UPPSALA
UNIVERSITET

Digital Comprehensive Summaries of Uppsala Dissertations from the Faculty of Science and Technology 885

Computational Techniques for Coupled Flow-Transport Problems

MARTIN KRONBICHLER

ACTA
UNIVERSITATIS
UPSALIENSIS
UPPSALA
2011

ISSN 1651-6214
ISBN 978-91-554-8232-9
urn:nbn:se:uu:diva-162215

Dissertation presented at Uppsala University to be publicly examined in room 2446, building 2, Polacksbacken, Lägerhyddsvägen 2, Uppsala. Friday, January 13, 2012 at 14:00 for the degree of Doctor of Philosophy. The examination will be conducted in English.

Abstract
Kronbichler, M. 2011. Computational Techniques for Coupled Flow-Transport Problems. Acta Universitatis Upsaliensis. *Digital Comprehensive Summaries of Uppsala Dissertations from the Faculty of Science and Technology* 885. 61 pp. Uppsala. ISBN 978-91-554-8232-9.

This thesis presents numerical techniques for solving problems of incompressible flow coupled to scalar transport equations using finite element discretizations in space. The two applications considered in this thesis are multi-phase flow, modeled by level set or phase field methods, and planetary mantle convection based on the Boussinesq approximation.

A systematic numerical study of approximation errors in evaluating the surface tension in finite element models for two-phase flow is presented. Forces constructed from a gradient in the same discrete function space as used for the pressure are shown to give the best performance. Moreover, two approaches for introducing contact line dynamics into level set methods are proposed. Firstly, a multiscale approach extracts a slip velocity from a micro simulation based on the phase field method and imposes it as a boundary condition in the macro model. This multiscale method is shown to provide an efficient model for the simulation of contact-line driven flow. The second approach combines a level set method based on a smoothed color function with a the phase field method in different parts of the domain. Away from contact lines, the additional information in phase field models is not necessary and it is disabled from the equations by a switch function. An in-depth convergence study is performed in order to quantify the benefits from this combination. Also, the resulting hybrid method is shown to satisfy an a priori energy estimate.

For the simulation of mantle convection, an implementation framework based on modern finite element and solver packages is presented. The framework is capable of running on today's large computing clusters with thousands of processors. All parts in the solution chain, from mesh adaptation over assembly to the solution of linear systems, are done in a fully distributed way. These tools are used for a parallel solver that combines higher order time and space discretizations. For treating the convection-dominated temperature equation, an advanced stabilization technique based on an artificial viscosity is used.

For more efficient evaluation of finite element operators in iterative methods, a matrix-free implementation built on cell-based quadrature is proposed. We obtain remarkable speedups over sparse matrix-vector products for many finite elements which are of practical interest. Our approach is particularly efficient for systems of differential equations.

Keywords: Finite element method, saddle point systems, two-phase flow, level set method, phase field method, stabilization, contact line dynamics

Martin Kronbichler, Uppsala University, Department of Information Technology, Division of Scientific Computing, Box 337, SE-751 05 Uppsala, Sweden. Department of Information Technology, Numerical Analysis, Box 337, SE-751 05 Uppsala, Sweden.

ISSN 1651-6214
ISBN 978-91-554-8232-9
urn:nbn:se:uu:diva-162215 (http://urn.kb.se/resolve?urn=urn:nbn:se:uu:diva-162215)

List of papers

This thesis is based on the following papers, which are referred to in the text by their Roman numerals.

I S. Zahedi, M. Kronbichler, and G. Kreiss (2011) Spurious Currents in Finite Element Based Level Set Methods for Two-Phase Flow. In *International Journal for Numerical Methods in Fluids*. DOI: 10.1002/fld.2643. In press.
Contributions: Implementation of and experiments with the coupled two-phase flow solver. The author of this thesis has contributed the first draft of the manuscript. Ideas were developed in close cooperation between the first and the second author in discussion with the third author.

II M. Kronbichler, C. Walker, G. Kreiss, and B. Müller (2011) Multiscale Modeling of Capillary-Driven Contact Line Dynamics. *Technical report 2011-024, Department of Information Technology, Uppsala University.* (Submitted)
Contributions: Ideas were developed in collaboration between the authors. The author of this thesis implemented the micro solver and performed all the experiments with that model.

III M. Kronbichler and G. Kreiss (2011) A Hybrid Level-Set-Phase-Field Method for Two-Phase Flow with Contact Lines. *Technical report 2011-026, Department of Information Technology, Uppsala University.*
Contributions: Ideas and analysis were developed in close discussion between the authors. The author of this thesis performed and evaluated all experiments and wrote the manuscript.

IV M. Kronbichler, T. Heister, and W. Bangerth (2011) High Accuracy Mantle Convection Simulation Through Modern Numerical Methods. *IAMCS preprint 2012-286, Texas A&M University.* (Submitted)
Contributions: Initial ideas and implementation were developed in collaboration between the first and third author, the second author contributed with the parallel tests and compressible results. The author of this thesis contributed most work in Sec. 3 on numerical methods and the benchmark tests (Sec. 4.1 and 4.2). The manuscript was prepared in collaboration between the authors.

V W. Bangerth, C. Burstedde, T. Heister, and M. Kronbichler (2011) Algorithms and Data Structures for Massively Parallel Generic Adaptive Finite Element Codes. To appear in *ACM Transactions on Mathematical Software* **38**(2).
Contributions: Work related to the implementation of constraints and systems of PDEs. Major parts of implementation of the Boussinesq solver in Sec. 6.2 and interfaces to Trilinos linear algebra. The author of this thesis contributed to the manuscript.

VI M. Kronbichler and K. Kormann (2011) A Generic Interface for Parallel Cell-Based Finite Element Operator Application. *Technical report 2011-025, Department of Information Technology, Uppsala University.* (Submitted)
Contributions: Ideas and implementation were done in close cooperation between the authors.

Reprints were made with permission from the publishers.

Related work

Although not explicitly discussed in the comprehensive summary, the following papers are related to the contents of this thesis.

- M. Kronbichler and G. Kreiss (2008) A Hybrid Level–Set–Cahn–Hilliard Model for Two-Phase Flow. In *Proceedings of the 1st European Conference on Microfluidics*, Bologna, Italy.
- K. Kormann and M. Kronbichler (2011) Parallel finite element operator application: Graph partitioning and coloring. Accepted for publication in *Proceedings of the 7th IEEE International Conference on e-Science*, Stockholm, Sweden.
- T. Heister, M. Kronbichler, and W. Bangerth (2010) Massively parallel finite element programming. In *Recent Advances in the Message Passing Interface*, volume 6305 of Lecture Notes in Computer Science, 122–131, Springer-Verlag, Berlin.

Essential parts of this thesis rely on the implementation of finite element solvers. Some of these solvers have been made publicly available through the widely used deal.II library, enabling other researchers to reproduce results or extend the implementation to new areas. The author of this thesis contributed to parts of the library and the deal.II example programs step-22 (solution of Stokes equations), step-31 (serial Boussinesq solver), step-32 (parallel Boussinesq solver), step-37 (multigrid solver based on cell-based finite element operator application), step-40 (a massively parallel Poisson solver), step-48 (parallel cell-based finite element operator application).

Contents

1. Introduction

Numerical simulation is considered the third pillar in engineering science, complementing theory and experiments. Computational fluid dynamics is one of its major fields and helps in understanding and predicting the behavior of fluid flow. The use of simulation promises better control of technical processes and insight into systems for which experiments are expensive or impossible.

An import discipline in computational fluid dynamics is multi-phase flow, a setting that involves two or more different fluids. There is a wide range of applications that involve multiple fluids and where numerical simulations are a fundamental tool. This thesis presents numerical techniques for the setting where the two fluids are incompressible and do not mix. One application is intravenous therapy in medicine where vesicles consisting of oleaginous substances enter the circulation of the blood [80]. These flows are usually characterized by high surface tensions that tend to make vesicles spherical. Similarly, simulation is a tool in the understanding of the shape of red blood cells [1]. In the biomedical industry, so-called lab-on-a-chip processes also involve the simulation of capillary-dominated multi-phase flow [96, 84]. For the development of polymer materials like bicontinuous interfacially-jammed emulsion gels [66], different fluids are separated by barrier materials, which can be modeled with multi-phase flow approaches for representing interface physics. Likewise, liquid phase sintering can be controlled by tracking mixtures of different fluids and their solidification.

Subsurface flow is another field where multiple fluids are present. The interaction of different fluids with the rock material is one of the main physical processes in this setting, but its modeling represents a considerable challenge. Due to the small length scales of pores on the order of millimeters, wetting effects dominate the flow. These small length scales need to be seen in the light of the size of the global structures modeled, which are groundwater basins and oil reservoirs that extend over thousands of square kilometers. The wide range of scales requires model reductions, like the combination of small-scale flow simulations with detailed rock structures to find the permeability of rock materials, and large scale simulations of Darcy flow with these parameters.

The implementation of the numerical models for multi-phase flow requires the use of advanced programming techniques. Particularly for reliable three-dimensional simulations, hundreds of millions or even billions of degrees of freedom are required to provide sufficient resolution. In order to tackle such enormous problems, efficient parallel implementations are necessary. In this thesis, a parallel adaptive finite element framework is presented for constructing solvers for the equations of incompressible flow coupled to a description

of the interface between the fluids. The framework is also applied to simulate planetary mantle convection where the goal is to understand the dynamics of mantle materials at different temperatures, flow that is mainly driven by buoyancy. At first sight, the problems of mantle convection and of capillary-dominated two-phase flow are very different, in particular with respect to the length scales of thousands of kilometers in mantle simulations and millimeters or less for capillary-dominated flow. However, the programming techniques applied in this thesis are indeed very similar.

This comprehensive summary consists of eight chapters. In the next chapter, we discuss the fundamental numerical aspects in the simulation of two-phase flow. Chapter 3 gives an introduction to the simulation of mantle convection. Chapter 4 discusses numerical and implementation issues underlying both application problems. Chapter 5 gives a short outline of software aspects in finite element programming. The papers included in this thesis are summarized in Chapter 6. Chapter 7 identifies possible directions of future research, and Chapter 8 gives a Swedish summary of this work.

2. Simulation of Two-Phase Flow

We consider the dynamics of two immiscible incompressible fluids separated by an interface Γ according to Fig. 2.1. By Ω_1 we denote the region occupied by the first fluid, and by Ω_2 the region occupied by the second one. We assume laminar flow at small Reynolds numbers.

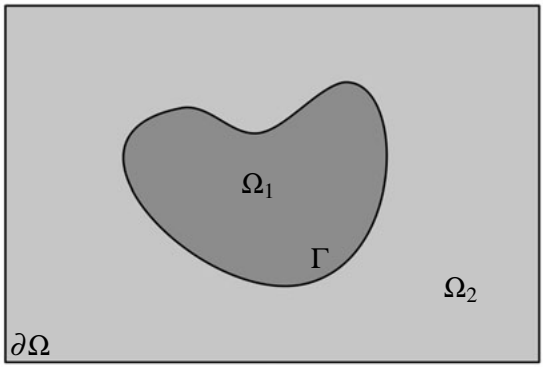


Figure 2.1. Schematic view of a two-phase flow problem. Fluid 1 occupies region Ω_1, and fluid 2 occupies Ω_2. The interface Γ separates the two fluids.

The motion of each fluid for two-phase flow in d-dimensional space is described by the *incompressible Navier–Stokes equations*,

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}\right) - \nabla\cdot(2\mu\varepsilon(\mathbf{u})) + \nabla p = \rho\mathbf{f}, \tag{2.1}$$

$$\nabla\cdot\mathbf{u} = 0. \tag{2.2}$$

The vector $\mathbf{u}$ denotes the d components of fluid velocity and p denotes the fluid's (dynamic) pressure. The rank-2 tensor $\varepsilon(\mathbf{u}) = \frac{1}{2}\left(\nabla\mathbf{u} + (\nabla\mathbf{u})^{\mathsf{T}}\right)$ represents the rate-of-deformation tensor. The fluid density is denoted by ρ and the (dynamic) viscosity by μ. The term $\mathbf{f}$ specifies external forces acting on the fluid, for instance gravity or electromagnetic forces.

The system of Navier–Stokes equations (2.1)–(2.2) is completed by a divergence-free initial velocity field $\mathbf{u}(\cdot,0)$ and application-specific boundary conditions at the boundary $\partial\Omega$ (cf. Gresho and Sani [51, 52] for a discussion of various possibilities). No-slip boundary conditions at solid walls are of the form

$$\mathbf{u} = \mathbf{0} \quad \text{on} \quad \partial\Omega_{\mathrm{w}} \subset \partial\Omega. \tag{2.3}$$

The incompressible Navier–Stokes equations (2.1)–(2.2) hold on each subdomain Ω_i for the two fluids and their solution is denoted by the variables $\mathbf{u}_i$ and p_i, $i = 1,2$. Extra conditions are necessary to describe the behavior at the interface $\Gamma = \Gamma(t)$, which is a free surface evolving in time according to the underlying fluid flow $\mathbf{u}|_\Gamma$. If we neglect mass transfer over the interface, the following *jump conditions* over the interface hold (see, e.g., [33, 85]),

$$[\![\mathbf{u}]\!]_\Gamma = \mathbf{0}, \tag{2.4}$$

$$[\![-2\mu\varepsilon(\mathbf{u})\cdot\mathbf{n} + p\mathbf{n}]\!]_\Gamma = \sigma\kappa\mathbf{n}, \tag{2.5}$$

$$[\![-2\mu\varepsilon(\mathbf{u})\cdot\mathbf{n}]\!]_\Gamma \cdot \mathbf{t} = 0, \tag{2.6}$$

where σ is a constant specifying the relative strength of the surface tension, κ denotes the interface curvature, $\mathbf{n}$ the direction normal to the interface (pointing into Ω_1), and $\mathbf{t}$ denotes vectors tangential to the interface. The jump of quantities from Ω_1 to Ω_2 is denoted by $[\![\mathbf{u}]\!] = \mathbf{u}_1 - \mathbf{u}_2$. A widespread formulation is to pose the two-phase flow problem on the whole domain $\Omega = \Omega_1 \cup \Gamma \cup \Omega_2$, based on the combined variables $\mathbf{u}$ and p. Generally, the fluid densities ρ_i and viscosities μ_i are different for the two fluids and jump at the interface. In addition, the jump in normal stress (2.5) can be modeled by adding a forcing of the form

$$\mathbf{f}_{\text{st}} = \sigma\kappa\mathbf{n}\,\delta_\Gamma \tag{2.7}$$

to the momentum equation, where δ_Γ is a delta function that localizes the force to the interface, i.e.,

$$\int_\Omega h(\mathbf{x})\delta_\Gamma(\mathbf{x})\,d\mathbf{x} = \int_\Gamma h(\mathbf{s})\,d\mathbf{s},$$

for $h : \mathbb{R}^d \to \mathbb{R}$.

In addition to solving the incompressible Navier–Stokes equations, the two main aspects in the numerical simulation of two-phase flow are the representation of the interface as it evolves in time, and the evaluation of interfacial tension. Some applications also require the evaluation of additional physical processes on the interface. The interface representation and evolution is discussed in Sections 2.1 to 2.3, and the evaluation of interface forces in Sec. 2.4.

2.1 Overview of numerical models for interface representation

The geometry of the interface is usually complex and can change topology, like the breakup and reconnection of bubbles. Several discrete approaches have been proposed during the last decades. There are two main strategies to couple the interface evolution problem to the Navier–Stokes equations discretized on a fixed grid, interface tracking and interface capturing approaches.

Interface tracking

Interface tracking is based on an explicit description of the interface. Boundary integral methods use potential theory to reduce the problem to the interface [68]. However, these methods are only valid for the limited physics of a Stokes flow. In the more general setting where the Navier–Stokes equations are formulated on the whole domain, one strategy is to embed marker points or elements for representing the interface. These so-called front tracking methods were introduced in different variants by Peskin [102] and Glimm et al. [47], see also the review articles [112, 131]. The evolution of the interface in front tracking schemes is accomplished by Lagrangian advection of the marker points. This process combines the information from the d-dimensional fluid grid and the $(d-1)$-dimensional interface grid. When the flow deforms the interface, the marker points need to be redistributed for retaining accurate interface representations. Also, straight-forward implementations of tracking methods are prone to unphysical changes of the area/volume of the respective fluid.

Interface capturing

In interface capturing approaches, the interface is implicitly defined through other quantities. Historically, the first scheme of this kind was the marker-and-cell method proposed by Harlow and Welch [63], considered to be a volume-tracking method. Here, one fluid is colored by marker particles whose location is advected with the flow field. The position of the interface can then be reconstructed from the particle field. An extension of this approach is to replace the marker particles by a marker function in terms of the volume-of-fluid (VOF) method [93, 67]. VOF includes an additional variable that stores the fraction of the first fluid on the total fluid for each cell of the computational grid. The advection of the interface is usually implemented by increasing or decreasing the volume fraction depending on the velocity field and the composition of neighboring cells. The advantage of VOF methods is volume conservation and a natural mechanism for breakup and fusion of bubbles. However, the evaluation of surface tension forces along the interface requires the reconstruction of the interface location from the discrete volume fractions. This reconstruction makes VOF schemes considerably more complicated to implement compared to front tracking methods. For details of the reconstruction, including higher order schemes, we refer to [112, 114] and references therein.

While VOF methods operate on the discrete level, a continuous front capturing framework is provided by the level set method [100, 124] and the phase field method [76]. The methods discussed in Paper I–III are based on these two approaches, which are discussed in detail in the following two sections.

Since all the different approaches have their pros and cons, hybrid approaches have been presented in order to combine attractive features of several methods. Examples are coupled level set-boundary integral methods [41],

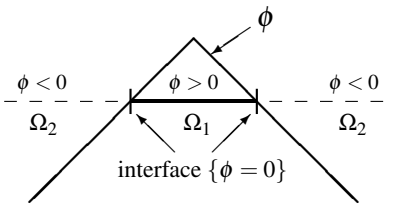


Figure 2.2. Representing an interface by a signed distance level set function ϕ.

combined level set-front tracking methods [115, 116], and combined level set-VOF approaches [123].

2.2 Level set methods

The basic idea of level set methods is to define the interface implicitly as the zero contour line of a function ϕ that is defined in the whole domain Ω. These methods have been introduced in [100] and applied to incompressible two-phase flow in [124]. Level set methods allow for a straight-forward evaluation of normal vectors, $\mathbf{n}(\phi) = \nabla\phi/|\nabla\phi|$, and curvature, $\kappa(\phi) = -\nabla \cdot \mathbf{n}(\phi)$. Both these quantities can be computed locally from ϕ on the same computational mesh as is used for the representation of fluid flow. The mechanism for moving the interface in level set methods is advection of the function ϕ with local fluid velocity, i.e.,

$$\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi = 0. \tag{2.8}$$

This is an additional partial differential equation coupled to the Navier–Stokes equations. The advection equation (2.8) can be solved with standard tools for hyperbolic transport equations, like upwind finite difference/volume schemes or stabilized finite element methods.

The standard choice for the level set function is a signed distance function ϕ, depicted in Fig. 2.2, see [114, 99] for a presentation of this concept. A signed distance function encodes the distance of a particular point to the interface and distinguishes the fluids by the sign. It has the property $|\nabla\phi| = 1$ almost everywhere. Since the information from ϕ is only needed in a region around the interface, so-called narrow-band level set implementations are often used. These methods calculate ϕ only on a few grid cells around the interface, usually combined with high-resolution grids for ϕ. We refer to [2] for the presentation of such a method.

To start a level set calculation, a profile $\phi(\cdot, 0)$ needs to be generated given the initial position of the interface. Moreover, the flow field as well as inaccuracies in the numerical scheme deform the signed distance function during

the simulation. Therefore, algorithms have been proposed to (re-)initialize the signed distance function. These algorithms enforce the property $|\nabla\phi| = 1$ [110, 72]. One option is to solve the PDE

$$\frac{\partial\phi}{\partial\tau} + S_\varepsilon(\phi_0)(|\nabla\phi| - 1) = 0, \tag{2.9}$$

to steady state, where $S_\varepsilon(\phi_0)$ is a smoothed sign function with value 1 in the first fluid and value -1 in the other one [99]. However, discretizations of the reinitialization scheme do not preserve mass in the sense that the area $\mathrm{vol}(\Omega_1) = \mathrm{vol}(\{\mathbf{x}; \phi(\mathbf{x}) > 0\})$ is altered, so that the overall level set implementation based on signed distance functions usually suffers from unphysical volume changes in the fluid phases.

Many methods have been proposed to improve mass conservation, like constrained reinitialization that penalizes volume changes during reinitialization [121, 64] and the conservative level set method [94, 95]. The conservative level set method is based on a smoothed color function instead of the signed distance function, see Fig. 2.3(b). Around the interface, there is a transition region where the function smoothly switches from value $+1$ to -1. This function ϕ can be calculated from the reinitialization equation

$$\frac{\partial\phi}{\partial\tau} + \nabla\cdot\left(\mathbf{n}\left(1-\phi^2\right)\right) - \nabla\cdot(\mathbf{n}\varepsilon\nabla\phi\cdot\mathbf{n}) = 0. \tag{2.10}$$

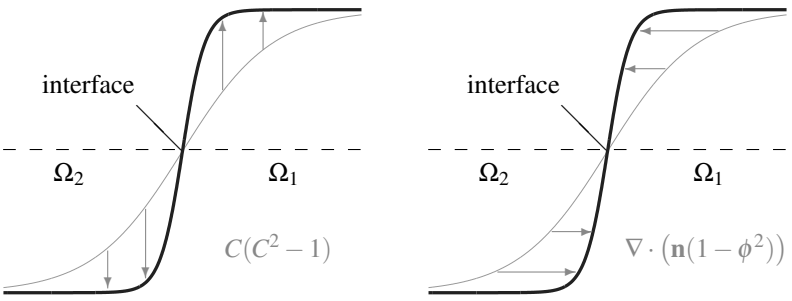
In this equation, diffusion in direction normal to the interface is balanced by a compressive flux. The parameter ε is typically chosen to be proportional to the mesh size. For a plane interface located at $x = \xi$, the steady-state value of (2.10) is given by

$$\phi(x) = \tanh\left(\frac{x-\xi}{\varepsilon}\right), \tag{2.11}$$

see also Fig. 2.3(b). Since the reinitialization equation (2.10) is posed as a conservation law, an implementation that exactly preserves $\int_\Omega \phi d\mathbf{x}$ is possible [95]. A disadvantage of a smoothed color function is that the evaluation of quantities along the interface is less accurate because of the steep profile, which increases resolution requirements considerably. Another drawback of the smoothed indicator function (2.11) is the necessity to consider it in the whole domain for conserving $\int_\Omega \phi d\mathbf{x}$. Nonetheless, the high resolution requirements around the interface can be alleviated by the use of adaptive meshes, as has been used in Paper III. Alternatively, the conservation properties of this method can be combined with the superior interface representation of signed distance functions as proposed in [82].

2.3 Phase field methods

The concept of all the methods presented above is to use some interface description as a mathematical tool. The phase field method gives the function



(a) Cahn–Hilliard equation: sharpness through nonlinear reaction-like term

(b) Conservative level set method: sharpness through nonlinear compressive flux

Figure 2.3. Illustration of mechanism driving interface sharpness in the Cahn–Hilliard equation [29] and in the conservative level set method [94].

representing the interface a physical meaning as diffuse interface in the whole domain Ω. It is derived from the van der Waals hypothesis stating that immiscible fluids actually mix on molecular level. The profile of the interface is given by a balance of random molecular motion and molecular attraction in terms of the molecular free energy, as formulated by Cahn and Hilliard [29],

$$f(C) = \beta\Psi(C) + \frac{\alpha}{2}|\nabla C|^2, \tag{2.12}$$

where C denotes the concentration and $\Psi(C) = (C+1)^2(C-1)^2$. Similar to the smoothed color function (2.11), the function seeks to attain the two equilibrium values $C = \pm 1$, which is driven by the double-well potential $\Psi(C)$, and the gradient term ∇C make the transition smooth.
The physical system seeks to minimize the energy (2.12), a process described by the Cahn–Hilliard equation

$$\frac{\partial C}{\partial t} = \nabla\cdot(M\nabla\psi) = \nabla\cdot\left(M\nabla\left(\beta\Psi'(C) - \alpha\nabla^2 C\right)\right), \tag{2.13}$$

with the chemical potential ψ and the mobility factor M. The phase field method [76] (see also the reviews [6, 35, 118]) includes transport of the interface by convection, i.e., a term $\mathbf{u}\cdot\nabla C$. The mean diffuse interface thickness in the phase field method is given by $\sqrt{\alpha/\beta}$. Even though physical mixing happens only in the range of some tens of nanometers [33], the phase field method has been used successfully for the simulation of multi-phase flow where the interface thickness is rather dictated by numerical resolution.

The phase field method relies on a similar function C as the conservative level set method [94]. For a plane interface, the steady state for the phase field method is indeed exactly the same as for the conservative level set method, namely (2.11) with $\varepsilon = \sqrt{2\alpha/\beta}$. However, the two methods rely on different nonlinear sharpening mechanism, as illustrated in Fig. 2.3.

The diffuse interface approach of the phase field method includes a formulation for interfacial tension as a volume force

$$\sigma\kappa\mathbf{n}\delta_\Gamma \approx \psi\nabla C. \tag{2.14}$$

Boyer [20] formulated a phase field model for fluids with different densities and viscosities. Different forms of the surface tension force have recently been studied numerically on a benchmark problem [5].

2.4 Interface forces

For evaluating the surface tension force (2.7) and other quantities along the interface for non-diffuse interface models, two main approaches can be distinguished. The first class of methods uses discrete delta functions approximating δ_Γ in the force definition in order to localize the force to a volume fraction close to the interface [103]. This continuous surface tension approach [23] (see also the review in [131]) is conceptually similar to the volume force (2.14) for the phase field method. A general framework for the evaluation of surface tension and other quantities along the interface is provided by the immersed boundary method, see the review paper [103]. Normal vectors and curvatures for evaluating surface tension (2.7) are computed from straightforward geometric identities with the information provided by the respective interface description. For color functions c that are $+1$ in one fluid and -1 in the other, the following continuous surface tension model can be used,

$$\mathbf{n}\delta_\Gamma \approx \frac{1}{2}\nabla c. \tag{2.15}$$

The factor one half comes from the fact that the function c switches from -1 to $+1$. Paper I addresses the accuracy of interface forces constructed from this concept in the context of finite element discretizations. In general, the definition of discrete delta functions needs to be done with care in order to maintain consistency of the method. For instance, the approach to combine a signed distance function with a scalar discrete delta function of the form $\delta_h(\phi(\mathbf{x}))$ has been demonstrated to lead to $\mathscr{O}(1)$ errors [36]. Several consistent delta function approximations have been proposed, see [36, 138] and references therein.

The second class of methods avoids introducing volume forces by considering jump conditions of the type (2.4)–(2.6) directly along the $(d-1)$-dimensional interface. This can be accomplished by embedding the jumps directly into finite difference stencils [38]. These so-called ghost fluid methods have been generalized to finite element space discretizations in the framework of extended finite element methods, where discontinuities are embedded into the solution spaces [54]. Recently, an extended finite element method

for simulating a three-dimensional rising bubble has been used [107]. A general formulation for evaluating interface forces and other quantities along the interface in a sharp way is provided by the immersed interface [85, 87].

2.5 Modeling of contact lines

When the interface between two fluids is in contact with solid walls, wetting driven by contact line dynamics needs to be included in the numerical model. A contact line is defined as the $(d-2)$-dimensional manifold where the interface between the two fluids is in contact with the solid. The conventional no-slip boundary condition does not hold at contact lines since that would yield a singularity in shear stress [70]. In order to resolve this singularity, the contact line needs to be allowed to slip, for example based on a Navier condition

$$\mathbf{u}_{\text{slip}} = \lambda \mathbf{n} \cdot \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^{\mathsf{T}} \right), \tag{2.16}$$

where the parameter λ is the so-called slip length. Many models have been proposed to define this parameter, like the ad-hoc approach in [119] or relations derived from multiscale simulations [109]. In Paper II, a new multiscale model is proposed for resolving this relation.

In the phase field method, there is a built-in mechanism for contact line behavior, which is used in Papers II and III. By setting a nonlinear Robin-type boundary condition on the concentration variable C,

$$\mathbf{n} \cdot \nabla C + \cos(\theta_S)(1 - C^2) = 0,$$

a problem-specific static contact line angle θ_S can be imposed [76, 134]. However, the behavior of flow driven by contact lines depends on the value of the mobility M. Yue et al. [136] showed convergence of contact dynamics with respect to the interface thickness $\sqrt{2\alpha/\beta}$ but the dependence on resolving small mobilities M sets high demands on numerical methods.

3. Mantle convection

Convection processes in planetary mantles can be described by incompressible flow driven by density differences, which in turn are caused by differences in temperature. The buoyancy forces are much lower than viscous friction forces, so the fluid flow is so slow that inertial terms can be neglected [113, Ch. 6]. The resulting fluid flow is described by the Stokes equations. In combination with the temperature evolution equation, this yields the Boussinesq approximation, a system of coupled differential equations,

$$-\nabla \cdot (2\mu\varepsilon(\mathbf{u})) + \nabla p = \rho(T)\mathbf{g}, \tag{3.1}$$

$$\nabla \cdot \mathbf{u} = 0, \tag{3.2}$$

$$\frac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla T - \nabla \cdot (\kappa \nabla T) = \gamma, \tag{3.3}$$

which is solved on a spherical domain like the 2D illustration in Fig. 3.1. The problem variables are the fluid velocity $\mathbf{u}$, the dynamic pressure p, and the temperature T. Rock viscosity is denoted by μ, thermal conductivity by κ, and the direction and magnitude of gravity is given by $\mathbf{g}$. The density $\rho(T)$ depends on temperature. Simple models assume an expansion around some reference values, $\rho(T) = \rho_{\text{ref}}(1 - \beta(T - T_{\text{ref}}))$, where β is the thermal expansion coefficient. More elaborate models rely on tabulated density values for mantle materials at various temperatures. The heat source γ is typically modeled to contain effects from radioactive decay and viscous heating,

$$\gamma = \gamma_{\text{radiogenic}} + 2\mu\varepsilon(\mathbf{u}) : \varepsilon(\mathbf{u}) = \gamma_{\text{radiogenic}} + \mu \sum_{ij} \frac{\partial u_i}{\partial x_j} \frac{\partial u_i}{\partial x_j}.$$

Other heat generating phenomena like chemical separation within the mantle or crystallization effects are not as straight-forward to model, see [113].
Boundary conditions in mantle convection simulations are typically no-slip velocity conditions at the inner core boundary and free-slip conditions at the outer mantle boundary, see Fig. 3.1. In global mantle simulations, Dirichlet-type conditions for the temperature are supplied both on the hot inner and the cold outer boundaries. Alternatively, adiabatic conditions on the outer boundary or prescribed heat fluxes are sometimes also used, especially in the process of validation of numerical models based on simplified smaller geometries.

Numerical simulation is essential in the understanding of the dynamics in the mantle of planets, as the flow quantities cannot be accessed directly. For

the Earth's mantle, numerical models can be verified by measurements close to the surface, like thermal fluxes, glacial rebound, seismic activities, or the gravitational field.

Typical mantle materials are characterized by high dynamic viscosities μ, often in the range of $10^{19} - 10^{23}$ $\frac{\text{kg}}{\text{m s}}$. For length scales in the range of planetary dimensions, i.e., 10^6 m, and buoyancy forces $\rho \mathbf{g}$ in the range of 10^4 $\frac{\text{kg}}{\text{m}^2\text{s}^2}$, typical flow velocities are of magnitude 10^{-8} $\frac{\text{m}}{\text{s}}$, i.e., 10–100 centimeters per year. The dynamic pressure is of magnitude 10^8 Pa, and temperatures are in the range of a few thousands of Kelvin. Handling quantities at these disparate orders of magnitude introduces requires special care, e.g., in the choice of iterative linear solvers where solutions of systems of equations are only performed approximately and suitable criteria to decide when a discrete residual is small enough need to be established. To cope with very large or very small numbers, numerical techniques must rely on residuals that are measured relative to the magnitude in matrix and right-hand-side entries. In addition, disparate scales of the components in a block system like when solving the Stokes equations (3.1)–(3.2) with a coupled linear solver need to be avoided. The approach taken in Paper IV is to rescale the continuity equation (3.2) and the pressure variable to obtain well-balanced magnitudes in the numerical computations.

An alternative approach is to introduce a non-dimensional form of the equations and quantify the strength of buoyancy forces $\rho \mathbf{g}$ by the non-dimensional Rayleigh number. If we consider convection in the earth's mantle driven only by internal heating, we can define a Rayleigh number

$$\text{Ra}_\gamma = \frac{|\mathbf{g}| \rho_0 \beta \gamma D^5}{c_p \mu \kappa^2},$$

where $|\mathbf{g}|$ denotes the magnitude of gravitational acceleration, ρ_0 a reference density, γ denotes the internal heat generation per unit mass, $D = R_{\text{outer}} - R_{\text{inner}}$ the depth of the mantle, and c_p is the specific heat at constant pressure, see [25]. For convection that is driven by the temperature difference between the hot inner (core) boundary and the cold outer boundary instead, an alternative Rayleigh number can be defined,

$$\text{Ra}_{\Delta T} = \frac{|\mathbf{g}| \rho_0 \beta (\Delta T) D^3}{\mu \kappa},$$

where ΔT is the temperature difference between inner and outer boundary. Since the real situation in the mantle is a combination of different driving mechanisms, one often chooses the most significant contribution when defining a characteristic Rayleigh number [113].

For the Boussinesq equations in non-dimensionalized form, the Rayleigh number is the only parameter left (μ, κ, and γ, if present, are of unit value). As the Rayleigh number in typical mantle convection simulations is of magnitude

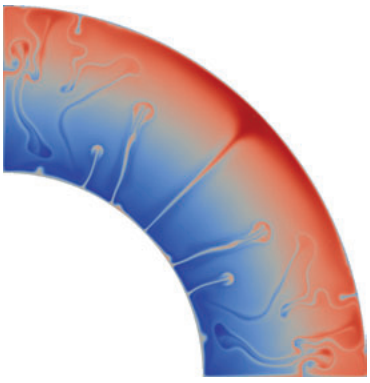


Figure 3.1. Spot at temperature distribution in a mantle convection simulation on a two-dimensional quarter shell. The inner boundary is hot and the outer boundary cold, and buoyancy transports the hot material towards the outer boundary. The small-scale plumes in this simulation are non-stationary and localized. Image from Paper IV.

between 10^6 and 10^8 or even larger, the ratio between convective transport of temperature, $\mathbf{u}T$, and diffusive transport, $\kappa\nabla T$, is very large. This implies that the temperature equation is strongly convection dominated, which needs to be taken into account when choosing the numerical discretization, see Sec. 4.2 below. A physical consequence of high Rayleigh number flow is the fact that the flow becomes non-stationary, opposed to flow at low Rayleigh numbers where the solution develops a steady state. At moderate Rayleigh numbers in the range of 10^5 and on simple problems, mantle convection develops periodic orbits as observed in one of the benchmark cases in [18]. At even higher Rayleigh numbers or on realistic shell geometries, the flow becomes completely unstable and so-called turbulent mantle convection develops [113]. Fig. 3.1 shows a numerical results for a typical two-dimensional mantle convection process in the high Rayleigh number regime. After an initial transition period, plumes develop which represent non-stationary rise of hot material from the bottom.

Extended mantle convection models

One way to make mantle convection models more realistic compared to the configuration considered in Paper IV is to introduce a dependence of the viscosity μ on the temperature. Usual viscosity contrasts between hot and cold regions can be 10^6 and even larger as remarked in [125]. In numerical models, an exponential relation of the form

$$\mu = \mu_0 \, \mathrm{e}^{E(T_0 - T)}$$

is popular [139], where E denotes a viscosity factor and μ_0 the value of the viscosity at the reference temperature T_0. Heterogeneity in the material and

temperature distributions can lead to sharp changes in μ, which requires robust discretization methods and linear solvers [45, 75].
Also, the Boussinesq limit of small density variations in the underlying equations is not completely realistic either, and one can allow for compressibility in the flow equations. This yields a modified continuity equation $\nabla \cdot (\rho \mathbf{u}) = 0$ with temperature- and pressure-dependent density ρ, for which the Stokes system becomes non-symmetric and, hence, numerically even more challenging.

4. Computational aspects of coupled flow-transport problems

The structure of the model used for the multi-phase flow application presented in Chapter 2 and the mantle convection from Chapter 3 enables the use of similar implementation concepts. Both applications are described by incompressible flow coupled to a scalar transport-type equation, with the transport driven by the flow field. For level set models, the level set variable is simply advected according to (2.8), whereas the equation is of convection-diffusion type in phase field models (2.13) and mantle convection (3.3). In all cases, the scalar quantity enters the momentum equation as a source, either through a surface tension term (2.15) or the temperature-dependent density. This nonlinear coupling of the equations poses several numerical challenges. In this chapter, we address the main steps in the numerical treatment of these coupled systems. Compared to the limited description provided in Papers I–III, the discussion included here focuses on all the essential steps in the discrete approximation for the Navier–Stokes equations, including efficient linear solvers.

For spatial discretization, there are various methods available. The most popular schemes are the finite volume method, the finite difference method, and the finite element method. A comparative introduction of these methods can be found, e.g., in Quartapelle [105]. The algorithms developed in this thesis are based on a finite element space discretization because of the flexibility of this method with respect to geometry representation and adaptive mesh refinement. There is a vast literature discussing various aspects of finite elements for the incompressible flow equations (2.1)–(2.2). Mathematical aspects are discussed, e.g., in Glowinski [48] and Gunzburger [59]. The books by Gresho and Sani [51, 52] focus on practical issues for implementation such as boundary conditions, discretization schemes, and stabilization. The work by Elman, Silvester & Wathen [34], Quarteroni & Valli [106], and Turek [133] put special emphasis on algorithmic matters like efficient numerical linear algebra. Throughout this section, we will mainly discuss the steps for discretizing the incompressible Navier–Stokes equations (2.1)–(2.2) coupled to the level set equation (2.8). Similar steps are taken for the Boussinesq system (3.1)–(3.3). Deviations in the treatment are explicitly addressed.

4.1 Weak form and spatial discretization

The equations of incompressible flow (2.1)–(2.2) depend on both space and time. Most implementations split the discretization into separate approximations for the spatial and temporal part.

For setting up the spatial finite element approximation, the first step is to rewrite the system of equations (2.1)–(2.2) in variational form. To this end, we define the space of admissible velocity solutions at a given instant in time by $\mathcal{V}_{\mathbf{u}} = \left\{\mathbf{u} \in H^1(\Omega); \mathbf{u} = 0 \text{ on } \partial\Omega_{\mathrm{w}}, \mathbf{n}\cdot\mathbf{u} = 0 \text{ on } \partial\Omega_{\mathrm{s}}\right\}$, where $\partial\Omega_{\mathrm{w}}$ denotes wall boundaries and $\partial\Omega_{\mathrm{s}}$ boundaries where the flow field may slip but no normal flux through the boundary is allowed (this condition is also used for symmetry boundaries). Hence, $\mathcal{V}_{\mathbf{u}}$ is the space of all square integrable vector-valued functions on Ω with integrable first derivatives that satisfy the boundary conditions. For simplicity, we here assume that the boundary is covered by these two types of boundaries, see [52] for additional forms for inflow and outflow boundaries. The space of admissible pressure solutions is defined by $\mathcal{V}_p = L_2(\Omega)$. Let us denote the standard L_2 inner product on Ω by $(\cdot,\cdot)_\Omega$. The variational problem corresponding to (2.1)–(2.2) is to find, at each instant in time, a pair $(\mathbf{u}(\cdot,t), p(\cdot,t)) \in \mathcal{V}_{\mathbf{u}} \times \mathcal{V}_p$ such that

$$\begin{aligned}\left(\mathbf{v}, \rho\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}\right)\right)_\Omega + (\varepsilon(\mathbf{v}), 2\mu\varepsilon(\mathbf{u}))_\Omega \\ - (\nabla\cdot\mathbf{v}, p)_\Omega - (q, \nabla\cdot\mathbf{u})_\Omega = \left(\mathbf{v}, \mathbf{f}\right)_\Omega\end{aligned} \tag{4.1}$$

holds for all test functions $(\mathbf{v}, q) \in \mathcal{V}_{\mathbf{u}} \times \mathcal{V}_p$. Similarly, the weak equation for the level set variable is to find $\phi \in H^1(\Omega)$ such that

$$\left(\xi, \frac{\partial \phi}{\partial t} + \mathbf{u}\cdot\nabla\phi\right)_\Omega = 0 \quad \text{for all } \xi \in \mathcal{V}_\phi = H^1(\Omega). \tag{4.2}$$

In the spatial discretization of systems (4.1) and (4.2), the (infinite-dimensional) solution function spaces $\mathcal{V}_{\mathbf{u}}$, $\mathcal{V}_p$, and $\mathcal{V}_\phi$ are replaced by finite-dimensional subspaces $\mathcal{V}_{\mathbf{u}}^h$, $\mathcal{V}_p^h$, and $\mathcal{V}_\phi^h$. Then, only the projection of the equations onto finite dimensional test function spaces $\mathcal{V}_{\mathbf{u}}^h \times \mathcal{V}_p^h$ is considered, resulting in numerical approximations $\mathbf{u}^h$ and p^h, and similarly for ϕ^h. The discretizations considered in this thesis use a decomposition of the computational domain into elements of characteristic size h. We mainly focus on quadrilateral elements in 2D and hexahedral (brick) elements in 3D in this thesis. The basis functions that span $\mathcal{V}_{\mathbf{u}}$ and $\mathcal{V}_p$ are chosen to be piecewise Lagrange polynomials within the elements. Globally, the basis functions are nonzero in exactly one node, and zero on all the others. The support of the basis functions is confined to a patch of elements around the respective nodes. For velocity, continuity of the approximation over the whole domain Ω is enforced to make sure that $\mathcal{V}_{\mathbf{u}}^h \subset \mathcal{V}_{\mathbf{u}}$. In the pressure, discontinuities over element boundaries are allowed. Apart from some results in Paper IV, all the computations were performed on

continuous pressure spaces, though. For the representation of the velocity $\mathbf{u}^h$, equally shaped basis functions are used for each spatial component. For instance, we will denote a finite element approximation with quadratic basis functions for velocity and linear basis functions for pressure by $\mathcal{Q}_2^d\mathcal{Q}_1$. For this finite element, the basis functions for $\mathbf{u}^h$ are defined as a tensor product of Lagrangian interpolation polynomials of degree two in each coordinate direction, and the pressure basis functions as tensor product of linear Lagrangian interpolants.

With constant coefficients, the viscous term is often reformulated using the identity $\nabla \cdot 2\mu\varepsilon(\mathbf{u}^h) = \mu\nabla\cdot\nabla\mathbf{u}^h + \mu\nabla(\nabla\cdot\mathbf{u}^h)$, dropping the second term because of incompressibility. This gives a block Laplacian operator, for which the final matrix is sparser by a factor $\frac{1}{d}$, which translates to a similar speedup when solving the linear systems. Since we consider systems where viscosity is *variable* because of multiple phases or temperature-dependent viscosity, we keep the original form. The incompressibility condition also allows to use different discrete forms of the convective term $\mathbf{u}^h\cdot\nabla\mathbf{u}^h$. Sometimes, the skew-symmetric form $\mathbf{u}^h\cdot\nabla\mathbf{u}^h + \frac{1}{2}\mathbf{u}^h(\nabla\cdot\mathbf{u}^h)$ is preferred over the convective form used in (4.1) as it ensures discrete energy conservation. We refer to Quarteroni and Valli [106] for a discussion of different forms.

4.2 Stability of spatial discretization

For stability of the spatial discretization of the incompressible flow equations (4.1), a compatibility condition between the approximation of velocity and pressure must be satisfied, namely

$$\inf_{q^h\in\mathcal{V}_p^h}\sup_{\mathbf{v}^h\in\mathcal{V}_\mathbf{u}^h}\frac{\left(q^h,\nabla\cdot\mathbf{v}^h\right)_\Omega}{\left\|q^h\right\|_{L_2}\left\|\mathbf{v}^h\right\|_{H^1}} \geq \beta > 0, \tag{4.3}$$

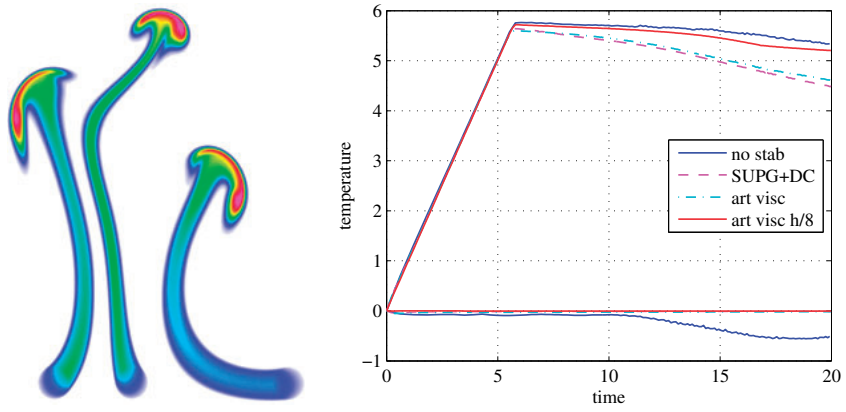
where the constant β does not depend on the mesh size parameter h. This condition is called *LBB* (Ladyzhenskaya, Babuška, Brezzi) or *inf–sup condition* [46, 34]. In particular, the condition also ensures that the discrete divergence matrix discretizing the term $\left(q^h,\nabla\cdot\mathbf{u}^h\right)_\Omega$ is of full row rank, guaranteeing a unique solution for velocity and pressure. The inf–sup condition restricts the choice of stable element combinations for velocity and pressure, for instance the element combination $\mathcal{Q}_1^d\mathcal{Q}_1$ is unstable, whereas the element combination $\mathcal{Q}_2^d\mathcal{Q}_1$, the so-called Taylor–Hood element pair, satisfies the inf–sup condition (see [46, 52, 34]).

The standard Galerkin method leads to central difference approximations of first derivatives. The solution to this approximation gets spoiled by spurious node-to-node oscillations around steep gradients once the element Péclet number $\mathrm{Pe} = h|\mathbf{u}^h|/\kappa$ is significantly larger than one [31, 51, 52]. Among the first systematic approaches to address the insufficiencies in the Galerkin

finite element method for flow problems is the streamline upwind/Petrov–Galerkin (SUPG) method for dominating convection [24], and the pressure-stabilization/Petrov–Galerkin (PSPG) method for circumventing the inf–sup condition [69]. The idea of this class of stabilized methods is to add additional terms based on the element-wise residual of the differential operator tested against $\mathbf{v}^h \cdot \nabla \mathbf{v}^h + \nabla q^h$ in the Navier–Stokes case, which can be interpreted as testing with modified test functions compared to the solution functions, a so-called Petrov–Galerkin approach. SUPG/PSPG stabilization techniques have been refined over the years [128], including theoretical support through the framework of the variational multiscale method, see, e.g., [14]. The variational multiscale method interprets the terms in residual-based stabilization as the representation of information from unresolved scales and was, e.g., used as a framework for turbulence simulations based on high order [14] and low-order finite element methods [49, 50]. Extensive work has been done on similar stabilization techniques, like the streamline diffusion method [62], edge stabilization [26], and local projection stabilization [21]. We refer to [22] and references therein for a critical assessment of these methods.

In Paper I, an SUPG/PSPG stabilization for the Navier–Stokes system has been used to perform experiments with the a priori unstable element pair $\mathcal{Q}_1^d \mathcal{Q}_1$. Apart from this setting, all other parts in this thesis consider laminar Navier–Stokes or Stokes flow with inf–sup stable elements where no stabilization in velocity and pressure is necessary. In the mantle convection, Péclet numbers in the temperature equation are often 10^3 and larger even for highly resolved computations and, hence, stabilization for temperature is essential, though. In the level set evolution equation as well as reinitialization (2.9), no physical diffusion at all is present, leading to similar difficulties. In the conservative level set method, the diffusion in the reinitialization equation (2.10) ensures that the discretization remains stable. Nonetheless, we observed that fewer reinitialization steps were necessary when advection was performed with a stabilized formulation.

On problems with sharp gradients like in mantle convection, the stabilization provided by SUPG-type methods is sometimes not enough. To address this problem, so-called discontinuity capturing has been proposed [127, 15]. This approach adds artificial diffusion whose amount depends on the gradient of the solution or its residual. The idea of artificial viscosity based on residuals has been considered recently without any further stabilizing terms like SUPG. In [56], a stabilization based on an entropy associated to the solution variable was proposed, whereas [92] considered the residual of the equation itself. Opposed to global stabilization methods like SUPG, methods with diffusion proportional to residuals allow to concentrate numerical dissipation to regions where it is actually needed, namely around steep gradients where a diffusion term similar to the first-order upwind schemes [86] is introduced. In smooth regions, on the other hand, residuals are small and little or no dissipation is added.



(a) Temperature field at $t = 20$ (b) Temperature minimum and maximum over time

Figure 4.1. Comparison of stabilization approaches for mantle convection at $\mathrm{Ra} = 10^6$, driven by three circular heat sources near the bottom. The computation is based on a domain of unit size with $\mathscr{Q}_2$ elements for temperature on an adaptively refined mesh with finest mesh size $h = 0.004$. An unstabilized method (no stab), a residual-based stabilization with YZβ discontinuity capturing with $\beta = 1$ as in [15] (SUPG+DC), and an artificial viscosity based on the residual of the temperature entropy according to [56] (art visc) are compared. As a reference, a highly resolved computation with $h = 0.0005$ and the artificial viscosity stabilization is included.

Fig. 4.1 gives an illustration of the effect of stabilization on a convection-dominated problem in mantle convection. It can be seen that an unstabilized computation leads to artificial over- and undershoots in the temperature extrema. A closer inspection of the solution field shows oscillations in considerable portions of the solution, especially close to the front of high temperature. If stabilization is applied, the quality of the solution is considerably improved, which can be seen by the minimum temperature that is very close to zero (around -0.03 in this example). In this case, SUPG stabilization alone did not approximations free from oscillations, so a discontinuity capturing term according to [15] was included. For the example displayed in Fig. 4.1, the two stabilized approaches of artificial viscosity [56] and SUPG with discontinuity capturing show a similar performance.

4.3 Adaptive mesh refinement

The accurate simulation of both two-phase flow and mantle convection relies on resolving relatively small features in the problem variables. To design efficient numerical methods, one needs to be able to concentrate high resolution to regions where they have the biggest impact on the quality of the solution. For the simulations in this thesis, meshes are dynamically refined and coars-

ened as the solution evolves in time. This allows the finest regions of the mesh to be considerably finer than would be possible with uniformly refined meshes. Adaptive strategies have been demonstrated to improve computational efficiency with up to a factor 100 in 2D and about 1000 in 3D compared to uniform meshes [4, 7, 12].

Two-phase flow

In two-phase flow, the region around the interface is a candidate for high resolution. On the one hand, changes in material parameters from one fluid to the other or the surface tension reduce the regularity in the solution fields. Moreover, for continuous surface tension models in the spirit of Brackbill et al. [23], one wishes to make the region over which the force is spread as thin as possible. This is because smoothed approaches are justified in the limit of sharp interfaces, i.e., $h \to 0$. For phase field models, the physical diffuse interface width needs to be smaller than the size of bubbles, which makes adaptive mesh refinement an indispensable tool already for two-dimensional simulations like the studies for the sharp interface limit by Yue et al. [137, 136].

In many two-phase flow applications, simple strategies are used as refinement indicators, like refining cells that are sufficiently close to the interface as for phase field methods [134] or the conservative level set method [95]. A similar strategy has been used for many of the results presented in this thesis. Other studies base the refinement on jump residuals [5].

Mantle convection

Similar to turbulent incompressible Navier–Stokes flow at high Reynolds numbers [104], the flow features in mantle convection become smaller with increasing Rayleigh number, requiring increasing spatial and temporal resolution. As opposed to turbulent flow away from boundary layers where eddies are evenly distributed and require high resolution basically everywhere in the computational domain for direct numerical simulation, the plumes in mantle convection are more localized, which allows for effective use of adaptive mesh refinement, as pointed out, e.g., in [27]. The temperature plot in Fig. 3.1 also illustrates that jumps in the temperature field indicate regions with high resolution requirements. A different view on this refinement strategy is the aim to reduce the impact of artificial dissipation of magnitude $|\mathbf{u}|h$ near gradients by reducing the mesh size in these regions. The evaluation of changes in the scalar field can be realized with similar implementation techniques as for two-phase flow. In Paper IV, a standard tool from error estimation on elliptic problems is used, the so-called Kelly error indicator [39] that measures the jump in gradients over cell boundaries.

4.4 Time discretization

The system of the incompressible Navier–Stokes equations (2.1)–(2.2) is coupled nonlinearly to the level set evolution equation (2.8). In addition, there is no time derivative on the pressure variable (and none on the velocity either, for the Stokes equations). This mixture of differential equations and algebraic equations yields a differential–algebraic system after discretization which is more difficult to solve than pure ordinary differential equations (see Hairer et al. [60, 61] for derivation and analysis of numerical schemes).

A popular approach is to split the advection-type equation from the incompressible flow equations and to treat them after one another. Splitting approaches combine explicit and implicit discretization of different terms in the equations. For higher order accuracy, several values from old time levels can be combined in the explicit stage, e.g., in stages of Runge–Kutta schemes or using Adams–Bashforth methods [60]. If we assume that the velocity in the advection equation is treated explicitly, the level set variable can be propagated in time, and the new values are available for the propagation of the Navier–Stokes system from the old to the new time level.

For Navier–Stokes time stepping schemes that treat both convection and diffusion explicitly, linear stability theory sets a restrictive limit on the time step Δt in terms of the spatial mesh size h, namely

$$\Delta t_{\max} \sim \min \left\{ \frac{h}{\|\mathbf{u}\|_\infty}, \frac{\rho h^2}{\mu} \right\}, \tag{4.4}$$

see, e.g., the discussion in [79]. Therefore, for applications where the viscosity is not too small compared to the velocity, (semi-)implicit time stepping schemes are often preferred over fully explicit ones. Popular schemes are variants of the one-step theta method [78] and the Backward Difference Formula of second order (BDF-2). Most results in this thesis have been obtained with the BDF-2 method. Stable time stepping on the Navier–Stokes block system requires implicit discretization of the (differential–algebraic) terms $(\nabla \cdot \mathbf{v}, p)_\Omega$ and $(q, \nabla \cdot \mathbf{u})_\Omega$. These terms enforce the divergence-free condition on the velocity in Lagrangian multiplier form, see, e.g., [59]. In such a setting, a coupled linear system of saddle-point structure arises and needs to be solved for each time step. The problem of finding optimal linear solvers for these problems is challenging and will be discussed in Sec. 4.5 below.

In order to avoid saddle-point linear systems, Harlow and Welch [63] proposed to perform time propagation of the Navier–Stokes equations by a *fractional time stepping strategy*, which was then developed further to a projection scheme by Chorin and Temam [30, 126]. Fractional-step schemes first advance velocities subject to the momentum equation with pressure extrapolated from old time levels. The resulting velocity is in general not divergence-free, so a pressure Poisson equation with forcing given by the divergence of the intermediate-step velocity is solved, in order to correct the velocity. The two

linear subproblems arising after spatial and temporal discretization are of considerably easier structure than the original saddle-point system in the coupled approach. In particular when convection is treated explicitly so that the terms at the new time level in the momentum equation are linear and have constant coefficients [79], this plays an important role. For these reasons, projection schemes are to date the most popular strategy for the numerical approximation of the instationary Navier–Stokes equations, especially in the medium and high Reynolds number regime [105, 79]. However, each of the subproblems needs to be equipped with suitable boundary conditions for well-posedness. Indeed, many naïve implementations of projection methods suffer from non-consistent approximations close to boundaries. We refer to the review paper [55] and references therein for an elaboration of the state-of-the-art. In the solution of multi-phase flow systems, the densities in different fluids are in general different. In straight-forward projection schemes as the one proposed in [57], the variation in density propagates into the Poisson equation of the pressure, giving rise to a Poisson operator of the form $\left(\nabla q, \frac{1}{\rho}\nabla p\right)_\Omega$. Strong variations in density like for liquid-gas systems make fast Poisson solvers less efficient. Recently, a projection method based on a constant-coefficient Poisson problem has been proposed [58].

In case the convection term $\mathbf{u}\cdot\nabla\mathbf{u}$ is treated implicitly, the nonlinearity needs to be resolved. One variant is the semi-implicit treatment of the form $\mathbf{u}^{n-1}\cdot\nabla\mathbf{u}^n$. For fully implicit methods, the nonlinearity can be treated by a Picard or full Newton iteration [34]. The Picard iteration or the semi-explicit form have the advantage that the contribution of the convection term to the velocity matrix becomes block-diagonal, enabling faster solution of linear systems. However, implicit treatment of convection is not always necessary. For problems where small time steps are needed for accuracy reasons (like in turbulent flow simulations or many of the popular fractional step solvers), the CFL condition (4.4) is no real restriction and it avoids evaluating the convective term in the linear system, gaining overall efficiency as pointed out, e.g., in [79].

We note that solutions from old time steps can be used to make iterative linear solvers more efficient by creating good initial guesses by extrapolation. As demonstrated in Paper IV, the iteration count for a linear Stokes solver can be roughly halved by using second-order extrapolated values instead of zero initial guess or constant extrapolation.

In many situations, the time step limit introduced by the splitting between transport equation and the incompressible flow equations leads to acceptable step sizes of the order $\Delta t \sim h/\|\mathbf{u}\|_\infty$. However, in two-phase flow simulations with the level set methods and explicit treatment of surface tension terms, an additional stability condition arises as pointed out in [23]. In its full generality,

the time step limit imposed by explicit treatment of surface tension reads

$$\Delta t_{\sigma,\max} = c_1 \frac{\mu}{\sigma} h + \sqrt{c_2 \frac{\rho}{\sigma} h^3}, \tag{4.5}$$

where c_1 and c_2 are constants that do not depend on the mesh size h and the material parameters, cf. [40]. The viscosity μ and density ρ represent the arithmetic mean of the values of the two fluids. If surface tension is large, the term $\sqrt{\rho/\sigma} h^{3/2}$ sets a more restrictive limit on the time step when resolving the propagation of so-called capillary waves. This limit is related to the interaction of inertial terms in the momentum equation and the surface tension term and appears also when velocity is treated explicitly in the level set equation (2.8) and surface tension "pseudo-implicitly" with the newly computed level set values. Several techniques have been proposed to overcome this limit, like the solution of an additional evolution equation for the mean curvature [122, 120] or the semi-implicit treatment of surface tension [71] based on the Laplace–Beltrami formulation [32].

4.5 Numerical linear algebra for discrete systems

After time discretization and linearization, the discrete problem for Navier–Stokes equations in weak form (4.1) is to find discrete approximations

$$\begin{aligned} \mathbf{u}^h &= \sum_{i=1}^{N_\mathbf{u}} \varphi_\mathbf{u}^{(i)} u_i, \\ p^h &= \sum_{j=1}^{N_p} \varphi_p^{(j)} p_j, \end{aligned} \tag{4.6}$$

which satisfy the weak form for test functions $\mathbf{v} = \varphi_\mathbf{u}^{(i)}$, $i = 1, \ldots, N_\mathbf{u}$ and $q = \varphi_p^{(j)}$, $j = 1, \ldots N_p$. Here, $\varphi_\mathbf{u}^{(i)}$ denotes the finite element basis functions for the velocity and $\varphi_q^{(i)}$ the basis functions for pressure. For simplicity of notation, let us assume backward Euler time stepping on the coupled Navier–Stokes operator (i.e., no projection) and a semi-implicit treatment of the convection term of the form $\mathbf{u}^{h,n-1} \cdot \nabla \mathbf{u}^{h,n}$ in this section. Assume further that the material parameters ρ and μ are constant. The result is the following linear system of equations in saddle point form

$$\begin{pmatrix} A & B^\mathsf{T} \\ B & 0 \end{pmatrix} \begin{pmatrix} x_\mathbf{u} \\ x_p \end{pmatrix} = \begin{pmatrix} b_\mathbf{u} \\ b_p \end{pmatrix}, \tag{4.7}$$

where the matrix elements are given by the integrals

$$\begin{aligned}
A &= \frac{\rho}{\Delta t} M + \rho C\left(\mathbf{u}^{h,n-1}\right) + \mu L, \\
M_{ij} &= \left(\varphi_{\mathbf{u}}^{(i)}, \varphi_{\mathbf{u}}^{(j)}\right)_{\Omega} && \text{(mass matrix)}, \\
C_{ij}\left(\mathbf{u}^{h,n-1}\right) &= \left(\varphi_{\mathbf{u}}^{(i)}, \mathbf{u}^{h,n-1} \cdot \nabla \varphi_{\mathbf{u}}^{(j)}\right)_{\Omega} && \text{(convection matrix)}, \\
L_{ij} &= \left(\varepsilon\left(\varphi_{\mathbf{u}}^{(i)}\right), 2\varepsilon\left(\varphi_{\mathbf{u}}^{(j)}\right)\right)_{\Omega} && \text{(viscosity matrix)}, \\
B &= -\left(\nabla \cdot \varphi_{\mathbf{u}}^{(i)}, \varphi_{p}^{(j)}\right)_{\Omega} && \text{(divergence matrix)},
\end{aligned}$$

the vectors $x_{\mathbf{u}} = (u_1, \ldots, u_{N_{\mathbf{u}}})^{\mathsf{T}}$, $x_p = (p_1, \ldots, p_{N_p})^{\mathsf{T}}$ denote the discrete solution vectors for $\mathbf{u}$ and p, respectively, and the right hand side vectors are given by

$$b_{\mathbf{u}} = \left(\varphi_{\mathbf{u}}^{(i)}, \mathbf{f}\right)_{\Omega}, \quad b_p = 0.$$

In system (4.7), the pressure gradient matrix B^{T} is the transpose of the divergence matrix. We note that rows and columns corresponding to essential boundary conditions are eliminated from the matrix, ensuring full rank of the resulting system. Since the basis functions are localized to a patch of elements around the node in which they are non-zero, the resulting matrices are sparse.

The solution of these linear system is the final task in finding a discrete approximation. In most flow codes, the solution of linear systems takes the largest proportion of computing time. Hence, efficient solvers are an essential tool in the numerical simulation. For realistic problem sizes, direct solvers are not an option because of excessive storage requirements and non-optimal scaling with the problem size. Instead, iterative Krylov subspace methods are the preferred option [111]. If paired with effective preconditioners, iterative solvers allow to solve a linear problem of size N in $\mathcal{O}(N)$ operations (though usually with quite large proportionality constants between 1,000 and 100,000).

The discretization of the level set or phase field equation results in a linear system with a system matrix that is dominated by a scalar mass matrix, which can be solved by straight-forward iterative techniques [111]. For the solution of the Cahn–Hilliard equation, an efficient method has been proposed in [19].

For solving the saddle-point Stokes and Navier–Stokes systems (4.7), a vast number of different techniques have been proposed. The review paper by Benzi, Golub, and Liesen [16] collects the many significant contributions for these systems, see also [133, 34]. Efficient schemes include multigrid methods with suitably defined smoothers (cf. [130]), segregated solution approaches, or use preconditioners based on an approximate Schur complement of the block 2×2 matrix system defined by velocity and pressure matrices. Here, we summarize the methods used in the work in this thesis.

4.5.1 Stokes equations

We start the discussion with the Stokes matrix, i.e., the system

$$\begin{pmatrix} \mu L & B^{\mathsf{T}} \\ B & 0 \end{pmatrix} \begin{pmatrix} x_{\mathbf{u}} \\ x_p \end{pmatrix} = \begin{pmatrix} b_{\mathbf{u}} \\ b_p \end{pmatrix}, \tag{4.8}$$

solved with an iterative Krylov method. Because of the zero pressure-pressure block, the matrix system is indefinite and simple solvers like conjugate gradient are not applicable. Instead, viable solvers are GMRES, SymmLQ, MinRes, and BiCGStab [111]. Since the preconditioner applied in the thesis is non-symmetric (see below), SymmLQ and MinRes are not possible either. In our simulations, the BiCGStab and GMRES show similar performance, and convergence speed depends more on the quality of the preconditioner and less on the specific iterative solver. One advantage of GMRES is that it allows for right preconditioning where the true residual is readily available as a stopping criterion, compared to the preconditioned residual that arises with left preconditioning [34]. Therefore, we choose GMRES in this work.

The condition number of matrix L increases as the mesh size h becomes smaller. Since the number of iterations needed by GMRES increases with increasing condition number, solvers without preconditioner deteriorate for increasing problem size. For the Stokes equations, an optimal preconditioner based on the Schur complement of the 2×2 block system was presented by Silvester and Wathen [117]. One defines a preconditioner by the block triangular matrix

$$P = \begin{pmatrix} \mu L & B^T \\ 0 & -S \end{pmatrix} \Leftrightarrow P^{-1} = \begin{pmatrix} (\mu L)^{-1} & (\mu L)^{-1} B^{\mathsf{T}} S^{-1} \\ 0 & -S^{-1} \end{pmatrix}, \tag{4.9}$$

where $S = B(\mu L)^{-1} B^{\mathsf{T}}$ denotes the Schur complement. If we apply P^{-1} as a right preconditioner, we obtain

$$\begin{pmatrix} \mu L & B^T \\ B & 0 \end{pmatrix} P^{-1} = \begin{pmatrix} I & 0 \\ B(\mu L)^{-1} & I \end{pmatrix}.$$

The preconditioned matrix thus only has eigenvalues of unit value with a minimum polynomial of degree two, for which GMRES can be shown to converge in at most two iterations [117, 16]. In order to avoid explicit inversion of the matrix μL for the Schur complement S, an approximation of the Schur complement is needed. Since B represents a discrete divergence operator, μL a discrete Laplacian, and B^{T} a discrete gradient, the resulting operator is spectrally equivalent to an identity operation on the pressure space. In the finite element setting, the operator is represented by the pressure mass matrix M_p. In case the viscosity is variable, an optimal approximation to the Schur complement is the mass matrix scaled by the inverse of the viscosity, i.e., $M_p^{(ij)} = \left(\varphi_p^{(i)}, \frac{1}{\mu}\varphi_p^{(j)}\right)_\Omega$, cf. [53].

For practical use of the preconditioner, the inverses $(\mu L)^{-1}$ and M_p^{-1} need to be approximated in (4.9). For the latter, either the diagonal of the pressure mass matrix or an incomplete LU (ILU) decomposition restricted to the sparsity structure of M_p yield optimal performance [16, Sec. 10.1.3]. For constant coefficients, the velocity matrix μL is dominated by the vector Laplacian, for which multigrid methods [130] give high quality approximations that do not depend on the mesh size. As pointed out in [44], geometric multigrid methods are less robust when coefficients vary strongly. Also, geometric multigrid is not straight-forward to design when using adaptively refined meshes, even though progress has been made recently [77]. Therefore, algebraic multigrid (AMG) methods are often the preferred preconditioner for approximating $(\mu L)^{-1}$. AMG has been demonstrated to be very robust to large contrasts in coefficients [42] and to scale well on massively parallel computers [132, 3]. AMG methods do not rely on a hierarchy of grids as geometric multigrid, but build a hierarchy based on the connectivity between different rows in the matrix.

Our work on Stokes systems in Papers II and IV is based on an AMG preconditioner based on the implementation in [43], computed from the entries of the vector Laplacian also in the case of variable coefficients. Even though omitting the coupling between components may reduce the quality of the approximation in the variable-coefficient case, it generally improves performance of algebraic multilevel preconditioners which perform best for scalar Poisson-type problems. In addition, the comparatively expensive operations when evaluating the preconditioner (smoothing, restriction, and prolongation) are performed on a sparser matrix, which reduces computational costs in practice. Another variant to significantly reduce computational costs for higher order finite elements is to build the AMG preconditioner for linear elements on a subdivided mesh with the same number of degrees of freedom. The resulting matrix is sparser but has the same spectral properties, as proposed, e.g., in [98, 16].

For certain problems, it is enough to simply use one or a few V-cycles of the algebraic multigrid preconditioner (see, e.g., [34]), whereas in other cases more accurate inversion of the velocity matrix (e.g., using CG with a loose tolerance) is necessary to obtain robust methods. In that case, the application of a preconditioner is a nonlinear operation and generally different from one iteration to the next. The flexible GMRES method (FGMRES) tolerates such variable preconditioners [111].

4.5.2 Navier–Stokes equations

The structure of the Navier–Stokes system (4.7) is similar to the one of the Stokes system (4.8). However, the velocity matrix A is non-symmetric because of convection, and its presence makes finding good realizable approximations

to the Schur complement $S = BA^{-1}B^{\mathsf{T}}$ difficult. In addition, a vector mass matrix on the velocity is introduced. For small time steps, the mass matrix dominates, whereas for $\Delta t \to \infty$ the discrete time-dependent Navier–Stokes system approaches the stationary Navier–Stokes system. For the limit case (the Oseen system), the approximation of the Schur complement is particularly intricate. Many proposed preconditioners depend on the magnitude of viscosity and deteriorate as $\mu \to 0$. This is the case, e.g., for the Schur complement approximation using a pressure convection-diffusion operator as in [81]. A different approach is to add an additional contribution to the velocity-velocity block in order to control the Schur complement, a so-called augmented Lagrangian approach [17]. Other directions of research have been to avoid the construction of Schur complements and explicitly embed the structure of divergence and pressure gradient matrices into a multilevel factorization process [129]. We refer to [16] for a discussion of various techniques.

In this thesis, we focus on the simpler case of the instationary Navier–Stokes equations where the mass matrix term is significant. We construct an approximation of the inverse of the Schur complement matrix S^{-1} as

$$\hat{S}^{-1} = \frac{\rho}{\Delta t} L_p^{-1} + \mu M_p^{-1}, \tag{4.10}$$

i.e., a sum of inverses of Laplace and mass matrix on the pressure space. This Schur complement approximation neglects the convective part and has been proposed for the time-dependent Stokes operator, $\widetilde{A} = \frac{\rho}{\Delta t} M + \mu L$, in [133, 90]. In this work, we found the approximation to perform extremely well also for Navier–Stokes systems at low and moderate Reynolds numbers. The approach to use a sum of inverse operators is motivated by the scalar analogy $s = \beta_{\mathrm{l}}(\nu + \kappa)^{-1}\beta_{\mathrm{r}}$, for which $s^{-1} = \beta_{\mathrm{l}}^{-1}\nu\beta_{\mathrm{r}}^{-1} + \beta_{\mathrm{l}}^{-1}\kappa\beta_{\mathrm{r}}^{-1}$, where the first term symbolically resembles the velocity mass matrix and the second term the viscous terms.

Since the pressure Laplace matrix in (4.10) is singular without boundary conditions, suitable conditions need to be applied to L_p. In our work, we choose to use natural (Neumann) conditions on L_p on boundaries where no-slip or free-slip conditions are applied, and Dirichlet conditions on inflow and outflow boundaries, motivated by similar choices sometimes made for projection schemes, see also the discussion in [34, Sec. 8.2].

As before, the inverses A^{-1} as well as L_p^{-1} and M_p^{-1} need to be approximated for an actual implementation. For the matrix M_p^{-1}, an ILU factorization is used, and an AMG V-cycle for approximating L_p^{-1}. For the matrix A^{-1}, we choose to select between two different strategies depending on the size of the time step and the spatial mesh size. If $\Delta t\mu < C\rho h^2$ for some constant $C \approx 1$, the influence of the viscosity matrix is small and an ILU decomposition on the block-diagonal part of A is used. Otherwise, the eigenvalue spectrum is significantly influenced by viscosity and an AMG V-cycle is applied. Since the quality of the ILU becomes worse when used in parallel, the constant C

Table 4.1. *Iteration numbers and computing time for one time step of the Navier–Stokes solver on 3D Beltrami flow from [37] on the domain $[-1,1]^3$ for different values of the viscosity μ, mesh size h, and time step Δt. If only a single number is given, the Newton iteration converged after one step. Otherwise, the iteration numbers to each iteration are given. If a superscript "A" is given next to the iteration numbers, an AMG preconditioner is used for approximately inverting the velocity matrix, otherwise ILU.*

μ	Δt	# its	time	# its	time
		$h = 0.167$ DoFs: 49,072		$h = 0.042$ DoFs: 2,855,668	
1	0.2	$17 + 19^{\mathrm{A}}$	0.77 s	$27 + 30^{\mathrm{A}}$	96 s
	0.02	16+6	0.18 s	$24 + 7^{\mathrm{A}}$	52 s
	0.002	8	0.069 s	19	11 s
	0.0002	4	0.038 s	4	2.5 s
0.01	0.2	50+24	0.74 s	50	39 s
	0.02	9	0.076 s	5	3.0 s
	0.002	4	0.037 s	2	1.5 s
	0.0002	2	0.022 s	2	1.5 s
0.0001	0.2	150+100	3.2 s	fail	—
	0.02	13+4	0.14 s	32	22 s
	0.002	6	0.051 s	4	2.5 s
	0.0002	2	0.022 s	2	1.5 s

is made smaller in the case of parallel computations in order to enable AMG earlier. The strategy of using a simple preconditioner on the velocity subsystem and an efficient Laplace preconditioner for pressure is also found in the design of efficient projection solvers.

Table 4.1 shows iteration counts for a 3D laminar flow with analytic solution according to [37]. Spatial discretization was performed with $\mathcal{Q}_2^d\mathcal{Q}_1$ elements. For time stepping, a fully implicit BDF-2 scheme was used with a Newton iteration to resolve the nonlinear convection in skew-symmetric form. The linear solver with preconditioners described above has been used, employing the fast matrix-vector products presented in Paper VI. The nonlinear solver was run until the absolute l_2 residual of discrete vectors was smaller than 10^{-8}, using relative residuals of 10^{-5} for the linear solver. Note that no inner iterations are performed in this example, i.e., each iteration corresponds to one matrix-vector product and one preconditioner application for velocity and pressure only. It can be seen that the solver is very efficient at small time steps, which is because the ILU preconditioner approximation for the velocity matrix A becomes better the more the mass matrix is dominating. In addition, the Schur complement approximation is improved. Lastly, the extrapolation gives better initial guesses for the linear solver when the time step is small (indeed, the residual needs to be reduced only by about 10^{-2} for small time steps). For high diffusion and large time steps, the solver is also quite efficient,

Table 4.2. *Iteration numbers in 2D Navier–Stokes solver for one time step of light bubble rising in heavy fluid [73] for different values of the mesh size h and time step Δt. The results for the finest mesh use adaptive mesh refinement around the interface with mesh size far away from interfaces $h_{\max} = 0.0063$. The simulation on the adaptive mesh at $\Delta t = 0.03$ is unstable because of the violation of the capillary time step limit* (4.5)*. If a superscript "A" is given next to the iteration numbers, an AMG preconditioner is used for the velocity-velocity block, otherwise ILU.*

Δt	# its	time	# its	time	# its	time
	$h = 0.0063$ DoFs: 463,203		$h = 0.0031$ DoFs: 1,848,003		$h_{\min} = 0.00078$ DoFs: 988,317	
0.03	$21 + 7^{\mathrm{A}}$	4.5 s	$24 + 12^{\mathrm{A}}$	21 s	unstable	—
0.003	11	0.79 s	20	5.0 s	$27 + 5^{\mathrm{A}}$	11 s
0.0003	4	0.24 s	4	0.92 s	17	3.2 s

but it breaks down for small diffusion and large time steps. This is because the approximation to the Schur complement does not include any contribution for the convective part. We note that for moderate to small time steps, the computational costs for one time step with the fully coupled solver are only slightly higher compared to efficient projection solvers. On the other hand, we observe that the fully coupled solver allows larger time steps on application problems of two-phase flow.

Table 4.2 shows the number of iterations and computing times of the Navier–Stokes solver for one time step on a two-dimensional rising bubble according to [73]. Note that the Navier–Stokes solver consumes about 80% of the total run-time in this test case, with the rest spent on level set computations. In this test case, both density and viscosity differ by a factor 10 between the two fluids. The test is performed at time $t = 0.93$ when the rise velocity of the bubble $|\mathbf{u}| = 0.242$ is close to its maximum. As before, the tolerance for the nonlinear solver is set to an l_2 residual smaller than 10^{-8} with 10^{-5} relative tolerance for the linear solver. The results show the efficiency of the given preconditioner on a relatively wide range of time steps and for different mesh sizes.

5. Software for finite element programming

The implementation of advanced numerical models for both two-phase flow and mantle convection requires a substantial programming effort. The infrastructure to handle adaptivity, finite element shape functions, as well as efficient linear algebra requires on the order of 100,000 lines of code. Therefore, libraries have been designed that provide building blocks for finite element programming, so that only the actual application (PDEs, time stepping, preconditioners) needs to be implemented. Among the hundreds of open-source projects addressing finite element programming, two successful projects used by many researchers all over the world are the deal.II library [9, 10] and the FEniCS/Dolfin project [88, 89]. The work in this thesis is based on the deal.II library. For numerical linear algebra with distributed storage of matrices or algebraic multilevel preconditioners, the PETSc [8] and Trilinos [65] libraries are common. The finite element libraries mentioned above provide interfaces to these linear algebra packages.

5.1 The deal.II library

The finite element library deal.II (short for Differential Equations Analysis Library) is a collection of various finite elements and a lot of frequently used functionality, written in C++. One of the main design objectives of deal.II is a unified interface for code in different space dimensions by the use of templates in the C++ programming language. This makes programming for different spatial dimensions transparent to the user and enables the development of programs that target both 2D and 3D. deal.II restricts the element type to quadrilaterals/bricks and implements both continuous and discontinuous element types. It supports arbitrary geometries by internal grid generators for simple geometries and interfaces to data formats from mesh generators, and allows for deforming meshes. deal.II comes with full support for adaptive mesh refinement and handles continuity in solutions as required for H^1-conforming approximations through introduction of algebraic constraints [9, 11]. The adaptive features include anisotropic meshes and h, p adaptivity.

5.2 Parallelization in deal.II

Also when using adaptive mesh refinement and coarsening to concentrate the work to the most important regions, many three-dimensional problems and

complex physical systems cannot be simulated on today's serial computers. Therefore, deal.II comes with support for parallelism for both shared memory computations as present on modern laptop or workstation computers, and distributed computing to make use of big computing clusters.

Shared memory parallelization in deal.II builds on the framework from Intel's Threading Building Blocks (TBB) [108, 74]. TBB is a C++ library that is centered around a dynamic task scheduler. Parallel work (like a loop over all vector entries in an operation of the form $z = \alpha x + y$ for vectors x, y, z and a scalar α) is subdivided into tasks which are then distributed to available worker threads. Task-based parallelism differs from concepts like Pthreads or OpenMP [97] that are centered around the available hardware threads. Another task-based library is SMP Superscalar [13], which tries to automatically exploit parallelism in C/C++ code. The library looks for data dependencies between tasks and schedules the work in a way that avoids write access to the same data by two or more threads simultaneously.

Shared-memory parallelization is employed for deal.II's internal linear algebra routines. In addition, interfaces for parallel assembly are provided. In this way, the two most computationally intensive parts of the library and user code are done in parallel [10]. One advantage of shared memory parallelism is that it can be applied incrementally, starting from the most time-consuming functions. However, this incremental approach has the consequence that parts which are not important to the developers (or too difficult to be easily implemented) do not profit immediately from the increase in computing power through parallelism. In deal.II, this applies, e.g., to enumeration of degrees of freedom, grid refinement, and coarsening as well as resolution of constraints.

The second level of parallelism in deal.II addresses distributed storage. In this concept, the grid, solution matrices, and vectors are partitioned among the processors (domain decomposition), as opposed to shared memory parallelism where each participating process has access to all data but only performs a subset of the operations. In addition to the locally owned partition of the global fields, each processor needs access to some data from neighbors, so-called ghost data. During the solution of linear systems or when changing the mesh, ghost data must be exchanged between the processors. This is done by explicit send and receive commands, which are organized by the Message Passing Interface (MPI) [91]. When using this approach to scale computations to massively parallel supercomputers with tens of thousands processor cores and meshes consisting of hundreds of millions or billions of cells, it is essential that all data structures describing the global solution are fully distributed. This is because only storing a few bytes per cell could saturate all memory available to an individual process. Paper IV presents a mantle convection solver using distributed parallelization, and Paper V presents the computational infrastructure to enable such large-scale computations.

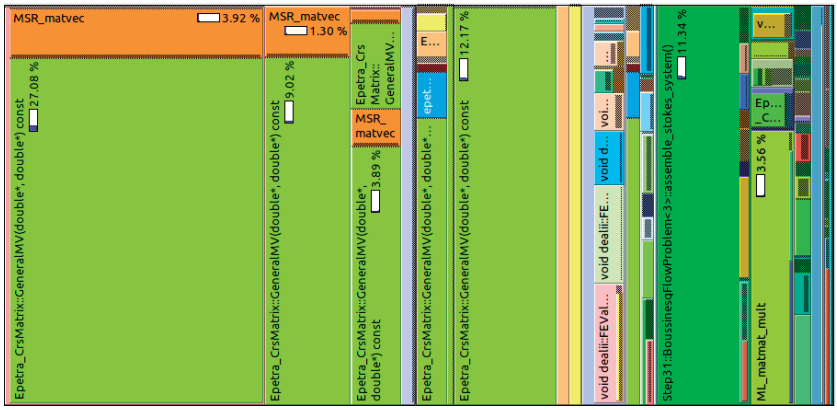


Figure 5.1. Illustration of CPU instructions spent for a 3D mantle convection solver. The light green and orange boxes in the left of the panel are sparse matrix-vector products, spending more than 60% of total instructions, within a GMRES solver preconditioned by a block-triangular scheme according to Sec. 4.5.1, using AMG on the velocity-velocity block. Several boxes of the same color indicate the same function called in different contexts (e.g., matrix-vector products in solver and on different levels in the AMG smoother). Picture created with the tool KCachegrind.

5.3 Efficiency on modern computer architectures

The ultimate task in finite element programming is to make simulations as fast as possible. The largest amount of time is usually spent in linear solvers and assembly routines, often 70–95% of total run-time. Efficient linear solvers like the ones presented in Sec. 4.5 often have a complex internal structure, as they combine fast preconditioners (e.g., using algebraic multilevel methods) on different components, orthonormalization, and the like. Nonetheless, most of the run time is spent on simple computational kernels like sparse matrix-vector products, inner products and sparse matrix substitutions. Fig. 5.1 shows a graph displaying the number of CPU instructions for running a three-dimensional mantle convection solver presented in Paper IV. More than 60% of the instructions (and about 80% of the run time) are spent on sparse matrix-vector products. Matrix-vector products are used by the Krylov subspace solver on the Stokes system as well as for algebraic multigrid on the velocity matrix (smoother, restriction, prolongation). In comparison, only about 10% of time is spent on assembly of matrices. In applications where operators are nonlinear, this proportion is usually somewhat larger.

Despite their fundamental importance for finite element analysis, sparse matrix-vector products show a relatively poor performance on modern computers because they are memory bandwidth limited rather than limited by computations [101]. This means that the processor cannot fully utilize arithmetic units because data access to the matrix and vector entries is not fast enough. For faster computations, data access needs to be organized in such a way that

data once loaded into computation units of the processor can be reused, see, e.g., the work in [135]. Paper VI discusses a strategy to improve the performance of matrix-vector products. The idea is not to store the global matrices explicitly but to evaluate the matrix-vector product element by element using finite element shape function information and transformations from real to unit cell. Such an approach can be faster than traditional matrix-vector products because considerably less data transfer is necessary. In addition, it also obviates matrix assembly. In flow solvers, we observed a speedup by a factor two to three for $\mathcal{Q}_2^d\mathcal{Q}_1$ elements when replacing sparse matrices by the method from Paper VI.

6. Summary of Papers

6.1 Paper I

This paper provides an in-depth numerical study of inaccuracies in the discrete evaluation of surface tension forces for the conservative level set model [95]. A circular bubble at rest is considered as a model problem. The force due to surface tension should ideally be balanced by a jump in pressure. The study is performed by applying a complete Navier–Stokes/two-phase flow solver to the problem and recording the magnitude of spurious velocities.

For an interface force that is constructed from a gradient of a color function in the same finite element space as pressure, we demonstrate that the velocity errors are caused by inaccuracies in the numerically calculated curvature. If the exact curvature is prescribed, there are no imbalances in the scheme. However, when the element order of the function representing the color gradient differs from the one for pressure, an additional imbalance is introduced. In other words, if the surface tension force is constructed from a gradient similar to the pressure term, exact force balance can be achieved.

We also study the effect of the continuous representation of surface tension on the prediction of the jump in pressure between the inside and the outside of the bubble. For regularized delta functions, our experiments show that the variation of the curvature between the inside and outside of the regularized length $\varepsilon \sim h$ has the largest impact on pressure accuracy, considerably more than the accuracy of the curvature. In addition, the paper studies the accuracy of sharp surface tension forces based on the evaluation of a line integral. The results illustrate that continuous pressure space cannot represent the sharp force appropriately, leading to a classical Gibbs phenomenon with oscillations in pressure that do not diminish as the mesh size is reduced. Also, there are large spurious velocities for this approach. These results indicate that discontinuities in the discrete pressure space need to be allowed.

The tests have been performed with two different flow solvers in order to evaluate the propagation of the force balance between different components of the solver. The results for a fully coupled implicit method and a decoupled (projection) solver are similar, which shows that the choice of the solver does not affect the balance of forces. Also, residual-based stabilization for enabling the use of equal order elements on velocity and pressure according to the discussion in Sec. 4.2 does not destroy the exact force balance.

6.2 Paper II

Paper II presents a multiscale approach for the modeling of contact line dynamics. Many of the classical cheap numerical methods for two-phase flow, like interface tracking schemes or the level set method, have no built-in mechanism for the representation of wetting flow driven by contact lines. Previously, ad-hoc approaches like slip conditions have been used to embed this information. The motivation for this work has been to derive a physically based formulation for the computation of contact line dynamics.

Our approach combines the capabilities of a detailed phase field simulation for representing contact line dynamics and a conventional solver for the bulk of the domain. On the macro scale, our approach moves the contact line by a slip velocity that depends on the apparent contact angle of the interface at the solid walls. In order to find numerical values for this relation, the physics around the contact point is represented by a micro model based on the phase field method. The dimensions of the micro domain are chosen to comprise the region of diffusive transport around the interface. In global phase field approaches, the correct physical value for these length scales cannot be set because of resolution restrictions, and numerically dictated values are very common. In our approach, the correct physical value for the diffusion length is easily embedded into the micro domain. In order to restrict the computational domain for the micro simulations, a similarity velocity field based on the creeping flow model by Huh and Scriven [70] is chosen as a far field. The creeping flow equations are based on the apparent contact angle. Close to the contact point, the physical static contact angle is specified as a boundary condition for the phase field method. The slip velocity is then defined as the velocity that is required to obtain a steady state between the apparent contact angle in the far field and the static contact angle at the contact line. This nonlinear relation is solved by a secant method, where a function evaluation corresponds to the solution of the phase field method in time with a given slip velocity. Since the apparent contact angle is a one-dimensional input and the slip velocity a one-dimensional output, the results from the micro solver can be tabulated prior to running the macro solver.

Numerical experiments for a macro solver based on a level set implementation with the ghost fluid method for evaluating surface tension demonstrate that the multiscale method produces converging results for two test cases of contact-line driven flow, namely, a capillary rise and an advancing droplet. We compare with results of global phase field simulations and theoretical predictions and obtain good agreement.

6.3 Paper III

Paper III presents a hybrid method that combines a conservative level set method and a phase field method. The method aims at using the phase field

model to represent wetting but to use the cheaper level set method away from the contact region. The method makes use of the similar formulation of these two methods, namely the same shape of the regularized color function illustrated in Fig. 2.3. The combination of the two models is realized by a switch function that is equal to one close to contact lines and becomes zero in regions far away. This way, the additional terms in the phase field model are disabled away from contact lines and simple advection of the color function according to (2.8) is used. Also, surface tension forces are evaluated through the usual relation with curvature and gradient of the color function away from the contact line.

The benefits of the combination are demonstrated by two benchmark tests of the conservative level set method and the phase field method in situations without contact lines, namely on a rising bubble [73] and a bubble oscillating by surface tension. It is shown that the conservative level set method is more efficient for a given mesh size in absence of contact dynamics. For the phase field method, the interface thickness and the mobility are chosen in order to give the best results.

Additionally, we provide an a priori energy estimate to show that the switch function is implemented in a stable way. We provide numerical tests on channel flow which show that the hybrid method gives good results.

6.4 Paper IV

Paper IV presents an efficient solver for mantle convection problems in geodynamics. As opposed to techniques currently used in the geodynamics community, like Citcom [139] or Conman [83], the solver uses modern numerical software as building blocks. Generic operations like definition of finite element shape functions and adaptive mesh refinement are handled through the finite element library deal.II [9, 10], and numerical linear algebra is provided by distributed matrices and vectors from Trilinos [65]. This combination of building blocks both reduces the complexity of the solver program and allows to more easily incorporate future developments in the software packages.

The method applies a coherent set of numerical techniques for efficiently solving the problem on large scale parallel computers. The algorithms for parallel adaptive mesh refinement follow the concept presented in Paper V. The solver is based on higher order $\mathscr{Q}_2$ elements for the temperature and the velocity, and linear elements for the pressure. This promises better accuracy than linear elements for all solution variables that are the current state-of-the-art. In order to stabilize the convection-dominated temperature equation, an advanced stabilization approach based on a residual of the temperature entropy [56] is used. This method is more stable on sharp gradients and also often less diffusive than SUPG approaches that add diffusion almost constantly throughout the domain. In order to solve the large systems of linear equations, efficient

(parallel) numerical linear algebra is discussed. For the Stokes equations, the Schur complement preconditioner presented in Sec. 4.5.1 is applied and its efficiency demonstrated.

The implementation is verified on two benchmarks, one for 2D and one for 3D. Additionally, parallel scalability tests are performed that show the good scaling of all the components in the implementation.

6.5 Paper V

This paper presents a software framework for enabling finite element simulations on computing clusters consisting of tens of thousands processor cores. The algorithms are presented in a modular way and interact with a distributed mesh storage scheme, in this case the software p4est [28], through a small set of queries. These queries are related to ownership of cells and ensure that each processor can construct a representation of the part of the mesh that is locally relevant, including the locally owned cells plus a layer of ghost cells. The local representation is created by refinement of the respective cells from a common coarse mesh that is present on all processors. The paper discusses the whole chain of finite element programming, including matrix assembly, representation of boundary conditions, solution of linear systems and postprocessing like the evaluation of error indicators, adaptive mesh refinement and transfer of solution components when the owning processor changes during parallel adaptive refinement.

The main innovation of the paper is the introduction of scalable algorithms to generic finite element programming, as opposed to specialized parallel implementation techniques that have previously been used. This is important because parallel capabilities in numerical software like deal.II will enable a broader community to make use of large computing clusters with relatively simple means. Previous approaches to "parallel" generic finite element computations used a computational mesh that was replicated on each processor, and only matrices and vectors had been distributed. Particular emphasis is put on practical problems like efficient indexing when working with distributed data sets where locally owned degrees of freedom and ghost indices are to be represented. Moreover, the computation of constraints for enforcing continuity at hanging nodes and representing Dirichlet boundary conditions is discussed in detail. The algorithm for computing constraints operates only on the locally stored part of the triangulation (locally owned cells and ghost cells), without communication with other processors. The paper presents scaling results on up to 16,384 processors that illustrate the effectiveness of the implementation.

6.6 Paper VI

Paper VI is concerned with the kernel that usually consumes most computing time in finite element codes, namely matrix-vector products. In the paper, an implementation for the efficient evaluation of finite element operators is presented without using a sparse matrix. Sparse matrix-vector products have unfavorable properties on modern computer systems, as their performance is usually given by available memory bandwidth instead of arithmetic performance of the processors. This is particularly true for modern computer clusters where several cores share access to memory, which limits the parallel scalability of linear solvers in finite element analysis.

This paper presents an alternative implementation that avoids storage of all the matrix elements and evaluates the finite element operator cell-wise. This approach has several advantages: Firstly, its memory requirements are significantly lower than for sparse matrices, which offsets the increase in operation count for low order methods. Secondly, it allows for more efficient use of the particular form of finite element shape functions, which reduces the computational effort per degree of freedom from $\mathcal{O}(p^d)$ in sparse matrices to $\mathcal{O}(d^2 p)$, where p is the polynomial order and d the spatial dimension, in case of hexahedral elements. This so-called sum-factorization approach can be interpreted as an evaluation of shape functions in one dimension at a time. Thirdly, the treatment of systems of PDEs is cheaper than for sparse matrices since the coupling between components is only introduced at the level of quadrature points, which is of lower complexity than the coupling introduced in sparse matrices. Also, nonlinear operators can be evaluated as efficiently as linear ones, which facilitates the evaluation of residuals, e.g., in the incompressible Navier–Stokes equations with implicit treatment of convection.

As an effect, the presented method outperforms sparse matrix-vector products for scalar problems by about a factor two for a scalar Laplace operator and by a factor five for the system matrix representing the linearized Navier–Stokes equations with $\mathcal{Q}_2$ finite elements in three space dimensions. In addition, it avoids expensive assembly of sparse matrices in nonlinear solvers or time-dependent problems. This improves overall performance of the Navier–Stokes solver by more than a factor two on the problems considered in this thesis. The performance gain for higher order elements is demonstrated to be even larger.

7. Future Work

In this thesis, efficient adaptive finite element techniques have been presented for solving problems where a scalar transport equation is coupled to incompressible flow. A significant part of the flow is influenced by the scalar field, like two-phase flow driven by surface tension and contact lines and mantle convection driven by buoyancy.

The results in Paper I highlight the importance of accurate evaluation of surface tension forces. One direction of future research is the implementation of sharp interface treatments for finite element discretization in the spirit of XFEM. Such an approach promises more accurate simulations on coarser meshes compared to a diffuse force representation that is sandwiched between the geometrical features of the bubbles and the mesh size. This is important especially for 3D simulations where resolution is more limited. Sharp interface treatment also allows for higher accuracy of numerical schemes compared to regularized interface forces. Another direction of future research is to improve the quality of curvature approximation. One way to achieve this is to use finer meshes around the interface only for the level set variable, and coarsen the mesh more aggressively in the far field. Such an approach of different meshes for the level set and flow variables has been employed for phase field simulations [137]. However, there are implementation hurdles to be taken when using domain decomposition to parallelize the computations, since an optimal partitioning for the flow variables might create a considerable imbalance on the level set degrees of freedom. In addition, a comparison of the benchmark results in Paper III with conventional level set methods based on signed distance approaches in [73] shows that the conservative level set method requires considerably more resolution to represent the interface. It will be necessary to analyze mass conservation techniques like the ones presented in [82, 64] for signed distance functions and compare with the conservative level set method in terms of overall computational efficiency.

The techniques from Papers II and III that combine efficient level set approaches for the bulk of the domain, enriched by phase field information close to contact lines, are potential tools for larger simulations of contact-line driven flows. Using the parallel framework used for mantle convection presented in Paper IV and V, the necessary degree of resolution can be provided to tackle these problems. These tools are of interest for the simulation of multi-phase flow in porous media: A first step in that direction would be to apply a solver that includes an efficient wetting mechanism to more complicated geometries, like channels with curved walls or the flow around obstacles. The flow field induced by an imposed pressure gradient on a small-scale material configuration

can be used to determine values for its permeability. An effective permeability tensor is the main material parameter in coarse-scale simulations based on Darcy's law and is thus useful for simulation of flow in oil/gas reservoirs or groundwater basins.

The multiscale approach presented in Paper II needs further analysis for the case without a clear separation in scales of contact line reactions and global fluid velocity. This situation is relevant when wetting is not the only driving force in the flow, but when also external forces are present. In that case, it might be necessary to exchange more information between the micro model and the macro model, possibly leading to a situation where the micro model needs to be evaluated during the macro simulation.

Also on the software side, further progress needs to be made. Firstly, the massive parallel framework presented in Paper V is limited by the requirement that each processor needs to hold the common coarse mesh through the p4est implementation, i.e., the mesh from which each processor refines different partitions to its local representation. When mesh generation is done outside of the deal.II library, no hierarchy of cells is available and a mesh consisting of millions of cells is necessary. In this situation, the largest possible problem size is still limited. Furthermore, the parallelization strategy needs to be evaluated for h, p adaptivity and anisotropic mesh refinement.

An implementation using h, p adaptivity, i.e., adaptivity in both the mesh size and the polynomial order of the elements, might be a tool to make the representation of multi-phase flow more efficient. It is conceivable to use fine meshes and low order elements close to the interface and coarser meshes with higher element orders away from the interface where the solutions are smooth. The framework in Paper VI will be very efficient for the regions where higher order elements are employed, as opposed to sparse matrices.

The approach from Paper VI for fast matrix-vector products has been shown to provide considerable speedups for matrix-vector products on second order elements and higher. However, the impact on the whole solution chain of finite element methods has not been fully analyzed. In particular, high-quality preconditioners for difficult problems, like the Stokes velocity operator with strongly varying viscosity or the Navier–Stokes matrix with variable densities, usually require explicit knowledge of matrix entries like aggregation-based algebraic multigrid. One solution already mentioned in Sec. 4.5.1 is to combine the matrix-free implementation of matrix-vector products on higher order finite elements with AMG based on matrices from linear finite elements with improved sparsity. However, this approach needs careful analysis, as initial tests revealed that it is efficient only for problems where the operator is dominated by a Laplacian like the Stokes system. System matrices as arising from the time-dependent Navier–Stokes operator in Sec. 4.5.2 show not as good a performance compared to ILU preconditioners on the high-order elements. However, these preconditioners are not trivially parallelized, so it will be necessary to find sufficiently simple yet efficient preconditioners for

systems where the mass matrix part is large but is not large enough to be approximated by diagonal Jacobi-type preconditioners.

Another direction of future research of the methods in Paper VI is to replace compiled code by code generators like the FEniCS form compiler [88]. Such a tool allows to more easily specialize the generated code to the computer hardware in use, in particular using optimizations as done for dense matrix-matrix multiplies. This is particularly the case for modern and future computer systems with graphical processing units, which require re-implementation of our techniques. Also, it is necessary to keep pace with development of algebraic multigrid solvers for modern computer architectures. Within the Trilinos project, the future package MueLu is targeted to provide algebraic multilevel preconditioners on the emerging computer architectures, and ShyLU is targeting algebraic factorizations.

8. Sammanfattning på svenska

Denna avhandling diskuterar numeriska approximationer för inkompressibelt flöde kopplat till skalära transportekvationer. De två tillämpningar som betraktas är flerfasflöde modellerat med levelsetmetoden och konvektion i planetmantel baserad på Boussinesq-approximationen.

Tvåfasflöde

Första delen av avhandlingen presenterar modeller för flödet av minst två inkompressibla, ej blandbara fluider. Denna typ av flöde är av betydelse i många områden inom teknik och medicin och numerisk simulering är ett grundläggande verktyg för att analysera och styra dessa processer. Ett exempel är intravenösa behandlingar där läkemedel i form av oljeliknande substanser injiceras i blodflödet. I dessa sammanhang är ytspänningen mellan de olika vätskorna vanligtvis stark, vilket gör att bubblor sträver efter att bli sfäriska. Liknande numeriska tekniker kan också användas för att undersöka formen av röda blodceller och för mikrofluidiska system, så kallade lab-on-a-chip-processer. Bläckstråleskrivare och sintring är tekniska tillämpningar där flödet av flera fluider måste kontrolleras för att göra processerna så effektiva som möjligt. Förståelsen av tvåfasflödet är också grundläggande för att uppnå god effektivitet i oljeutvinning.

I den numeriska modellen som betraktas i denna avhandling representeras ytan mellan olika fluider med hjälp av en nivåkurva hos en indikatorfunktion. Ytspänningskraften ingår i rörelsemängdekvationen och är proportionellt mot krökningen av ytan och riktad i normalriktningen av ytan. Denna information kan beräknas från indikatorfunktionen. Ett bidrag av denna avhandling är en systematisk numerisk evaluering av diskretiseringsfelen som görs när ytspänningskraften implementeras med finitaelementmetoden. För evalueringen betraktas en cirkulär bubbla som är i ett stationärt tillstånd och där alla hastigheter som mäts i en numerisk beräkning beror på approximationsfel. Resultaten visar att krafterna är i exakt balans om ett analytiskt värde av krökningen används och ytspänningen representeras genom en gradient i samma funktionsrum som används för approximationen av trycket i Navier-Stokes-ekvationerna. För praktiska beräkningar betyder det att den enda felkällan är approximationsfelet i krökningen. Dessutom visas att finitaelementapproximationer som föreskriver kontinuerliga funktioner inte kan representera en kraft som endast verkar på ytan utan måste approximeras med en utspridd volymkraft.

Ett annat bidrag av avhandlingen är två modeller för effektiv simulering av flöde där ytan mellan fluiderna är i kontakt med väggar, ett så kallad kontaktlinjeproblem. Den traditionella modellen som föreskriver att fluider inte kan röra sig vid väggar är otillräcklig i detta avseende. Den första metoden är en multiskalansats. En mikrosimulering extraherar information från det fysikaliska beteendet i ett begränsat område omkring kontaktpunkten. Denna simulering görs med en phasefieldmetod vilket innehåller en mekanism för att förflytta kontaktpunkterna. Informationen överförs sedan till en makromodell i form av ett slip-randvillkor, ett randvillkor som tillåter avgränsningsytan att röra sig. Denna sliphastighet beror på den makroskopiska kontaktvinkeln som uppmäts i varje tidssteg i makrosimuleringen.

Den andra föreslagna metoden för kontaktlinjeproblem använder olika modeller i olika delar av beräkningsområdet. I närheten av kontaktlinjer är vanliga levelsetmetoder otillräckliga för att representera fysiken, så här används den mer allmäna phasefieldmetoden. I delar som ligger längre bort från kontaktområdet vore en phasefieldmetod betydlig dyrare och där räcker det med att använda en levelsetmetod. Lösningen som fås med hjälp av en speciell indikatorfunktion och lösningen till phasefieldmetoden har en liknande form. Därför kan levelsetmetoden tolkas som en phasefieldmetod där några termer sätts till noll. En konvergensstudie på två testfall har genomförts för att kvantifiera besparingsmöjligheter med hybridmetoden jämfört med en vanlig phasefieldimplementering. Dessutom visas stabilitet av kombinationen med hälp av en a-priori energiuppskattning.

Mantelkonvektion och mjukvara för finitaelementkoder

Den intressanta längdskalan för konvektion i jordens mantel är helt annorlunda och mycket större jämfört med de små längderna i tvåfasflöde. Den matematiska modellen som används i detta sammanhang, Stokes-ekvationerna kopplade till en skalär ekvation för transporten av temperaturen, har dock många likheter till den som används för tvåfasflöde. Förutom liknande struktur i modellen kräver båda problem hög upplösning för att representera fysiken. Detta är särskilt viktigt för tredimensionella beräkningar.

En del av avhandlingen beskriver en parallell implementering av mantelkonvektionsmodellen för moderna beräkningskluster med tiotusentals processorer, vilket möjliggör simuleringar med hundratals miljoner till några miljarder frihetsgrader. Koden använder sig av parallell adaptiv nätförfining och noggranna tids- och rumsdiskretiseringar. För att kunna simulera den advektionsdominerade temperaturekvationen som karakteriserar mantelkonvektionen använder vi en avancerad stabiliseringsmetod baserad på en artificiell viskositet som beror på residualen av entropin i temperaturekvationen. Implementeringen använder sig av byggstenar för generisk finitaelementprogrammering och effektiva iterativa linjära lösare samt förkonditionerare. Denna modularitet gör det möjligt att på ett relativt enkelt sätt kunna anpassa ko-

den till framtida utvecklingar på mjukvarusidan, till exempel utvecklingar för nya sorters parallelldatorer.

Ett allmänt ramverk som möjliggör parallella adaptiva finitaelementberäkningar för godtyckliga partiella differentialekvationer presenteras. Inga delar som beskriver det globala problemet (som nätstrukturen eller information om alla frihetsgrader) behöver hållas i sin helhet på en beräkningskärna. Tvärtom delas all information upp mellan processorna, och lämpliga funktioner för utbytet av denna information tillhandahålls. Detta berör förutom nätstrukturen också vektorerna och matriserna som beskriver de linjära systemen, såsom felindikatorer som beskriver vilka celler som ska förfinas eller förgrovas.

För att förbättra prestandan av högre ordningens finita elementmetoder har en ny matrisfri implementering föreslagits. Denna metod utför elementbaserad kvadratur istället för att assemblera en global gles matris. För att utföra de lokala kvadraturberäkningar på ett så effektivt sätt som möjligt utnyttjas strukturen i basfunktionerna. Implementeringen är betydligt snabbare på att utföra matris-vektorprodukter än glesa matriser, dagens standardmetod, för finita element med kvadratiska och ännu högre ordningens basfunktioner. Dessutom är detta koncept mycket generellt och kan tillämpas till både linjära och ickelinjära problem såsom evalueringen av residualer i Navier-Stokes-ekvationerna. Eftersom denna metod inte använder sig av matriser kan ytterligare besparingar uppnås eftersom man undviker assembleringen som annars måste utföras i varje tidsteg för ickelinjära ekvationer. Det finns dock fortfarande behov av assemblering av matriser i viss mån för att konstruera effektiva förkonditionerare som algebraiska multinivåförkonditionerare, men dessa behöver inte uppdateras i varje tidssteg.

Acknowledgments

I would like to thank Gunilla Kreiss for introducing me to the exciting subject of multiphase flow and guiding me through my studies. The discussions with you have encouraged me to work on new ideas and try out various approaches. I have enjoyed an increasing degree of freedom in my work which has helped me to develop my own interests.

I am grateful to Wolfgang Bangerth for acting like a second adviser from afar, developing my knowledge on finite element programming. I have appreciated the inspiring environment when I visited you at Texas A&M University.

Many thanks to Claudio Walker, Bernhard Müller, Katharina Kormann, Timo Heister, Sara Zahedi, Carsten Burstedde, Bärbel Janssen, and Volker Gravemeier for our joint work and for giving me the opportunity to visit you. Also, the Navier–Stokes group and the Parallel Scientific Computing Colloquium are acknowledged for being a platform for discussion and ideas.

I thank Petia Boyanova, Kenneth Duru, Katharina Kormann, and Karl Ljungkvist for proofreading this introductory summary and giving many valuable comments.

My work was supported by the Graduate School in Mathematics and Computing (FMB).

References

[1] M. Abkarian, M. Faivre, and A. Viallat. Swinging of red blood cells under shear flow. *Phys. Rev. Lett.*, 98:188302, 2007.

[2] D. Adalsteinsson and J. A. Sethian. A fast level set method for propagating interfaces. *J. Comput. Phys.*, 118(2):269–277, 1995.

[3] M. Adams, M. Brezina, J. Hu, and R. Tuminaro. Parallel multigrid smoothing: polynomial versus Gauss–Seidel. *J. Comput. Phys.*, 188:593–610, 2003.

[4] M. Ainsworth and J. T. Oden. *A Posteriori Error Estimation in Finite Element Analysis*. John Wiley and Sons, 2000.

[5] S. Aland and A. Voigt. Benchmark computations of diffuse interface models for two-dimensional bubble dynamics. *Int. J. Numer. Meth. Fluids*, to appear, 2011.

[6] D. M. Anderson, G. B. McFadden, and A. A. Wheeler. Diffuse-interface methods in fluid mechanics. *Annu. Rev. Fluid Mech.*, 30:139–165, 1998.

[7] I. Babuška and T. Strouboulis. *The Finite Element Method and its Reliability*. Clarendon Press, New York, 2001.

[8] S. Balay, K. Buschelman, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. Curfman McInnes, B. F. Smith, and H. Zhang. PETSc users manual. Technical Report ANL-95/11 - Revision 3.2, Argonne National Laboratory, 2011.

[9] W. Bangerth, R. Hartmann, and G. Kanschat. deal.II — a general purpose object oriented finite element library. *ACM Trans. Math. Softw.*, 33(4):article no. 24, 2007.

[10] W. Bangerth and G. Kanschat. *deal.II Differential Equations Analysis Library, Technical Reference*. http://www.dealii.org.

[11] W. Bangerth and O. Kayser-Herold. Data structures and requirements for *hp* finite element software. *ACM Trans. Math. Softw.*, 36(1):4/1–4/31, 2009.

[12] W. Bangerth and R. Rannacher. *Adaptive Finite Element Methods for Differential Equations*. Birkhäuser Verlag, 2003.

[13] Barcelona Supercomputing Center. *SMP Superscalar (SMPSs) User's Manual, Version 2.4*, September 2011. http://www.bsc.es/media/4783.pdf.

[14] Y. Bazilevs, V. M. Calo, J. A. Cottrell, T. J. R. Hughes, A. Reali, and G. Scovazzi. Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows. *Comput. Meth. Appl. Mech. Engrg.*, 197:173–201, 2007.

[15] Y. Bazilevs, V. M. Calo, T. E. Tezduyar, and T. J. R. Hughes. YZβ discontinuity capturing for advection-dominated processes with application to arterial drug delivery. *Int. J. Numer. Meth. Fluids*, 54:593–608, 2007.

[16] M. Benzi, G. H. Golub, and J. Liesen. Numerical solution of saddle point problems. *Acta Numerica*, 14:1–137, 2005.

[17] M. Benzi and M. A. Olshanskii. An augmented Lagrangian-based approach to the Oseen problem. *SIAM J. Sci. Comput.*, 28(6):2095–2113, 2006.

[18] B. Blankenbach, F. Busse, U. Christensen, L. Cserepes, D. Gunkel, U. Hansen, H. Harder, G. Jarvis, M. Koch, G. Marquart, D. Moore, P. Olson, H. Schmeling, and T. Schnaubelt. A benchmark comparison for mantle convection codes. *Geophys. J. Int.*, 98:23–38, 1989.

[19] P. Boyanova, M. Do-Quang, and M. Neytcheva. Efficient preconditioners for large scale binary Cahn–Hilliard models. *Comput. Meth. Appl. Math.*, to appear, 2011.

[20] F. Boyer. A theoretical and numerical model for the study of incompressible mixture flows. *Comput. & Fluids*, 31:41–68, 2002.

[21] M. Braack and E. Burman. Local projection stabilization for the Oseen problem and its interpretation as a variational multiscale method. *SIAM J. Numer. Anal.*, 43(6):2544–2566, 2006.

[22] M. Braack, E. Burman, V. John, and G. Lube. Stabilized finite element methods for the generalized Oseen problem. *Comput. Meth. Appl. Mech. Engrg.*, 196:853–866, 2007.

[23] J. U. Brackbill, D. B. Kothe, and C. Zemach. A continuum method for modeling surface tension. *J. Comput. Phys.*, 100:335–354, 1992.

[24] A. N. Brooks and T. J. R. Hughes. Streamline upwind Petrov–Galerkin formulation for convection dominated flows with particular emphasis on the incompressible Navier–Stokes equations. *Comput. Meth. Appl. Mech. Engrg.*, 32:199–259, 1982.

[25] H.-P. Bunge, M. A. Richards, and J. R. Baumgardner. A sensitivity study of three-dimensional spherical mantle convection at 10^8 Rayleigh number: Effects of depth-dependent viscosity, heating mode, and an endothermic phase change. *J. Geophys. Res.*, 102(B6):11991–12008, 1997.

[26] E. Burman, M. A. Fernández, and P. Hansbo. Continuous interior penalty method for the Oseen's equations. *SIAM J. Numer. Anal.*, 44(3):1248–1274, 2006.

[27] C. Burstedde, O. Ghattas, M. Gurnis, E. Tan, T. Tu, G. Stadler, L. C. Wilcox, and S. Zhong. Scalable adaptive mantle convection simulation on petascale supercomputers. In *SC '08: Proceedings of the International Conference for High Performance Computing, Networking, Storage, and Analysis*. ACM/IEEE, 2008.

[28] C. Burstedde, L. C. Wilcox, and O. Ghattas. p4est: Scalable algorithms for parallel adaptive mesh refinement on forests of octrees. *SIAM J. Sci. Comput.*, 33(3):1103–1133, 2011.

[29] J. W. Cahn and J. E. Hilliard. Free energy of a nonuniform system. i. interfacial free energy. *J. Chem. Phys.*, 28(2):258–267, 1958.

[30] A. J. Chorin. Numerical solution of the Navier–Stokes equations. *Math. Comput.*, 22:745–762, 1968.

[31] J. Donea and A. Huerta. *Finite Element Methods for Flow Problems*. J. Wiley & Sons, Chichester, 2003.

[32] G. Dzuik. An algorithm for evolutionary surfaces. *Numer. Math.*, 58(6):603–611, 1991.

[33] D. A. Edwards, H. Brenner, and D. T. Wasan. *Interfacial Transport Processes and Rheology*. Butterworth–Heinemann, Stoneham, 1991.

[34] H. Elman, D. Silvester, and A. Wathen. *Finite Elements and Fast Iterative*

Solvers with Applications in Incompressible Fluid Dynamics. Oxford Science Publications, Oxford, 2005.

[35] H. Emmerich. *The Diffuse Interface Approach in Materials Science: Thermodynamic Concepts and Applications of Phase-Field Models*. Springer, Berlin, 2003.

[36] B. Engquist, A.-K. Tornberg, and R. Tsai. Discretization of Dirac delta functions in level set methods. *J. Comput. Phys.*, 207:28–51, 2005.

[37] C. R. Ethier and D. A. Steinman. Exact fully 3d Navier Stokes solution for benchmarking. *Int. J. Numer. Meth. Fluids*, 19:369–375, 1994.

[38] R. P. Fedkiw, T. Aslam, B. Merriman, and S. Osher. A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the Ghost Fluid Method). *J. Comput. Phys.*, 152(2):457–492, 1999.

[39] J. P. de S. R. 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. *Int. J. Num. Meth. Engrg.*, 19:1621–1656, 1983.

[40] C. Galusinski and P. Vigneaux. On stability condition for bifluid flows with surface tension: Application to microfluidics. *J. Comput. Phys.*, 227:6140–6164, 2008.

[41] M. Garzon, L. J. Gray, and J. A. Sethian. Numerical simulation of non-viscous liquid pinch-off using a coupled level set-boundary integral method. *J. Comput. Phys.*, 228:6079–6106, 2009.

[42] M. W. Gee, J. J. Hu, and R. S. Tuminaro. A new smoothed aggregation multigrid method for anisotropic problems. *Numer. Linear Algebra Appl.*, 16:19–37, 2009.

[43] M. W. Gee, C. M. Siefert, J. J. Hu, R. S. Tuminaro, and M. G. Sala. ML 5.0 Smoothed Aggregation User's Guide. Technical Report 2006-2649, Sandia National Laboratories, 2006.

[44] T. Geenen, M. ur Rehman, S. P. MacLachlan, G. Segal, C. Vuik, A. P. van den Berg, and W. Spakman. Scalable robust solvers for unstructured FE geodynamic modeling applications: Solving the Stokes equation for models with large localized viscosity contrasts. *Geochem. Geophys. Geosyst.*, 10(9):Q09002, 2009.

[45] T. Gerya. *Introduction to Numerical Geodynamic Modelling*. Cambridge University Press, 2010.

[46] V. Girault and P.-A. Raviart. *Finite Element Methods for the Navier–Stokes Equations*. Springer Verlag, New York, 1986.

[47] J. Glimm, J. Grove, B. Lendquist, O. A. McBryan, and G. Tryggvason. The bifurcation of tracked scalar waves. *SIAM J. Sci. Stat. Comput.*, 9(1):61–79, 1988.

[48] R. Glowinski. Finite element methods for incompressible viscous flow. In J. L. Lions and P. G. Ciarlet, editors, *Numerical Methods for Fluids (3), Handbook of Numerical Analysis*, volume 9, pages 3–1176. North-Holland, Amsterdam, 2003.

[49] V. Gravemeier, M. W. Gee, M. Kronbichler, and W. A. Wall. An algebraic variational multiscale–multigrid method for large eddy simulation of turbulent flow. *Comput. Meth. Appl. Mech. Engrg.*, 199(13–16):853–864, 2010.

[50] V. Gravemeier, M. Kronbichler, M. W. Gee, and W. A. Wall. An algebraic variational multiscale–multigrid method for large-eddy simulation: generalized-

α time integration, Fourier analysis and application to turbulent flow past a square-section cylinder. *Comput. Mech.*, 47:217–233, 2011.

[51] P. M. Gresho and R. L. Sani. *Incompressible Flow and the Finite Element Method*, volume one: Advection–Diffusion. John Wiley & Sons, Chichester, 2000.

[52] P. M. Gresho and R. L. Sani. *Incompressible Flow and the Finite Element Method*, volume two: Isothermal Laminar Flow. John Wiley & Sons, Chichester, 2000.

[53] P. P. Grinevich and M. A. Olshanskii. An iterative method for the Stokes-type problem with variable viscosity. *SIAM J. Sci. Comput.*, 31(5):3959–3978, 2009.

[54] S. Groß and A. Reusken. An extended pressure finite element space for two-phase incompressible flows with surface tension. *J. Comput. Phys.*, 224:40–58, 2007.

[55] J.-L. Guermond, P. Minev, and J. Shen. An overview of projection methods for incompressible flows. *Comput. Meth. Appl. Mech. Engrg.*, 195:6011–6045, 2006.

[56] J.-L. Guermond, R. Pasquetti, and B. Popov. Entropy viscosity method for nonlinear conservation laws. *J. Comput. Phys.*, 230:4248–4267, 2011.

[57] J.-L. Guermond and L. Quartapelle. A projection FEM for variable density incompressible flows. *J. Comput. Phys.*, 165(1):167–188, 2000.

[58] J.-L. Guermond and A. Salgado. A splitting method for incompressible flows with variable density based on a pressure Poisson equation. *J. Comput. Phys.*, 228:2834–2846, 2009.

[59] M. D. Gunzburger. *Finite Element Methods for Viscous Incompressible Flows*. Academic Press, Boston, 1989.

[60] E. Hairer, S. P. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations I. Nonstiff Problems*. Springer Verlag, Berlin, 2nd edition, 1993.

[61] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II. Stiff and Differential-Algebraic Problems*. Springer Verlag, Berlin, 1991.

[62] P. Hansbo and A. Szepessy. A velocity–pressure streamline diffusion FEM. *Comput. Meth. Appl. Mech. Engrg.*, 84(2):175–192, 1990.

[63] F. H. Harlow and J. E. Welch. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Phys. Fluids*, 8(12):2182–2189, 1965.

[64] D. Hartmann, M. Meinke, and W. Schröder. The constrained reinitialization equation for level set methods. *J. Comput. Phys.*, 229:1514–1535, 2010.

[65] M. A. Heroux, R. A. Bartlett, V. E. Howle, R. J. Hoekstra, J. J. Hu, T. G. Kolda, R. B. Lehoucq, K. R. Long, R. P. Pawlowski, E. T. Phipps, A. G. Salinger, H. K. Thornquist, R. S. Tuminaro, J. M. Willenbring, Williams A., and K. S. Stanley. An overview of the Trilinos project. *ACM Trans. Math. Softw.*, 31(3):397–423, 2005.

[66] E. M. Herzig, K. A. White, A. B. Schofield, W. C. K. Poon, and P. S. Clegg. Bicontinuous emulsions stabilized solely by colloidal particles. *Nature Materials*, 6:966–971, 2007.

[67] C. W. Hirt and B. D. Nichols. Volume of fluid (VOF) method for the dynamics of free boundaries. *J. Comput. Phys.*, 39(1):201–225, 1981.

[68] T. Y. Hou, J. S. Lowengrub, and M. J. Shelley. Boundary integral methods for

multicomponent fluids and multiphase materials. *J. Comput. Phys.*, 169:302–362, 2001.

[69] T. J. R. Hughes, L. P. Franca, and M. Balestra. A new finite element formulation for computational fluid dynamics: V. Circumventing the Babuska–Brezzi condition: A stable Petrov–Galerkin formulation of the Stokes problem accomodating equal-order interpolations. *Comput. Meth. Appl. Mech. Engrg.*, 59:85–99, 1986.

[70] C. Huh and L. E. Scriven. Hydrodynamic model of steady movement of a solid/liquid/fluid contact line. *J. Colloid Interf. Sci.*, 35(1):85–101, 1971.

[71] S. Hysing. A new implicit surface tension implementation for interfacial flows. *Int. J. Numer. Meth. Fluids*, 51:659–672, 2006.

[72] S. Hysing and S. Turek. The Eikonal equation: Numerical efficiency vs. algorithmic complexity on quadrilateral grids. In *Proceedings of Algorithmy*, pages 22–31, 2005.

[73] S. Hysing, S. Turek, D. Kuzmin, N. Parolini, E. Burman, S. Ganesan, and L. Tobiska. Quantitative benchmark computations of two-dimensional bubble dynamics. *Int. J. Numer. Meth. Fluids*, 60:1259–1288, 2009.

[74] Intel Corporation. *Intel Threading Building Blocks*, 2011. http://www.threadingbuildingblocks.org.

[75] A. Ismail-Zadeh and P. Tackley. *Computational Methods for Geodynamics*. Cambridge University Press, 2010.

[76] D. Jacqmin. Calculation of two-phase Navier–Stokes flows using phase-field modeling. *J. Comput. Phys.*, 155(1):96–127, 1999.

[77] B. Janssen and G. Kanschat. Adaptive multilevel methods with local smoothing for H^1- and H^{curl}-conforming high order finite element methods. *SIAM J. Sci. Comput.*, 33(4):2095–2114, 2011.

[78] V. John, G. Matthies, and J. Rang. A comparison of time-discretization/linearization approaches for the incompressible Navier–Stokes equations. *Comput. Meth. Appl. Mech. Engrg.*, 195(44-47):5995–6010, 2006.

[79] H. Johnston and J.-G. Liu. Accurate, stable and efficient Navier–Stokes solvers based on explicit treatment of the pressure term. *J. Comput. Phys.*, 199(1):221–259, 2004.

[80] V. Kantsler, E. Segre, and V. Steinberg. Vesicle dynamics in time-dependent elongation flow: Wrinkling instability. *Phys. Rev. Lett.*, 99:178102, 2007.

[81] D. A. Kay, D. Loghin, and A. Wathen. A preconditioner for the steady-state Navier–Stokes equations. *SIAM J. Sci. Comput.*, 24(1):237–256, 2002.

[82] C. E. Kees, I. Akkerman, M. W. Farthing, and Y. Bazilevs. A conservative level set method suitable for variable-order approximations and unstructured meshes. *J. Comput. Phys.*, 230:4536–4558, 2011.

[83] S. D. King, A. Raefsky, and B. H. Hager. Conman: Vectorizing a finite element code for incompressible two-dimensional convection in the Earth's mantle. *Phys. Earth Planet. Inter.*, 59:195–207, 1990.

[84] B. J. Kirby. *Micro- and Nanoscale Fluid Mechanics: Transport in Microfluidic Devices*. Cambridge University Press, New York, 2010.

[85] L. Lee and R. LeVeque. An immersed interface method for incompressible Navier–Stokes equations. *SIAM J. Sci. Comput.*, 25(3):832–856, 2003.

[86] R. J. LeVeque. *Finite Volume Methods for Hyberbolic Problems*. Cambridge

Texts in Applied Mathematics, Cambridge, 2002.

[87] Z. Li and K. Ito. *The Immersed Interface Method: Numerical Solution of PDEs Involving Interfaces and Irregular Domains*. SIAM, Philadelphia, 2006.

[88] A. Logg. Automating the finite element method. *Arch. Comput. Meth. Eng.*, 14(2):93–138, 2007.

[89] A. Logg and G. N. Wells. DOLFIN: Automated finite element computing. *ACM Trans. Math. Softw.*, 37(2):1–28, 2010.

[90] K.-A. Mardal and R. Winther. Uniform preconditioners for the time dependent Stokes problem. *Numer. Math.*, 38:305–327, 2004.

[91] Message Passing Interface Forum. MPI: A message-passing interface standard (version 2.2). Technical report, http://www.mpi-forum.org, 2009.

[92] M. Nazarov. Convergence of a residual based artificial viscosity finite element method. Technical Report KTH-CTL-4016, Royal Institute of Technology, Stockholm, 2011.

[93] W. Noh and P. Woodwards. SLIC (simple line interface calculation). In *Proceedings of the 5th International Conference on Numerical Methods in Fluid Dynamics*, pages 330–340, Enschede, 1976. Springer Verlag.

[94] E. Olsson and G. Kreiss. A conservative level set method for two phase flow. *J. Comput. Phys.*, 210:225–246, 2005.

[95] E. Olsson, G. Kreiss, and S. Zahedi. A conservative level set method for two phase flow II. *J. Comput. Phys.*, 225:785–807, 2007.

[96] E. Oosterbroek and A. van den Berg, editors. *Lab-on-a-Chip: Miniaturized Systems for (Bio)Chemical Analysis and Synthesis*. Elsevier Science, Amsterdam, second edition, 2003.

[97] OpenMP Architecture Review Board. *OpenMP Application Program Interface, Version 3.1*, July 2011. http://www.openmp.org/mp-documents/OpenMP3.1.pdf.

[98] S. A. Orszag. Spectral methods for problems in complex geometries. *J. Comput. Phys.*, 37:70–92, 1980.

[99] S. Osher and R. P. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Applied Mathematical Sciences. Springer, New York, 2003.

[100] S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton–Jacobi formulations. *J. Comput. Phys.*, 79(1):12–49, 1988.

[101] D. A. Patterson and J. L. Hennessy. *Computer Organization and Design*. Morgan Kaufmann, Burlington, 4th edition, 2009.

[102] C. S. Peskin. Numerical analysis of blood flow in the heart. *J. Comput. Phys.*, 25(3):220–252, 1977.

[103] C. S. Peskin. The immersed boundary method. *Acta Numerica*, 11:479–517, 2002.

[104] S. B. Pope. *Turbulent Flows*. Cambridge University Press, Cambridge, 2000.

[105] L. Quartapelle. *Numerical solution of the incompressible Navier–Stokes equations*, volume 113 of *International Series of Numerical Mathematics*. Birkhäuser-Verlag, Basel, 1993.

[106] A. Quarteroni and A. Valli. *Numerical Approximation of Partial Differential Equations*. Springer Series in Computational Mathematics. Springer, Berlin, 1994.

[107] U. Rasthofer, F. Henke, W. A. Wall, and V. Gravemeier. An extended residual-based variational multiscale method for two-phase flow including surface tension. *Comput. Meth. Appl. Mech. Engrg.*, 200:1866–1876, 2011.

[108] J. Reinders. *Intel Threading Building Blocks*. O'Reilly Media, Sebastopol, CA, 2007.

[109] W. Ren and W. E. Boundary conditions for the moving contact line problem. *Phys. Fluids*, 19:022101, 2007.

[110] G. Russo and P. Smereka. A remark on computing distance functions. *J. Comput. Phys.*, 163:51–67, 2000.

[111] Y. Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, Philadelphia, second edition, 2003.

[112] R. Scardovelli and S. Zaleski. Direct numerical simulation of free-surface and interfacial flow. *Ann. Rev. Fluid Mech.*, 31:567–603, 1999.

[113] G. Schubert, D. L. Turcotte, and P. Olson. *Mantle Convection in the Earth and Planets, Part 1*. Cambridge University Press, Cambridge, 2001.

[114] J. A. Sethian. *Level Set Methods and Fast Marching Methods. Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*. Cambridge University Press, 2000.

[115] S. Shin and D. Juric. Modeling three-dimensional multiphase flow using a level contour reconstruction method for front tracking without connectivity. *J. Comput. Phys.*, 180:427–470, 2002.

[116] S. Shin, I. Yoon, and D. Juric. The local front reconstruction method for direct simulation of two- and three-dimensional multiphase flow. *J. Comput. Phys.*, 230:6605–6646, 2011.

[117] D. Silvester and A. Wathen. Fast iterative solution of stabilised Stokes systems. Part II: Using general block preconditioners. *SIAM J. Numer. Anal.*, 31:1352–1367, 1994.

[118] I. Singer-Loginova and H. M. Singer. The phase field technique for modeling multiphase materials. *Rep. Prog. Phys.*, 71:106501, 2008.

[119] P. D. M. Spelt. A level-set approach for simulations of flows with multiple moving contact lines with hysteresis. *J. Comput. Phys.*, 207:389–404, 2005.

[120] M. Sussman. A method for overcoming the surface tension time step constraint in multiphase flow II. *Int. J. Numer. Meth. Fluids*, to appear, 2011.

[121] M. Sussman and E. Fatemi. An efficient, interface-preserving level set redistancing algorithm and its application to interfacial incompressible fluid flow. *SIAM J. Sci. Comput.*, 20(4):1165–1191, 1999.

[122] M. Sussman and M. Ohta. A stable and efficient method for treating surface tension in incompressible two-phase flow. *SIAM J. Sci. Comput.*, 31(4):2447–2471, 2009.

[123] M. Sussman and E. G. Pucket. A coupled level set and volume-of-fluid method for computing 3d and axisymmetric incompressible two-phase flows. *J. Comput. Phys.*, 162:301–337, 2000.

[124] M. Sussman, P. Smereka, and S. Osher. A level set method for computing solutions to incompressible two-phase flow. *J. Comput. Phys.*, 114(1):146–159, 1994.

[125] P. J. Tackley. Effects of strongly variable viscosity on three-dimensional compressible convection in planetary mantles. *J. Geophys. Res.*, 101(B2):3311–

3332, 1996.

[126] R. Temam. Une méthode d'approximation de la solution des équations de Navier-Stokes. *Bull. Soc. Math. France*, 96:115–152, 1968.

[127] T. E. Tezduyar. Finite element methods for fluid dynamics with moving boundaries and interfaces. In E. Stein, R. De Borst, and T. J. R. Hughes, editors, *Encyclopedia of Computational Mechanics, Vol. 3: Fluids*, chapter 17. Wiley, New York, 2004.

[128] T. E. Tezduyar. Finite elements in fluids: Stabilized formulation and moving boundaries and interfaces. *Comput. Fluids*, 36:191–206, 2007.

[129] J. Thies. *Scalable algorithms for fully implicit ocean models*. PhD thesis, University of Groningen, 2011.

[130] U. Trottenberg, C. Oosterlee, and A. Schüller. *Multigrid*. Elsevier Academic Press, London, 2001.

[131] G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, S. Nas, and Y.-J. Jan. A front tracking method for the computations of multiphase flow. *J. Comput. Phys.*, 169(2):708–759, 2001.

[132] R. Tuminaro and C. Tong. Parallel smoothed aggregation multigrid: aggregation strategies on massively parallel machines. In J. Donnelley, editor, *Super Computing 2000 Proceedings*, 2000.

[133] S. Turek. *Efficient Solvers for Incompressible Flow Problems: An Algorithmic and Computational Approach*. Springer, Berlin, 1999.

[134] W. Villanueva and G. Amberg. Some generic capillary-driven flows. *Int. J. Multiphase Flow*, 32:1072–1086, 2006.

[135] D. Wallin, H. Löf, E. Hagersten, and S. Holmgren. Reconsidering algorithms for iterative solvers in the multicore era. *Int. J. Comput. Sci. Eng.*, 4(4):270–282, 2009.

[136] P. Yue, C. Zhou, and J. J. Feng. Sharp-interface limit of the Cahn–Hilliard model for moving contact lines. *J. Fluid Mech.*, 645:279–294, 2010.

[137] P. Yue, C. Zhou, J. J. Feng, C. F. Ollivier-Gooch, and H. H. Hu. Phase-field simulations of interfacial dynamics in viscoelastic fluids using finite elements with adaptive meshing. *J. Comput. Phys.*, 219:47–67, 2006.

[138] S. Zahedi and A.-K. Tornberg. Delta function approximations in level set methods by distance function extension. *J. Comput. Phys.*, 229:2199–2219, 2010.

[139] S. Zhong, A. McNamara, E. Tan, L. Moresi, and M. Gurnis. A benchmark study on mantle convection in a 3-D spherical shell using CitcomS. *Geochem. Geophys. Geosyst.*, 9:Q10017, 2008.

Acta Universitatis Upsaliensis

Digital Comprehensive Summaries of Uppsala Dissertations from the Faculty of Science and Technology 885

Editor: The Dean of the Faculty of Science and Technology

A doctoral dissertation from the Faculty of Science and Technology, Uppsala University, is usually a summary of a number of papers. A few copies of the complete dissertation are kept at major Swedish research libraries, while the summary alone is distributed internationally through the series Digital Comprehensive Summaries of Uppsala Dissertations from the Faculty of Science and Technology.

Distribution: publications.uu.se
urn:nbn:se:uu:diva-162215

ACTA
UNIVERSITATIS
UPSALIENSIS
UPPSALA
2011