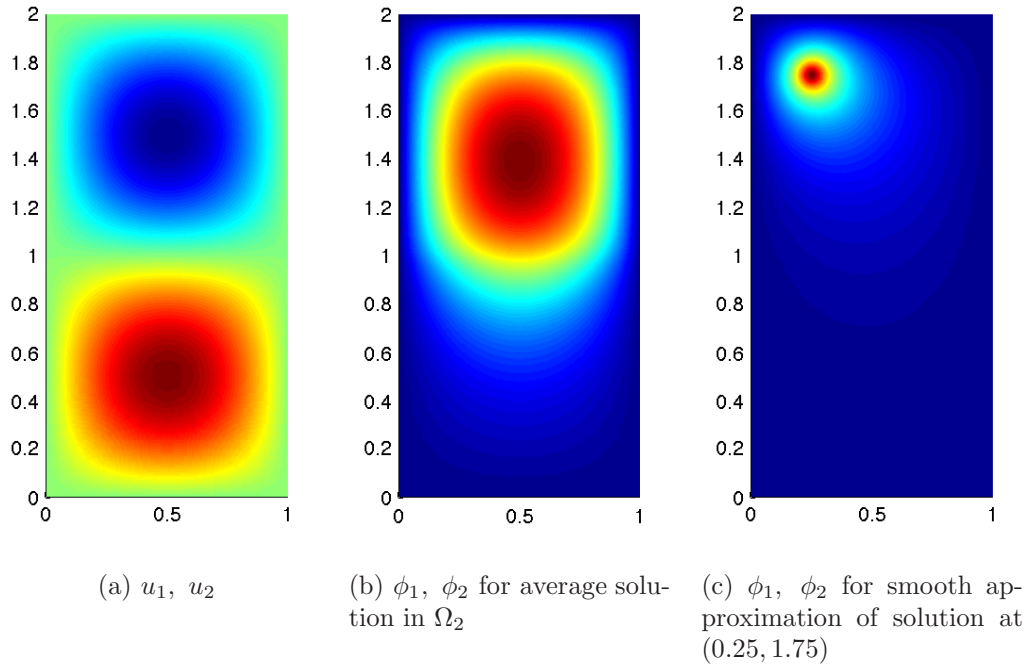


Figure 5.1: Original and adjoint solutions for the problems with the quantities of interest measuring the solution in the domain.

| Average error in $\Omega_2$ |                     |               | Point error, $A = 1000$ |                     |               | Point error, $A = 10000$ |                     |               |
|-----------------------------|---------------------|---------------|-------------------------|---------------------|---------------|--------------------------|---------------------|---------------|
| $DOF$                       | $\hat{\mathcal{E}}$ | $\mathcal{R}$ | $DOF$                   | $\hat{\mathcal{E}}$ | $\mathcal{R}$ | $DOF$                    | $\hat{\mathcal{E}}$ | $\mathcal{R}$ |
| 453                         | -0.002682           | 0.963         | 453                     | -0.003641           | 1.775         | 453                      | -0.002920           | 1.424         |
| 4301                        | -0.000631           | 0.972         | 6959                    | -0.000321           | 0.867         | 6959                     | -0.000364           | 0.983         |

Table 5.1: Estimated average error in  $\Omega_2$  and approximation to point error, and corresponding effectivity ratios.  $DOF$  is the total number of degrees of freedom in the system. Higher value of parameter  $A$  corresponds to a better approximation of the point error.

Next, we solve the same problem with  $\psi_2 = (A/\pi) \exp(-A(x - 0.25)^2 - A(y - 1.75)^2)$ .

This weighting is chosen as a smooth approximation to the Dirac delta function

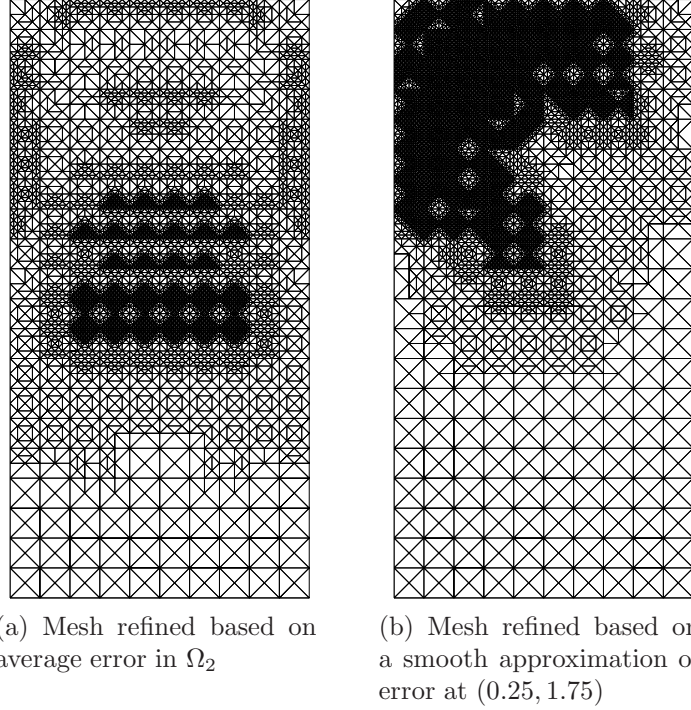


Figure 5.2: Refined mesh for the problems with quantities of interest measuring the solution in the domain.

$\delta_{(0.25, 1.75)}$ . The quantity of interest approximates a point value of the solution, and the error is

$$\mathcal{E} = \int_{\Omega_2} \psi_2 e_2 \, d\mathbf{x} \approx e_2(0.25, 1.75). \quad (5.15)$$

The refined mesh for  $A = 1000$  is shown in Figure 5.2b, and the adjoint in Figure 5.1c. As is to be expected, the highest concentration of elements is around the point  $(0.25, 1.75)$  where the quantity of interest is located and where the adjoint solution is concentrated, while virtually no refinements are required in  $\Omega_1$ . The estimated error values  $\hat{\mathcal{E}}$  are given in Table 5.1. Note that we are now computing approximate effectivity ratios, since the exact error is calculated as  $\mathcal{E} = u_2(0.25, 1.75) - u_2^h(0.25, 1.75)$ , while the approximate error is calculated in the integral form, according to (5.15). The more closely  $\psi_2$  represents the delta function, the closer the effectivity ratio approaches one. This is illustrated by comparing the error and effectivity values for

$A = 1000$  with  $A = 10000$ . The latter gives a better approximation for a delta function (see Table 5.1), and consequently better effectivity ratios. Note also that the effectivity ratios computed on the original coarse uniform mesh are relatively poor. However, the refinement strategy improves them significantly.

#### 5.1.4.2 Error along the interface

We consider a test problem where the quantity of interest is the average of the Lagrange multiplier on the interface. Letting  $\psi_1 = \psi_2 = 0$  and  $\psi = 1$ , we have

$$\mathcal{E} = \int_{\Gamma} e_{\lambda} \, ds.$$

For this choice of the quantity of interest the strong form of the coupling condition is  $\phi_1 - \phi_2 = 1$  on  $\Gamma$ . Consequently we cannot have zero Dirichlet boundary conditions at points  $(0, 1)$  and  $(1, 1)$ . We avoid the inconsistency by changing the boundary conditions to zero Neumann boundary conditions on  $\Gamma_1^N = \{(0, y) \mid y \in (0, 2)\}$  and  $\Gamma_2^N = \{(1, y) \mid y \in (0, 2)\}$  in both the original and the adjoint problems. Alternatively, the weighting function  $\psi$  could be chosen so that the Dirichlet boundary conditions at  $x = 0$  and  $x = 1$  are satisfied, e.g.  $\psi = \sin(\pi x)$ .

We now choose different diffusion coefficients in the sub-domains, specifically  $k_1 = 3$  and  $k_2 = 1$ . The right-hand sides are such that  $u_1 = \frac{1}{3} \cos(\pi x) \sin(\pi y)$  and  $u_2 = \cos(\pi x) \sin(\pi y)$ . In this example the exact adjoint solution is linear  $\phi_1 = 0.25x$ ,  $\phi_2 = 0.75(x - 2)$ ,  $\lambda = 0.75$  (see Figure 5.3a), and thus lies in the finite element space of the original problem. The exact error must therefore be zero by Galerkin orthogonality, and the estimated error value is  $\sim 10^{-17}$ , hence the effectivity ratio for this example was not computed.



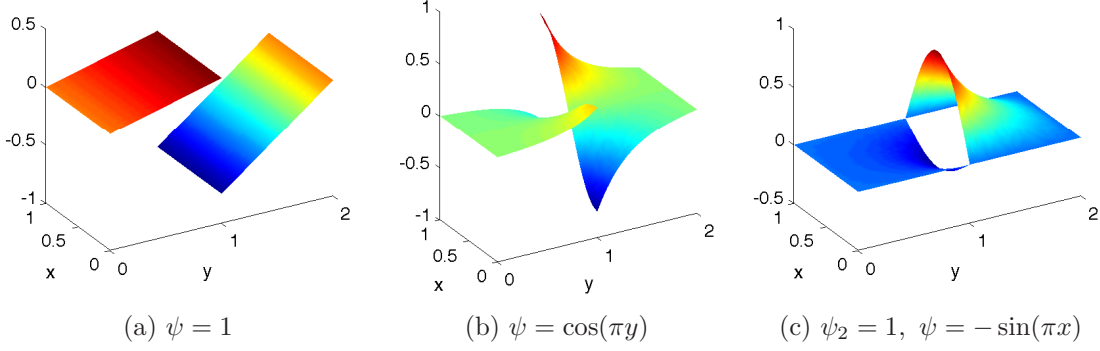


Figure 5.3: Adjoint solutions for a problem with the Lagrange multiplier as the quantity of interest.

If we take  $\psi = \cos(\pi y)$ , the adjoint solution is no longer linear, and the error is no longer zero. We start with a uniform mesh with 453 degrees of freedom, and use  $TOL = 0.1$ . Both the approximate errors and the corresponding effectivity ratios can be seen in Table 5.2. The adjoint solution and the refined mesh are presented in Figures 5.3b and 5.4a respectively. The interface region is the most refined area and more refinements are needed in  $\Omega_2$  where solution gradients are larger than in  $\Omega_1$ .

| Weighted average on $\Gamma$ |               |               | Mixed $\Omega_2$ and $\Gamma$ |               |               |
|------------------------------|---------------|---------------|-------------------------------|---------------|---------------|
| $DOF$                        | $\mathcal{E}$ | $\mathcal{R}$ | $DOF$                         | $\mathcal{E}$ | $\mathcal{R}$ |
| 453                          | -0.006629     | 1.015         | 453                           | 0.003947      | 1.055         |
| 5431                         | 0.000350      | 0.890         | 5605                          | -0.001233     | 0.946         |

Table 5.2: Estimated values of the error in the Lagrange multiplier and the corresponding effectivity ratios.  $DOF$  is the total number of degrees of freedom in the system.

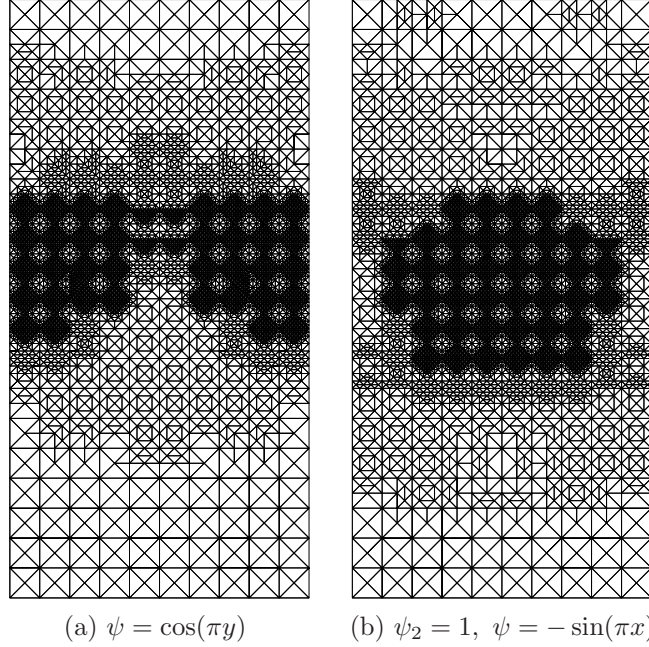


Figure 5.4: Adapted meshes for problems with error on the interface  $\Gamma$  as the quantity of interest.

#### 5.1.4.3 Mixed error

Finally we consider a quantity of interest that includes both domain and interface components. We take  $\psi_1 = 0$ ,  $\psi_2 = 1$  and  $\psi = -\sin(\pi x)$ , so that the error is

$$\mathcal{E} = \int_{\Gamma} \psi e_{\lambda} \, ds + \int_{\Omega_2} \psi_2 e_2 \, d\mathbf{x}.$$

Further,  $k_1 = k_2 = 1$ ,  $f_1, f_2$  and the boundary conditions are such that  $u_1 = u_2 = \sin(\pi x) \sin(\pi y)$ . We start with the same initial mesh (453 degrees of freedom), and use a global tolerance  $TOL = 0.1$ .

The refined mesh is presented in Figure 5.4b, the adjoint solution in Figure 5.3c, the error values and the effectivity ratios in Table 5.2. The mesh is refined most around the interface  $\Gamma$ , particularly at its centre, where  $|\psi| = \sin(\pi x)$  achieves its maximum value and the weighting of the error is greatest. Refinements are more pronounced in

$\Omega_2$  than in  $\Omega_1$ , as is to be expected given the quantity of interest.

#### 5.1.4.4 Accuracy of the approximate error representation formula

We consider the effect that the accuracy of the adjoint solution has on the accuracy of the approximate quantity of interest. Based on the definition of effectivity ratio (5.14), we compute the error in the representation formula as  $\epsilon = \mathcal{E}(1 - \mathcal{R})$ . We considered the following two problems.

- (i) The  $(QoI)$  is equal to the average value of the solution in the domain, i.e.  $\psi_1 = \psi_2 = 1$  and  $\psi = 0$ . Here  $k_1 = k_2 = 1$ , and  $f_1 = f_2$  and boundary conditions are chosen so that  $u_1 = u_2 = 0.5y(2 - y)$ ,
- (ii) The  $(QoI)$  is the average of the solution  $u_2$  in  $\Omega_2$  and sin-weighted average of the Lagrange multiplier on  $\Gamma$ , i.e.  $\psi_1 = 0$ , with  $\psi_2 = 1$  and  $\psi = -\sin(\pi x)$ . Here  $k_1 = k_2 = 1$  and  $f_1 = f_2$  and boundary conditions are chosen so that  $u_1 = u_2 = \sin(\pi x) \sin(\pi y)$ .

The original problem was solved on a uniform mesh with 453 degrees of freedom. The computed values are presented in Table 5.3. In both examples the error in the representation formula decayed with  $\mathcal{O}(h_{\text{adjoint}}^2)$ , where  $h_{\text{adjoint}} = h/(\text{adjoint refinements})$  is the adjoint mesh size.

|                     | (i) domain error |                         | (ii) mixed error |                         |
|---------------------|------------------|-------------------------|------------------|-------------------------|
| adjoint refinements | $\mathcal{R}$    | $10^5 \cdot  \epsilon $ | $\mathcal{R}$    | $10^4 \cdot  \epsilon $ |
| 2                   | 0.931            | 7.6526                  | 1.217            | -8.1244                 |
| 3                   | 0.983            | 1.9491                  | 1.055            | -2.0425                 |
| 4                   | 0.996            | 0.4873                  | 1.014            | -0.5116                 |
| 5                   | 0.999            | 0.1220                  | 1.003            | -0.1280                 |

Table 5.3: Effectivity ratio and error in the approximate representation formula for the adjoint problem solved on the original mesh refined 2, 3, 4 and 5 times uniformly.

### 5.1.5 Non-matching finite element spaces in $\Omega_1$ and $\Omega_2$

The next issue that we address in this results section is that of the error component associated with the adjoint Lagrange multiplier  $\mu$ . In all the above experiments the computed value of  $\hat{\mathcal{E}}_\mu = \langle \mu^h, u_1^h - u_2^h \rangle_\Gamma = R_\Gamma(\mu^h)$  term in (5.13) was zero up to machine precision, in other words, the continuity condition on the interface was satisfied strongly by the numerical solution. This follows from the conformity of the finite element spaces in the two domains.

The relevant part of the matrix problem (3.4) is the equation

$$-C_1^T U_1 + C_2^T U_2 = 0. \quad (5.16)$$

If the finite element spaces in  $\Omega_1$  and  $\Omega_2$  match on the interface the non-zero block of the coupling matrix  $C_1$  is identical to the non-zero block of  $C_2$ . In this case (5.16) can be rewritten as

$$C_{1,I}^T (U_{2,I} - U_{1,I}) = 0,$$

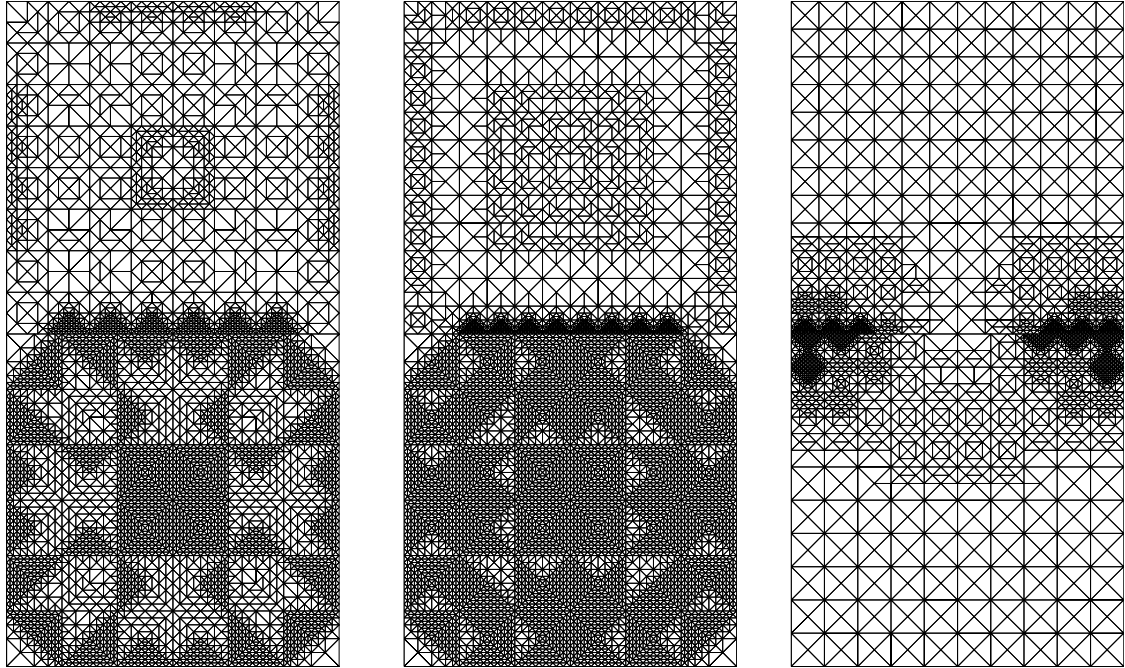
where  $I$  is the set of indices corresponding to interface nodes. This equation has a unique zero solution, leading to strong continuity of the finite element solution:  $U_{1,I} = U_{2,I}$ .

The condition will not be satisfied strongly if the finite element spaces in the two-sub-domains do not match on the interface, and  $C_1 \neq C_2$ . To further illustrate the applicability of the error representation formula we consider a problem where the meshes on the two sub-domains do not conform on  $\Gamma$ , leading to the case just described. Non-conforming meshes are commonly used, for instance, to solve fluid-structure interaction problems, as fluid equations normally require higher resolution than structural equations.

The only different feature of the non-conforming case is in the choice of the finite-dimensional Lagrange multiplier space. Whilst previously the subdivision defining the basis of this space was the projection from either meshes in  $\Omega_1$  or  $\Omega_2$ , it is now the ‘richer’ subdivision, i.e. that coming from a finer mesh.

The problem we consider is as follows:  $k_1 = k_2 = 1$ , and  $f_1 = f_2$  and boundary conditions such that  $u_1 = u_2 = \sin(\pi x) \sin(\pi y)$ . The quantity of interest computed is the average value of the solutions in respective sub-domains (taken with opposite signs to prevent cancellation), i.e.  $\psi_1 = -\psi_2 = 1$  and  $\psi = 0$ .  $TOL = 0.02$  was used as the global tolerance.

Two initial meshes were considered: with respective numbers of degrees of freedom in  $\Omega_1$  and  $\Omega_2$   $DOF_1 = 85$ ,  $DOF_2 = 145$  and  $DOF_1 = 85$ ,  $DOF_2 = 313$ . The refined meshes are shown in Figures 5.5a and 5.5b, and the error estimates  $\hat{\mathcal{E}}$  and  $\hat{\mathcal{E}}_\mu$ , as well as effectivity ratios  $\mathcal{R}$  are given in Table 5.4.



(a)  $\psi_1 = -\psi_2 = 1$ ,  $\psi = 0$ ,  $DOF_1 = 85$ ,  $DOF_2 = 145$  (b)  $\psi_1 = -\psi_2 = 1$ ,  $\psi = 0$ ,  $DOF_1 = 85$ ,  $DOF_2 = 313$  (c)  $\psi_1 = \psi_2 = 0$ ,  $\psi = \cos(\pi x)$ ,  $DOF_1 = 221$ ,  $DOF_2 = 313$

Figure 5.5: Refinements of the initial meshes that do not match on the interface  $\Gamma$ .

The refinement strategy creates more elements in  $\Omega_1$ , where the initial error is higher (see Table 5.4). Note that the resulting mesh in  $\Omega_1$  is more refined when the disparity between the original meshes is more pronounced.

| $DOF_{1,2}$ | $\hat{\mathcal{E}}$ | $10^5 \cdot \hat{\mathcal{E}}_\mu$ | $\mathcal{R}$ | $DOF_{1,2}$ | $\hat{\mathcal{E}}$ | $10^5 \cdot \hat{\mathcal{E}}_\mu$ | $\mathcal{R}$ |
|-------------|---------------------|------------------------------------|---------------|-------------|---------------------|------------------------------------|---------------|
| 85, 145     | 0.011669            | -2.60                              | 0.962         | 85, 313     | 0.009333            | 9.35                               | 0.962         |
| 2917, 1060  | 0.001842            | -1.58                              | 0.970         | 4077, 1229  | 0.001525            | 1.99                               | 0.969         |

Table 5.4: Estimated values of the error for initial non-conforming meshes and corresponding effectivity ratios. The quantity of interest is measured in the domains.  $DOF_{1,2}$  are the numbers of degrees of freedom in each sub-domain.

Next we consider the effect of non-conforming meshes in the case of the quantity of interest on the interface. We take  $k_1 = k_2 = 1$ , and  $f_1 = f_2$  such that  $u_1 = u_2 = \sin(\pi x) \cos(\pi y)$ . We set  $TOL = 0.3$  and use initial meshes with  $DOF_1 = 221$  and  $DOF_2 = 313$ . The quantity of interest is the cos-weighted average error on  $\Gamma$ , with  $\psi_1 = \psi_2 = 0$  and  $\psi = \cos(\pi x)$ . The refined mesh is shown in Figure 5.5c, and the errors,  $\mu$ -components, and effectivity values are given in Table 5.5.

| Weighted average error on $\Gamma$ |                     |                              |               |
|------------------------------------|---------------------|------------------------------|---------------|
| $DOF_{1,2}$                        | $\hat{\mathcal{E}}$ | $10^5 \cdot \mathcal{E}_\mu$ | $\mathcal{R}$ |
| 221, 313                           | -0.005615           | -6.48                        | 1.015         |
| 1089, 937                          | -0.001714           | 6.68                         | 1.043         |

Table 5.5: Estimated values of the error for initial non-conforming meshes and corresponding effectivity ratios. The quantity of interest is measured on the interface.  $DOF_1$  and  $DOF_2$  are the numbers of degrees of freedom in each sub-domain.

Again, the sub-domain  $\Omega_1$ , which originally has a coarser mesh, is refined more by the algorithm in order to equilibrate error contributions. The centre of the interface is not refined because the weighting function is close to zero around  $x = 0.5$  on  $\Gamma$ . The absolute value of  $\mathcal{E}_\mu$  does not decrease with refinement, since it is not the leading order term in the total error.

An alternative case where different finite element spaces in  $\Omega_1$  and  $\Omega_2$  can arise

is the coupling of different order elements. As an example, we solve a problem using piecewise-quadratic basis functions in  $\Omega_1$  and piecewise-linear basis functions in  $\Omega_2$ .

As before, we consider two types of quantities of interest. First, we consider a  $(QoI)$  in the whole domain by setting  $\psi_1 = 1$ ,  $\psi_2 = -1$  and  $\psi = 0$ . For  $k_1 = k_2 = 1$ , we choose  $f_1 = f_2$  and boundary conditions so that  $u_1 = u_2 = \sin(\pi x) \sin(\pi y)$ . Second, we consider a  $(QoI)$  on the interface by setting  $\psi_1 = \psi_2 = 0$  and  $\psi = \cos(\pi x)$  for  $k_1 = k_2 = 1$ , and choosing  $f_1 = f_2$  and boundary conditions so that  $u_1 = u_2 = \cos(\pi x) \sin(\pi y)$ . The initial meshes in  $\Omega_1$  and  $\Omega_2$  are identical.

Since higher order elements are utilised in  $\Omega_1$ , we expect the error there to be significantly lower than in  $\Omega_2$ , and for the meshes to be refined accordingly. Indeed, it can be seen from Figure 5.6a that in both problems the only refinements in  $\Omega_1$  occur near the interface, where the errors are high due to non-matching finite element spaces. In contrast, the number of degrees of freedom in  $\Omega_2$  rises by a factor of  $\sim 10$ . The errors and corresponding effectivity ratios for the two problems are given in Table 5.6.

| Average error in $\Omega_1 \cup \Omega_2$ |                     |                              |               | Weighted average error on $\Gamma$ |                     |                              |               |
|-------------------------------------------|---------------------|------------------------------|---------------|------------------------------------|---------------------|------------------------------|---------------|
| $DOF_{1,2}$                               | $\hat{\mathcal{E}}$ | $10^5 \cdot \mathcal{E}_\mu$ | $\mathcal{R}$ | $DOF_{1,2}$                        | $\hat{\mathcal{E}}$ | $10^4 \cdot \mathcal{E}_\mu$ | $\mathcal{R}$ |
| 841, 221                                  | 0.002681            | 4.40                         | 0.963         | 841,221                            | 0.003428            | -9.40                        | 0.975         |
| 1447, 2265                                | 0.000485            | 0.09                         | 1.153         | 1239, 1968                         | 0.000110            | -3.43                        | 0.831         |

Table 5.6: Estimated errors for initial non-conforming meshes and corresponding effectivity ratios.  $DOF_1$  and  $DOF_2$  are the numbers of degrees of freedom in each sub-domain.

### 5.1.5.1 Error-based vs uniform refinements

As a final numerical example of error estimation and mesh refinement in steady coupled problems we consider a problem with a solution that has large curvature localised within a small region inside the domain. In this case local refinement based

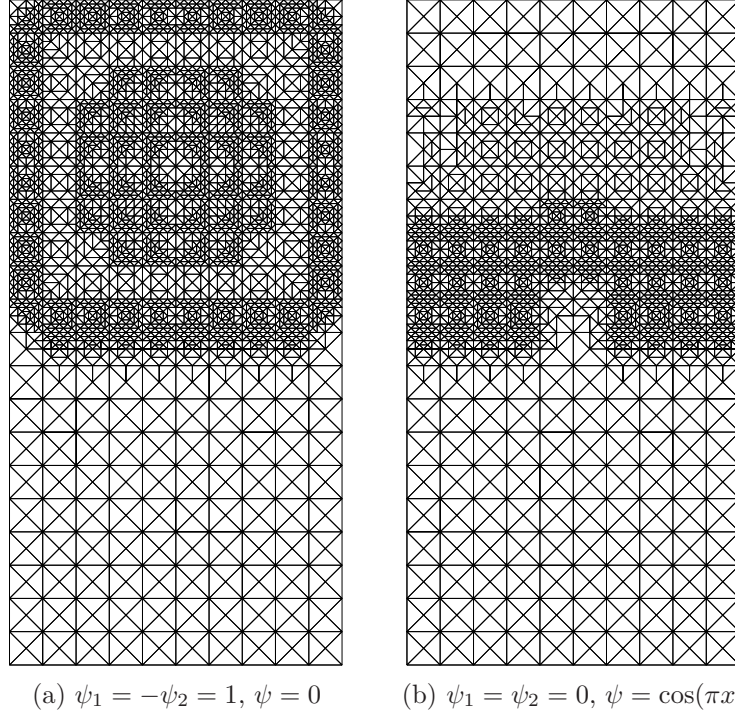


Figure 5.6: Refinements of the meshes for piecewise-quadratic finite element functions in  $\Omega_1$  and piecewise-linear finite element functions in  $\Omega_2$ .

on local error estimates should be far more efficient than uniform refinement.

We choose  $k_i = 1$ ,  $f_i$ ,  $i = 1, 2$  so that  $u_1 = u_2 = \exp(-a(x - b)^2 - a(y - c)^2)$ , which is a Gaussian centered at  $(x, y) = (b, c)$ . If  $a \gg 1$  the solution undergoes a rapid change around the peak. The quantity of interest is the sin-weighted average error on the interface, i.e. we take  $\psi_1 = \psi_2 = 0$  and  $\psi = \sin(\pi x)$ .

By locating the center of the Gaussian near the interface we ensure that it affects the interface errors. We first consider  $a = 100$ ,  $b = 0.25$ ,  $c = 1.25$ . The initial number of degrees of freedom in the mesh is  $DOF = 453$  and the global tolerance is  $TOL = 0.005$ . The error-based refinement algorithm produces the mesh shown in Figure 5.7a and the error values in Table 5.7. For comparison we show the number of degrees of freedom in a uniform mesh required to obtain a similarly sized error. This amounted to refining the whole mesh three times, while the error-based algorithm selects the



area where additional elements are necessary – a circle around  $(x, y) = (0.25, 1.25)$ .

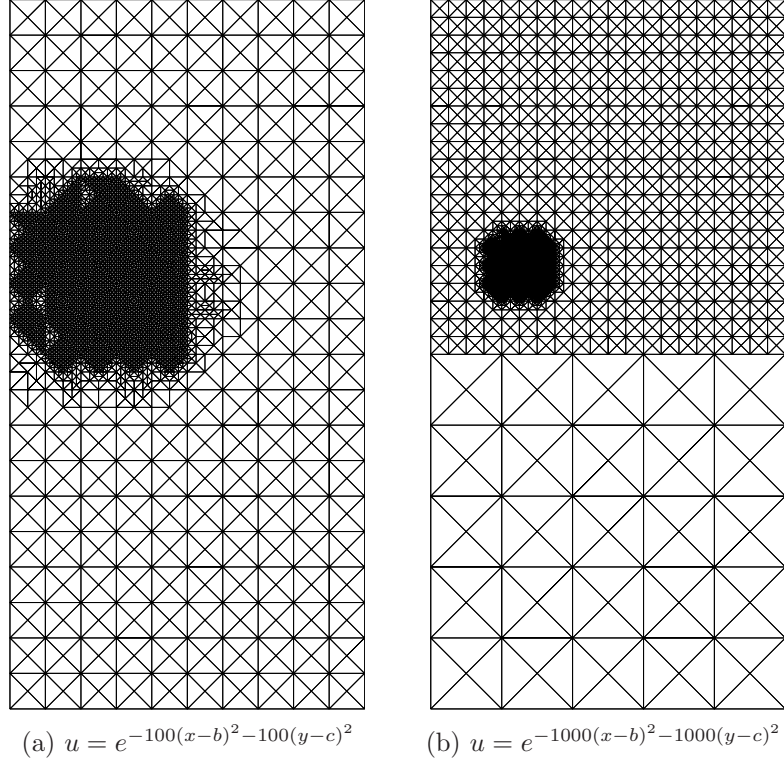


Figure 5.7: Refined meshes obtained using the error-representation formula for the error on the interface  $\Gamma$ .

We then solve the problem with  $a = 1000$ , which makes the Gaussian more localised and more difficult to approximate in the original finite element space. We start with an initial mesh that is finer in  $\Omega_2$ , where the peak resides. The refined mesh is presented in Figure 5.7b, and the error values in Table 5.7, along with the size of the uniform mesh (in each sub-domain) required to obtain similar errors.

|            | $DOF_{in}$ | $\hat{\mathcal{E}}$ | $DOF_u$ | $\hat{\mathcal{E}}$ | $DOF_e$ | $\hat{\mathcal{E}}$ |
|------------|------------|---------------------|---------|---------------------|---------|---------------------|
| $a = 100$  | 453        | -0.000416           | 26003   | -0.000006           | 3175    | 0.000006            |
| $a = 1000$ | 923        | -0.003119           | 54963   | -0.000050           | 2778    | -0.000046           |

Table 5.7: Estimated errors in the quantity of interest on the interface before and after refinement of the mesh.  $DOF_{in}$ , uniform  $DOF_u$  and error-based  $DOF_e$  are the total numbers of degrees of freedom in the initial mesh, the mesh refined uniformly, and the mesh produced by our refinement strategy. Parameter  $a$  regulates the solution shape, namely the diameter of the peak.

In both scenarios refinements based on local errors provide a clear advantage in terms of the size of the problem that needs to be solved to obtain errors of the same order. Of course, in practical terms the method requires that the adjoint problem is solved to higher accuracy than the original problem, thus incurring an additional computational expense. It is important to note that the main benefit of the method comes from producing an optimal mesh (with respect to a given quantity of interest), allowing the best possible use of available computational resources. The solution computed on a locally refined mesh using the total  $DOF$  degrees of freedom is expected to produce a more accurate quantity of interest than the solution computed on a uniform mesh with  $DOF$  degrees of freedom.

## 5.2 Time-dependent problems

The fundamental ideas behind deriving the error representation formula for time-dependent coupled problems are similar to those described above. However, introducing temporal variation gives rise to additional complications. The adjoint problem has to be defined backwards in time from a prescribed condition at  $T$  to 0. The order of accuracy of the adjoint solution must exceed that of the original solution not only in space, but also in time. In this section these and some other issues are addressed in detail.

Recall the formulation of the weak parabolic problem (3.6) from Section 3.4.1:

*Find  $(u_1, u_2, \lambda) \in X_{1,D} \times X_{2,D} \times Y$  such that*

$$\int_0^T \left\{ \mp \langle \lambda, v_i \rangle + \left( \frac{\partial u_i}{\partial t}, v_i \right) + a_i(u_i, v_i) \right\} dt$$

$$= \int_0^T \left\{ (f_i, v_i) + \langle g_{i,N}, v_i \rangle_{\Gamma_i^N} \right\} dt \quad \forall v_i \in X_{i,0}, \quad i = 1, 2, \quad (5.17a)$$

$$\int_0^T \langle \nu, u_1 - u_2 \rangle dt = 0 \quad \forall \nu \in Y, \quad (5.17b)$$

$$u_i|_{t=0} = u_{i,0}, \quad i = 1, 2. \quad (5.17c)$$

In this section we omit the subscripts  $\Gamma$  and  $\Omega_i$ ,  $i = 1, 2$  on  $\langle \cdot, \cdot \rangle$  and  $(\cdot, \cdot)$ , and it is always assumed that the appropriate inner product is used. As before, for simplicity we assume that  $g_{i,N} = 0$  for  $i = 1, 2$ .

In addition to the finite element spaces defined in Section 3.4.2, which are used to discretise this problem, we introduce semi-discrete sub-spaces of  $X_1$ ,  $X_2$  and  $Y$ , and the projections  $\pi^h$  and  $\theta$  onto these semi-discrete sub-spaces:

$$X_i^x = \{v_i \in X_i \mid v_i(\cdot, t) \in P_c^k(\tau_i) \quad \forall t \in \mathcal{T}\}, \quad i = 1, 2,$$

$$Y^x = \{\nu \in Y \mid \nu(\cdot, t) \in P_c^k(\tau) \quad \forall t \in \mathcal{T}\},$$

$$X_i^t = \{v \in X_i \mid v_i(x, \cdot) \in P_d^q(\{\mathcal{T}_n\}) \quad \forall x \in \Omega_i\}, \quad i = 1, 2,$$

$$Y^t = \{\nu \in Y \mid \nu(x, \cdot) \in P_d^q(\{\mathcal{T}_n\}) \quad \forall s \in \Gamma\},$$

$$\pi^h : \begin{cases} X_i \rightarrow X_i^x & \text{such that } \forall v \in X_i, w \in X_i^x : (v - \pi^h v, w) = 0, \quad i = 1, 2, \\ Y \rightarrow Y^x & \text{such that } \forall \nu \in Y, w \in Y_j^x : \langle \nu - \pi^h \nu, w \rangle = 0, \end{cases}$$

$$\theta : \begin{cases} X_i \rightarrow X_i^t & \text{such that } \forall v \in X_i, w \in X_i^t : (v - \theta v, w) = 0, \quad i = 1, 2, \\ Y \rightarrow Y^t & \text{such that } \forall \nu \in Y, w \in X_j^t : \langle \nu - \theta \nu, w \rangle = 0, \end{cases}$$

where  $j$  is the index of the domain that defines the finite element space on  $\Gamma$  (i.e. that where the mesh in the domain coincides with the mesh on the interface).

The finite element method for (5.17) can be written as follows:

Find  $(u_1^h, u_2^h, \lambda^h) \in X_{1,D}^h \times X_{2,D}^h \times Y^h$  with  $u_{i,0-}^h = \pi^h u_{i,0}$ ,  $i = 1, 2$ , such that

$$\sum_{n=1}^N \left\{ \left( [u_i^h]_{n-1}, v_{i,(n-1)+}^h \right) - \int_{I_n} (R_i, v_i^h) \, dt \right\} = 0 \quad \forall v_i^h \in X_{i,0}^h, \quad i = 1, 2, \quad (5.18a)$$

$$\sum_{n=1}^N \int_{I_n} \langle \nu^h, u_1^h - u_2^h \rangle \, dt = 0 \quad \forall \nu^h \in Y^h, \quad (5.18b)$$

where  $R_i$  denote the residuals

$$(R_i, v_i) = \pm \langle \lambda^h, v_i \rangle + (f_i, v_i) - \left( \frac{\partial u_i^h}{\partial t}, v_i \right) - a_i(u_i^h, v_i) \quad \forall v_i \in X_i^h, \quad i = 1, 2. \quad (5.19)$$

### 5.2.1 The error representation formula

Let the discretisation errors in the domains and on the interface be defined respectively as

$$e_1 = u_1 - u_1^h, \quad e_2 = u_2 - u_2^h, \quad \text{and} \quad e_\lambda = \lambda - \lambda^h.$$

These are now functions of both space and time. The quantity of interest

$$(QoI) = \int_0^T \{ - \langle \lambda, \psi \rangle + (u_1, \psi_1) + (u_2, \psi_2) \} \, dt, \quad (5.20)$$

is a linear functional of the solution with coefficients  $\psi_1$ ,  $\psi_2$  and  $\psi$ , which is averaged over the time interval  $[0, T]$ . The error in the  $(QoI)$  is defined as

$$\mathcal{E} = \int_0^T \{ - \langle e_\lambda, \psi \rangle + (e_1, \psi_1) + (e_2, \psi_2) \} \, dt. \quad (5.21)$$

The adjoint problem for (5.17) corresponding to this quantity of interest is:

Find  $(\phi_1, \phi_2, \mu) \in X_{1,0} \times X_{2,0} \times Y$  such that

$$\int_0^T \left\{ \mp \langle \mu, v_i \rangle - \left( \frac{\partial \phi_i}{\partial t}, v_i \right) + a_i(\phi_i, v_i) - (\psi_i, v_i) \right\} dt = 0$$

$$\forall v_i \in X_{i,0}, i = 1, 2, \quad (5.22a)$$

$$\int_0^T \langle \rho, \phi_1 - \phi_2 \rangle dt = \int_0^T \langle \rho, \psi \rangle dt \quad \forall \rho \in Y, \quad (5.22b)$$

$$\phi_i(\cdot, T) = 0, \text{ in } \Omega_i, i = 1, 2. \quad (5.22c)$$

As in the steady case, we present the derivation of the error representation formula as a theorem.

**Theorem 8.** *The error  $\mathcal{E}$  defined by (5.21) can be computed as*

$$\mathcal{E} = \sum_{i=1,2} \left\{ (\phi_{i,0}, u_{i,0} - \pi^h u_{i,0}) - \sum_{n=2}^N (\phi_{i,n-1}, [u_i^h]_{n-1}) \right. \\ \left. + \sum_{n=1}^N \int_{I_n} \{ \pm \langle \mu, u_i^h \rangle + (R_i, \phi_i) \} dt \right\}, \quad (5.23)$$

where  $(\phi_1, \phi_2, \mu)$  is the solution of the adjoint problem (5.22), and the residuals  $R_i$ ,  $i = 1, 2$ , are defined in (5.19).

*Proof.* We use the errors  $e_1$ ,  $e_2$  and  $e_\lambda$  as test functions in the weak formulation of the adjoint problem (5.22a) and (5.22b), yielding

$$\sum_{n=1}^N \int_{I_n} \left\{ \mp \langle \mu, u_i - u_i^h \rangle - \left( \frac{\partial \phi_i}{\partial t}, u_i - u_i^h \right) + a_i(\phi_i, u_i - u_i^h) \right\} dt = \sum_{n=1}^N \int_{I_n} (\psi_i, e_i) dt,$$

$$i = 1, 2,$$

$$\sum_{n=1}^N \int_{I_n} \{ - \langle \lambda, \phi_1 - \phi_2 \rangle + \langle \lambda^h, \phi_1 - \phi_2 \rangle \} dt = - \sum_{n=1}^N \int_{I_n} \langle e_\lambda, \psi \rangle dt.$$

Adding these equations together, integrating by parts in time, and recalling the resi-

dual definition (5.19) and the piecewise constant dependence of the finite-element solution in time, we obtain

$$\begin{aligned} \mathcal{E} = & \sum_{n=1}^N \int_{I_n} \{ - \langle \mu, u_1 - u_2 \rangle + \langle \mu, u_1^h - u_2^h \rangle \} dt - \sum_{i=1,2} \sum_{n=1}^N (\phi_i, u_i - u_i^h) \Big|_{t_{n-1}^+}^{t_n^-} \\ & + \sum_{i=1,2} \sum_{n=1}^N \int_{I_n} \left\{ \mp \langle \lambda, \phi_i \rangle + \left( \frac{\partial u_i}{\partial t}, \phi_i \right) + a_i(u_i, \phi_i) + (R_i, \phi_i) - (f_i, \phi_i) \right\} dt. \end{aligned}$$

We can also use  $\phi_1$ ,  $\phi_2$  and  $\mu$  as test functions in the weak formulation of the original problem (5.17), resulting in

$$\begin{aligned} \int_0^T \left\{ \mp \langle \lambda, \phi_i \rangle - (f_i, \phi_i) + \left( \frac{\partial u_i}{\partial t}, \phi_i \right) + a_i(u_i, \phi_i) \right\} dt &= 0, \quad i = 1, 2, \\ - \int_0^T \langle \mu, u_1 - u_2 \rangle dt &= 0. \end{aligned}$$

Subtracting both expressions from the above formula for  $\mathcal{E}$ ,

$$\mathcal{E} = \sum_{i=1,2} \sum_{n=1}^N \left\{ - (\phi_i, u_i - u_i^h) \Big|_{t_{n-1}^+}^{t_n^-} + \int_{I_n} \{ \pm \langle \mu, u_i^h \rangle + (R_i, \phi_i) \} dt \right\}. \quad (5.24)$$

We can re-write the term due to jumps in the solution, assuming the exact solutions to the original and to the adjoint problems are continuous in time (within their respective subdomains), and using the initial and final conditions (5.17c) and (5.22c). Noting that

$$\sum_{n=1}^N (\phi_i, u_i - u_i^h) \Big|_{t_{n-1}^+}^{t_n^-} = -(\phi_{i,0}, u_{i,0}) - \sum_{n=1}^N (\phi_i, u_i^h) \Big|_{t_{n-1}^+}^{t_n^-},$$

the error measure can be written as

$$\mathcal{E} = \sum_{i=1,2} \left\{ (\phi_{i,0}, u_{i,0}) + \sum_{n=1}^N (\phi_i, u_i^h) \Big|_{t_{n-1}^+}^{t_n^-} + \sum_{n=1}^N \int_{I_n} \{ \pm \langle \mu, u_i^h \rangle + (R_i, \phi_i) \} dt \right\}.$$

Using the initial conditions  $u_{i,0+}^h = \pi u_{i,0}$  and  $\phi_{i,N} = 0$  to rearrange the first two terms we obtain the representation formula (5.23),

$$\mathcal{E} = \sum_{i=1,2} \left\{ (\phi_{i,0}, u_{i,0} - \pi u_{i,0}) - \sum_{n=2}^N (\phi_{i,n-1}, [u_i^h]_{n-1}) + \sum_{n=1}^N \int_{I_n} \{ \pm \langle \mu, u_i^h \rangle + (R_i, \phi_i) \} dt \right\}.$$

□

For convenience in writing we identify

$$\begin{aligned} \mathcal{E}_0 &= \sum_{i=1,2} (\phi_{i,0}, u_{i,0} - \pi^h u_{i,0}) && \text{-- the initial error,} \\ \mathcal{E}_J &= - \sum_{i=1,2} \sum_{n=2}^N (\phi_{i,n-1}, [u_i^h]_{n-1}) && \text{-- the error due to solution jumps in time} \\ \mathcal{E}_\Gamma &= \sum_{n=1}^N \int_{I_n} \langle \mu, u_1^h - u_2^h \rangle dt && \text{-- the interface error,} \\ \mathcal{E}_\Omega &= \sum_{i=1,2} \sum_{n=1}^N \int_{I_n} (R_i, \phi_i) dt && \text{-- the domain error.} \end{aligned}$$

The sum of these errors makes up the total error in the quantity of interest:

$$\mathcal{E} = \mathcal{E}_0 + \mathcal{E}_J + \mathcal{E}_\Gamma + \mathcal{E}_\Omega.$$

**Remark 4.** *The strong form of the adjoint problem is as follows:*

$$\begin{aligned} \frac{\partial \phi_i}{\partial t} - \nabla \cdot (k_i \nabla \phi_i) + c_i \phi_i &= \psi_i \quad \text{in } \Omega_i \times (0, T], \quad i = 1, 2, \\ \phi_i &= 0 \quad \text{on } \Gamma_i^D \times (0, T], \quad i = 1, 2, \\ k_i \nabla \phi_i \cdot \mathbf{n}_i &= 0 \quad \text{on } \Gamma_i^N \times (0, T], \quad i = 1, 2, \end{aligned}$$

$$\begin{aligned}
\phi_1 - \phi_2 &= \psi \quad \text{on } \Gamma \times (0, T], \\
k_1 \nabla \phi_1 \cdot \mathbf{n}_1 + k_2 \nabla \phi_2 \cdot \mathbf{n}_2 &= 0 \quad \text{on } \Gamma \times (0, T], \\
\phi_i &= 0 \quad \text{in } \Omega_i \times \{T\}, \quad i = 1, 2.
\end{aligned}$$

### 5.2.2 The effect of discretization error on the boundary

The proof of Theorem 8 utilised errors  $e_1$  and  $e_2$  as test functions in the adjoint problem. This is permissible only if the boundary conditions lie in the appropriate finite element spaces, i.e., if  $u_i^D(x, t) \in X_{i,D}^h, i = 1, 2$ . In particular, the proof of Theorem 8 assumes that  $u_i^D(x, t)$  is piecewise-constant in time.

In order to account for the circumstances when  $u_i^D(x, t) \notin X_{i,D}^h, i = 1, 2$ , we introduce functions  $e_i^D = u_i^D - \pi^h \theta u_i^D$  on  $\Gamma_i^D, i = 1, 2$  for all  $t \in [0, T]$ . We extend this function to the whole domain  $\Omega_1 \cup \Omega_2$  by setting it to zero at all mesh points other than those on the Dirichlet boundaries, and allowing it to be piecewise linear in space. Then we can write

$$\int_0^T (\psi_i, e_i) \, dt = \int_0^T (\psi_i, e_i - e_i^D) \, dt + \int_0^T (\psi_i, e_i^D) \, dt.$$

Now,  $e_i - e_i^D$  is a suitable test function for the adjoint. Substituting for the first term on the right-hand side from (5.22a),

$$\begin{aligned}
\int_0^T (\psi_i, e_i) \, dt &= \int_0^T \left\{ \mp \langle \mu, e_i - e_i^D \rangle - \left( \frac{\partial \phi_i}{\partial t}, e_i - e_i^D \right) + a_i(\phi_i, e_i - e_i^D) \right\} \, dt \\
&\quad + \int_0^T (\psi_i, e_i^D) \, dt.
\end{aligned}$$



This introduces an additional error term in the error representation formula, namely

$$\mathcal{E} = \mathcal{E}_0 + \mathcal{E}_\Gamma + \mathcal{E}_J + \mathcal{E}_\Omega + \mathcal{E}_D,$$

with

$$\mathcal{E}_D = \sum_{i=1,2} \sum_{n=1}^N \int_{I_n} \left\{ \pm \langle \mu, e_i^D \rangle + \left( \frac{\partial \phi_i}{\partial t}, e_i^D \right) - a_i(\phi_i, e_i^D) + (\psi_i, e_i^D) \right\} dt. \quad (5.26)$$

**Remark 5.** *In a practical problem Dirichlet boundary conditions are likely to be based on discrete data, rather than given in functional form. Such boundary conditions will automatically reside in the finite element space. Unless stated explicitly, we do not include the boundary error component in the calculations below.*

### 5.2.3 The effect of adjoint approximation

As in the steady case, the derivation of the error representation formula (5.23) was based on the assumption that the exact adjoint solution was available. We will now consider the effect of using a finite element approximation to the adjoint solution in the calculation of the quantity of interest.

Let the adjoint finite element spaces  $\hat{X}_1$ ,  $\hat{X}_2$  and  $\hat{Y}$  be such that  $X_i^h \subset \hat{X}_i^h \subset X_i$  and  $Y^h \subset \hat{Y}^h \subset Y$ , and let  $\hat{X}_{i,0}^h = \hat{X}_i^h \cap X_{i,0}$ ,  $i = 1, 2$ . Let the finite element method used to solve the adjoint problem be as follows:

*Find  $(\phi_1^h, \phi_2^h, \mu^h) \in \hat{X}_{1,0}^h \times \hat{X}_{2,0}^h \times \hat{Y}^h$  such that*

$$\begin{aligned} \sum_{n=1}^N \left( -([\phi_i^h]_n, v_{i,n-}^h) + \int_{I_n} \left\{ \mp \langle \mu^h, v_i^h \rangle - \left( \frac{\partial \phi_i^h}{\partial t}, v_i^h \right) + a_i(\phi_i^h, v_i^h) \right\} dt \right) \\ = \sum_{n=1}^N \int_{I_n} (\psi_i, v_i^h) dt \quad \forall v_i^h \in \hat{X}_{i,0}^h, \quad i = 1, 2, \end{aligned} \quad (5.27a)$$

$$\sum_{n=1}^N \int_{I_n} \langle \rho^h, \phi_1^h - \phi_2^h \rangle \, dt = \sum_{n=1}^N \int_{I_n} \langle \rho^h, \psi \rangle \, dt \quad \forall \rho^h \in \hat{Y}^h, \quad (5.27b)$$

$$\phi_i^h(\cdot, T) = 0 \quad \text{in } \Omega_i, \quad i = 1, 2. \quad (5.27c)$$

**Remark 6.** *The partition of the time interval  $I$  in the above finite element formulation has to remain the same, since a finer partition would introduce additional jump terms in the derivation below. In practice higher temporal accuracy of the adjoint solution is achieved by employing a higher order method, namely  $dG(1)$ , for time stepping. At the same time, higher spatial accuracy is achieved by uniform mesh refinement.*

**Theorem 9.** *Let  $(u_1^h, u_2^h, \lambda^h)$  be the finite element solution of (5.18), and  $(\phi_1^h, \phi_2^h, \mu^h)$  be the finite element solution of (5.27). Then the error in the quantity of interest defined by (5.23) can be represented as a sum of the estimated error  $\mathcal{E}_{est}$  and a remainder  $\epsilon$ , i.e.,*

$$\mathcal{E} = \mathcal{E}_{est} + \epsilon, \quad (5.28)$$

where

$$\mathcal{E}_{est} = \sum_{i=1,2} \left\{ (\phi_{i,0+}^h, u_{i,0} - \pi^h u_{i,0}) - \sum_{n=2}^N (\phi_{i,(n-1)+}^h, [u_i^h]_{n-1}) + \sum_{n=1}^N \int_{I_n} \{ \pm \langle \mu^h, u_i^h \rangle + (R_i, \phi_i^h) \} \, dt \right\}, \quad (5.29)$$

$$\epsilon = \sum_{i=1,2} \left\{ (\phi_{i,0} - \phi_{i,0+}^h, u_{i,0}) + \sum_{n=1}^N \int_{I_n} (f_i, \phi_i - \phi_i^h) \, dt \right\}. \quad (5.30)$$

*Proof.* Since the spaces  $X_i^h \subset \hat{X}_i$ ,  $i=1,2$ , and  $Y^h \subset \hat{Y}$  are nested, we can use the original finite element solution as the test function in the adjoint finite element problem:

$$\begin{aligned} \sum_{i=1,2} \sum_{n=1}^N \left( -([\phi_i^h]_n, u_{i,n-}^h) + \int_{I_n} \left\{ \mp \langle \mu^h, u_i^h \rangle - \left( \frac{\partial \phi_i^h}{\partial t}, u_i^h \right) + a_i(\phi_i^h, u_i^h) \right\} dt \right) \\ = \sum_{i=1,2} \sum_{n=1}^N \int_{I_n} (\psi_i, u_i^h) dt, \\ \sum_{n=1}^N \int_{I_n} \langle \lambda^h, \phi_1^h - \phi_2^h \rangle dt = \sum_{n=1}^N \int_{I_n} \langle \lambda^h, \psi \rangle dt. \end{aligned}$$

Galerkin orthogonality then gives

$$\begin{aligned} \sum_{i=1,2} \sum_{n=1}^N \left( ([\phi_i^h]_n, u_{i,n-}^h) + \int_{I_n} \left\{ \mp \langle \mu - \mu^h, u_i^h \rangle - \left( \frac{\partial(\phi_i - \phi_i^h)}{\partial t}, u_i^h \right) \right. \right. \\ \left. \left. + a_i(\phi_i - \phi_i^h, u_i^h) \right\} dt \right) = 0, \\ - \sum_{n=1}^N \int_{I_n} \langle \lambda^h, \phi_1 - \phi_2 - \phi_1^h + \phi_2^h \rangle dt = 0. \end{aligned}$$

We note that, by the divergence theorem,

$$- \int_{I_n} \left( \frac{\partial(\phi_i - \phi_i^h)}{\partial t}, u_i^h \right) dt = - (\phi_i - \phi_i^h, u_i^h) \Big|_{t_{n-1}^+}^{t_n^-} + \int_{I_n} \left( \frac{\partial u_i^h}{\partial t}, \phi_i - \phi_i^h \right) dt,$$

and using the definition of the residual (5.19) we obtain,

$$\begin{aligned} \int_{I_n} \left\{ - \left( \frac{\partial(\phi_i - \phi_i^h)}{\partial t}, u_i^h \right) dt + a_i(\phi_i - \phi_i^h, u_i^h) \right\} = - (\phi_i - \phi_i^h, u_i^h) \Big|_{t_{n-1}^+}^{t_n^-} \\ + \int_{I_n} \{ \pm \langle \lambda^h, \phi_i - \phi_i^h \rangle + (f_i, \phi_i - \phi_i^h) - (R_i, \phi_i - \phi_i^h) \} dt \quad \forall n = 1, 2, \dots, N. \end{aligned}$$

Combining,

$$\begin{aligned} \sum_{i=1,2} \sum_{n=1}^N \left\{ ([\phi_i^h]_n, u_{i,n-}^h) - (\phi_i - \phi_i^h, u_i^h) \Big|_{t_{n-1}^+}^{t_n^-} \right. \\ \left. + \int_{I_n} (\mp \langle \mu - \mu^h, u_i^h \rangle + (f_i, \phi_i - \phi_i^h) - (R_i, \phi_i - \phi_i^h)) \, dt \right\} = 0. \end{aligned}$$

Adding this expression to the error representation formula (5.24), we have

$$\begin{aligned} \mathcal{E} = \sum_{i=1,2} \sum_{n=1}^N \left\{ -(\phi_i, u_i - u_i^h) \Big|_{t_{n-1}^+}^{t_n^-} + ([\phi_i^h]_n, u_{i,n-}^h) - (\phi_i - \phi_i^h, u_i^h) \Big|_{t_{n-1}^+}^{t_n^-} \right\} \\ + \sum_{i=1,2} \sum_{n=1}^N \int_{I_n} \left\{ \pm \langle \mu^h, u_i^h \rangle + (R_i, \phi_i^h) + (f_i, \phi_i - \phi_i^h) \right\} \, dt. \quad (5.31) \end{aligned}$$

Assuming the exact solutions are continuous in time, and using the initial conditions, we can re-arrange the terms in the first sum as

$$\begin{aligned} \sum_{n=1}^N \left\{ -(\phi_i, u_i - u_i^h) \Big|_{t_{n-1}^+}^{t_n^-} + ([\phi_i^h]_n, u_{i,n-}^h) - (\phi_i - \phi_i^h, u_i^h) \Big|_{t_{n-1}^+}^{t_n^-} \right\} \\ = (\phi_{i,0+}^h, u_{i,0} - \pi^h u_{i,0}) + (\phi_{i,0} - \phi_{i,0+}^h, u_{i,0}) - \sum_{n=2}^N \left( \phi_{i,(n-1)+}^h, [u_i^h]_{n-1} \right). \end{aligned}$$

Substituting this expression into (5.31), and using definitions (5.29)-(5.30), we can write the exact error as the sum of the approximate error in the quantity of interest as computed using the finite element approximation to the adjoint solution and the error in the representation formula, specifically  $\mathcal{E} = \mathcal{E}_{est} + \epsilon$ .  $\square$

The value of  $\epsilon$  depends on the exact adjoint solution and is not computable. However, it can be bounded *a priori* if  $f_i$  are sufficiently regular. For the numerical examples considered below we ascertain that the remainder  $\epsilon$  is negligible provided the approximate solution to the adjoint problem is sufficiently accurate.

### 5.2.4 An adaptive strategy

Our goal is to use the approximate error representation formula (5.29) in order to obtain a mesh which is well-suited for a given problem and which achieves a specified error tolerance in a given quantity of interest. We consider the projection of the approximate adjoint solution onto the original finite element space  $(\pi^h \theta \phi_1^h, \pi^h \theta \phi_2^h, \pi^h \theta \mu^h)$ , which can be used as a test function in the original finite element problem, and therefore satisfies

$$\begin{aligned} & \sum_{i=1,2} \sum_{n=1}^N \left\{ \left( \pi^h \theta \phi_{i,(n-1)+}^h, [u_i^h]_{n-1} \right) - \int_{I_n} (R_i, \pi^h \theta \phi_i^h) \, dt \right\} = 0, \\ & - \sum_{n=1}^N \int_{I_n} \langle \pi^h \theta \mu^h, u_1^h - u_2^h \rangle \, dt = 0. \end{aligned}$$

Adding these equations together and rearranging the terms to resemble the approximate error representation formula (5.29), we have

$$\sum_{i=1,2} \sum_{n=1}^N \left\{ \left( \pi^h \theta \phi_{i,(n-1)+}^h, [u_i^h]_{n-1} \right) - \int_{I_n} \left\{ \pm \langle \pi^h \theta \mu^h, u_i^h \rangle + (R_i, \pi^h \theta \phi_i^h) \right\} \, dt \right\} = 0.$$

Adding this expression to the representation formula (5.29), while also adding and subtracting the term

$$\sum_{i=1,2} \sum_{n=1}^N \left\{ \left( \theta \phi_{i,(n-1)+}^h, [u_i^h]_{n-1} \right) - \int_{I_n} \left\{ \pm \langle \theta \mu^h, u_i^h \rangle + (R_i, \theta \phi_i^h) \right\} \, dt \right\},$$

yields

$$\mathcal{E}_{est} = \sum_{i=1,2} \left\{ \left( \phi_{i,0+}^h, u_{i,0} - \pi^h u_{i,0} \right) - \sum_{n=1}^N \left( \phi_{i,(n-1)+}^h - \theta \phi_{i,(n-1)+}^h, [u_i^h]_{n-1} \right) \right\},$$

$$\begin{aligned}
& - \sum_{n=1}^N \left( \theta \phi_{i,(n-1)+}^h - \pi^h \theta \phi_{i,(n-1)+}^h, [u_i^h]_{n-1} \right) + \sum_{n=1}^N \int_{I_n} \{ \pm < \mu^h - \theta \mu^h, u_i^h > \} \, dt \\
& + \sum_{n=1}^N \int_{I_n} \{ \pm < \theta \mu^h - \pi^h \theta \mu^h, u_i^h > + (R_i, \phi_i^h - \theta \phi_i^h) + (R_i, \theta \phi_i^h - \pi^h \theta \phi_i^h) \} \, dt \Big\}.
\end{aligned}$$

This form of the error measure allows us to distinguish between the effects of approximation in time and in space. We group all terms involving  $\phi^h - \theta \phi^h$  to produce the temporal error and all terms involving  $\theta \phi^h - \pi^h \theta \phi^h$  to produce the spatial error, and write the total error as the sum of these two components. Clearly, the initial error is part of the spatial approximation error. We compute the contributions to the temporal error at each time step  $I_n$ , and the contributions to the spatial error on each element  $\Delta$  at each time step  $I_n$  in the following manner:

$$\mathcal{E}_{est} = \mathcal{E}_t + \mathcal{E}_x,$$

where

$$\begin{aligned}
\mathcal{E}_t &= \sum_{n=1}^N \mathcal{E}_t^n = \sum_{n=1}^N \left( \sum_{i=1,2} \sum_{\Delta_i \in \tau_i} \mathcal{E}_t^{n,\Delta_i} + \sum_{\gamma \in \tau} \mathcal{E}_t^{n,\gamma} \right), \\
\mathcal{E}_x &= \sum_{i=1,2} \sum_{\Delta_i \in \tau_i} \mathcal{E}_0^{\Delta_i} + \sum_{n=1}^N \left( \sum_{i=1,2} \sum_{\Delta_i \in \tau_i} \mathcal{E}_x^{n,\Delta_i} + \sum_{\gamma \in \tau} \mathcal{E}_x^{n,\gamma} \right),
\end{aligned}$$

$$\begin{aligned}
\mathcal{E}_t^{n,\Delta_i} &= - \left( \phi_{i,(n-1)+}^h - \theta \phi_{i,(n-1)+}^h, [u_i^h]_{n-1} \right)_{\Delta_i} \\
&+ \int_{I_n} \left\{ (f_i, \phi_i^h - \theta \phi_i^h)_{\Delta_i} - \left( \frac{\partial u_i^h}{\partial t}, \phi_i^h - \theta \phi_i^h \right)_{\Delta_i} - a_i(u_i^h, \phi_i^h - \theta \phi_i^h)_{\Delta_i} \right\} \, dt,
\end{aligned}$$

$$n = 1, \dots, N, \quad i = 1, 2,$$

$$\mathcal{E}_t^{n,\gamma} = \sum_{i=1,2} \int_{I_n} \{ \pm < \mu^h - \theta \mu^h, u_i^h >_{\gamma} \pm < \lambda^h, \phi_i^h - \theta \phi_i^h >_{\gamma} \} \, dt, \quad n = 1, \dots, N,$$

$$\begin{aligned}
\mathcal{E}_0^{\Delta_i} &= (\phi_{i,0+}^h, u_{i,0} - \pi^h u_{i,0})_{\Delta_i}, \quad i = 1, 2, \\
\mathcal{E}_x^{n,\Delta_i} &= - \left( \theta \phi_{i,(n-1)+}^h - \pi^h \theta \phi_{i,(n-1)+}^h, [u_i^h]_{n-1} \right)_{\Delta_i} \\
&\quad + \int_{I_n} \left\{ (f_i, \theta \phi_i^h - \pi^h \theta \phi_i^h)_{\Delta_i} - \left( \frac{\partial u_i^h}{\partial t}, \theta \phi_i^h - \pi^h \theta \phi_i^h \right)_{\Delta_i} - a_i(u_i^h, \theta \phi_i^h - \pi^h \theta \phi_i^h)_{\Delta_i} \right\} dt, \\
n &= 1, \dots, N, \quad i = 1, 2, \\
\mathcal{E}_x^{n,\gamma} &= \sum_{i=1,2} \int_{I_n} \left\{ \pm \langle \theta \mu^h - \pi^h \theta \mu^h, u_i^h \rangle_\gamma \pm \langle \lambda^h, \theta \phi_i^h - \pi^h \theta \phi_i^h \rangle_\gamma \right\} dt, \\
n &= 1, \dots, N,
\end{aligned}$$

with the local operators for  $i = 1, 2$  defined as follows:

$$(u, v)_{\Delta_i} = \int_{\Delta_i} uv \, d\mathbf{x}, \quad a_i(u, v)_{\Delta_i} = \int_{\Delta_i} \nabla u \cdot \nabla v \, d\mathbf{x}, \quad \langle \nu, v \rangle_\gamma = \int_\gamma \nu v \, ds.$$

Given a global tolerance  $TOL$ , the total number of elements in the domain and on the interface  $M$ , and the number of time steps  $N$ , we compute the local tolerance  $LATOL = 2 TOL/(NM)$ . A time step  $n$  is marked for refinement if  $|\mathcal{E}_t^n| > M \cdot LATOL$ , an element  $\Delta_i$  at time step  $n$  is marked for refinement if  $|\mathcal{E}_x^{n,\Delta_i}| > LATOL$ , and an interface edge  $\gamma$  at time step  $n$  is marked for refinement if  $|\mathcal{E}_x^{n,\gamma}| > LATOL$ . If an interface edge is refined then the neighbouring domain elements are refined as well.

This strategy presents a downside from the practical point of view. Storing a different mesh for each time step and projecting the solutions from one mesh to another can become extremely costly. Moreover, projection introduces additional errors, and any gains from non-uniform refinements can be lost in the process of repeated projections. To overcome this difficulty we employ one of the blockwise

time refinement techniques introduced in [31]. We group time steps into blocks so that a single refined spatial mesh is used to compute solutions within each space-time block. We prescribe an upper bound on the number of refinements allowed in a block,  $NMAX$ , introducing an effective upper bound on the number of elements in the refined mesh, and hence a restriction on the memory and operations required.

Starting with the original mesh we apply the required refinements at each time step until a block is ‘full’, i.e. the number of required refinements reaches  $NMAX$ . Then a new block is formed from the original mesh and the process is repeated beginning at the next time step.

This algorithm can result in more than  $NMAX$  refinements in a block, firstly, because a new block is created only after this upper limit is exceeded, and secondly, because the resulting triangulation is regularised: additional edges need to be introduced to remove hanging nodes. Depending on the refinements limit and the behaviour of the solution, this strategy can result in anything from a single time block to a block every time step. The nature of the algorithm will be illustrated in the following section.

### 5.2.5 Numerical examples of the effectivity of the error estimate

We test the representation formula in a variety of scenarios for which analytic solutions can be obtained, allowing the computation of the effectivity ratio,  $\mathcal{R} = \mathcal{E}_{est}/\mathcal{E}$ , where  $\mathcal{E}_{est}$  is calculated using the representation formula (5.29), and

$$\mathcal{E} = \int_0^T \{ - \langle \psi, e_\lambda \rangle + (\psi_1, e_1) + (\psi_2, e_2) \} dt,$$



which can be computed given the exact solution  $(u_1, u_2, \lambda)$ .

### 5.2.5.1 Domain error

We first address problems in which the quantity of interest is a function of solutions  $u_1$  and  $u_2$  only.

**Single domain problems** First we consider a simple problem with identical components 1 and 2,  $k_i = 1$ ,  $c_i = 0$ ,  $i = 1, 2$ ,  $\Omega_1 = (0, 1)^2$  and  $\Omega_2 = (1, 2) \times (0, 1)$ . The right-hand side functions  $f_i$ ,  $i = 1, 2$ , the initial and the boundary conditions are obtained from the given analytic solution.

The quantity of interest is a weighted average solution in the domain, so that the adjoint solution can be obtained in closed form as  $\phi_i = 0.5(T - t)x(2 - x)$ ,  $i = 1, 2$ .

The problem was solved with  $N_t = 5$  time steps, final time  $T = 1$ , and the original mesh with  $DOF = 78$  degrees of freedom.

The following five exact solutions were used:

(i)  $u_i = 0.5x(2 - x)$ ,  $i = 1, 2$ ,

(ii)  $u_i = 0.5tx(2 - x)$ ,  $i = 1, 2$ ,

(iii)  $u_i = t \sin(0.5\pi x)$ ,  $i = 1, 2$ ,

(iv)  $u_i = x + t$ ,  $i = 1, 2$ , and

(v)  $u_i = x + \sin(t)$ ,  $i = 1, 2$ .

For each of the problems we report effectivity ratios in Table 5.8 for varying the spatial accuracy of the adjoint solution. In all problems the adjoint finite element solution was computed on a mesh obtained from the original one by refining it uniformly

$r = 1, 2$ , or 3 times. This way we satisfy the condition of the original and adjoint finite element spaces being nested:  $X_i^h \subset \hat{X}_i^h$ ,  $i = 1, 2$  and  $Y^h \subset \hat{Y}^h$ .

| $r$ | (i)    | (ii)   | (iii)  | (iv)   | (v)    |
|-----|--------|--------|--------|--------|--------|
| 1   | 0.8499 | 1.0151 | 1.0129 | 1.0000 | 1.0000 |
| 2   | 0.9625 | 1.0038 | 1.0033 | 1.0000 | 1.0000 |
| 3   | 0.9906 | 1.0009 | 1.0009 | 1.0000 | 1.0000 |

Table 5.8: Example 5.2.5.1. Effectivity ratios for model problems (i)-(v) varying the spatial accuracy of the adjoint solution. The value of  $r$  corresponds to the number of uniform spatial refinements used to obtain the adjoint mesh from the original mesh. The problem was solved with  $N_t = 5$  time steps, final time  $T = 1$ , and the original mesh of  $DOF = 78$  degrees of freedom.

The calculated effectivity ratios for problems (i)-(iii) agree with the predictions of Theorem 9, namely they approach one as the accuracy of the adjoint solution improves. The exact adjoint solution  $\phi_i = 0.5(T - t)x(2 - x)$ ,  $i = 1, 2$  is linear in time and quadratic in space, hence the finite element method used (dG(1) in time, cG(1) in space) can capture temporal variation exactly, while spatial refinements of the adjoint mesh reduce the error in the adjoint solution.

Problems (iv) and (v) were chosen so that the boundary conditions do not lie in the finite element space (are not piecewise constant in time). The effectivity ratios are exactly one since the remainder in the representation formula  $\epsilon$  is precisely zero. The adjoint solution can be captured exactly in time, and its spatial errors are symmetric about the interface  $\Gamma$ , leading to cancellations in the terms  $(f_i, \phi_i - \phi_i^h)$ ,  $i = 1, 2$ .

It was also verified that the boundary error terms should not be neglected. If the error was calculated according to the formula with no Dirichlet boundary terms, the effectivity ratio in both problems remained  $\mathcal{R} \approx 0.4$  even as the accuracy of the adjoint problem improved. Despite the fact that the Dirichlet boundary error  $\mathcal{E}_D$  itself is small, it is significant in relation to the overall error. The effectivity ratios shown in Table 5.8 were all obtained using the formula (5.29) for  $\mathcal{E}_{est}$ , which includes

the Dirichlet boundary terms.

**Distinct problems in  $\Omega_1$  and  $\Omega_2$**  The case of non-trivial coupling, where the components in  $\Omega_1$  and  $\Omega_2$  are distinct, and when the meshes in the two domains do not necessarily conform across the interface is considered next. The following three problems are studied:

- (vi) Common elliptic operator, distinct forcing:  $k_1 = k_2 = 1$ ,  $c_1 = c_2 = 0$  with exact solutions  $u_1 = 0.5tx(2 - x)$  and  $u_2 = -0.5t \cos(0.5\pi(3 - x))$ .
- (vii) Distinct elliptic operators:  $k_1 = 1$ ,  $k_2 = -0.1$ ,  $c_1 = 0$ ,  $c_2 = 1$  with exact solutions  $u_1 = t \sin(\pi x)$  and  $u_2 = 10t \sin(\pi x)$ .
- (viii) Elliptic-parabolic coupling:

$$\begin{aligned} -\Delta u_1 &= f_1, \\ \frac{\partial u_2}{\partial t} - \Delta u_2 &= f_2, \end{aligned}$$

with the usual initial, boundary and coupling conditions, and the right-hand sides such that  $u_1 = u_2 = t \sin(\pi x)$ .

Note that in (vii) and (viii) the differential operators are different, and hence the adjoint problem and the residuals change their form accordingly. As before, the quantity of interest is such that  $\phi_i = 0.5(T - t)x(2 - x)$ ,  $i = 1, 2$ .

All problems were solved with  $N_t = 5$  time steps for final time  $T = 1$ . We consider both the cases of triangulations in  $\Omega_1$  and  $\Omega_2$  that conform on the interface, with 36 degrees of freedom in each domain and 6 on the interface, and triangulations that do not conform on  $\Gamma$ , with 36 degrees of freedom in  $\Omega_1$ , 64 in  $\Omega_2$  and 8 on  $\Gamma$ .

The results are shown in Table 5.9. Spatial errors dominate in all three cases, and therefore the effectivity ratio converges to one with spatial refinements for both conforming and non-conforming meshes in  $\Omega_1$  and  $\Omega_2$ .

| $r$ | (vi)<br>conf | (vi)<br>non-conf | (vii)<br>conf | (vii)<br>non-conf | (viii)<br>conf | (viii)<br>non-conf |
|-----|--------------|------------------|---------------|-------------------|----------------|--------------------|
| 1   | 1.0140       | 1.0091           | 1.0056        | 0.9974            | 1.0253         | 1.0137             |
| 2   | 1.0035       | 1.0023           | 1.0023        | 0.9993            | 1.0063         | 1.0034             |
| 3   | 1.0009       | 1.0006           | 1.0015        | 0.9997            | 1.0016         | 1.0008             |

Table 5.9: Example 5.2.5.1. Effectivity ratios for model problems (vi)-(viii) varying the spatial accuracy of the adjoint solution. The value of  $r$  corresponds to the number of uniform spatial refinements used to obtain the adjoint mesh from the original mesh. The problem was solved with  $N_t = 5$  time steps, final time  $T = 1$ , and the original mesh of  $DOF = 78$  degrees of freedom for the conforming (conf) meshes and  $DOF = 108$  for the non-conforming (non-conf) meshes.

### 5.2.5.2 Estimating the error on the interface

In order to compute the effectivity ratio for the error along the interface, we require  $\psi \neq 0$ . We consider mixed domain and interface error, and then error in the Lagrange multiplier space alone.

**Mixed domain and interface error** Let  $k_1 = k_2 = 1$  and  $c_1 = c_2 = 0$ ,  $\Omega_1 = (0, 1)^2$  and  $\Omega_2 = (1, 2) \times (0, 1)$ . We choose  $f_1$  and  $f_2$  and appropriate initial and boundary conditions so that  $u_1 = u_2 = t \sin(\pi x)$ . Again, both conforming and non-conforming meshes are considered, with respectively 121 degrees of freedom in each domain and 11 on the interface, and 121 degrees of freedom in  $\Omega_1$ , 256 in  $\Omega_2$  and 16 on  $\Gamma$ . The mesh is better refined than in the problems for domain error, so as to better capture the original solution, and its integrals in error calculations.

In order to obtain an adjoint solution in closed form, specifically  $\phi_1 = 0.5(T - t)x$  and  $\phi_2 = 0.5(T - t)(x - 2)$ , which satisfies zero Dirichlet conditions at  $x = 0, 2$ , zero

Neumann conditions at  $y = 0, 1$  and zero conditions at  $t = T$ , the weighting functions for the quantity of interest are chosen to be  $\psi_1 = 0.5x$ ,  $\psi_2 = (0.5x - 1)$  and  $\psi = T - t$ . These adjoint weights correspond to mixed domain and interface error. In order to establish the significance of the interface error component we computed

$$\frac{\mathcal{E}_\Gamma}{\mathcal{E}} = \frac{\left| \int_0^T \langle \psi, e_\lambda \rangle dt \right|}{\left| \int_0^T \{ - \langle \psi, e_\lambda \rangle + (\psi_1, e_1) + (\psi_2, e_2) \} dt \right|} = 0.7669,$$

where  $\mathcal{E}$  is the actual error based on the exact solution. The fact that this ratio is non-negligible means that the interface error will cause a significant proportion of refinements.

We report effectivity ratios for this problem in columns two and three of Table 5.10. The adjoint solution is captured exactly by the finite element method, and the effectivity ratios are therefore equal to one.

| $r$ | Mixed error |          | Interface error |          |
|-----|-------------|----------|-----------------|----------|
|     | conf        | non-conf | conf            | non-conf |
| 1   | 1.0000      | 1.0000   | 0.9925          | 0.9946   |
| 2   | 1.0000      | 1.0000   | 0.9982          | 0.9987   |
| 3   | 1.0000      | 1.0000   | 0.9996          | 0.9997   |

Table 5.10: Example 5.2.5.2. Effectivity ratios varying the spatial accuracy of the adjoint solution. The value of  $r$  corresponds to the number of uniform spatial refinements used to obtain the adjoint mesh from the original mesh. The problem was solved with  $N_t = 5$  time steps, final time  $T = 1$ , and the original mesh of  $DOF = 253$  degrees of freedom for the conforming (conf) meshes and  $DOF = 393$  for the non-conforming (non-conf) meshes.

**Error on the interface** In order to calculate the error solely on the interface, we set  $\psi_1 = \psi_2 = 0$  and  $\psi = T - t$  (to satisfy the condition at  $t = T$ ). The problem is solved both on conforming (conf) and non-conforming (non-conf) meshes in  $\Omega_1$  and  $\Omega_2$ . As seen in columns four and five of Table 5.10, the calculated effectivity ratios

for both conforming and non-conforming cases converge to one as the mesh on which the adjoint solution is computed gets finer.

## 5.2.6 Numerical examples of adaptive mesh refinement

We apply the adaptive strategy described Section 5.2.4 to a sequence of problems and discuss the results obtained.

### 5.2.6.1 Equal diffusion coefficients

Let  $\Omega_1 = (0, 1) \times (0, 1)$ ,  $\Omega_2 = (0, 1) \times (1, 2)$ ,  $k_1 = k_2 = 1$ , and  $c_1 = c_2 = 0$ . The right-hand sides, the initial and the Dirichlet boundary conditions are chosen so that

$$u_i = \exp(-100(x - 0.5 - 0.25 \sin(\pi t))^2 - 100(y - 1 - 0.5 \cos(\pi t))^2), \quad i = 1, 2.$$

This solution is a Gaussian with a peak located at  $(0.5, 1.5)$  at time  $t = 0$ , and moving counterclockwise along an elliptic trajectory inside  $\Omega_1 \cup \Omega_2$ . At the final time  $T = 2$  the peak returns to its initial position after making one complete orbit. A view of eight snapshots of the solution is presented in Figure 5.8. The adaptive procedure is applied to a mesh with 453 degrees of freedom in total and 20 time steps.

**Adaptivity based on error in the domain only** First, we consider the quantity of interest representing the average in the whole domain, so that

$$\mathcal{E} = \int_0^T \left\{ \int_{\Omega_1} e_1 \, dx + \int_{\Omega_2} e_2 \, dx \right\} dt,$$

i.e.  $\psi_1 = \psi_2 = 1$  and  $\psi = 0$ . The time-space adaptive refinement strategy was applied to the original uniform mesh, producing six temporal blocks. The global tolerance was

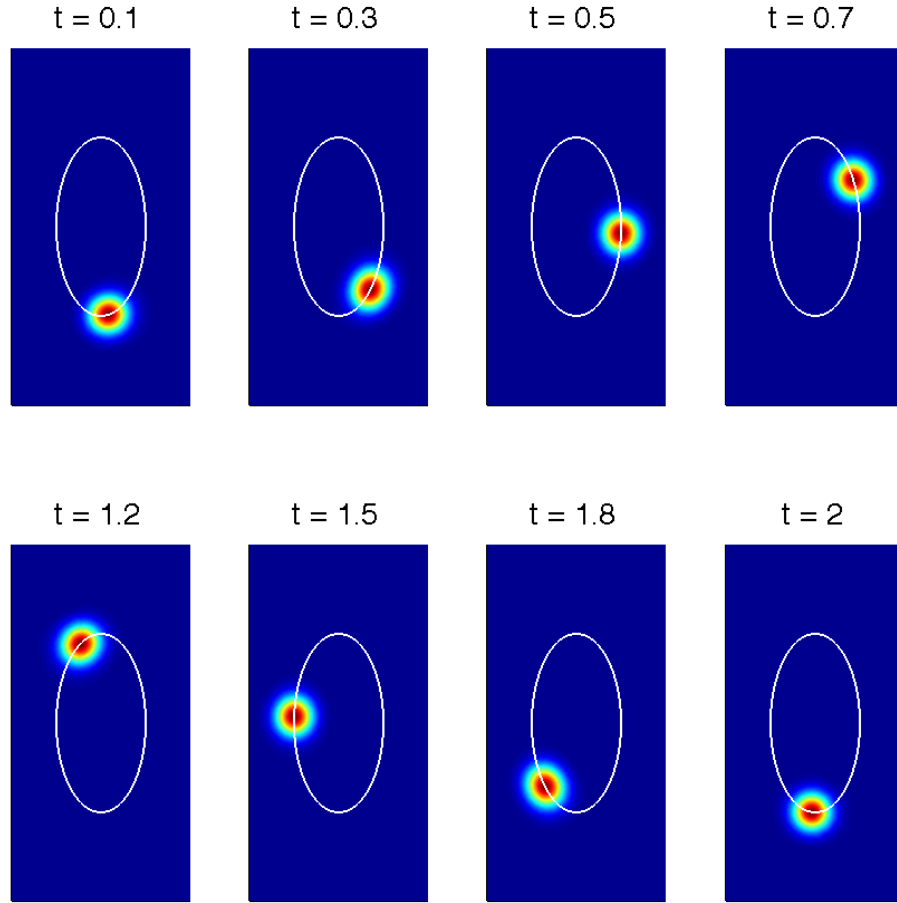


Figure 5.8: Snapshots of a Gaussian travelling along an elliptic trajectory.

set to  $TOL = 0.01$ . At most three refinements of an element and a single refinement of a time step were allowed. The maximum number of triangles marked for refinements based on local errors in each temporal block was set at  $NMAX = 350$ . However more triangles are refined in practice in order to keep the triangulations regular.

The refined meshes are shown in Figure 5.9, and the estimated errors in the quantity of interest are shown in Table 5.11. As expected, the spatial mesh refinements trace the movement of the solution in the domain. Away from the peak of the Gaussian the solution is almost constant, and therefore does not require any refinements.

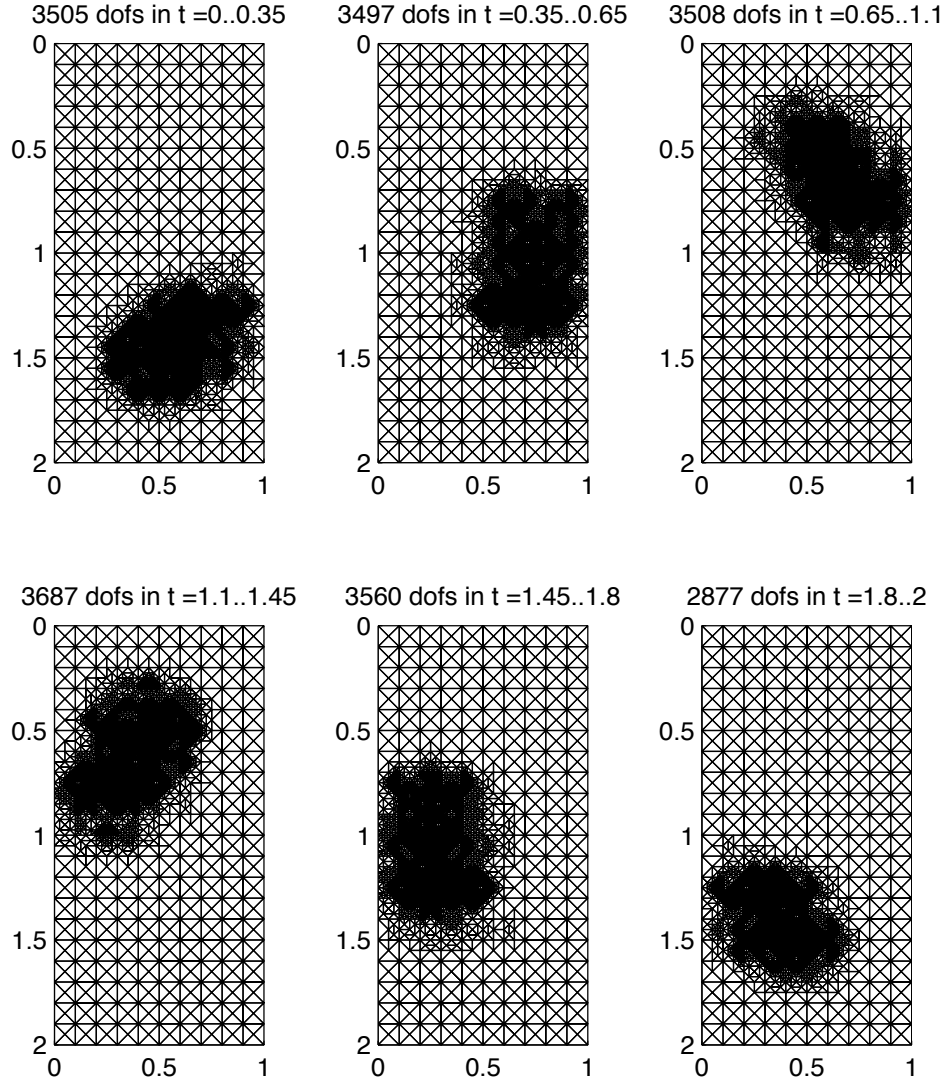


Figure 5.9: Example 5.2.6.1. Meshes obtained after a time block-wise refinement for the quantity of interest in the whole domain  $\Omega_1 \cup \Omega_2$  (at most 350 elements allowed to be refined in a block based on local errors).

The total error was reduced by an order of magnitude. Similar accuracy can be achieved after three uniform refinements, producing a mesh with 26003 degrees of freedom. Even the non-restrictive block with  $NMAX = 1500$  is half this size. For smaller values of  $NMAX$  we can reduce the maximum number of degrees of freedom



|                          | original | refined |
|--------------------------|----------|---------|
| $10^4 \mathcal{E}_{est}$ | -3.3234  | -0.2857 |
| $10^4 \mathcal{E}_t$     | -0.2335  | -0.0545 |
| $10^4 \mathcal{E}_x$     | -3.0899  | -0.2311 |
| $e$                      | 0.0895   | 0.0621  |

Table 5.11: Example 5.2.6.1. Estimated total error  $\mathcal{E}_{est}$ , its temporal and spatial components  $\mathcal{E}_t$  and  $\mathcal{E}_x$ , and the corresponding  $L^2$ -error  $e$  calculated on the original uniform mesh (453 degrees of freedom), and on the meshes obtained after an error-based refinement (at most 3687 degrees of freedom in a block) for the quantity of interest in the whole domain.

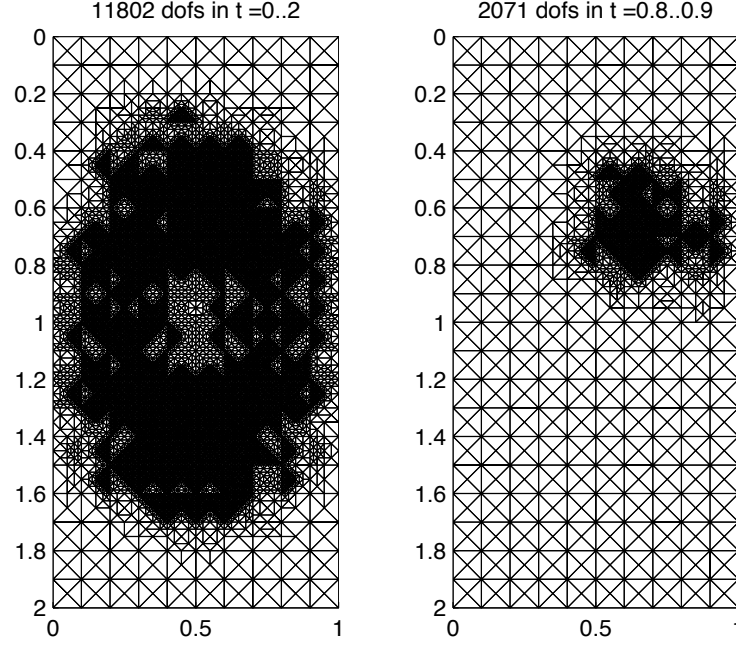
in the domain. In the case of  $NMAX = 350$  the size of the locally refined mesh is six times smaller than that of the uniformly refined mesh, producing a similar reduction of the error in the quantity of interest.

For comparison we show sample blocks for the values of  $NMAX = 1500$  and  $NMAX = 100$  in Figure 5.10. The former limit is not restrictive at all, and the algorithm creates a single mesh block for all time steps. The latter is very restrictive, and results in one block per one or two (original) time steps. In both cases time steps are refined independently of the number of mesh blocks produced.

To further illustrate the method's performance, we compute the  $L^2$ -norm errors:

$$e = \left( \sum_{i=1,2} \int_0^T \int_{\Omega_i} (u_i - u_i^h)^2 \, d\mathbf{x} \, dt \right)^{1/2}.$$

The knowledge of the exact solution makes this calculation possible, and we can thus verify that the errors have been reduced globally. This is not obvious if the total error in the quantity of interest decreases, because its definition allows cancellation of errors. As we can see from Table 5.11, the reduction in  $\mathcal{E}_{est}$  is more pronounced than in the  $L^2$ -norm error  $e$ . This is to be expected, since the algorithm is geared towards reducing the error in the quantity of interest.



(a)  $NMAX = 1500$

(b)  $NMAX = 100$

Figure 5.10: Example 5.2.6.1. Meshes in sample blocks obtained after a time block-wise refinement for the quantity of interest in the whole domain  $\Omega_1 \cup \Omega_2$  (at most  $NMAX$  elements allowed to be refined in a block based on local errors).

**Adaptivity based on error along the interface** The weighted averaged error along the interface is calculated as

$$\mathcal{E} = \int_0^T \int_{\Gamma} \psi e_{\lambda} \, ds \, dt,$$

which corresponds to the adjoint right-hand sides  $\psi_1 = \psi_2 = 0$  and  $\psi \neq 0$ . The adjoint solution has a jump across the interface  $\phi_1 - \phi_2 = \psi$  on  $\Gamma$ . In order for this to conform with the Dirichlet boundary conditions for  $\phi_1 = \phi_2 = 0$ , we need to have  $\psi = 0$  at  $(0, 1)$  and  $(1, 1)$ . We also require  $\phi_1 - \phi_2 = 0$  at  $t = T$ . To satisfy these conditions we chose  $\psi = (T - t) \sin(\pi x)$ . The refined meshes are shown in Figure 5.11.

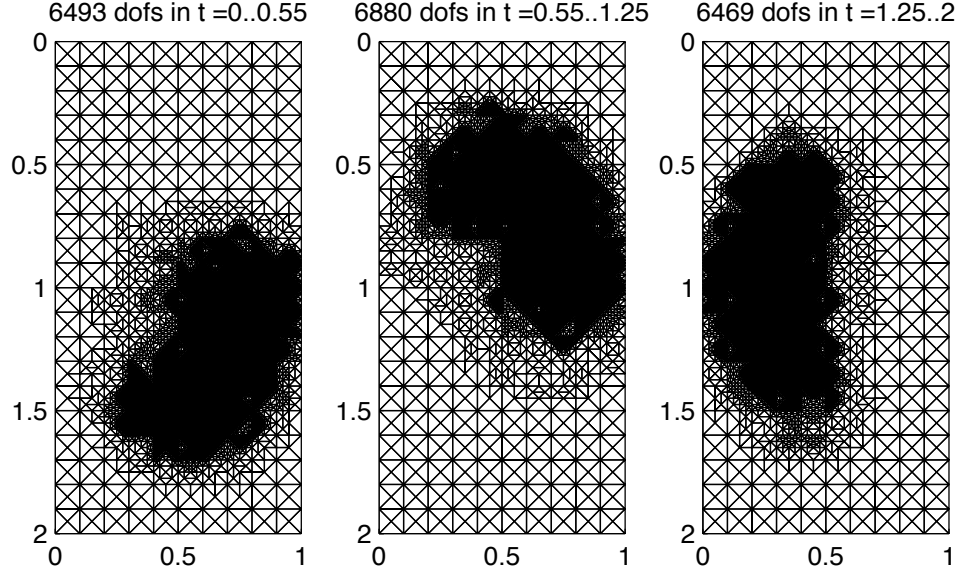


Figure 5.11: Example 5.2.6.1. Meshes obtained after a time block-wise refinement for the quantity of interest on the interface  $\Gamma$  (at most 700 elements allowed to be refined in a block based on local errors).

In this example we set block capacity to  $NMAX = 700$  elements, and obtained three temporal blocks, each with at most 6880 degrees of freedom. In the second block  $t \in (0.55, 1.25]$  the peak trajectory is fully contained in  $\Omega_1$  and does not intersect  $\Gamma$ , but the mesh is still refined around the interface in  $\Omega_2$ . This was not the case for the quantity of interest in the whole domain, and arose due to the greater relative importance of errors near the interface. The last block  $t \in (1.25, 2]$  is not as refined around the point  $(0.5, 1.5)$  as the first block. The reason is that while the accuracy of the solution for  $t < 0.55$  directly affects the accuracy of the solution on the interface through which the Gaussian passes at  $t = 0.5$ , the accuracy with which the Gaussian is resolved for  $t > 1.5$  will have little effect on the interface values.

The estimated error values for this quantity of interest are given in Table 5.12. The refinement reduces the interface error by two orders of magnitude. The corresponding

$L^2$ -norm error is calculated as

$$e = \left( \int_0^T \int_{\Gamma} (\lambda - \lambda^h)^2 \, ds \, dt \right)^{1/2}.$$

|                          | original | refined |
|--------------------------|----------|---------|
| $10^4 \mathcal{E}_{est}$ | -3.6439  | 0.0291  |
| $10^4 \mathcal{E}_t$     | 0.0004   | 0.0077  |
| $10^4 \mathcal{E}_x$     | -3.6443  | 0.0214  |
| $e$                      | 0.0206   | 0.0039  |

Table 5.12: Example 5.2.6.1. Estimated total error  $\mathcal{E}_{est}$ , its temporal and spatial components  $\mathcal{E}_t$  and  $\mathcal{E}_x$ , and the corresponding  $L^2$ -error  $e$  calculated on the original uniform mesh (453 degrees of freedom), and on the meshes obtained after an error-based refinement (at most 6880 degrees of freedom in a block).

Once again we observe a significant reduction in the total error in the quantity of interest. The temporal error increased, most likely due to loss of symmetry, however this does not affect the overall error, since the spatial component dominates on both the original and the refined meshes. The  $L^2$ -norm of the error was also reduced.

### 5.2.6.2 Different diffusion coefficients

We modify the problem to consider the case of different diffusion coefficients. In this case the right-hand sides and the initial and boundary conditions are chosen so that

$$u_i = \exp \left( -100 (x - 0.5 - 0.25 \sin(\pi t))^2 - 100 \left( \frac{y-1}{k_i} - 0.5 \sin(\pi t) \right)^2 \right), \quad i = 1, 2.$$

This choice of the exact solution allows the coupling conditions to hold for  $k_1 \neq k_2$ . The Gaussian and its trajectory are now compressed in the  $y$  direction by the factor of  $1/k_i$  in  $\Omega_i$ ,  $i = 1, 2$ . The solution for  $k_1 = 0.2$  and  $k_2 = 1$  is shown in Figure 5.12.

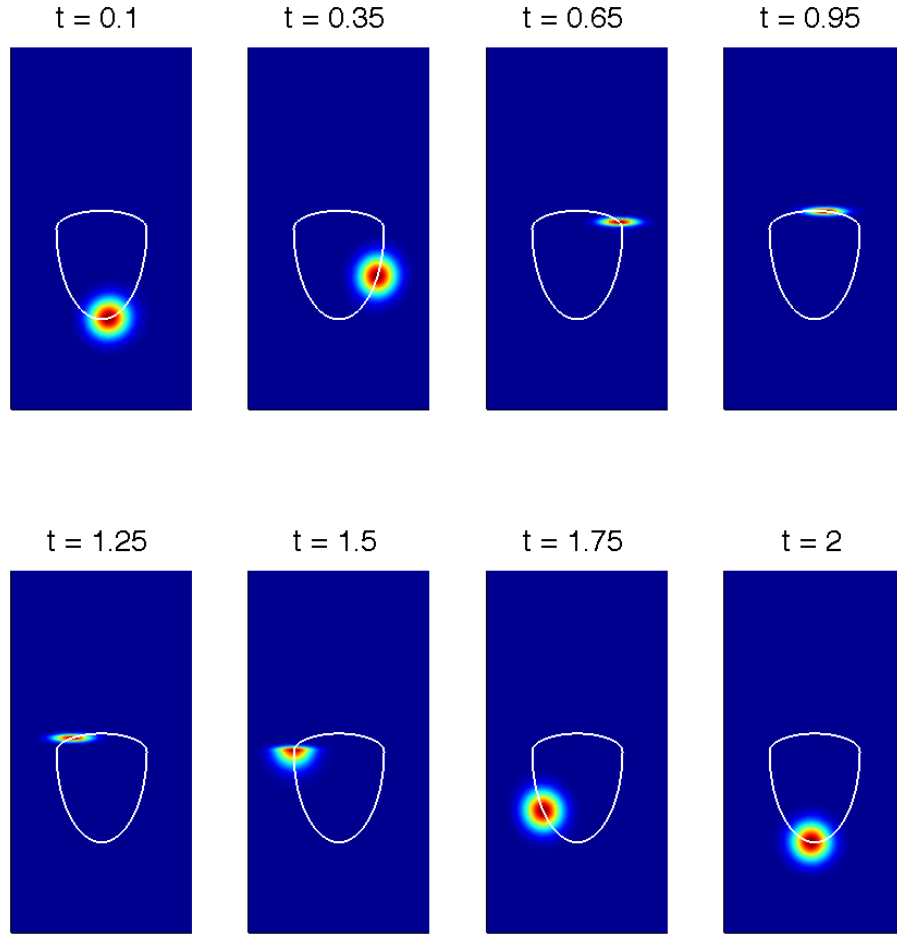


Figure 5.12: Snapshots of a Gaussian travelling along a trajectory constructed of two semi-ellipses.

The meshes obtained using the adaptive strategy are shown in Figure 5.13, and the estimated error values in Table 5.13.

For both quantities of interest we observe several common features of the refined mesh. A single temporal block is required to trace the trajectory of the solution in most of  $\Omega_1$ , where the solution is now compressed relative to the case with  $k_1 = k_2 = 1$ , and travels closer to the interface. The highly refined region is noticeably thinner in the  $y$ -direction in the second block as compared to the other blocks. There is error

|                            | Domain   |         | Interface |         |
|----------------------------|----------|---------|-----------|---------|
|                            | original | refined | original  | refined |
| $10^{-3}\mathcal{E}_{est}$ | 1.7913   | 0.0997  | -7.8804   | -1.5004 |
| $10^{-3}\mathcal{E}_t$     | 1.3593   | 0.1104  | -5.6943   | -1.5120 |
| $10^{-3}\mathcal{E}_x$     | 0.4320   | -0.0107 | -2.1862   | 0.0116  |
| $e$                        | 0.0881   | 0.0355  | 0.0153    | 0.0017  |

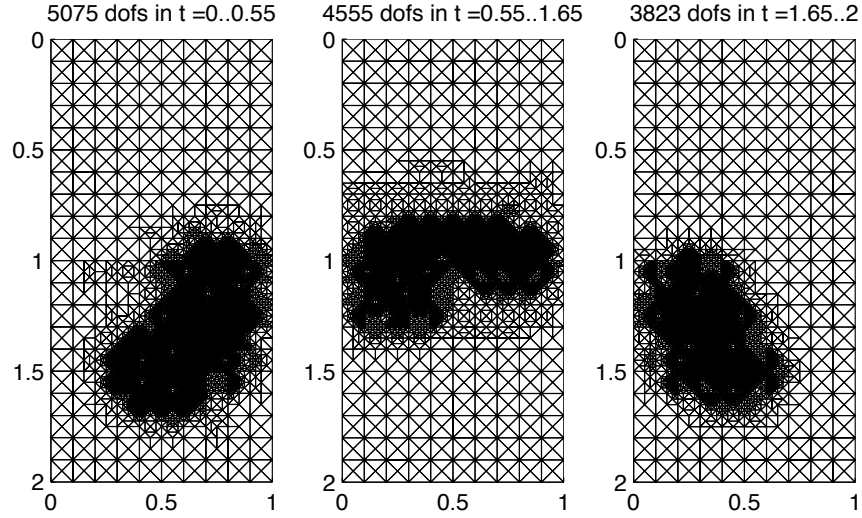
Table 5.13: Example 5.2.6.2. Estimated total error  $\mathcal{E}_{est}$ , its temporal and spatial components  $\mathcal{E}_t$  and  $\mathcal{E}_x$ , and the corresponding  $L^2$ -error  $e$  calculated on the original uniform mesh (453 degrees of freedom), and on the meshes obtained after an error-based refinement (with a block of at most 5075 degrees of freedom for domain error, 4609 degrees of freedom for interface error).

cancellation between the temporal and the spatial components on the refined mesh.

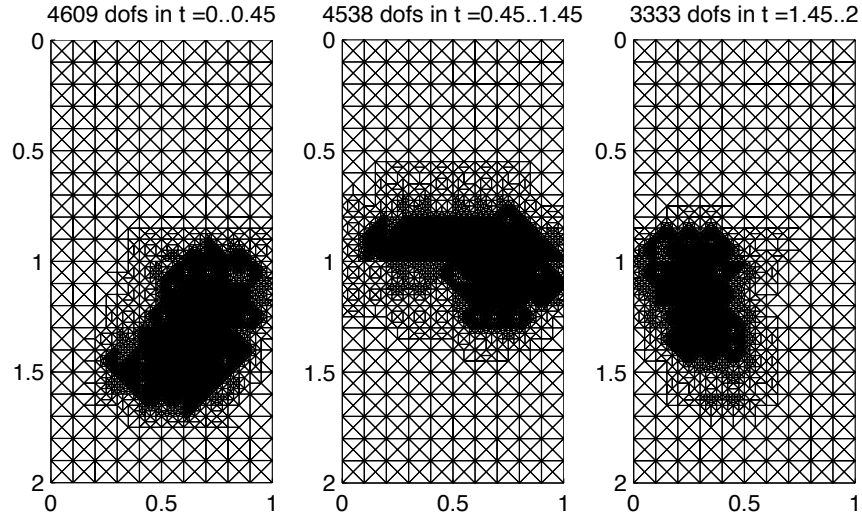
In the case of interface refinement the temporal error dominates strongly. Since we only allow a single temporal refinement per step, the error does not improve as dramatically as in the other cases. The size of the last mesh block  $t \in (1.45, 2]$  is much smaller than that of the previous two. The reason is twofold. Firstly, the accuracy of the solution does not have a strong effect on the interface values after the peak has crossed  $\Gamma$  the second time at  $t = 1.5$ , and secondly, the weight of the interface error decreases linearly with  $T - t$ , and therefore it is more important to capture the solution at the beginning of  $(0, T]$  as compared to the end.

### 5.3 Discussion

The *a posteriori* error representation formula derived and studied in this chapter provides a useful tool for controlling the error in multi-physics systems that are coupled across a common interface. An important feature of our approach is the use of Lagrange multipliers to represent coupling conditions. This allows for calculation of the error in the interface condition which follows naturally from the weak formulation of the problem. This error is of interest in many practical applications, for example



(a) Average error in the domain  $\Omega_1 \cup \Omega_2$  (at most 500 elements allowed to be refined in a block based on local errors).



(b) Weighted average error along the interface  $\Gamma$  (at most 500 elements allowed to be refined in a block based on local errors).

Figure 5.13: Example 5.2.6.2. Meshes obtained after a time block-wise refinement for the quantity of interest in the whole domain  $\Omega_1 \cup \Omega_2$  with  $k_1 = 0.2$ ,  $k_2 = 1$ .

in fluid-structure interaction problems, where the Lagrange multiplier represents the normal stress at the boundary.

The accuracy of the error estimate depends on the accuracy of the approximate adjoint solution. In the numerical examples that were considered here effectivity ratios demonstrated that the use of approximate adjoint solutions results in good estimates of the actual value of the error when the adjoint problem is solved using a richer finite element space as compared to the original problem. This can be achieved either by using a method of the same order, but defined on a finer mesh, or using a higher order method defined on the original mesh.

A block time-space refinement strategy based on the error representation formula for time-dependent problems was implemented taking advantage of our ability to distinguish between temporal and spatial errors. The performance of the algorithm was investigated in the case of a travelling wave-like solution. The refinement strategy was shown to produce efficient refinements to reduce the errors in a wide range of quantities of interest. Imposing an upper limit on the number of degrees of freedom in a time block provides a useful technique for restricting the computational overhead associated with the algorithm.

The extension of this framework to multiphysics systems that are coupled across fixed boundaries is a relatively straightforward procedure. In the next chapter we illustrate this by considering a problem where unimpeded fluid flow within rigid boundaries is coupled to flow through a porous medium. We show that the benefits of using the efficient solver developed in Chapter 4, as well as the local adjoint-based error estimation technique and the associated adaptive mesh refinement algorithm developed in Chapter 5, are retained in a physical setting which is different to the ones considered originally.



# Chapter 6

## Coupling free flow and flow through porous media

Our main purpose for developing mathematical tools for the efficient treatment of coupling between abstract systems is to enable the solution of practical problems involving such coupling. The block solver and the error estimation/mesh refinement strategies presented in Chapters 4 and 5 do not explicitly rely on the functional form of the coupled equations within the system, and should therefore be applicable in a setting which is different to those already considered. In this chapter we investigate the methods' performance in a problem where Stokes flow is coupled with flow through a porous medium.

We consider the steady problem, and assume an incompressible Newtonian fluid where the flow is sufficiently slow that it can be modelled by the Stokes equations. The flow through the permeable block is modelled using the Darcy equations. The natural condition prescribing continuity of normal velocities at the fluid-porous interface is often supplemented by the Beavers-Joseph-Saffman condition. Originally this condition was proposed in [15], where the authors experimentally verified linear

dependence of the difference between tangential velocities on the shear rate of the free fluid flow. This was further developed in [106], where the tangential component of the porous velocity was established to be negligible, and in this form the condition is often used in mathematical models [28, 40, 51, 77].

The numerical treatment of the coupled Stokes-Darcy system needs to be considered carefully, as, generally speaking, the velocities in the porous medium need not have the same regularity as the velocities in the Stokes region. The use of the standard stable finite element spaces for the Stokes problem, e.g. Taylor-Hood element, generally speaking violates the inf – sup condition, and results in loss of convergence in the Darcy problem. The problem has been treated by the use of different types of stable elements for the two components, and the appropriate choice for the Lagrange multiplier space [51, 61], or alternatively by stabilising a unified element method [11, 27].

Solution of the Stokes-Darcy system often relies on decoupling. The two-grid approach, which is based on solving the monolithic system on a coarse mesh, the decoupled problem on a fine mesh, and then combining the solutions, is discussed in [28, 33]. An iterative partitioned approach to solving the coupled system is applied in [40]. The application of FETI (finite element by tearing and interconnecting) algorithms based on domain decomposition is studied in [62, 63]. Block preconditioning for the coupled Stokes-Darcy system was considered in [29], where fully decoupled (diagonal-block) preconditioners are compared to the Schur complement-type preconditioners also studied here, with a scaled discrete identity approximation to the Schur complement block.

*A posteriori* error estimation for Stokes-Darcy coupled problems is performed in [7], where the authors take the residual-based approach, and derive estimates for

stable combinations of elements. The results highlight the need for a joint estimate, which does not immediately follow from independent estimates for components. In this work we also focus on deriving a joint estimate, in this case for an error in a quantity of interest. Moreover, we are able to use the estimate in a mesh refinement strategy for a number of numerical problems.

In this chapter we write out the equations making up the Stokes-Darcy coupled model, and the finite element discretisation of the system. We revisit the methods for efficient solution of the resulting block matrix equations, as presented in Chapter 4, and for adjoint-based error estimation and mesh refinement, as presented in Chapter 5. The derivation of both follows naturally from the presentation in these earlier chapters, but is sufficiently different due to the underlying geometry and physics, and is therefore obtained explicitly. A set of examples is considered at the end of the chapter, where the performance of both methods is evaluated.

## 6.1 Weak equations and the finite element method

The problem considered here is inspired by flow through a cartridge filter, where the two-dimensional cross-section of the domain can be schematically represented by the three-compartment domain shown in Figure 6.1.

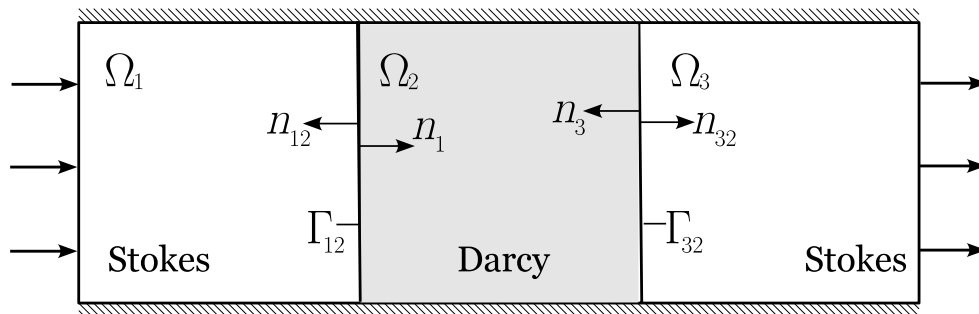


Figure 6.1: Schematic representation of the subdomains for the problem modelling coupling of free fluid flow and flow through a porous medium.

We study steady flow through a channel composed of three subdomains:  $\Omega_1$ ,  $\Omega_2$  and  $\Omega_3 \in \mathbb{R}^2$ . The Darcy model 6.1e is used for the flow in  $\Omega_2$ , which is assumed to be filled with a porous material.  $\Omega_1$  and  $\Omega_3$  only contain the fluid, which is represented by the Stokes model (6.1a). The interfaces between the respective domains are denoted by  $\Gamma_{12}$  and  $\Gamma_{32}$ .

The boundary conditions are as follows. The Dirichlet conditions in the fluid domains (6.1c) prescribe no-slip on the sides of the channel and a given inflow profile at the inlet. The corresponding boundaries  $\Gamma_1^D$  and  $\Gamma_3^D$  comprise the top, left and bottom edges of  $\Omega_1$ , and the top and bottom edges of  $\Omega_3$  respectively. The functions  $\mathbf{u}_{i,D} = 0$  for  $i = 1, 3$  on the top and bottom edges of  $\Omega_i$ , and  $u_{1,D}$  is a known function (e.g. parabolic in  $y$ ) on the left edge of  $\Omega_1$ . At the outlet  $\Gamma_3^N$  (the right edge of  $\Omega_3$ ) we impose zero flux (6.1d). In the porous domain we require no outflow (6.1g) through the sides of the channel  $\Gamma_2^D$ , i.e. the top and bottom edges of  $\Omega_2$ .

The interface conditions enforce continuity of the normal component of the velocity (6.1h) and stress (6.1j) on  $\Gamma_{12}$  and  $\Gamma_{32}$ . Additionally we impose the empirically obtained equation (6.1i), referred to as the Beavers-Joseph-Saffman condition.

The full coupled system (6.1a)-(6.1j) is given on the next page.

In the below equations  $\mu$  is the fluid viscosity (assumed constant),  $\mathbf{f}_i$ ,  $i = 1, 3$  are the external forces,  $f_2$  is the source term,  $\mathbf{K}$  is the permeability tensor for the porous medium, and  $G$  is the friction constant. On the domain boundaries the outward-facing unit normal vector is denoted by  $\mathbf{n}$ . On the interfaces the outward-facing normals for  $\Omega_i$  on  $\Gamma_{i2}$  are denoted by  $\mathbf{n}_i$ , the outward-facing normals for  $\Omega_2$  on  $\Gamma_{i2}$  are denoted by  $\mathbf{n}_{i2}$ , and the tangential unit vectors for  $\Omega_i$  on  $\Gamma_{i2}$  are denoted by  $\boldsymbol{\tau}_i$ ,  $i = 1, 3$ .

$$-\nabla \cdot (\mu \nabla \mathbf{u}_i - p_i \mathbf{I}) = \mathbf{f}_i \quad \text{in } \Omega_i, \ i = 1, 3, \quad (6.1a)$$

$$\nabla \cdot \mathbf{u}_i = 0 \quad \text{in } \Omega_i, \ i = 1, 3, \quad (6.1b)$$

$$\mathbf{u}_i = \mathbf{u}_{i,D} \quad \text{on } \Gamma_i^D, \ i = 1, 3, \quad (6.1c)$$

$$(\mu \nabla \mathbf{u}_3 - p_3 \mathbf{I}) \mathbf{n} = 0 \quad \text{on } \Gamma_3^N, \quad (6.1d)$$

$$\mathbf{K}^{-1} \mathbf{u}_2 + \nabla p_2 = 0 \quad \text{in } \Omega_2, \quad (6.1e)$$

$$\nabla \cdot \mathbf{u}_2 = f_2 \quad \text{in } \Omega_2, \quad (6.1f)$$

$$\mathbf{u}_2 \cdot \mathbf{n} = 0 \quad \text{on } \Gamma_2^D, \quad (6.1g)$$

$$\mathbf{u}_i \cdot \mathbf{n}_i = -\mathbf{u}_2 \cdot \mathbf{n}_{i2} \quad \text{on } \Gamma_{i2}, \ i = 1, 3, \quad (6.1h)$$

$$G[(\mu \nabla \mathbf{u}_i - p_i \mathbf{I}) \mathbf{n}_i] \cdot \boldsymbol{\tau}_i = -\mathbf{u}_i \cdot \boldsymbol{\tau}_i \quad \text{on } \Gamma_{i2}, \ i = 1, 3, \quad (6.1i)$$

$$[(\mu \nabla \mathbf{u}_i - p_i \mathbf{I}) \mathbf{n}_i] \cdot \mathbf{n}_i = -p_2 \quad \text{on } \Gamma_{i2}, \ i = 1, 3. \quad (6.1j)$$

The following Sobolev spaces are used to construct the weak form of the equations:

$$X_i = (H^1(\Omega_i))^2, \ i = 1, 3,$$

$$X_2 = H(\text{div}; \Omega_2) = \left\{ \mathbf{u}_2 \in (L^2(\Omega_2))^2 \mid \nabla \cdot \mathbf{u}_2 \in L^2(\Omega_2) \right\},$$

$$X_{i,D} = \{ \mathbf{w} \in X_i \mid \mathbf{w} = \mathbf{u}_{i,D} \text{ on } \Gamma_i^D \}, \ i = 1, 3,$$

$$X_{i,0} = \{ \mathbf{w} \in X_i \mid \mathbf{w} = 0 \text{ on } \Gamma_i^D \}, \ i = 1, 3,$$

$$X_{2,0} = \{ \mathbf{w} \in X_2 \mid \mathbf{w} \cdot \mathbf{n} = 0 \text{ on } \Gamma_2^D \},$$

$$Y_i = L^2(\Omega_i), \ i = 1, 2, 3,$$

$$Z_i = L^2(\Gamma_{i2}), \ i = 1, 3.$$

We obtain the weak form of the Stokes equations (6.1a) by testing them with

$\mathbf{w}_i \in X_{i,0}$  and integrating over  $\Omega_i$  respectively:

$$-\int_{\Omega_i} \nabla \cdot (\mu \nabla \mathbf{u}_i - p_i \mathbf{I}) \cdot \mathbf{w}_i \, d\mathbf{x} = \int_{\Omega_i} \mathbf{f}_i \cdot \mathbf{w}_i \, d\mathbf{x}, \quad i = 1, 3.$$

The divergence theorem yields:

$$-\int_{\partial\Omega_i} [(\mu \nabla \mathbf{u}_i - p_i \mathbf{I}) \mathbf{n}_i] \cdot \mathbf{w}_i \, ds + \int_{\Omega_i} (\mu \nabla \mathbf{u}_i : \nabla \mathbf{w}_i - p_i \nabla \cdot \mathbf{w}_i) \, d\mathbf{x} = \int_{\Omega_i} \mathbf{f}_i \cdot \mathbf{w}_i \, d\mathbf{x}, \quad i = 1, 3,$$

where  $\partial\Omega_1 = \Gamma_1^D \cup \Gamma_{12}$  and  $\partial\Omega_3 = \Gamma_3^D \cup \Gamma_3^N \cup \Gamma_{32}$  are the boundaries of  $\Omega_1$  and  $\Omega_3$  respectively.

The terms on  $\Gamma_i^D$ ,  $i = 1, 3$ , and  $\Gamma_3^N$  disappear due to boundary conditions:  $\mathbf{w}_i = 0$  on  $\Gamma_i^D$  for  $\mathbf{w}_i \in X_i$ , and  $(\mu \nabla \mathbf{u}_i - p_i \mathbf{I}) \mathbf{n} = 0$  on  $\Gamma_i^N$  from (6.1d). Thus the only non-zero boundary terms are those on  $\Gamma_{i2}$ ,  $i = 1, 3$ :

$$-\int_{\partial\Omega_i} [(\mu \nabla \mathbf{u}_i - p_i \mathbf{I}) \mathbf{n}_i] \cdot \mathbf{w}_i \, ds = -\int_{\Gamma_{i2}} [(\mu \nabla \mathbf{u}_i - p_i \mathbf{I}) \mathbf{n}_i] \cdot \mathbf{w}_i \, ds, \quad i = 1, 3.$$

This interface term can be further transformed using equations (6.1i) and (6.1j). We introduce two Lagrange multipliers  $\lambda_i = [(\mu \nabla \mathbf{u}_i - p_i \mathbf{I}) \mathbf{n}_i] \cdot \mathbf{n}_i$ ,  $i = 1, 3$ , and also note that  $[(\mu \nabla \mathbf{u}_i - p_i \mathbf{I}) \mathbf{n}_i] \cdot \boldsymbol{\tau}_i$  can be replaced with  $-\frac{1}{G} \mathbf{u}_i \cdot \boldsymbol{\tau}_i$ . This allows us to rewrite the above integral as follows:

$$-\int_{\Gamma_{i2}} [(\mu \nabla \mathbf{u}_i - p_i \mathbf{I}) \mathbf{n}_i] \cdot \mathbf{w}_i \, ds = -\int_{\Gamma_{i2}} \left[ \lambda_i \mathbf{w}_i \cdot \mathbf{n}_i - \frac{1}{G} (\mathbf{u}_i \cdot \boldsymbol{\tau}_i) (\mathbf{w}_i \cdot \boldsymbol{\tau}_i) \right] \, ds, \quad i = 1, 3.$$

In addition (6.1j) implies  $p_2 = -\lambda_i$  on  $\Gamma_{i2}$  for  $i = 1, 3$ .

The weak Stokes equations have the following form:

$$\begin{aligned}
& - \int_{\Gamma_{i2}} \left[ \lambda_i \mathbf{w}_i \cdot \mathbf{n}_i - \frac{1}{G} (\mathbf{u}_i \cdot \boldsymbol{\tau}_i) (\mathbf{w}_i \cdot \boldsymbol{\tau}_i) \right] ds + \int_{\Omega_i} (\mu \nabla \mathbf{u}_i : \nabla \mathbf{w}_i - p_i \nabla \cdot \mathbf{w}_i) d\mathbf{x} \\
& \qquad \qquad \qquad = \int_{\Omega_i} \mathbf{f}_i \cdot \mathbf{w}_i d\mathbf{x}, \quad i = 1, 3.
\end{aligned}$$

We define bilinear forms  $a_i : X_i \times X_i \rightarrow \mathbb{R}$ ,  $b_i : X_i \times Y_i \rightarrow \mathbb{R}$  and  $c_i : X_i \times Z_i \rightarrow \mathbb{R}$  such that  $\forall \mathbf{v}_i, \mathbf{w}_i \in X_i$ ,  $q_i \in Y_i$  and  $\nu_i \in Z_i$

$$\begin{aligned}
a_i(\mathbf{v}_i, \mathbf{w}_i) &= \int_{\Gamma_{i2}} \frac{1}{G} (\mathbf{v}_i \cdot \boldsymbol{\tau}_i) (\mathbf{w}_i \cdot \boldsymbol{\tau}_i) ds + \int_{\Omega_i} \mu \nabla \mathbf{v}_i : \nabla \mathbf{w}_i d\mathbf{x}, \quad i = 1, 3, \\
b_i(\mathbf{w}_i, q_i) &= \int_{\Omega_i} q_i \nabla \cdot \mathbf{w}_i d\mathbf{x}, \quad i = 1, 3, \\
c_i(\mathbf{w}_i, \nu_i) &= \int_{\Gamma_{i2}} \nu_i \mathbf{w}_i \cdot \mathbf{n}_i ds, \quad i = 1, 3.
\end{aligned}$$

This notation allows us to write the weak Stokes equations in shorthand form as

$$a_i(\mathbf{u}_i, \mathbf{w}_i) - b_i(\mathbf{w}_i, p_i) - c_i(\mathbf{w}_i, \lambda_i) = (\mathbf{f}_i, \mathbf{w}_i), \quad i = 1, 3. \quad (6.2)$$

In order to obtain the weak form of the porous flow equations we test (6.1e) with  $\mathbf{w}_2 \in X_{2,0}$ , integrate over  $\Omega_2$ , and use the divergence theorem, which leads to the following equation:

$$\int_{\partial\Omega_2} p_2 \mathbf{w}_2 \cdot \mathbf{n} ds + \int_{\Omega_2} (\mathbf{K}^{-1} \mathbf{u}_2 \cdot \mathbf{w}_2 - p_2 \nabla \cdot \mathbf{w}_2) d\mathbf{x} = 0,$$

where  $\partial\Omega_2 = \Gamma_2^D \cup \Gamma_{12} \cup \Gamma_{32}$ . By construction of the space  $X_{2,0}$  we have that  $\mathbf{w}_2 \cdot \mathbf{n} = 0$

on  $\Gamma_2^D$ . As noted earlier,  $p_2 = -\lambda_i$  on  $\Gamma_{i2}$  for  $i = 1, 3$ , and therefore

$$\int_{\Gamma_{i2}} p_2 \mathbf{w}_2 \cdot \mathbf{n}_{i2} \, ds = - \int_{\Gamma_{i2}} \lambda_i \mathbf{w}_2 \cdot \mathbf{n}_{i2} \, ds, \quad i = 1, 3.$$

Hence the weak porous flow equation can be written as follows:

$$- \int_{\Gamma_{12}} \lambda_1 \mathbf{w}_2 \cdot \mathbf{n}_{12} \, ds - \int_{\Gamma_{32}} \lambda_3 \mathbf{w}_2 \cdot \mathbf{n}_{32} \, ds + \int_{\Omega_2} (\mathbf{K}^{-1} \mathbf{u}_2 \cdot \mathbf{w}_2 - p_2 \nabla \cdot \mathbf{w}_2) \, d\mathbf{x} = 0.$$

As before, we define bilinear forms  $a_2 : X_2 \times X_2 \rightarrow \mathbb{R}$ ,  $b_2 : X_2 \times Y_2 \rightarrow \mathbb{R}$  and  $c_2 : X_2 \times Z_2 \rightarrow \mathbb{R}$  such that  $\forall \mathbf{v}_2, \mathbf{w}_2 \in X_2, q_2 \in Y_2, \nu_1 \in Z_1$  and  $\nu_3 \in Z_3$

$$\begin{aligned} a_2(\mathbf{v}_2, \mathbf{w}_2) &= \int_{\Omega_2} \mathbf{K}^{-1} \mathbf{u}_2 \cdot \mathbf{w}_2 \, d\mathbf{x}, \\ b_2(\mathbf{w}_2, q_2) &= \int_{\Omega_2} q_2 \nabla \cdot \mathbf{w}_2 \, d\mathbf{x}, \\ c_{i2}(\mathbf{w}_2, \nu_i) &= \int_{\Gamma_{i2}} \nu_i \mathbf{w}_2 \cdot \mathbf{n}_{i2} \, ds, \quad i = 1, 3. \end{aligned}$$

This leads to the following shorthand form of the above equation:

$$a_2(\mathbf{u}_2, \mathbf{w}_2) - b_2(\mathbf{w}_2, p_2) - c_{12}(\mathbf{w}_2, \lambda_1) - c_{32}(\mathbf{w}_2, \lambda_3) = 0. \quad (6.3)$$

Next we test (6.1b) and (6.1f) with  $-q_i \in Y_i$  for  $i = 1, 2, 3$  respectively:

$$\begin{aligned} - \int_{\Omega_i} q_i \nabla \cdot \mathbf{u}_i \, d\mathbf{x} &= 0, \quad i = 1, 3, \\ - \int_{\Omega_2} q_2 \nabla \cdot \mathbf{u}_2 \, d\mathbf{x} &= - \int_{\Omega_2} f_2 q_2 \, d\mathbf{x}. \end{aligned}$$



Using the notation introduced above we can rewrite these equations in operator form:

$$-b_i(\mathbf{u}_i, q_i) = 0, \quad i = 1, 3, \quad (6.4)$$

$$-b_2(\mathbf{u}_2, q_2) = -(f_2, q_2). \quad (6.5)$$

Finally, we test the coupling conditions (6.1h) with  $\nu_i \in Z_i$ ,  $i = 1, 3$ :

$$-\int_{\Gamma_{i2}} \nu_i (\mathbf{u}_i \cdot \mathbf{n}_i + \mathbf{u}_2 \cdot \mathbf{n}_{i2}) \, ds = 0,$$

or, equivalently,

$$-c_i(\mathbf{u}_i, \nu_i) - c_{i2}(\mathbf{u}_2, \nu_i) = 0, \quad i = 1, 3.$$

Combining the weak equations (6.2)-(6.1), we arrive at the following weak formulation of the coupled Stokes-Darcy-Stokes problem:

$$Find (\mathbf{u}_1, p_1, \mathbf{u}_2, p_2, \mathbf{u}_3, p_3, \lambda_1, \lambda_3) \in X_{1,D} \times Y_1 \times X_{2,0} \times Y_2 \times X_{3,D} \times Y_3 \times Z_1 \times Z_3$$

such that

$$a_1(\mathbf{u}_1, \mathbf{w}_1) - b_1(\mathbf{w}_1, p_1) - c_1(\mathbf{w}_1, \lambda_1) = (\mathbf{f}_1, \mathbf{w}_1) \quad \forall \mathbf{w}_1 \in X_{1,0}, \quad (6.6a)$$

$$-b_1(\mathbf{u}_1, q_1) = 0 \quad \forall q_1 \in Y_1, \quad (6.6b)$$

$$a_2(\mathbf{u}_2, \mathbf{w}_2) - b_2(\mathbf{w}_2, p_2) - c_{12}(\mathbf{w}_2, \lambda_1) - c_{32}(\mathbf{w}_2, \lambda_3) = 0 \quad \forall \mathbf{w}_2 \in X_{2,0}, \quad (6.6c)$$

$$-b_2(\mathbf{u}_2, q_2) = -(f_2, q_2) \quad \forall q_2 \in Y_2, \quad (6.6d)$$

$$a_3(\mathbf{u}_3, \mathbf{w}_3) - b_3(\mathbf{w}_3, p_3) - c_3(\mathbf{w}_3, \lambda_3) = (\mathbf{f}_3, \mathbf{w}_3) \quad \forall \mathbf{w}_3 \in X_{3,0}, \quad (6.6e)$$

$$-b_3(\mathbf{u}_3, q_3) = 0 \quad \forall q_3 \in Y_3, \quad (6.6f)$$

$$-c_1(\mathbf{u}_1, \nu_1) - c_{12}(\mathbf{u}_2, \nu_1) = 0 \quad \forall \nu_1 \in Z_1, \quad (6.6g)$$

$$-c_3(\mathbf{u}_3, \nu_3) - c_{32}(\mathbf{u}_2, \nu_3) = 0 \quad \forall \nu_3 \in Z_3. \quad (6.6h)$$

We introduce the finite-dimensional approximations for the test and trial spaces as  $X_i^h \subset X_i$ ,  $X_{i,D}^h = X_i^h \cap X_{i,D}$ ,  $X_{i,0}^h = X_i^h \cap X_{i,0}$ ,  $Y_i^h \subset Y_i$ ,  $i = 1, 2, 3$  and  $Z_i^h \subset Z_i$ ,  $i = 1, 3$  to write the finite element method:

*Find*  $(\mathbf{u}_1^h, p_1^h, \mathbf{u}_2^h, p_2^h, \mathbf{u}_3^h, p_3^h, \lambda_1^h, \lambda_3^h) \in X_{1,D}^h \times Y_1^h \times X_{2,0}^h \times Y_2^h \times X_{3,D}^h \times Y_3^h \times Z_1^h \times Z_3^h$   
*such that*

$$\begin{aligned}
a_1(\mathbf{u}_1^h, \mathbf{w}_1^h) - b_1(\mathbf{w}_1^h, p_1^h) - c_1(\mathbf{w}_1^h, \lambda_1^h) &= (\mathbf{f}_1, \mathbf{w}_1^h) \quad \forall \mathbf{w}_1^h \in X_{1,0}^h, \\
-b_1(\mathbf{u}_1^h, q_1^h) &= 0 \quad \forall q_1^h \in Y_1^h, \\
a_2(\mathbf{u}_2^h, \mathbf{w}_2^h) - b_2(\mathbf{w}_2^h, p_2^h) - c_{12}(\mathbf{w}_2^h, \lambda_1^h) - c_{32}(\mathbf{w}_2^h, \lambda_3^h) &= 0 \quad \forall \mathbf{w}_2^h \in X_{2,0}^h, \\
-b_2(\mathbf{u}_2^h, q_2^h) &= -(f_2, q_2^h) \quad \forall q_2^h \in Y_2^h, \\
a_3(\mathbf{u}_3^h, \mathbf{w}_3^h) - b_3(\mathbf{w}_3^h, p_3^h) - c_3(\mathbf{w}_3^h, \lambda_3^h) &= (\mathbf{f}_3, \mathbf{w}_3^h) \quad \forall \mathbf{w}_3^h \in X_{3,0}^h, \\
-b_3(\mathbf{u}_3^h, q_3^h) &= 0 \quad \forall q_3^h \in Y_3^h, \\
-c_1(\mathbf{u}_1^h, \nu_1^h) - c_{12}(\mathbf{u}_2^h, \nu_1^h) &= 0 \quad \forall \nu_1^h \in Z_1^h, \\
-c_3(\mathbf{u}_3^h, \nu_3^h) - c_{32}(\mathbf{u}_2^h, \nu_3^h) &= 0 \quad \forall \nu_3^h \in Z_3^h.
\end{aligned}$$

Assuming  $\{\varphi_j^i\}_{j=1,\dots,N_i}$  are the bases of  $X_i^h$ , and  $\mathbf{U}_i$  are the vectors of coefficients in the basis expansions of  $\mathbf{u}_i^h$ ,  $\{\psi_m^i\}_{m=1,\dots,M_i}$  are the bases of  $Y_i^h$ , and  $\mathbf{P}_i$  are the vectors of coefficients in the basis expansions of  $p_i^h$ ,  $i = 1, 2, 3$ ,  $\{\xi_r^i\}_{r=1,\dots,R_i}$  are the bases of  $Z_i$ , and  $\mathbf{A}_i$  are the vectors of coefficients in basis expansions of  $\lambda_i^h$ ,  $i = 1, 3$ , the finite element method can be written in matrix form (6.7), with component matrices and vectors constructed as follows:

$$\begin{aligned}
\mathbf{A}_i &\in \mathbb{R}^{N_i \times N_i} : (\mathbf{A}_i)_{j,k} = a_i(\varphi_j^i, \varphi_k^i), \quad i = 1, 2, 3, \\
\mathbf{B}_i &\in \mathbb{R}^{N_i \times M_i} : (\mathbf{B}_i)_{j,m} = b_i(\varphi_j^i, \psi_m^i), \quad i = 1, 2, 3, \\
\mathbf{C}_i &\in \mathbb{R}^{N_i \times R_i} : (\mathbf{C}_i)_{j,r} = c_i(\varphi_j^i, \xi_r^i), \quad i = 1, 3,
\end{aligned}$$

$$\mathbf{C}_{i2} \in \mathbb{R}^{N_2 \times R_i} : (\mathbf{C}_{i2})_{j,r} = c_{i2}(\varphi_j^2, \xi_r^i), \quad i = 1, 3,$$

$$\mathbf{F}_i \in \mathbb{R}^{N_i} : \quad (\mathbf{F}_i)_j = (f_i, \varphi_j^i), \quad i = 1, 2, 3,$$

$$\left( \begin{array}{cc|cc|cc|cc} \mathbf{A}_1 & -\mathbf{B}_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{C}_1 & \mathbf{0} \\ -\mathbf{B}_1^T & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{A}_2 & -\mathbf{B}_2 & \mathbf{0} & \mathbf{0} & -\mathbf{C}_{12} & -\mathbf{C}_{32} \\ \mathbf{0} & \mathbf{0} & -\mathbf{B}_2^T & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A}_3 & -\mathbf{B}_3 & \mathbf{0} & -\mathbf{C}_3 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{B}_3^T & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline -\mathbf{C}_1^T & \mathbf{0} & -\mathbf{C}_{12}^T & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -\mathbf{C}_{32}^T & \mathbf{0} & -\mathbf{C}_3^T & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{array} \right) \begin{pmatrix} \mathbf{U}_1 \\ \mathbf{P}_1 \\ \mathbf{U}_2 \\ \mathbf{P}_2 \\ \mathbf{U}_3 \\ \mathbf{P}_3 \\ \mathbf{\Lambda}_1 \\ \mathbf{\Lambda}_3 \end{pmatrix} = \begin{pmatrix} \mathbf{F}_1 \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{F}_2 \\ \mathbf{F}_3 \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix}, \quad (6.7)$$

The system has clear saddle-point block structure. While the diagonal saddle-point blocks for the Stokes and Darcy flows appear similar, the difference in operators  $a_1$ ,  $a_3$  and  $a_2$  results in matrices with elements of vastly different orders of magnitude for small mesh sizes.

## 6.2 Finite element spaces

As mentioned above, the choice of the approximation spaces for the coupled Stokes-Darcy problem requires either a combination of different stable elements, as used in [7, 40, 51, 61, 86], or a stabilised formulation using the same elements, but possibly different values for the stabilisation parameters, as presented in [11, 27, 34, 81]. While the effect of using a different type of finite elements and/or stabilisation on both the block solver and the error estimation formula requires further investigation, in this

work the priority was given to consideration of different physical laws. Assuming higher than  $H(\text{div})$ , specifically  $H^1$ , regularity of velocities in the porous region, we may apply the  $P2 - P1$  approximation in both the Stokes and the Darcy domains [28] in order to use the existing implementation of the finite element method.

### 6.3 Solver

In order to make the notation somewhat consistent with that used in Chapter 4 we define the following matrices and vectors:  $\hat{\mathbf{A}}_i \in \mathbb{R}^{(N_i+M_i) \times (N_i+M_i)}$ ,  $\hat{\mathbf{C}}_i \in \mathbb{R}^{(N_i+M_i) \times (R_1+R_3)}$ ,  $\hat{\mathbf{U}}_i \in \mathbb{R}^{N_i+M_i}$ ,  $\mathbf{F}_i \in \mathbb{R}^{N_i+M_i}$ ,  $i = 1, 2, 3$ , and  $\hat{\mathbf{\Lambda}} \in \mathbb{R}^{R_1+R_3}$  such that

$$\begin{aligned}\hat{\mathbf{A}}_i &= \begin{pmatrix} \mathbf{A}_i & -\mathbf{B}_i \\ -\mathbf{B}_i^T & \mathbf{0} \end{pmatrix}, \quad i = 1, 2, 3, \\ \hat{\mathbf{C}}_1 &= \begin{pmatrix} -\mathbf{C}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}, \quad \hat{\mathbf{C}}_2 = \begin{pmatrix} \mathbf{C}_{12} & \mathbf{C}_{32} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}, \quad \hat{\mathbf{C}}_3 = \begin{pmatrix} \mathbf{0} & -\mathbf{C}_3 \\ \mathbf{0} & \mathbf{0} \end{pmatrix}, \\ \hat{\mathbf{U}}_i &= \begin{pmatrix} \mathbf{U}_i \\ \mathbf{P}_i \end{pmatrix}, \quad i = 1, 2, 3, \quad \hat{\mathbf{\Lambda}} = \begin{pmatrix} \mathbf{\Lambda}_1 \\ \mathbf{\Lambda}_3 \end{pmatrix}, \\ \hat{\mathbf{F}}_1 &= \begin{pmatrix} \mathbf{F}_1 \\ \mathbf{0} \end{pmatrix}, \quad \hat{\mathbf{F}}_2 = \begin{pmatrix} \mathbf{0} \\ \mathbf{F}_2 \end{pmatrix}, \quad \hat{\mathbf{F}}_3 = \begin{pmatrix} \mathbf{F}_3 \\ \mathbf{0} \end{pmatrix}.\end{aligned}$$

The right preconditioner  $\mathbf{P}^{-1}$  for the block system, as used in equation (4.2) has

the following form:

$$\mathbf{P}^{-1} = \begin{pmatrix} \hat{\mathbf{A}}_1^{-1} & \mathbf{0} & \mathbf{0} & \hat{\mathbf{A}}_1^{-1}\hat{\mathbf{C}}_1\hat{\mathbf{S}}^{-1} \\ \mathbf{0} & \hat{\mathbf{A}}_2^{-1} & \mathbf{0} & \hat{\mathbf{A}}_2^{-1}\hat{\mathbf{C}}_2\hat{\mathbf{S}}^{-1} \\ \mathbf{0} & \mathbf{0} & \hat{\mathbf{A}}_3^{-1} & \hat{\mathbf{A}}_3^{-1}\hat{\mathbf{C}}_3\hat{\mathbf{S}}^{-1} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & -\hat{\mathbf{S}}^{-1} \end{pmatrix}, \quad (6.8)$$

where the approximation to the inverse of the Schur complement is computed based on  $\mathbf{S}_4$  and  $\mathbf{S}_5$  from Chapter 4.

$$\hat{\mathbf{S}}_4^{-1} = \hat{\mathbf{C}}_{2,I}^{-T} \left( \hat{\mathbf{A}}_{2,II} - \hat{\mathbf{A}}_{2,ID} \hat{\mathbf{A}}_{2,DD}^{-1} \hat{\mathbf{A}}_{2,DI} \right) \hat{\mathbf{C}}_{2,I}^{-1}. \quad (6.9)$$

Here  $\hat{\mathbf{C}}_{2,I}$  is the submatrix of  $\hat{\mathbf{C}}_2$  only containing the rows corresponding to the interface degrees of freedom, and similarly  $\hat{\mathbf{A}}_{2,II}$ ,  $\hat{\mathbf{A}}_{2,ID}$ ,  $\hat{\mathbf{A}}_{2,DD}$ ,  $\hat{\mathbf{A}}_{2,DI}$  are the submatrices of  $\hat{\mathbf{A}}_2$  containing the rows and columns corresponding to the respective degrees of freedom.

The porous block is more suitable for use in this geometric setup due to its coupling to both other domains. In terms of matrices this means that the coupling matrix  $\hat{\mathbf{C}}_2$  used in the approximation will have a non-singular block on the diagonal, corresponding to both Lagrange multipliers  $\lambda_1$  and  $\lambda_3$ . The other coupling blocks  $\mathbf{C}_1$  and  $\mathbf{C}_3$  only have nonzero entries in the blocks corresponding to  $\lambda_1$  or  $\lambda_3$  respectively. As in the fluid-structure interaction example in Chapter 4, only those degrees of freedom that correspond to velocities are included in the ‘interface’ indices  $I$ .

Defining

$$\hat{\mathbf{A}}_F = \begin{pmatrix} \hat{\mathbf{A}}_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \hat{\mathbf{A}}_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \hat{\mathbf{A}}_3 \end{pmatrix}, \quad \hat{\mathbf{C}}_F = \begin{pmatrix} \hat{\mathbf{C}}_1 \\ \hat{\mathbf{C}}_2 \\ \hat{\mathbf{C}}_3 \end{pmatrix},$$

we can also write

$$\hat{\mathbf{S}}_5^{-1} = (\hat{\mathbf{C}}_F^T \hat{\mathbf{C}}_F)^{-1} (\hat{\mathbf{C}}_F^T \hat{\mathbf{A}}_F \hat{\mathbf{C}}_F) (\hat{\mathbf{C}}_F^T \hat{\mathbf{C}}_F). \quad (6.10)$$

The blocks  $\hat{\mathbf{A}}_i$ ,  $i = 1, 2, 3$  are all saddle-point matrices in their own right. In the numerical examples in this section we compute the action of their inverses using a direct method due to implementation restrictions, specifically because the code was written in Matlab as proof of concept rather than as part of a scientific software development project. As mentioned previously, an efficient implementation relies on efficient component solvers.

## 6.4 Error estimation and mesh refinement

We write the quantity of interest for the Stokes-Darcy-Stokes coupled problem (6.6) as

$$(QoI) = \sum_{i=1}^3 \int_{\Omega_i} (\psi_{i,u} \cdot \mathbf{u}_i + \psi_{i,p} p_i) \, d\mathbf{x} + \sum_{i=1,3} \int_{\Gamma_{i2}} \psi_{i,\lambda} \lambda_i \, ds, \quad (6.11)$$

where  $\psi_{i,u}$ ,  $\psi_{i,p}$ ,  $i = 1, 2, 3$ , and  $\psi_{i,\lambda}$ ,  $i = 1, 3$  are coefficient functions of appropriate dimensions.

Let the errors in the finite element solutions for each of the problem variables be denoted respectively by  $\mathbf{e}_{i,u} = \mathbf{u}_i - \mathbf{u}_i^h$ ,  $e_{i,p} = p_i - p_i^h$ ,  $i = 1, 2, 3$ , and  $e_{i,\lambda} = \lambda_i - \lambda_i^h$ ,

$i = 1, 3$ . We aim to control the error in the quantity of interest, given by

$$\mathcal{E} = \sum_{i=1}^3 \int_{\Omega_i} (\boldsymbol{\psi}_{i,u} \cdot \mathbf{e}_{i,u} + \psi_{i,p} e_{i,p}) \, d\mathbf{x} + \sum_{i=1,3} \int_{\Gamma_{i2}} \psi_{i,\lambda} e_{i,\lambda} \, ds.$$

All equations in the problem (strong, weak, or discretised) are linear. Moreover, if the permeability tensor  $\mathbf{K}$  is symmetric, then all operators  $a_i(\cdot, \cdot)$ ,  $i = 1, 3$ , are symmetric, and therefore the matrix system (6.7) can be written so that  $\mathbf{A}_i$ ,  $i = 1, 3$  are symmetric\*. In the following derivation we assume that  $\mathbf{K}$  is indeed symmetric and the differential operator is self-adjoint, so as not to introduce additional notation in the adjoint problem. The representation formula for the non-symmetric case is obtained in exactly the same way.

The adjoint problem for (6.6) and the quantity of interest (6.11) can be written as follows:

*Find  $(\mathbf{v}_1, r_1, \mathbf{v}_2, r_2, \mathbf{v}_3, r_3, \xi_1, \xi_3) \in X_{1,0} \times Y_1 \times X_{2,0} \times Y_2 \times X_{3,0} \times Y_3 \times Z_1 \times Z_3$  such that*

$$\begin{aligned} a_1(\mathbf{v}_1, \mathbf{w}_1) - b_1(\mathbf{w}_1, r_1) - c_1(\mathbf{w}_1, \xi_1) &= (\boldsymbol{\psi}_{1,u}, \mathbf{w}_1) \quad \forall \mathbf{w}_1 \in X_{1,0}, \\ -b_1(\mathbf{v}_1, q_1) &= (\psi_{1,p}, q_1) \quad \forall q_1 \in Y_1, \\ a_2(\mathbf{v}_2, \mathbf{w}_2) - b_2(\mathbf{w}_2, r_2) + c_{12}(\mathbf{w}_2, \xi_1) + c_{32}(\mathbf{w}_2, \xi_3) &= (\psi_{2,u}, \mathbf{w}_2) \quad \forall \mathbf{w}_2 \in X_{2,0}, \\ -b_2(\mathbf{v}_2, q_2) &= (\psi_{2,p}, q_2), \quad \forall q_2 \in Y_2, \\ a_3(\mathbf{v}_3, \mathbf{w}_3) - b_3(\mathbf{w}_3, r_3) - c_3(\mathbf{w}_3, \xi_3) &= (\boldsymbol{\psi}_{3,u}, \mathbf{w}_3) \quad \forall \mathbf{w}_3 \in X_{3,0}, \\ -b_3(\mathbf{v}_3, q_3) &= (\psi_{3,p}, q_3) \quad \forall q_3 \in Y_3, \\ -c_1(\mathbf{v}_1, \nu_1) + c_{12}(\mathbf{v}_2, \nu_1) &= (\psi_{1,\lambda}, \nu_1) \quad \forall \nu_1 \in Z_1, \end{aligned}$$

---

\*The matrix adjustment required to achieve symmetry is due to Dirichlet boundary conditions, and is achieved via trivial linear transformations.

$$-c_3(\mathbf{v}_3, \nu_3) + c_{32}(\mathbf{v}_2, \nu_3) = (\psi_{3,\lambda}, \nu_3) \quad \forall \nu_3 \in Z_3.$$

Following Theorem 5.1.1, we can derive the error representation formula by using the error terms as test functions in the adjoint above. Namely, we have

$$\begin{aligned} a_i(\mathbf{v}_i, \mathbf{u}_i - \mathbf{u}_i^h) - b_i(\mathbf{u}_i - \mathbf{u}_i^h, r_i) - c_i(\mathbf{u}_i - \mathbf{u}_i^h, \xi_i) &= (\psi_{i,u}, \mathbf{e}_{i,u}), \quad i = 1, 3, \\ a_2(\mathbf{v}_2, \mathbf{u}_2 - \mathbf{u}_2^h) - b_2(\mathbf{u}_2 - \mathbf{u}_2^h, r_2) + c_{12}(\mathbf{u}_2 - \mathbf{u}_2^h, \xi_1) + c_{32}(\mathbf{u}_2 - \mathbf{u}_2^h, \xi_3) &= (\psi_{2,u}, \mathbf{e}_{2,u}), \\ -b_i(\mathbf{v}_i, p_i - p_i^h) &= (\psi_{i,p}, e_{i,p}), \quad i = 1, 2, 3, \\ -c_i(\mathbf{v}_i, \lambda_i - \lambda_i^h) + c_{i2}(\mathbf{v}_2, \lambda_i - \lambda_i^h) &= (\psi_{i,\lambda}, e_{i,\lambda}), \quad i = 1, 3. \end{aligned}$$

We add all these equations together, and use the symmetry of  $a_i(\cdot, \cdot)$ ,  $i = 1, 2, 3$ , and the linearity of all operators to obtain

$$\begin{aligned} \mathcal{E} &= \sum_{i=1}^3 (\psi_{i,u}, \mathbf{e}_{i,u}) + \sum_{i=1}^3 (\psi_{i,p}, e_{i,p}) + \sum_{i=1,3} (\psi_{i,\lambda}, e_{i,\lambda}) \\ &= a_1(\mathbf{v}_1, \mathbf{u}_1) - b_1(\mathbf{v}_1, p_1) - c_1(\mathbf{v}_1, \lambda_1) \quad [ = (\mathbf{f}_1, \mathbf{v}_1) \text{ from (6.6a)}] \\ &+ a_2(\mathbf{v}_2, \mathbf{u}_2) - b_2(\mathbf{v}_2, p_2) + c_{12}(\mathbf{v}_2, \lambda_1) + c_{32}(\mathbf{v}_2, \lambda_3) \quad [ = 0 \text{ from (6.6c)}] \\ &+ a_3(\mathbf{v}_3, \mathbf{u}_3) - b_3(\mathbf{v}_3, p_3) - c_3(\mathbf{v}_3, \lambda_3) \quad [ = (\mathbf{f}_3, \mathbf{v}_3) \text{ from (6.6e)}] \\ &- b_1(\mathbf{u}_1, r_1) - b_2(\mathbf{u}_2, r_2) - b_3(\mathbf{u}_3, r_3) \quad [ = -(f_2, r_2) \text{ from (6.6b,d,f)}] \\ &- c_1(\mathbf{u}_1, \xi_1) + c_{12}(\mathbf{u}_2, \xi_1) \quad [ = 0 \text{ from (6.6g)}] \\ &- c_3(\mathbf{u}_3, \xi_3) + c_{32}(\mathbf{u}_2, \xi_3) \quad [ = 0 \text{ from (6.6h)}] \\ &- a_1(\mathbf{u}_1^h, \mathbf{v}_1) + b_1(\mathbf{v}_1, p_1^h) + c_1(\mathbf{v}_1, \lambda_1^h) \\ &- a_2(\mathbf{u}_2^h, \mathbf{v}_2) + b_2(\mathbf{v}_2, p_2^h) - c_{12}(\mathbf{v}_2, \lambda_1^h) - c_{32}(\mathbf{v}_2, \lambda_3^h) \\ &- a_3(\mathbf{u}_3^h, \mathbf{v}_3) + b_3(\mathbf{v}_3, p_3^h) + c_3(\mathbf{v}_3, \lambda_3^h) \end{aligned}$$



$$\begin{aligned}
& + b_1(\mathbf{u}_1^h, r_1) + b_2(\mathbf{u}_2^h, r_2) + b_3(\mathbf{u}_3^h, r_3) \\
& + c_1(\mathbf{u}_1^h, \xi_1) - c_{12}(\mathbf{u}_2^h, \xi_1) \\
& + c_3(\mathbf{u}_3^h, \xi_3) - c_{32}(\mathbf{u}_2^h, \xi_3).
\end{aligned}$$

We define the following finite element residuals:

$$\begin{aligned}
R_{i,u} &= (\mathbf{f}_i, \mathbf{v}_i) - a_i(\mathbf{u}_i^h, \mathbf{v}_i) + b_i(\mathbf{v}_i, p_i^h) + c_i(\mathbf{v}_i, \lambda_i^h), \quad i = 1, 3, \\
R_{2,u} &= -a_2(\mathbf{u}_2^h, \mathbf{v}_2) + b_2(\mathbf{v}_2, p_2^h) - c_{12}(\mathbf{v}_2, \lambda_1^h) - c_{32}(\mathbf{v}_2, \lambda_3^h), \\
R_{i,p} &= b_i(\mathbf{u}_i^h, r_i), \quad i = 1, 3, \\
R_{2,p} &= -(f_2, r_2) + b_2(\mathbf{u}_2^h, r_2), \\
R_{i,\lambda} &= c_i(\mathbf{u}_i^h, \xi_i) - c_{i2}(\mathbf{u}_2^h, \xi_i), \quad i = 1, 3.
\end{aligned}$$

The error in the quantity of interest can therefore be represented as the sum of residuals

$$\mathcal{E} = \sum_{i=1}^3 (R_{i,u} + R_{i,p}) + \sum_{i=1,3} R_{i,\lambda}.$$

This formula is based on the exact adjoint solution, and therefore cannot be used in practice. Instead, we calculate all of the residuals using the approximate adjoint solution obtained on a finer mesh than the original solution, and compute the estimated error  $\mathcal{E}_{est}$ . The local mesh refinement strategy based on this estimate is applied exactly as described in Chapter 5.1.3.

## 6.5 Results

Below we present the results of numerical tests performed to evaluate the block solvers and the error estimation/mesh refinement techniques. A set of three-compartment geometries is studied: a straight channel, a channel with a linear expansion and contraction, and a channel with sudden expansion and contraction.

In all problems first we perform a GMRES convergence test using the block preconditioner (6.8) with  $\mathbf{S}_4^{-1}$  (6.9) and  $\mathbf{S}_5^{-1}$  (6.10) approximations to the Schur complement. We consider uniform meshes and the same orders of finite elements across the domains, and hence the choice is motivated purely by the structure of the coupling matrices.

Next we compute error estimates and perform local mesh refinement for a number of quantities of interest, based on these estimates. The values are then recomputed on refined meshes to assess how well the algorithm performed.

### 6.5.1 Straight channel

First we consider a straight channel with a rectangular cross-section, as shown in Figure 6.1. The components are as follows:  $\Omega_1 = (0, 1) \times (0, 1)$ ,  $\Omega_2 = (1, 2) \times (0, 1)$  and  $\Omega_3 = (2, 3) \times (0, 1)$ . No external forces are applied in the system, i.e.  $\mathbf{f}_1 = \mathbf{f}_3 = \mathbf{0}$  and  $f_2 = 0$ . The material parameters are all taken to be 1. We prescribe a parabolic inflow profile at  $x = 0$ :  $\mathbf{u}_1 = (0, y(1 - y))^T$ , and allow free flow at  $x = 3$ . The original mesh with  $DOF = 331$  in each compartment, and the solution on this mesh are shown in Figures 6.2b, 6.2a and 6.2c.

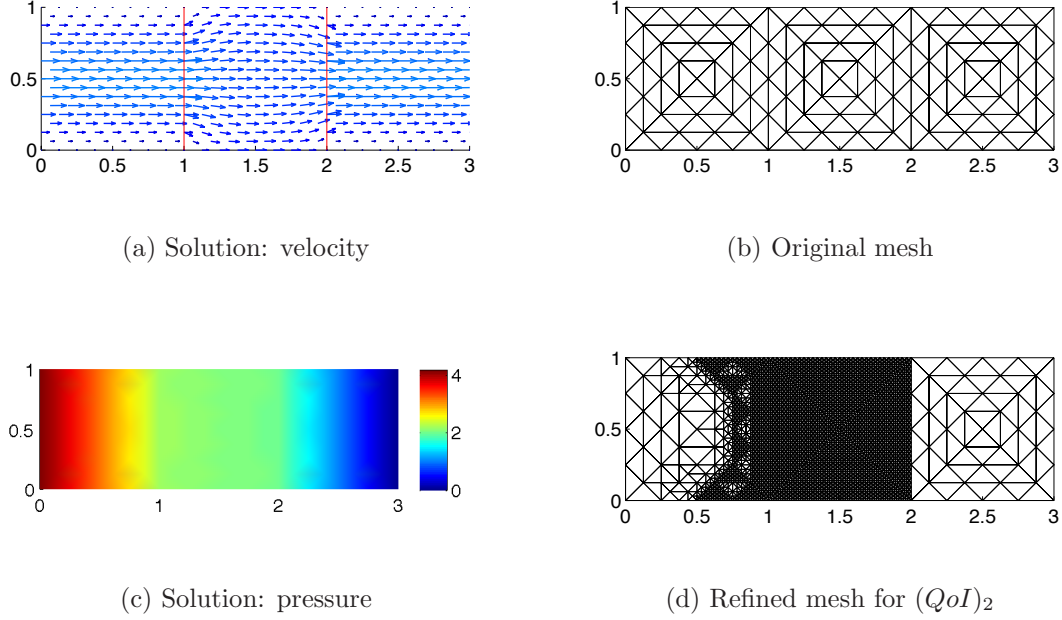


Figure 6.2: Straight channel with parabolic inflow,  $\mu = 1$ ,  $\mathbf{K} = \mathbf{I}$ .

#### 6.5.1.1 Solver

The preconditioned GMRES with  $\mathbf{S}_4^{-1}$  and  $\mathbf{S}_5^{-1}$  approximations to the inverse of the Schur complement was used to solve the problem on progressively finer meshes. Uniform refinements were used for convergence tests. The mesh sizes and iteration counts for  $TOL = 10^{-10}$  are given in Table 6.1.

| Mesh | $DOF$  | $DOF_\Gamma$ | $\mathbf{S}_4$ | $\mathbf{S}_5$ |
|------|--------|--------------|----------------|----------------|
| orig | 1011   | 18           | 11             | 11             |
| 1    | 3739   | 34           | 14             | 15             |
| 2    | 14379  | 66           | 17             | 21             |
| 3    | 56395  | 130          | 18             | 27             |
| 4    | 223371 | 258          | 17             | 34             |
| 5    | 630731 | 514          | 16             | 42             |

Table 6.1: Iteration counts for preconditioned GMRES convergence to  $TOL = 10^{-10}$  in the straight channel problem.  $DOF$  corresponds to the total number of degrees of freedom in the system,  $DOF_\Gamma$  to the size of the Schur complement.

The behaviour of the solver is consistent with that observed for problems in Chapter 4. The use of  $\mathbf{S}_5^{-1}$  leads to relatively low iteration counts, which grow with the

mesh size, while the use of  $\mathbf{S}_4^{-1}$  leads to mesh-independent convergence of preconditioned GMRES. The results suggest that the block preconditioner can be effectively used for systems with more than two components, and for systems of larger size, where the application of direct solvers is not feasible.

### 6.5.1.2 Error estimation and mesh refinement

For the mesh refinement algorithm we use the global tolerance  $TOL = 0.001$ , and allow at most three refinements of each element.

First we consider the quantity of interest producing the average velocity in the fluid domains:

$$(QoI)_1 = \int_{\Omega_1} \mathbf{u}_1 \, d\mathbf{x} + \int_{\Omega_3} \mathbf{u}_3 \, d\mathbf{x}.$$

The fluid velocities in this problem are parabolic throughout the domain (see Figure 6.2a), i.e.  $\mathbf{u}_1$  and  $\mathbf{u}_3$  lie in the finite element space. Consequently  $(QoI)_1$  should be computed up to machine precision. Indeed, the estimated error is  $\mathcal{E}_{est} \sim 10^{-17}$ .

Next we consider the quantity of interest producing the average normal stress/pressure on the interfaces:

$$\begin{aligned} (QoI)_2 &= \int_{\Gamma_{12}} \lambda_1 \, ds + \int_{\Gamma_{32}} \lambda_3 \, ds = - \int_{\Gamma_{12}} p_2 \, ds - \int_{\Gamma_{32}} p_2 \, ds \\ &= \int_{\Gamma_{12}} [(\mu \nabla \mathbf{u}_1 - p_1 \mathbf{I}) \mathbf{n}] \cdot \mathbf{n} \, ds + \int_{\Gamma_{32}} [(\mu \nabla \mathbf{u}_3 - p_3 \mathbf{I}) \mathbf{n}] \cdot \mathbf{n} \, ds. \end{aligned}$$

The values of the estimated error in this quantity of interest before and after refinement is given in Table 6.2, and the refined mesh is shown in Figure 6.2d. The refinement algorithm reduces the estimated error by almost an order of magnitude. Most of the refinements occur in  $\Omega_2$ , where the dual solution is non-zero. Since the flow is moving from left to right, there are refinements occurring in  $\Omega_1$  near the

interface  $\Gamma_{12}$ , where the accuracy of the solution affects the accuracy in the quantity of interest. At the same time, the flow in  $\Omega_3$  has already passed the interfaces, and thus does not affect the estimated error, and the domain does not need any refinement. The system based on the uniformly refined mesh producing the error of the same order contains 56395 degrees of freedom, which is more than twice as many as the adaptively refined system contains.

| Mesh | $DOF_{1/2/3}$  | $10^2 \mathcal{E}_{est}$ |
|------|----------------|--------------------------|
| orig | 331/331/331    | -3.0398                  |
| ref  | 6349/18755/331 | -0.4184                  |

Table 6.2: The estimated error in the quantity of interest on the original (orig) and adaptively refined (ref) mesh for the straight channel problem.  $DOF_{1/2/3}$  represents the number of degrees of freedom in each of the domains  $\Omega_1$ ,  $\Omega_2$  and  $\Omega_3$ .

### 6.5.2 Channel with linear expansion and contraction

Now the domain is changed to represent a channel that expands towards the porous block. The subdomains can be described as

$$\Omega_1 = \{(x, y) \mid 0 < x < 1, \min(0.25, 0.5 - 0.5x) < y < \max(0.75, 0.5 + 0.5x)\},$$

$$\Omega_2 = (1, 2) \times (0, 1),$$

$$\Omega_3 = \{(x, y) \mid 2 < x < 3, \min(0.25, 0.5x - 1) < y < \max(0.75, 2 - 0.5x)\}.$$

The inflow profile is changed to  $\mathbf{u}_1 = (0, (y - 0.25)(0.75 - y))^T$  to remain consistent with the no-slip boundary conditions at the walls of the channel.

The original mesh with  $DOF_1 = DOF_3 = 721$  and  $DOF_2 = 331$  is shown in Figure 6.3b, the velocity profiles in Figure 6.3a and 6.3c.

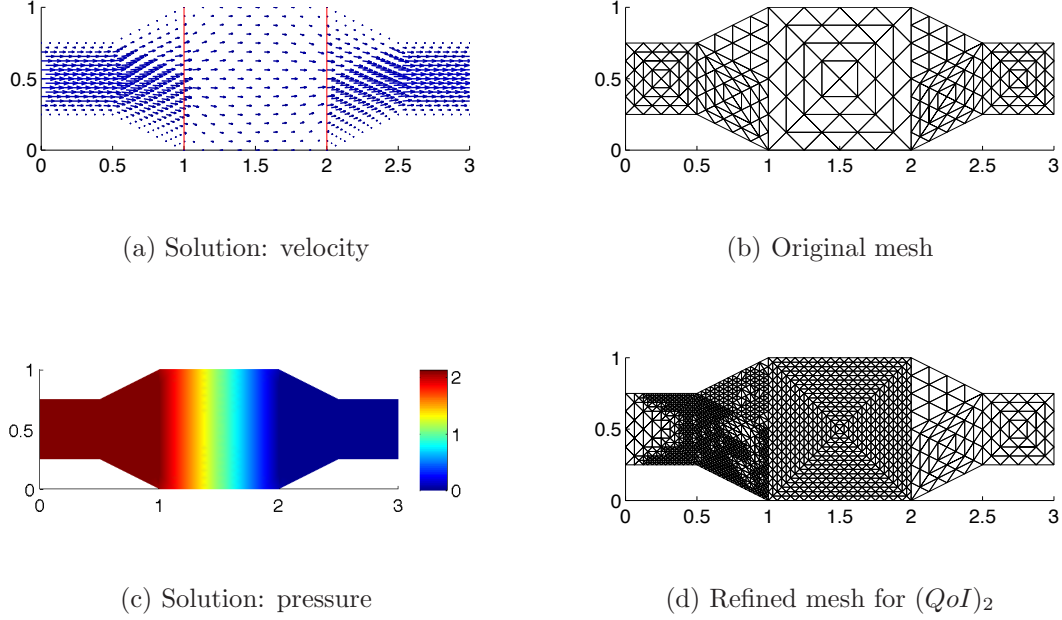


Figure 6.3: Channel with linear expansion and contraction, parabolic inflow,  $\mu = 1$ ,  $\mathbf{K} = \mathbf{I}$ .

### 6.5.2.1 Solver

The iteration counts for GMRES convergence, and the respective mesh sizes are presented in Table 6.3. The solvers were tested in two parameter regimes:  $\mu = 1$  and  $\mathbf{K} = \mathbf{I}$ , and  $\mu = 0.01$  and  $\mathbf{K} = 0.01\mathbf{I}$ .

|      |        |                | $\mu = 1, \mathbf{K} = \mathbf{I}$ |                | $\mu = 0.01, \mathbf{K} = 0.01\mathbf{I}$ |                |
|------|--------|----------------|------------------------------------|----------------|-------------------------------------------|----------------|
| Mesh | $DOF$  | $DOF_{\Gamma}$ | $\mathbf{S}_4$                     | $\mathbf{S}_5$ | $\mathbf{S}_4$                            | $\mathbf{S}_5$ |
| orig | 2711   | 34             | 17                                 | 21             | 35                                        | 27             |
| 1    | 10307  | 66             | 17                                 | 26             | 47                                        | 35             |
| 2    | 40187  | 130            | 19                                 | 31             | 48                                        | 44             |
| 3    | 158699 | 258            | 18                                 | 38             | 49                                        | 56             |
| 4    | 630731 | 514            | 15                                 | 45             | 47                                        | 69             |

Table 6.3: Iteration counts for preconditioned GMRES convergence to  $TOL = 10^{-10}$  in the linear expansion problem for two sets of parameters:  $\mu = 1$ ,  $\mathbf{K} = \mathbf{I}$  and  $\mu = 0.01$ ,  $\mathbf{K} = 0.01\mathbf{I}$ .  $DOF$  corresponds to the total number of degrees of freedom in the system,  $DOF_{\Gamma}$  to the size of the Schur complement.

As before, the  $\mathbf{S}_4^{-1}$  approximation leads to bounded iteration counts for a set of

uniformly refined meshes, while the iteration counts resulting from the application of  $\mathbf{S}_5^{-1}$  grow with mesh size. However, lowering the viscosity and permeability has a worse effect on the performance of the solver with  $\mathbf{S}_4$  than  $\mathbf{S}_5$ , as in the case of fluid-structure interaction.

### 6.5.2.2 Error estimation and mesh refinement

We keep the global tolerance at  $TOL = 0.001$ , but change the maximum number of refinements in an element to two, so as to keep the system size relatively small. The refined mesh for  $(QoI)_2$  as above is shown in Figure 6.3d, and the mesh sizes and estimated errors for the original and refined meshes are given in Table 6.4. The original mesh is more refined in the free flow compartments than the corresponding meshes in the straight channel, hence the lower original error. By reducing the allowed number of refinements within each element we kept the total number of degrees of freedom in the refined mesh lower than the corresponding number for the straight channel. Nevertheless it was possible to reduce the error by two orders of magnitude.

| $(QoI)_2$ |               |                          | $(QoI)_3$ |                  |                          |
|-----------|---------------|--------------------------|-----------|------------------|--------------------------|
| Mesh      | $DOF_{1/2/3}$ | $10^3 \mathcal{E}_{est}$ | Mesh      | $DOF_{1/2/3}$    | $10^3 \mathcal{E}_{est}$ |
| orig      | 721/331/721   | 2.4945                   | orig      | 721/1235/721     | -1.0587                  |
| ref       | 7178/4771/721 | -0.0155                  | ref       | 7739/12585/10561 | -0.0352                  |

Table 6.4: The estimated error in the quantity of interest on the original (orig) and adaptively refined (ref) mesh for the channel with linear expansion and contraction.  $DOF_{1/2/3}$  represents the number of degrees of freedom in each of the domains  $\Omega_1$ ,  $\Omega_2$  and  $\Omega_3$ .

The mesh refinement algorithm produces refinements in  $\Omega_1$  near the interface  $\Gamma_{12}$ , and in all of  $\Omega_2$ . The results are similar to the straight channel case: the direction of the flow means that the accuracy of the variables on the interfaces is not affected by the accuracy in  $\Omega_3$ .

For comparison, we consider a problem with lower viscosity and permeability:  $\mu = 0.01$ ,  $\mathbf{K} = 0.01\mathbf{I}$ , and use the friction constant  $G = \sqrt{\mu K}/\alpha = 0.005$  [103]. The quantity of interest is chosen to approximate the normal stress/pressure at the centre of the second interface  $\Gamma_{23}$ :

$$(QoI)_3 = 5 \int_{\Gamma_{32}} \exp(-100(s - 0.5)^2) \lambda_3 \, ds \approx 5 \int_{\Gamma_{32}} \delta_{0.5}(s) \lambda_3 \, ds,$$

where the Gaussian approximates the Kronecker delta function at  $y = 0.5$ . The value  $\exp(-25) \sim 10^{-11}$ , hence we can assume that the Dirichlet boundary conditions at the walls of the channel are satisfied exactly. The error in  $(QoI)_3$  pre- and post-refinement, and the sizes of the corresponding meshes are given in Table 6.4. Once again the error is reduced by almost two orders of magnitude.

The refined mesh is shown in Figure 6.4d, and the original mesh is shown for reference in Figure 6.4b.

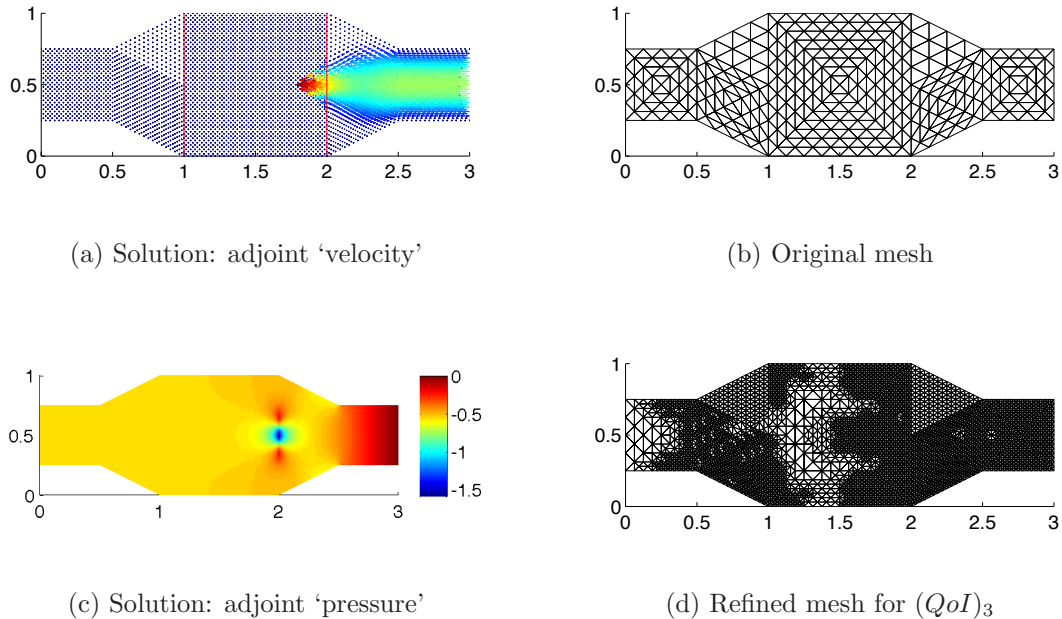


Figure 6.4: Channel with linear expansion and contraction, parabolic inflow,  $\mu = 0.01$ ,  $K = 0.01$ .



We can see that the point ‘source’ on the interface  $\Gamma_{32}$  introduces nontrivial behaviour in the adjoint solution. This behaviour is captured by the refinement in  $\Omega_3$ . At the same time,  $\Omega_2$  does not need to be refined as much as in the case of  $(QoI)_2$ , because the quantity of interest is only supported on a small section of  $\Gamma_{32}$ , not along the length of both interfaces.

### 6.5.3 Channel with sudden symmetric expansion and contraction

The last example we consider in this chapter looks at flow in a channel with sudden symmetric expansion. The geometry of the problem is as follows:

$$\Omega_1 = \{(x, y) \mid 0 < x < 0.5, 0.25 < y < 0.75\} \cup \{(x, y) \mid 0.5 < x < 1, 0 < y < 1\},$$

$$\Omega_2 = (1, 2) \times (0, 1),$$

$$\Omega_3 = \{(x, y) \mid 2 < x < 2.5, 0 < y < 1\} \cup \{(x, y) \mid 2.5 < x < 3, 0.25 < y < 0.75\}.$$

The inflow profile remains parabolic at the inlet:  $\mathbf{u}_1 = (0, (y - 0.25)(0.75 - y))^T$ .

Figure 6.5b shows the original mesh for this shape of the channel with  $DOF_1 = DOF_3 = 1173$  and  $DOF_2 = 1235$ . The velocity (Figure 6.3a) and the pressure solutions (Figure 6.3c) are similar to the previous cases, apart from the pressure pockets at the sudden expansion.

#### 6.5.3.1 Solver

Once again the block-preconditioned GMRES with  $\mathbf{S}_4^{-1}$  and  $\mathbf{S}_5^{-1}$  as approximations for the Schur complement was used to solve the discretised Stokes-Darcy-Stokes coupled equations. Table 6.5 shows the resulting iteration counts for two parameter regimes:

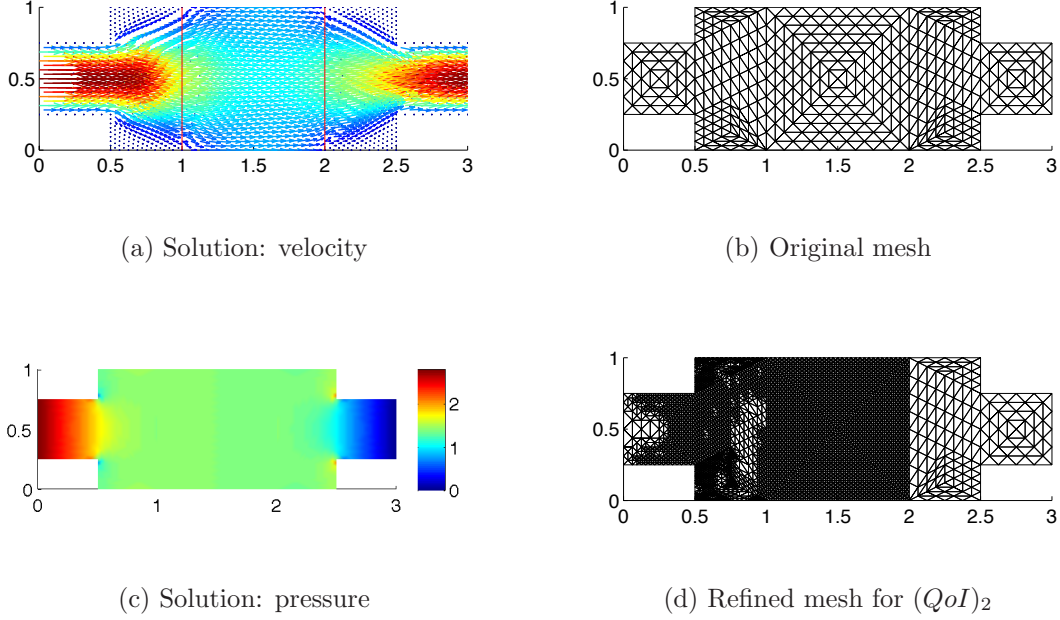


Figure 6.5: Channel with sudden symmetric expansion and contraction, parabolic inflow,  $\mu = 1$ ,  $\mathbf{K} = \mathbf{I}$ .

$\mu = 1$ ,  $\mathbf{K} = \mathbf{I}$  and  $\mu = 0.01$ ,  $\mathbf{K} = 0.01\mathbf{I}$ .

| Mesh | $DOF$  | $DOF_\Gamma$ | $\mu = 1, \mathbf{K} = \mathbf{I}$ |                | $\mu = 0.01, \mathbf{K} = 0.01\mathbf{I}$ |                |
|------|--------|--------------|------------------------------------|----------------|-------------------------------------------|----------------|
|      |        |              | $\mathbf{S}_4$                     | $\mathbf{S}_5$ | $\mathbf{S}_4$                            | $\mathbf{S}_5$ |
| orig | 3615   | 34           | 17                                 | 20             | 32                                        | 23             |
| 1    | 13843  | 66           | 18                                 | 23             | 36                                        | 31             |
| 2    | 54171  | 130          | 20                                 | 30             | 39                                        | 38             |
| 3    | 214315 | 258          | 18                                 | 36             | 40                                        | 45             |
| 4    | 852555 | 514          | 17                                 | 46             | 38                                        | 52             |

Table 6.5: Iteration counts for preconditioned GMRES convergence to  $TOL = 10^{-10}$  in the sudden symmetric expansion and contraction problem for two sets of parameters:  $\mu = 1$ ,  $\mathbf{K} = \mathbf{I}$  and  $\mu = 0.01$ ,  $\mathbf{K} = 0.01\mathbf{I}$ .  $DOF$  corresponds to the total number of degrees of freedom in the system,  $DOF_\Gamma$  to the size of the Schur complement.

The behaviour of the solvers is consistent with the previous tests. The use of  $\mathbf{S}_4^{-1}$  results in mesh-independent GMRES convergence, but leads to deterioration in iteration counts as viscosity and permeability decrease. The preconditioner based on  $\mathbf{S}_5^{-1}$  exhibits mesh-dependent convergence, while remaining competitive in absolute terms for the mesh sizes considered, and improves convergence with decreasing  $\mu$  and  $\mathbf{K}$ .

### 6.5.3.2 Error estimation and mesh refinement

For completeness we consider the same quantities of interest as in the previous section:

$$(QoI)_2 = \int_{\Gamma_{12}} \lambda_1 \, ds + \int_{\Gamma_{32}} \lambda_3 \, ds; \quad \mu = 1, \mathbf{K} = \mathbf{I}, G = 1,$$

$$(QoI)_3 = 5 \int_{\Gamma_{32}} \exp(-100(s - 0.5)^2) \lambda_3 \, ds; \quad \mu = 0.01, \mathbf{K} = 0.01\mathbf{I}, G = 0.005.$$

The behaviour of the refinement algorithm is consistent with the case of sloping walls, as attested by the refined meshes shown in Figures 6.5d and 6.6d, and the values of the error in the quantity of interest shown in Table 6.6

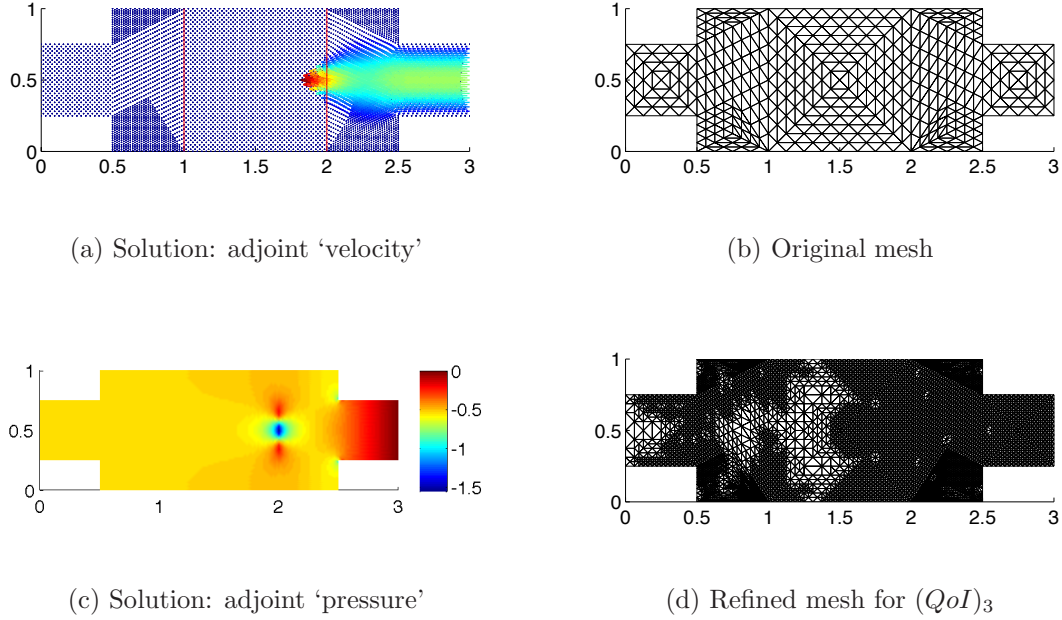


Figure 6.6: Channel with sudden symmetric expansion and contraction, parabolic inflow,  $\mu = 0.01$ ,  $K = 0.01$ .

A particularly interesting example for this configuration of the channel involves measuring the error in the pressure near a corner, where it changes rapidly as the

| $(QoI)_2$ |                  |                          | $(QoI)_3$ |                   |                          |
|-----------|------------------|--------------------------|-----------|-------------------|--------------------------|
| Mesh      | $DOF_{1/2/3}$    | $10^3 \mathcal{E}_{est}$ | Mesh      | $DOF_{1/2/3}$     | $10^3 \mathcal{E}_{est}$ |
| orig      | 1173/1235/1173   | -2.0031                  | orig      | 1173/1235/1173    | -1.0464                  |
| ref       | 12879/18755/1173 | -0.5511                  | ref       | 12557/13603/17021 | -0.0269                  |

Table 6.6: The estimated error in the quantity of interest on the original (orig) and adaptively refined (ref) mesh for the channel with sudden expansion and contraction.  $DOF_{1/2/3}$  represents the number of degrees of freedom in each of the domains  $\Omega_1$ ,  $\Omega_2$  and  $\Omega_3$ .

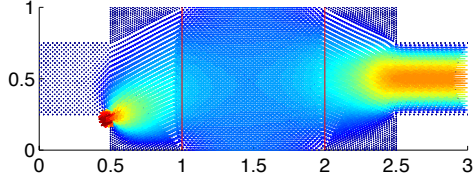
flow is allowed to expand. We use the following quantities of interest:

$$(QoI)_4 = 5 \int_{\Omega_1} \exp(-500(x - 0.55)^2 - 500(y - 0.25)^2) p_1 \, d\mathbf{x},$$

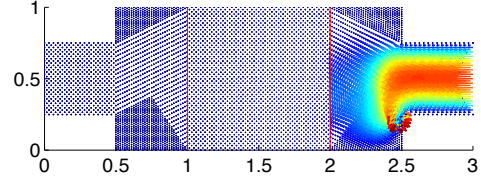
$$(QoI)_5 = 5 \int_{\Omega_3} \exp(-500(x - 2.45)^2 - 500(y - 0.25)^2) p_3 \, d\mathbf{x}.$$

The values of parameters are taken at  $\mu = 1$ ,  $\mathbf{K} = \mathbf{I}$ ,  $G = 1$ . The global tolerance  $TOL = 0.001$  and the number of refinements allowed per element (two) remained the same.

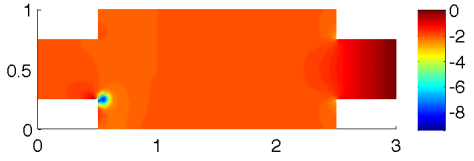
The adjoint solutions for each of the quantities of interest are shown in Figures 6.7a, 6.7c and 6.7b, 6.7d respectively. While the ‘pressure’ peak is highly localised in both cases, the ‘flow’ profiles driven by the forcing terms  $\psi_1$  and  $\psi_3$  are qualitatively different: in the case of  $(QoI)_4$  the adjoint solution propagates through all three subdomains, while in the case of  $(QoI)_5$  it is confined to  $\Omega_3$ . This translates directly into how the mesh is refined. Specifically, for the quantity of interest in  $\Omega_1$ , the refinements occur throughout the channel, while for the quantity of interest in  $\Omega_3$ , there is no need for much refinement in  $\Omega_2$ . In addition, the error  $\mathcal{E}$  is only reduced by a factor of  $\sim 1.5$  for  $(QoI)_4$ , as compared to  $\sim 20$  for  $(QoI)_5$  (see Table 6.7).



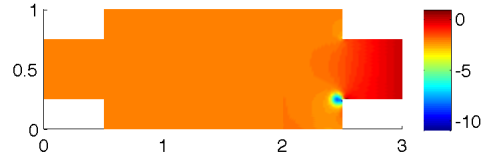
(a) Solution: adjoint 'velocity'



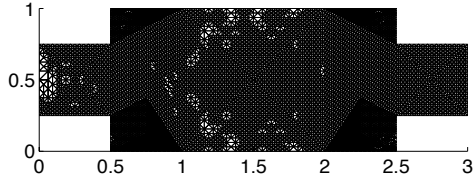
(b) Solution: adjoint 'velocity'



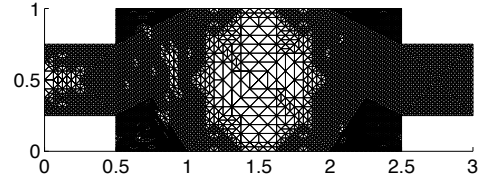
(c) Solution: adjoint 'pressure'



(d) Solution: adjoint 'pressure'



(e) Refined mesh for  $(QoI)_4$



(f) Refined mesh for  $(QoI)_5$

Figure 6.7: Channel with sudden symmetric expansion and contraction, parabolic inflow. Quantity of interest approximates point values of fluid pressure near the corners.

| $(QoI)_4$ |                   |                          | $(QoI)_5$ |                  |                          |
|-----------|-------------------|--------------------------|-----------|------------------|--------------------------|
| Mesh      | $DOF_{1/2/3}$     | $10^3 \mathcal{E}_{est}$ | Mesh      | $DOF_{1/2/3}$    | $10^3 \mathcal{E}_{est}$ |
| orig      | 1173/1235/1173    | -2.8805                  | orig      | 1173/1235/1173   | 1.2748                   |
| ref       | 16910/17487/17521 | -1.8467                  | ref       | 15990/8826/17476 | 0.0735                   |

Table 6.7: The estimated error in the quantity of interest on the original (orig) and adaptively refined (ref) mesh for the channel with sudden expansion and contraction.  $DOF_{1/2/3}$  represents the number of degrees of freedom in each of the domains  $\Omega_1$ ,  $\Omega_2$  and  $\Omega_3$ .

## 6.6 Discussion

In this chapter we showed that the theory developed in Chapters 4 and 5 can be naturally extended to other applications, and problems with more than two coupled

components. The Lagrange multiplier approach to coupling allows a great flexibility in solving the systems monolithically, i.e. without introducing decoupling in the method, yet taking full advantage of the block structure of such systems.

We considered a system where fluid flows in a channel with a permeable sector, like those in chemical or biological filters. The components were modelled as Stokes flow up to the filter, Darcy flow through the porous material, and Stokes flow beyond the filter. The coupling was achieved through the use of regular continuity conditions and the experimentally obtained Joseph-Beavers-Saffman slip condition on the interfaces.

The block preconditioner for GMRES based on inverse Schur complement approximations was shown to converge in accordance to the initial tests. The first approximation to the Schur complement inverse, denoted by  $\mathbf{S}_4^{-1}$ , results in mesh-independent convergence of preconditioned GMRES, thus providing a scalable strategy for solving large coupled systems. The second approximation, denoted by  $\mathbf{S}_5^{-1}$ , while lacking this property, is potentially better suited for flow systems with low viscosity, where its performance is improved.

The error estimator was used to compute the error in the traction on the interfaces between the components. This is potentially useful for engineering applications, where it is important to limit the error when assessing the stress the materials need to be able to withstand. The mesh refinement strategy allowed us to reduce the error by introducing additional degrees of freedom in the areas where the local errors were high for a given quantity of interest, instead of throughout the domain. For each problem this method produced smaller linear systems that needed to be solved in order to obtain a more accurate solution.

Both approaches have been successfully applied in the Stokes-Darcy-Stokes coupled problem, and show potential for use in other physics.

# Chapter 7

## Conclusion

### 7.1 Summary and conclusions

The task of modelling the behaviour of coupled system arises in a multitude of fields. In physiology, for example, one can consider the flow of blood in deformable human arteries as a combination of fluid and solid mechanics problems. In chemical engineering cartridge filters can be designed based on simulations of combined free and porous flow. The mathematical interpretation of the fundamental laws of physics allows computer simulation of these processes, most often with the use of numerical methods such as Galerkin finite elements. Detailed and accurate finite element modelling is achieved through using extremely large numbers of degrees of freedom in the discretisations of the systems considered, making their solution more challenging from the computational perspective. On the one hand, this challenge is met by increasing availability of high-performance systems. On the other hand, it is often possible to optimise the methods so that they produce a sufficiently accurate solution faster, or a more accurate solution in the same time, provided a certain set of computational resources. In this thesis we investigated strategies of improving the

efficiency of standard finite element solvers for coupled problems.

In Chapter 2 we gave a brief overview of the standard finite element methods, which were used throughout this thesis.

Chapter 3 described and compared the partitioned and the monolithic approaches to addressing coupling. We obtained explicit formulations for monolithic coupling through Lagrange multipliers in steady and time-dependent finite element methods. The robustness of the monolithic methods achieved through inclusion of all coupling terms in the problem, can in practice be overlooked due to the modularity of the partitioned approach, meaning that single physics solvers can be used without modification, as the problem is in some sense decoupled. We argued that modularity can be achieved in the monolithic case through the use of block preconditioners for linear solvers.

Such preconditioners were investigated in Chapter 4. Several approximations to the Schur complement block were proposed, and studied along with an existing approximation based on commutators [45]. An inverse approximation proposed here and denoted throughout as  $\mathbf{S}_4^{-1}$  was shown to have a relatively low associated computational cost if used in conjunction with an iterative linear solver, such as GMRES. The performance of this method was tested in a series of numerical tests for different types of coupled problems, including a two-dimensional example of fluid-structure interaction. Unlike the other approximations,  $\mathbf{S}_4^{-1}$  resulted in mesh-independent convergence of GMRES in all examples considered. We conclude that the block preconditioner based on  $\mathbf{S}_4^{-1}$  can provide an efficient solution method for large-scale monolithic coupled problems, where the number of GMRES iterations required for convergence would remain low.

In Chapter 5 we turned to adaptive mesh refinement, which addresses a different



aspect of improving the efficiency of coupled solvers. We derived *a posteriori* error estimators based on the solution of adjoint problems for steady, then time-dependent monolithic coupled problems, and employed it to drive adaptive space, then time-space refinement. The monolithic approach was shown to naturally lead to controls on errors on the coupling interface. The refinement strategy allowed us to build computational meshes tailored to specific problems by concentrating additional degrees of freedom where they helped reduce the errors in the quantities of interest. Generally speaking, such meshes have significantly smaller numbers of degrees of freedom than uniformly refined meshes leading to comparable errors. Consequently the computation of solutions on such meshes requires less memory and processing power. It is sometimes argued that the overhead of computing the adjoint solution is comparable to the overhead of solving the original problem on the finer mesh. However, a uniform mesh does not have the optimal distribution of degrees of freedom with regards to a specific quantity of interest, and therefore does not produce the most accurate solution that could be obtained using a given set of computational resources. We therefore concluded that the mesh refinement strategy developed here provides an advantage in solving problems where error control is required. The adjoint-based *a posteriori* error estimator is an effective tool for such control.

Chapter 6 revisited both methods in a different physical setting. Both the preconditioner and the error estimator were initially developed for generic coupled elliptic or parabolic problems. The extension of the methods to other types of equations was thought to be natural to a certain extent, but both the performance of the preconditioner and the form of the error estimator could change depending on the problem considered. Specifically we studied steady flow through a channel with a porous section in the middle modelling flow through a cartridge filter. Both methods were

reformulated in this setting, and the results in a series of numerical tests agreed with those obtained in the previous chapters. The use of the block preconditioner with the  $\mathbf{S}_4^{-1}$  approximation to the Schur complement led to mesh-independent convergence of GMRES. Similarly, the adjoint-based error estimator employed as part of the adaptive refinement strategy produced localised mesh refinements for various quantities of interest. The results showed that the methods developed in this thesis can be successfully employed in solving specific coupled problems.

## 7.2 Future work

### Applicability in more complex physical settings

When extending the methods developed in this thesis to the case of coupled Stokes-Darcy-Stokes equations in Chapter 6 we could see that the nature of the coupled components and the manner of coupling may necessitate modifications in either the approximation of the Schur complement used in the block preconditioner for the iterative linear solver, or the formula for the adjoint-based error estimate.

### Extensions for the solver

In the case of the preconditioner, its performance has only been established empirically, and is therefore not guaranteed for modified formulations. The efficiency of the method has been shown in a simple case of fluid-structure interaction in Chapter 4, but a further, more extensive investigation would be of practical value. The following aspects of a full fluid-structure interaction problem could have an effect on the solver performance, and would be worth studying:

- the nonlinearity in solid mechanics models applicable to large deformations,

- vastly different domain shapes for the fluid and solid equations, as in a model of a heart pumping blood,
- different types of elements used for approximation of the different components, e.g. a combination of elements of different orders, Taylor-Hood and Raviart-Thomas elements, tetrahedral and hexahedral elements.

### **Extensions for the error estimate and the mesh refinement strategy**

The error representation formulae derived in Chapters 5 and 6 are only directly applicable to linear differential operators. Building an adjoint for a nonlinear problem involves an additional approximation due to the necessary linearisation, which may change the estimate. Similarly, we have not considered here the effect of domain motion. Both complications arise in fluid-structure interaction problems, which would be an interesting application to consider. For example, when modelling blood flow in arteries it is often necessary to assess the stress exerted by the blood on arterial walls. The representation formula for quantities of interest on the interface would provide a way of directly estimating the error in this stress, and refining the meshes appropriately to reduce this error.

### **Implementation**

Practical use of the methods presented here relies heavily on their efficient implementation for large-scale problems. Since the focus of this work was in method development, the flexibility of the code and fast implementation were deemed more important than producing ready-to-use software. We conducted testing for two-dimensional problems with up to  $\sim 10^5$  degrees of freedom, where using Matlab on a standard personal computer was still feasible. In all such examples, the direct

solver implemented in Matlab (the backslash operator) outperformed user-designed preconditioned GMRES. The advantage of preconditioned iterative solvers becomes pronounced in larger (on more complex geometries, with  $\sim 10^8 - 10^9$  degrees of freedom) and denser (three-dimensional, possibly using higher order elements) problems. In such problems the use of direct methods is prohibitive due to memory restrictions and/or extremely high operation counts [26, 70, 95]. At the same time, the use of iterative methods with block preconditioners on this scale requires optimal implementation, including efficient memory management, parallelisation, matrix reuse in nonlinear and time-dependent problems, and careful consideration of software-related issues that are outside the scope of this work.

# References

- [1] R. A. Adams and J. J. F. Fournier. *Sobolev spaces*. Academic press New York, 1975.
- [2] M. Ainsworth and J. T. Oden. *A posteriori* error estimation in finite element analysis. *Computer Methods in Applied Mechanics and Engineering*, 142(1–2):1–88, 1997.
- [3] M. Ainsworth and J. T. Oden. *A posteriori error estimation in finite element analysis*, volume 37. John Wiley & Sons, 2011.
- [4] D. N. Arnold, F. Brezzi, and M. Fortin. A stable finite element for the Stokes equations. *Calcolo*, 21(4):337–344, 1984.
- [5] L. Asner, S. Tavener, and D. Kay. Adjoint-based *a posteriori* error estimation for coupled time-dependent systems. *SIAM Journal on Scientific Computing*, 34(4):A2394–A2419, 2012.
- [6] I. Babuška. Error-bounds for finite element method. *Numerische Mathematik*, 16(4):322–333, 1971.
- [7] I. Babuška and G. N. Gatica. A residual-based *a posteriori* error estimator for the stokes-darcy coupled problem. *SIAM Journal on Numerical Analysis*, 48(2):498–523, 2010.
- [8] I. Babuška and B. Q. Guo. Approximation properties of the  $h - p$  version of the finite element method. *Computer Methods in Applied Mechanics and Engineering*, 133(3):319–346, 1996.

- [9] I. Babuška and W. C. Rheinboldt. *A-posteriori* error estimates for the finite element method. *International Journal for Numerical Methods in Engineering*, 12:1597–1615, 1978.
- [10] I. Babuška and M. Suri. The  $p$  and  $h - p$  versions of the finite element method, basic principles and properties. *SIAM review*, 36(4):578–632, 1994.
- [11] S. Badia and R. Codina. Unified stabilized finite element formulations for the Stokes and the Darcy problems. *SIAM Journal on Numerical Analysis*, 47(3):1971–2000, 2009.
- [12] T. J. Baker. Mesh adaptation strategies for problems in fluid dynamics. *Finite Elements in Analysis and Design*, 25(3):243–273, 1997.
- [13] G. K. Batchelor. *An introduction to fluid dynamics*. Cambridge University Press, New York, 2000.
- [14] R. Beauwens. Iterative solution methods. *Applied Numerical Mathematics*, 51:437–450, 2004.
- [15] G. S. Beavers and D. D. Joseph. Boundary conditions at a naturally permeable wall. *Journal of Fluid Mechanics*, 30(1):197–207, 1967.
- [16] R. Becker, P. Hansbo, and R. Stenberg. A finite element method for domain decomposition with non-matching grids. *ESAIM: Mathematical Modelling and Numerical Analysis*, 37(02):209–225, 2003.
- [17] R. Becker and R. Rannacher. An optimal control approach to *a posteriori* error estimation in finite element methods. *Acta Numerica*, 10:1–102, 2001.
- [18] M. Benzi. Preconditioning techniques for large linear systems: a survey. *Journal of Computational Physics*, 182:418–477, 2002.

- [19] M. Benzi and A. J. Wathen. Some preconditioning techniques for saddle point problems. In *Model Order Reduction: Theory, Research Aspects and Applications*, pages 195–211. Springer, 2008.
- [20] C. Bernardi, Y. Maday, and A. T. Patera. Domain decomposition by the mortar element method. In *Asymptotic and numerical methods for partial differential equations with critical parameters*, pages 269–286. Springer, 1993.
- [21] S. C. Brenner and L. R. Scott. *The Mathematical Theory of Finite Element Methods*. Springer, New York, 2nd edition, 2002.
- [22] F. Brezzi. On the existence, uniqueness and approximation of saddle-point problems arising from Lagrangian multipliers. *RAIRO-Analyse Numérique*, 8:129–151, 1974.
- [23] F. Brezzi and M. Fortin. *Mixed and hybrid finite element methods*. Springer-Verlag New York, Inc., New York, NY, USA, 1991.
- [24] W. L. Briggs, V. E. Henson, and S. F. McCormin. *A Multigrid Tutorial*. SIAM, Philadelphia, 2nd edition, 2000.
- [25] G. Brussino and V. Sonnad. A comparison of direct and preconditioned iterative techniques for sparse, unsymmetric systems of linear equations. *International Journal for Numerical Methods in Engineering*, 28(4):801–815, 2005.
- [26] K. Burckhardt, D. Szczerba, J. Brown, K. Muralidhar, and G. Székely. Fast implicit simulation of oscillatory flow in human abdominal bifurcation using a schur complement preconditioner. In *Euro-Par 2009 Parallel Processing*, pages 747–759. Springer, 2009.
- [27] E. Burman and P. Hansbo. A unified stabilized method for Stokes’ and Darcy’s equations. *Journal of Computational and Applied Mathematics*, 198(1):35–51, 2007.

- [28] M. Cai, M. Mu, and J. Xu. Numerical solution to a mixed Navier-Stokes/Darcy model by the two-grid approach. *SIAM Journal on Numerical Analysis*, 47(5):3325–3338, 2009.
- [29] M. Cai, M. Mu, and J. Xu. Preconditioning techniques for a mixed Stokes/Darcy model in porous media applications. *Journal of Computational and Applied Mathematics*, 233(2):346–355, 2009.
- [30] G. F. Carey, M. Anderson, B. Carnes, and B. Kirk. Some aspects of adaptive grid technology related to boundary and interior layers. *Journal of Computational and Applied Mathematics*, 166(1):55–86, 2004.
- [31] V. Carey, D. Estep, A. Johansson, M. Larson, and S. Tavener. Blockwise adaptivity for time dependent problems based on coarse scale adjoint solutions. *SIAM Journal on Scientific Computing*, 32(4):2121–2145, 2010.
- [32] V. Carey, D. Estep, and S. Tavener. *A posteriori* analysis and adaptive error control for multiscale operator decomposition solution of elliptic systems I: Triangular systems. *SIAM Journal on Numerical Analysis*, 47:740–761, 2009.
- [33] P. Chidyagwai and B. Rivière. A two-grid method for coupled free flow with porous media flow. *Advances in Water Resources*, 34(9):1113–1123, 2011.
- [34] M. R. Correa and A. F. D. Loula. A unified mixed formulation naturally coupling Stokes and Darcy flows. *Computer Methods in Applied Mechanics and Engineering*, 198(33):2710–2722, 2009.
- [35] S. Cotin, H. Delingette, and N. Ayache. Real-time elastic deformations of soft tissues for surgery simulation. *IEEE Transactions on Visualization and Computer Graphics*, 5(1):62–73, 1999.
- [36] B. Dally. Power, programmability, and granularity: The challenges of exascale computing. In *Test Conference (ITC), 2011 IEEE International*, pages 12–12. IEEE, 2011.



- [37] S. R. De Gago, D. W. Kelly, O. C. Zienkiewicz, and I. Babuška. A posteriori error analysis and adaptive processes in the finite element method: Part II – Adaptive mesh refinement. *International Journal for Numerical Methods in Engineering*, 19(11):1621–1656, 1983.
- [38] J. De Hart, G. W. M. Peters, P. J. G. Schreurs, and F.P.T Baaijens. A three-dimensional computational analysis of fluid-structure interaction in the aortic valve. *Journal of Biomechanics*, 36:103–112, 2003.
- [39] E. S. Di Martino, G. Guadagni, A. Fumero, G. Ballerini, R. Spirito, P. Biglioli, and A. Redaelli. Fluid-structure interaction within realistic three-dimensional models of the aneurysmatic aorta as a guidance to assess the risk of rupture of the aneurysm. *Medical Engineering and Physics*, 23:647–655, 2001.
- [40] M. Discacciati, E. Miglio, and A. Quarteroni. Mathematical and numerical models for coupling surface and groundwater flows. *Applied Numerical Mathematics*, 43(1):57–74, 2002.
- [41] J. Donea, A. Huerta, J.-Ph. Ponthot, and A. Rodríguez-Ferran. Arbitrary Lagrangian-Eulerian methods. In E. Stein, R. De Borst, and T. J. R. Hughes, editors, *Encyclopedia of Computational Mechanics. Volume 1: Fundamentals*, pages 1–25. John Wiley and Sons Ltd, New York, 2004.
- [42] J. J. Dongarra and D. W. Walker. The quest for petascale computing. *Computing in Science and Engineering*, 3(3):32–39, 2001.
- [43] E. H. Dowell and K. C. Hall. Modeling of fluid-structure interaction. *Annual Review of Fluid Mechanics*, 33(1):445–490, 2001.
- [44] H. C. Elman. Preconditioning for the steady-state Navier-Stokes equations with low viscosity. *SIAM Journal on Scientific Computing*, 20:1299–1316, 1999.

- [45] H. C. Elman, V. Howle, J. Shadid, R. Shuttleworth, and R. Tuminaro. Block preconditioners based on approximate commutators. *SIAM Journal on Scientific Computing*, 27:1651–1668, 2006.
- [46] H. C. Elman, A. Ramage, and D. J. Silvester. IFISS: A Matlab toolbox for modelling incompressible flow. *ACM Transactions on Mathematical Software*, 33:14, 2007.
- [47] K. Eriksson, D. Estep, P. Hansbo, and C. Johnson. Introduction to adaptive methods for differential equations. *Acta Numerica*, 4:105–158, 1995.
- [48] K. Eriksson, D. Estep, P. Hansbo, and C. Johnson. *Computational differential equations*, volume 1. Cambridge University Press, 1996.
- [49] K. Eriksson and C. Johnson. Adaptive finite element methods for parabolic problems I: A linear model problem. *SIAM Journal on Numerical Analysis*, 28:43–77, 1991.
- [50] K. Eriksson and C. Johnson. Adaptive finite element methods for parabolic problems IV: Nonlinear problems. *SIAM Journal on Numerical Analysis*, 32:1729–1749, 1995.
- [51] V. J. Ervin, E. W. Jenkins, and S. Sun. Coupled generalized nonlinear Stokes flow with flow through a porous medium. *SIAM Journal on Numerical Analysis*, 47(2):929–952, 2009.
- [52] D. Estep, V. Carey, V. Ginting, S. Tavener, and T. Wildey. *A posteriori* error analysis of multiscale operator decomposition methods for multiphysics models. *John Desmond Bernal: Science and Society*, 125:012075, 2008.
- [53] D. Estep, S. Tavener, and T. Wildey. *A posteriori* analysis and improved accuracy for an operator decomposition solution of a conjugate heat transfer problem. *SIAM Journal on Numerical Analysis*, 46:2068–2089, 2008.
- [54] C. Farhat and M. Lesoinne. Two efficient staggered algorithms for the serial and parallel solution of three-dimensional nonlinear transient aeroelastic problems. *Computer Methods in Applied Mechanics and Engineering*, 182:499–515, 2000.

- [55] C. Farhat, M. Lesoinne, and P. Le Tallec. Load and motion transfer algorithms for fluid/structure interaction problems with non-matching discrete interfaces: momentum and energy conservation, optimal discretization and application to aeroelasticity. *Computer Methods in Applied Mechanics and Engineering*, 157(1):95–114, 1998.
- [56] C. Farhat, J. Mandel, and F. X. Roux. Optimal convergence properties of the FETI domain decomposition method. *Computer Methods in Applied Mechanics and Engineering*, 115(3):365–385, 1994.
- [57] C. A. Felippa, K. C. Park, and C. Farhat. Partitioned analysis of coupled mechanical systems. *Computer Methods in Applied Mechanics and Engineering*, 190:3247–3270, 2001.
- [58] J. Fish and T. Belytschko. *A First Course in Finite Element Methods*. John Wiley and Sons Ltd, Chichester, 2007.
- [59] L. P. Franca and S. L. Frey. Stabilized finite element methods: II. The incompressible Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 99(2):209–233, 1992.
- [60] L. P. Franca, S. L. Frey, and T. J. R. Hughes. Stabilized finite element methods: I. Application to the advective-diffusive model. *Computer Methods in Applied Mechanics and Engineering*, 95(2):253–276, 1992.
- [61] J. Galvis and M. Sarkis. Non-matching mortar discretization analysis for the coupling Stokes-Darcy equations. *Electronic Transactions on Numerical Analysis*, 26:350–384, 2007.
- [62] J. Galvis and M. Sarkis. FETI and BDD preconditioners for Stokes-Mortar-Darcy systems. *Communications in Applied Mathematics and Computational Science*, 5(1):1–30, 2010.

- [63] J. Galvis and M. Sarkis. FETI-DP for Stokes-Mortar-Darcy Systems. In *Domain Decomposition Methods in Science and Engineering XIX. Lecture Notes in Computational Science and Engineering*, volume 78, pages 221–228, 2011.
- [64] M. Giles and E. Süli. Adjoint methods for PDEs: *A posteriori* error analysis and postprocessing by duality. *Acta Numerica*, pages 145–236, 2002.
- [65] M. B. Giles. Stability analysis of numerical interface conditions in fluid-structure thermal analysis. *International Journal for Numerical Methods in Fluids*, 25:421–436, 1997.
- [66] R. Glowinski and M. F. Wheeler. Domain decomposition and mixed finite element methods for elliptic problems. In *First international symposium on domain decomposition methods for partial differential equations*, pages 144–172. SIAM, Philadelphia, 1988.
- [67] G. H. Golub and C. F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, 3rd edition, 1996.
- [68] N. I. M. Gould, J. A. Scott, and Y. Hu. A numerical evaluation of sparse direct solvers for the solution of large sparse symmetric linear systems of equations. *ACM Transactions on Mathematical Software*, 33(2):10, 2007.
- [69] N. S. Hanspal, A. N. Waghode, V. Nassehi, and R. J. Wakeman. Numerical analysis of coupled Stokes/Darcy flows in industrial filtrations. *Transport in Porous Media*, 64(1):73–101, 2006.
- [70] M. Heil, A. Hazel, and J. Boyle. Solvers for large-displacement fluid-structure interaction problems: segregated versus monolithic approaches. *Computational Mechanics*, 43:91–101, 2008.
- [71] C. S. Henriquez. Simulating the electrical behavior of cardiac tissue using the bidomain model. *Critical Reviews in Biomedical Engineering*, 21(1):1–77, 1992.

- [72] M. R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49(6):409–436, 1952.
- [73] J. S. Hesthaven and D. Gottlieb. A stable penalty method for the compressible Navier-Stokes equations: I. Open boundary conditions. *SIAM Journal on Scientific Computing*, 17(3):579–612, 1996.
- [74] V. Heuveline and R. Rannacher. Duality-based adaptivity in the *hp*-finite element method. *Journal of Numerical Mathematics*, 11(2):95–113, 2003.
- [75] P. Howell, G. Kozyreff, and J. Ockendon. *Applied Solid Mechanics*, volume 43 of *Cambridge Texts in Applied Mathematics*. Cambridge University Press, Cambridge, 2009.
- [76] O. Iliev and V. Laptev. On numerical simulation of flow through oil filters. *Computing and Visualization in Science*, 6(2-3):139–146, 2004.
- [77] W. Jäger and A. Mikelić. On the interface boundary condition of Beavers, Joseph, and Saffman. *SIAM Journal on Applied Mathematics*, 60(4):1111–1127, 2000.
- [78] C. Johnson. *Numerical solution of partial differential equations by the finite element method*. Dover Publications, 2012.
- [79] R. Kamakoti and W. Shyy. Fluid–structure interaction for aeroelastic applications. *Progress in Aerospace Sciences*, 40(8):535–558, 2004.
- [80] G. E. Karniadakis and S. J. Sherwin. *Spectral/*hp* element methods for CFD*. Oxford University Press, 1999.
- [81] T. Karper, K.-A. Mardal, and R. Winther. Unified finite element discretizations of coupled Darcy-Stokes flow. *Numerical Methods for Partial Differential Equations*, 25(2):311–326, 2009.

- [82] D. Kay, D. Loghin, and A. Wathen. A preconditioner for the steady-state Navier-Stokes equations. *SIAM Journal on Scientific Computing*, 24:237–256, 2002.
- [83] J. Keener and J. Sneyd. *Mathematical Physiology*. Springer Science + Business Media, New York, 2004.
- [84] S. J. Kollet and R. M. Maxwell. Integrated surface-groundwater flow modeling: A free-surface overland flow boundary condition in a parallel groundwater flow model. *Advances in Water Resources*, 29(7):945–958, 2006.
- [85] U. Küttler, M. Gee, C. Förster, A. Comerford, and W. A. Wall. Coupling strategies for biomedical fluid-structure interaction problems. *International Journal for Numerical Methods in Biomedical Engineering*, 26:305–321, 2010.
- [86] W. J. Layton, F. Schieweck, and I. Yotov. Coupling fluid flow with porous media flow. *SIAM Journal on Numerical Analysis*, 40:2195–2218, 2003.
- [87] R. J. LeVeque. *Finite Volume Methods for Hyperbolic Problems*. Cambridge Texts in Applied Mathematics. Cambridge University Press, 2002.
- [88] F. Liu, M. Sosonkina, and R. Bramley. A new approach: Component-based multi-physics coupling through CCA-LISI. *Computational Science and Its Applications – ICCSA 2010, Lecture Notes in Computational Science*, 6017:503–518, 2010.
- [89] H. G. Matthies and J. Steindorf. Partitioned strong coupling algorithms for fluid-structure interaction. *Computers and Structures*, 81:805–812, 2003.
- [90] K.W. Morton and D.F. Mayers. *Numerical Solution of Partial Differential Equations: An Introduction*. Cambridge University Press, 2005.
- [91] M. F. Murphy, G. H. Golub, and A. J. Wathen. Note on preconditioning for indefinite linear systems. *SIAM Journal on Scientific Computing*, 21:1969–1972, 2000.

- [92] F. Nobile. *Numerical approximation of fluid-structure interaction problems with application to haemodynamics*. PhD thesis, École Polytechnique Fédérale de Lausanne, 2001.
- [93] D. Nordsletten, P. Hunter, and N. Smith. Conservative and non-conservative arbitrary Lagrangian-Eulerian forms for ventricular flows. *International Journal for Numerical Methods in Fluids*, 56:1457–1463, 2008.
- [94] D. Nordsletten, D. Kay, and N. Smith. A non-conforming monolithic finite element method for problems of coupled mechanics. *Journal of Computational Physics*, 229(20):7571–7593, 2010.
- [95] D. A. Nordsletten. *Fluid-solid coupling for the simulation of left ventricular mechanics*. PhD thesis, University of Oxford, Computing Laboratory, 2009.
- [96] C. C. Paige and M. A. Saunders. Solution of sparse indefinite systems of linear equations. *SIAM Journal on Numerical Analysis*, 12(4):617–629, 1975.
- [97] K. Perktold, M. Hofer, G. Rappitsch, M. Loew, B. D. Kuban, and M. H. Friedman. Validated computation of physiologic flow in a realistic coronary artery branch. *Journal of Biomechanics*, 31(3):217–228, 1997.
- [98] A. Quarteroni and A. Valli. *Domain Decomposition Methods for Partial Differential Equations*. Numerical Mathematics and Scientific Computation Series. Clarendon Press, 1999.
- [99] R. Rannacher. Adaptive Galerkin finite element methods for partial differential equations. *Journal of Computational and Applied Mathematics*, 128(1–2):205–233, 2001.
- [100] J. N. Reddy. On penalty function methods in the finite-element analysis of flow problems. *International Journal for Numerical Methods in Fluids*, 2(2):151–171, 1982.
- [101] W. C. Rheinboldt. On a theory of mesh-refinement processes. *SIAM Journal on Numerical Analysis*, 17:766–778, 1980.

- [102] J. R. Rice, P. Tsompanopoulou, and E. Vavalis. Interface relaxation methods for elliptic differential equations. *Applied Numerical Mathematics*, 32:219–245, 2000.
- [103] B. Rivière and I. Yotov. Locally conservative coupling of Stokes and Darcy flows. *SIAM Journal on Numerical Analysis*, 42(5):1959–1977, 2005.
- [104] Y. Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, Philadelphia, 2nd edition, 1996.
- [105] Y. Saad and M. H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 7(3):856–869, 1986.
- [106] P. G. Saffman. On the boundary condition at the surface of a porous medium. *Studies in Applied Mathematics*, 50(2):93–101, 1971.
- [107] M. Sedef, E. Samur, and C. Basdogan. Real-time finite-element simulation of linear viscoelastic tissue behavior based on experimental data. *Computer Graphics and Applications, IEEE*, 26(6):58–68, 2006.
- [108] P. Seshaiyer and M. Suri. Uniform  $hp$  convergence results for the mortar finite element method. *Mathematics of Computation of the American Mathematical Society*, 69(230):521–546, 2000.
- [109] J. Shalf, S. Dosanjh, and J. Morrison. Exascale computing technology challenges. In *High Performance Computing for Computational Science–VECPAR 2010*, pages 1–25. Springer, 2011.
- [110] J. Shen. On error estimates of the penalty method for unsteady Navier-Stokes equations. *SIAM Journal on Numerical Analysis*, 32(2):386–403, 1995.
- [111] D. Silvester, H. Elman, D. Kay, and A. Wathen. Efficient preconditioning of the linearized Navier-Stokes equations for the incompressible flow. *Journal of Computational and Applied Mathematics*, 128:261–279, 2001.



- [112] B. Smith, P. Bjorstad, and W. Gropp. *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*. Cambridge University Press, 2004.
- [113] G. Stadler, M. Gurnis, C. Burstedde, L. C. Wilcox, L. Alisic, and O. Ghattas. The dynamics of plate tectonics and mantle flow: From local to global scales. *Science*, 329(5995):1033–1038, 2010.
- [114] J. Strikwerda. *Finite Difference Schemes and Partial Differential Equations*. Society for Industrial and Applied Mathematics, 2007.
- [115] K. Stüben. A review of algebraic multigrid. *Journal of Computational and Applied Mathematics*, 128(1):281–309, 2001.
- [116] J. W. Thomas. *Numerical Partial Differential Equations: Finite Difference Methods*. Texts in Applied Mathematics. Springer, 1995.
- [117] R. Torii, M. Oshima, T. Kobayashi, K. Takagi, and T. E. Tezduyar. Fluid–structure interaction modeling of aneurysmal conditions with high and normal blood pressures. *Computational Mechanics*, 38(4-5):482–490, 2006.
- [118] C. Truesdell and R. Toupin. *The classical field theories*. Springer, 1960.
- [119] H. A. Van der Vorst. Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 13(2):631–644, 1992.
- [120] R. Verfürth. A posteriori error estimation and mesh refinement techniques. *Journal of Computational and Applied Mathematics*, 50:67–83, 1994.
- [121] H. H. K. Versteeg and W. Malalasekera. *An introduction to computational fluid dynamics: the finite volume method*. Pearson Education Australia, 2007.
- [122] W. A. Wall and T. Rabczuk. Fluid-structure interaction in lower airways of CT-based lung geometries. *International Journal for Numerical Methods in Fluids*, 57(5):653–675, 2008.

- [123] Y. Wang and J. C. Ragusa. Standard and goal-oriented adaptive mesh refinement applied to radiation transport on 2D unstructured triangular meshes. *Journal of Computational Physics*, 230(3):763–788, 2011.
- [124] M. Wicke, D. Ritchie, B. M. Klingner, S. Burke, J. R. Shewchuk, and J. F. O’Brien. Dynamic local remeshing for elastoplastic simulation. *ACM Transactions on Graphics*, 29(4):49, 2010.
- [125] T. Wildey. A posteriori *analysis of operator decomposition on interface problems*. PhD thesis, Colorado State University, 2007.
- [126] G. Xia, M. H. Tawhai, E. A. Hoffman, and C.-L. Lin. Airway wall stiffening increases peak wall shear stress: a fluid-structure interaction study in rigid and compliant airways. *Annals of biomedical engineering*, 38(5):1836–1853, 2010.
- [127] N. Yan and A. Zhou. Gradient recovery type *a posteriori* error estimates for finite element approximations on irregular meshes. *Computer Methods in Applied Mechanics and Engineering*, 190:4289–4299, 2001.
- [128] O. C. Zienkiewicz and J. Z. Zhu. The superconvergent patch recovery and a posteriori error estimates. Part 2: Error estimates and adaptivity. *International Journal for Numerical Methods in Engineering*, 33(7):1365–1382, 1992.
- [129] O. C. Zienkiewicz and J. Z. Zhu. The superconvergent patch recovery (SPR) and adaptive finite element refinement. *Computer Methods in Applied Mechanics and Engineering*, 101(1):207–224, 1992.