ON DISCRETIZATION AND ITERATIVE TECHNIQUES FOR SECOND–ORDER PROBLEMS WITH APPLICATIONS TO MULTIPHASE FLOW IN POROUS MEDIA

A Dissertation
by
Apostol Todorov Vassilev

Submitted to the Office of the Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

May 1996

Major Subject: Mathematics

Abstract

There are three related topics which are considered in this dissertation: the discretization of second-order partial differential equations (PDEs), the development of new iterative techniques for solving the resulting systems of equations efficiently, and applications of this theory to the important problem of modeling multiphase fluid flow in porous media. New fully discrete finite element schemes of backward Euler type which utilize local refinement in time and space are constructed and analyzed. It is shown that these schemes are unconditionally stable and an error analysis in maximum norms is provided. New inexact nonoverlapping domain decomposition preconditioners, applied to the solution of problems arising from Galerkin, mixed, and locally refined finite element discretizations of second-order PDEs, are constructed and analyzed. The preconditioners are developed based only on the assumption that the interior solves are provided by uniform preconditioning forms. They exhibit the same asymptotic condition number growth as corresponding exact preconditioners but are much more efficient computationally. In addition, their preconditioning effect is independent of jumps of the operator coefficients across subdomain boundaries. An abstract analysis of inexact variants of the classical Uzawa iterative algorithm for solving saddle-point problems is developed. Both linear and nonlinear inexact algorithms are analyzed and special considerations for second-order PDEs are provided. Applications of the developed new discretizations and iterative algorithms to problems of fluid flow in porous media are considered. Emphasis is given to the two-phase fractional flow model in the context of a class of environmental applications. Illustrative numerical examples involving the new techniques developed as well as a computer simulation of groundwater flow and contaminant transport are included.

Dedication

To my parents Todor and Tanya,
wife Mariana,
and daughters Victoria and Teodora.

Acknowledgments

I am grateful to my advisor Richard Ewing for his support and guidance through the years of my graduate school. Dick has always been generous to me by providing excellent opportunities for research. Under his supervision I enjoyed a research assistantship for the entire period of my studies at the University of Wyoming and at Texas A&M where both of us moved in 1992. Because of him I was involved in some of the most exciting projects in numerical analysis and scientific computing which was a great source of knowledge and experience. I wish also to thank Dick for encouraging and helping me always when I needed it.

During the last two years I was privileged to work with James Bramble. I am sincerely grateful to him for sharing with me many ideas. The opportunity to do research with a mathematician of his class was an exceptional inspiration for me. I wish also to thank him for all the care and consideration he provided for me.

I am deeply indebted to Raytcho Lazarov for first inspiring me to study numerical analysis. When I won a scholarship at the Bulgarian Academy of Sciences in 1990, I was fortunate to begin my studies under Raytcho's supervision. His generous attitude, care, and devotion to his students helped me enormously in this period and played a key role in the continuation of my education in the United States. During the years at the University of Wyoming and at Texas A&M he was a person I could always turn to in a difficult or a happy moment. Raytcho's attention to detail and his valuable suggestions helped improve my dissertation considerably.

I am grateful to Joseph Pasciak for his support and guidance through the years of my graduate studies. I collaborated closely with him since the time we first met in Wyoming in the fall of 1991. The four wonderful summers I spent at the Brookhaven National Laboratory in New York thanks to all of his support are perhaps the most productive periods of my graduate student life. The knowledge and experience I gained during this collaboration I value as second to none. I wish to thank him for all the motivation and encouragement he provided for me just when I needed them most. He has always been my model for an extremely professional, totally dedicated and brilliant scientist. Even though Joe did not officially serve on my graduate committee, I view and respect him as my mentor.

I would like to thank Bart Childs for serving on my graduate committee and being extremely helpful to me in this regard.

I wish to thank Michael Mann, my friend and colleague from the Institute for Scientific Computation, for preparing all of the LaTeX figures in the text and many other typesetting tricks which he generously shared with me in the process of writing this thesis. I am also thankful for the nice times we had together on many occasions.

My final personal acknowledgment goes to my parents, my wife and children for everything they did for me in order to make this possible. I am deeply indebted to them for all of their love, sacrifice, help, and encouragement.

My graduate research was supported in part by the US Department of Energy under grant # DE–FG05–92ER25143 and by the National Science Foundation under grant # INT–89–14472. I am thankful for the valuable funding provided through these grants. However, no part of the research described in this dissertation was a subject to an official review by any of the organizations mentioned above and therefore does not necessarily reflect their views and no official endorsement should be inferred.

In my dissertation I have used portions of my joint papers with Ewing, Lazarov, Bramble, and Pasciak (cf. [23, 51, 52]). I would like to thank the Society for Industrial and Applied Mathematics and Elsevier Science for granting me permission to use the results published in these papers. Copies of the

permissions are provided in Appendices A and B.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Numerical approximation of partial differential equations is perhaps one of the most dynamically developing branches of numerical analysis. This area of mathematical research is the place where many different scientific disciplines meet in an attempt either to solve existing difficult problems or to set new challenges. Contemporary numerical analysis combines in a coherent way knowledge from mathematics, physics, chemistry, biology, and computer science in order to tackle problems of practical interest.

The typical path in solving interesting problems in numerical analysis starts with the formulation of a boundary (and initial) value problem. Most often these equations represent mathematical models of physical phenomena. The physical background is very important in understanding the behavior of the possible solutions.

The next step is a reformulation of the differential problem in a weak (variational) form. This leads to seeking generalized solutions in Hilbert (or Banach) functional spaces. The study of the weak forms results in estimates for the solution which reveal important information about the smoothness and other properties.

Based on this, a numerical approximation technique is selected. The key elements in making the decision are the stability and the error analyses. The Galerkin finite element methods are classical discretization methods. Other existing numerical techniques such as the finite difference and collocation methods will not be considered here. During the last two decades, the mixed finite element method has become very popular due to the advantages it offers in solving problems from elasticity and fluid flow. Other specialized techniques have emerged as well. Examples of these are locally-refined discretizations which lead to very accurate and efficient approximations. The physical background of the differential equations should also be taken into account in selecting the discretization method because some of them have features that are particularly attractive for a given physical application. Examples are the mass conservative properties of the mixed method which are very useful in fluid flow applications. For transient problems, a time discretization is applied. Most often this is performed using finite difference quotients to approximate the time derivatives. This approach is then combined with the existing Galerkin discretization in space. Discretizations that use local time stepping as an extension of the spatially refined discretizations are relatively new and quite important. Throughout this dissertation we consider standard Galerkin and mixed finite element discretizations of second-order elliptic partial differential equations. Combinations of these techniques with backward Euler time stepping are the methods considered for parabolic problems. Locally-refined backward Euler–Galerkin approximations are developed as well.

Once the discretization method has been chosen, the next problem to address is the solution of the corresponding discrete system of equations. It should be emphasized from the very beginning that there are critical differences between small and large systems of equations. The small systems can be solved with any of the known linear algebra methods. In the case of large systems, however, the amount of work needed in the standard direct solution methods, such as the classical Gaussian elimination, increases dramatically and leads to a fast deterioration in their performance. Finite element discretizations eventually lead to very large systems of linear equations. In the era of powerful computers, scientists are trying to solve much larger problems than those considered as impossible even half a century ago. Today, systems with hundreds of thousands (even millions) of equations are often encountered, and direct methods are

simply not practical when applied to them. The class of iterative methods provides a needed alternative. The history of these methods dates back more than 170 years. The first iterative methods are attributed to Gauss, Jacobi and Seidel. Even though these methods are very efficient computationally, their dominance over the direct algorithms for large-scale problems was established less than three decades ago. It was the astonishing progress in the computing technology that made this possible. After almost a century of stagnation, the theory of iterative methods also took the path of fast development. The modern period begins with the breakthrough of David Young [99] in 1950 which resulted in an accelerated variant of the Gauss–Seidel iteration, known today as the SOR method.

The contemporary approach to the development of new and more efficient iterative methods relies on a deep understanding of the mathematical nature of the system of equations to which the method is applied. In our setting these properties are determined by the properties of the differential equation and the discretization technique used for the derivation of the discrete equations. We shall consider two basic types of systems of equations, namely symmetric and positive definite (SPD) and symmetric and indefinite. The latter are also called saddle-point systems. In the case of SPD systems, the main emphasis is given to the development of effective preconditioners. The class of domain decomposition preconditioners has established itself as a favorable one. There are three factors that have contributed most to the popularity of these preconditioners. First, they result in very good conditioning. Second, these methods provide an efficient way of overcoming certain difficulties in designing robust iterative procedures coming from particular features of the differential equation or the physical background that often are an unsurmountable obstacle for most of the other preconditioning approaches available in the literature. Such examples are jumps in the operator coefficients, which can be handled efficiently within the domain decomposition paradigm but lead to deterioration of the effectiveness of the bulk of other preconditioning techniques. Third, these algorithms take full advantage of existing modern computer technology.

The saddle-point systems are much more difficult mathematical problems than SPD systems of equations. They arise, for example, when standard mixed finite element approximations to second-order differential equations are considered. Correspondingly, less efficient iterative methods for their solution are available. Naturally, the main emphasis in this case is given to the development of fast iterative algorithms. Among many other methods for saddle-point systems proposed in the literature, the classical Uzawa algorithm stands out for its simplicity, versatility, and effectiveness. Further improvement of the efficiency of this algorithm is achieved via the inexact Uzawa algorithm. However, the properties of the latter method are not well understood, and significant deficiencies exist in the theory available for the inexact method.

Progress in numerical analysis and scientific computing is due to a great extent to the demand for accurate and efficient techniques for obtaining numerical approximations to differential models in physics and the other natural sciences. Thus, the ultimate goal of most of the research devoted to developing numerical methods is to solve important practical problems. Mathematical modeling is another large area of active research. For the purposes of our considerations, we shall restrict ourselves to the problems of modeling multiphase fluid flow in porous media. Most of the mathematical models in this field have been developed because of the need for more sophisticated technologies for oil recovery in the petroleum industry. Recently, the interest in tackling the related class of environmental problems has contributed to the further improvement of these models. Typically, when mathematical modeling of physical phenomena is attempted, the attention is focused mainly on capturing the underlying physics as much as possible. It turns out, however, that the same physics can be modeled by quite different equations, some of which are better suited for numerical approximation than others. Thus, best results are obtained when the development of the mathematical model takes into account this fact.

This dissertation aims at making progress in four different but closely related areas in numerical analysis, mathematical modeling and scientific computing. More precisely, we shall develop and analyze a new discretization method of backward Euler–Galerkin type for parabolic problems which utilizes local refinement in time and space. We shall also construct and analyze new and very efficient nonoverlapping domain decomposition preconditioners with inexact subdomain solves. A new theory that provides the needed insights for the inexact Uzawa algorithms will be developed. Finally, we shall demonstrate that these new methods can be applied successfully in modeling two-phase fluid flow in porous media, when appropriate mathematical models are chosen.

The dissertation is organized as follows. Chapter 2 is devoted to finite element discretizations of second-order elliptic and parabolic problems. First, fundamental facts from the theory of Sobolev spaces are introduced. Next, the standard finite element discretizations of Galerkin and mixed types are defined together with their backward Euler variants. Important results from the theory of these discretizations, which form the basis for our considerations in the remaining part of the thesis, are included. The new results in this chapter are about locally-refined discretizations of backward Euler–Galerkin types of parabolic equations. In the literature there is a good understanding of how to construct locally-refined spatial discretizations. However, the effects of the implicit local time stepping on the stability and accuracy of the schemes are less well understood. The main goal of the analysis in Section 2.4 is to provide a rigorous theory for schemes with local time stepping when linear interpolation in time is applied next to the interfaces between the refined and unrefined regions. The results of Theorems 2.7, 2.8, and 2.9 establish unconditional stability and error estimates for fully implicit discretizations of backward Euler–Galerkin type with refinement in time and space. Interesting numerical experiments involving schemes with local time stepping are provided as well.

Chapter 3 contains the theory of two iterative techniques for second-order problems. In Section 3.2 we construct and analyze new nonoverlapping domain decomposition preconditioners with inexact subdomain solves for elliptic problems. The results of Theorems 3.1 and 3.3 provide estimates of the asymptotic condition number growth of these algorithms. The new algorithms exhibit the same asymptotic behavior as the corresponding algorithms with exact solves but are much more efficient computationally. They are also robust with respect to jumps of the operator coefficients across the subdomain boundaries. In addition, these preconditioners are quite versatile. We consider applications to parabolic problems, mixed methods and locally-refined discretizations and show that the new algorithms are guaranteed to perform equally well in these settings. In Section 3.3.1 we provide a new analysis of the inexact Uzawa algorithms for solving saddle-point problems. We consider two types of inexact methods: linear and nonlinear iterations. Theorem 3.4 and Corollary 3.1 establish a general result for convergence of the linear inexact algorithm under minimal assumptions. The main result for the nonlinear algorithm is a sufficient condition for convergence given in Theorem 3.5. Our approach to the analysis of these algorithms is very general and applies to a variety of concrete examples. We have considered applications to mixed discretizations of second order problems as well as applications to the Stokes equation.

In Chapter 4 we consider the modeling of two-phase fluid flow in porous media. This chapter contains a discussion of the underlying physical principles for developing flow models. We provide a hierarchy of different models and a discussion of their mathematical properties. We also demonstrate that the new algorithms developed in Chapters 2 and 3 can be applied successfully to the solution of complex flow models. The emphasis is given to the Richards equation and the fractional flow model. Important aspects of these models such as wells and boundary conditions are discussed. An iterative technique for imposing boundary conditions on the two-phase fractional flow model is proposed. The development of a sophisticated flow simulator is discussed and interesting results from a computer simulation are included.

Finally, in Chapter 5 conclusions and a discussion of interesting possibilities for future research are provided.

Chapter 2

Discretization of second-order problems

In this chapter we consider finite element approximations to second-order linear partial differential equations. In general, the finite element method is based on a few simple but very powerful ideas. First, one partitions the domain Ω, where a given differential problem is posed, into a set of subdomains, called elements. Typically, the elements are triangles (tetrahedra), quadrilaterals, etc. Second, based on such partitioning of Ω, a finite dimensional space of functions is defined so that when the differential problem is reformulated in this space it is "easy" to solve. In addition, one requires that the approximation obtained be close to the solution of the continuous problem in an appropriate sense. Several finite element approximations to second-order problems are defined here. Along with standard methods, we also develop a new discretization technique for parabolic problems and provide a corresponding error analysis. This chapter contains fundamental theoretical results concerning the finite element approximations considered. They will be the basis for our analysis in the subsequent chapters.

The chapter is organized as follows. We begin with definitions and basic facts from the theory of Sobolev spaces. In Section 2.2 we introduce a model elliptic problem and discuss standard discretizations. In particular, we define Galerkin and mixed finite element methods for elliptic problems and provide classical results concerning error estimates and properties of the discrete operators corresponding to the above discretizations. Subsequently, in Section 2.3 we define fully discrete schemes of backward Euler type for parabolic problems and outline important facts about the analysis of such schemes. Finally, in Section 2.4 a new discretization technique for parabolic problems using composite grids with refinement in time and space is developed and an error analysis is provided. Results from illustrative numerical experiments with locally refined discretizations are presented as well.

2.1 Sobolev spaces

In this section we provide definitions and basic properties of Sobolev spaces of real valued functions over bounded, simply connected domains $\Omega \subset \mathbb{R}^n$ with Lipshitz continuous boundary $\partial\Omega$ (cf. [37]). Here Ω denotes an open set in the n–dimensional Euclidean space $\mathbb{R}^n$, $n = 1, 2, 3$. Throughout this thesis we shall restrict our attention only to such functions and domains even though most of the theorems stated below hold in much greater generality. We refer to Adams [1] for the proofs of all results included in this section concerning Sobolev spaces.

Let $u(x)$ be a real valued function on Ω. We define a generalized derivative of u of order $|\alpha|$ by the function $D^\alpha u$ (provided that it exists) that satisfies

$$\int_\Omega D^\alpha u(x)\psi(x)\,dx = (-1)^{|\alpha|}\int_\Omega u(x)D^\alpha\psi\,dx,$$

for all infinitely continuously differentiable functions ψ with compact support in Ω. Here $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{N}^n$ is a multi-index and $|\alpha| = \sum_{i=1}^n \alpha_i$. We remark that the derivative in the right-hand side of the

above definition is understood in the classical sense.

Let m be a nonnegative integer and p be an extended real number satisfying $1 \le p \le \infty$. We define a functional $\|\cdot\|_{m,p}$ by

$$\|u\|_{m,p} = \left\{ \sum_{0 \le |\alpha| \le m} \|D^\alpha u\|_p^p \right\}^{1/p}, \quad \text{for all} \quad 1 \le p < \infty, \tag{2.1a}$$

$$\|u\|_{m,\infty} = \max_{0 \le |\alpha| \le m} \|D^\alpha u\|_\infty, \tag{2.1b}$$

for all functions u for which the right-hand side makes sense. Here $\|\cdot\|_p$ is the $L^p(\Omega)$–norm given by

$$\|u\|_p = \left\{ \int_\Omega |u|^p \, dx \right\}^{1/p}, \quad \text{for all} \quad u \in L^p(\Omega).$$

It is clear that $\|\cdot\|_{m,p}$ generates a norm over any vector space of functions for which the right-hand side of (2.1a) or (2.1b) takes finite values. Correspondingly, for $1 \le k \le m$, the functional

$$|u|_{k,p} = \left\{ \sum_{|\alpha| = k} \|D^\alpha u\|_p^p \right\}^{1/p}, \quad \text{for all} \quad 1 \le p < \infty, \tag{2.2}$$

defines a k-th semi-norm (obviously, $|\cdot|_{0,p} \equiv \|\cdot\|_p$).

We define three such spaces by

$$H^{m,p}(\Omega) \equiv \text{the completion of } \{u \in C^m(\Omega) \mid \|u\|_{m,p} < \infty\} \text{ with respect to the norm } \|\cdot\|_{m,p}, \tag{2.3a}$$

$$W^{m,p}(\Omega) \equiv \{u \in L^p(\Omega) \mid D^\alpha u \in L^p(\Omega) \text{ , for } 0 \le |\alpha| \le m\}, \tag{2.3b}$$

and

$$W_0^{m,p}(\Omega) \equiv \text{the closure of } C_0^\infty(\Omega) \text{ in the space } W^{m,p}(\Omega). \tag{2.3c}$$

Definition 2.1 *The spaces defined in* (2.3)*, equipped with the norms* (2.1)*, are called Sobolev spaces over the domain* Ω*.*

Theorem 2.1 (Adams [1]) *$W^{m,p}(\Omega)$ is a Banach space.*

Theorem 2.2 (Meyers and Serrin [75]) *If $1 \le p < \infty$, then*

$$W^{m,p}(\Omega) = H^{m,p}(\Omega).$$

The spaces $W^{s,p}(\Omega)$ where s is a positive real number can be defined by real interpolation of Banach spaces (cf. [70, 11]).

The dual space $(W^{r,p}(\Omega))'$ can be characterized, for any $p \in (0, \infty)$ and any $r \in (0, \infty)$, as the completion of $L^{p'}(\Omega)$ with respect to the norm

$$\|u\|_{-r,p'} = \sup_{\substack{v \in W^{r,p}(\Omega) \\ v \ne 0}} \frac{|(u,v)|}{\|v\|_{r,p}}, \tag{2.4}$$

where $1/p' + 1/p = 1$ and $(\cdot,\cdot)$ is defined by

$$(u,v) = \int_\Omega u(x)v(x) \, dx.$$

Let, for any integer $m \geq 0$ and any real number $\alpha \in (0,1]$, $C^{m,\alpha}(\bar{\Omega})$ be the space of all functions in $C^m(\bar{\Omega})$ whose m-th derivatives satisfy a Hölder condition with exponent α. We note that $C^{m,\alpha}(\bar{\Omega})$ is a Banach space when equipped with the norm

$$\|u\|_{C^{m,\alpha}(\bar{\Omega})} = \|u\|_{m,\infty,\bar{\Omega}} + \max_{|\beta|=m} \sup_{\substack{x,y\in\bar{\Omega} \\ x\neq y}} \frac{|D^\beta u(x) - D^\beta u(y)|}{\|x-y\|^\alpha},$$

where $\|\cdot\|$ is the Euclidean norm in $\mathbb{R}^n$.

A fundamental result (cf. [1, 37]) that characterizes the family of spaces defined in (2.3) is given in the next theorem.

Definition 2.2 *A normed linear space $X(\Omega)$ is imbedded continuously in another normed linear space $Y(\Omega)$ (denoted $X(\Omega) \hookrightarrow Y(\Omega)$) if the space X is contained in Y with continuous injection. In other words, there exists a positive constant C which depends only on Ω through the dimension n and the properties of $\partial\Omega$ such that*

$$\|u\|_Y \leq C\|u\|_X, \quad \textit{for all} \quad u \in X(\Omega).$$

Theorem 2.3 (The Sobolev imbedding theorem) *Let $\Omega \subset \mathbb{R}^n$ be a bounded, simply connected domain with Lipshitz continuous boundary $\partial\Omega$. Then the following imbeddings hold for all nonnegative integers m and all extended real numbers p such that $1 \leq p \leq \infty$:*

(2.5a) $$W^{m,p}(\Omega) \hookrightarrow L^{p^*}(\Omega), \quad \textit{with } \frac{1}{p^*} = \frac{1}{p} - \frac{m}{n}, \qquad \textit{if } \ m < \frac{n}{p};$$

(2.5b) $$W^{m,p}(\Omega) \hookrightarrow L^q(\Omega), \quad \textit{for all } q \in [1,\infty), \qquad \textit{if } \ m = \frac{n}{p};$$

(2.5c) $$W^{m,p}(\Omega) \hookrightarrow C^{0,m-(n/p)}(\bar{\Omega}), \qquad \textit{if } \ \frac{n}{p} < m < \frac{n}{p} + 1;$$

(2.5d) $$W^{m,p}(\Omega) \hookrightarrow C^{0,\alpha}(\bar{\Omega}), \quad \textit{for all } 0 < \alpha < 1, \qquad \textit{if } \ m = \frac{n}{p} + 1;$$

(2.5e) $$W^{m,p}(\Omega) \hookrightarrow C^{0,1}(\bar{\Omega}), \qquad \textit{if } \ \frac{n}{p} + 1 < m.$$

2.2 Elliptic problems

In this section we formulate a model elliptic problem and introduce two classical techniques for its discretization.

2.2.1 A model elliptic problem. Weak formulation

We consider the Dirichlet problem

(2.6a) $$\mathcal{L}u = f \qquad \text{in} \quad \Omega,$$

(2.6b) $$u = 0 \qquad \text{on} \quad \partial\Omega,$$

where f is a given function, $\Omega \subset \mathbb{R}^n$ $(n = 1, 2, 3)$ is a bounded polyhedral domain with Lipshitz boundary, and

(2.7) $$\mathcal{L}v = -\sum_{i,j=1}^{n} \frac{\partial}{\partial x_i}\left(a_{ij}\frac{\partial v}{\partial x_j}\right).$$

Here the $n \times n$ coefficient matrix $\{a_{ij}\}$ is symmetric, uniformly positive definite, and bounded above on Ω. This is a classical model problem for a second-order uniformly elliptic equation.

In this section and throughout the entire thesis, we shall use the notation $H^m(\Omega)$ and $H_0^m(\Omega)$ to denote the special cases of $W^{m,2}(\Omega)$ and $W_0^{m,2}(\Omega)$, respectively (cf. Theorem 2.2). Similarly, $\|\cdot\|_k$ and

$|\cdot|_k$, $0 \le k \le m$, will be used in place of $\|\cdot\|_{k,2}$ and $|\cdot|_{k,2}$, defined by (2.1a) and (2.2). $H^m(\Omega)$ is a Hilbert space for any positive integer m with inner product given by

$$(u,v)_m = \sum_{0\le|\alpha|\le m} (D^\alpha u, D^\alpha v), \quad \text{for all} \quad u,\ v\ \in H^m(\Omega).$$

It is convenient to adopt the notation $\partial_1 = D^{(1,0,\dots,0)}$, $\dots$, $\partial_n = D^{(0,0,\dots,1)}$ for the generalized first derivatives of u.

Let us define the generalized Dirichlet form on Ω by

$$\mathcal{A}(v,w) = \sum_{i,j=1}^{n} \int_\Omega a_{ij}\partial_i v\, \partial_j w\, dx. \tag{2.8}$$

This symmetric bilinear form is well defined for functions v and w in the Sobolev space $H^1(\Omega)$.

The weak formulation of (2.6) in $H_0^1(\Omega)$ is then given by the following.

Given $f \in L^2(\Omega)$, find $u \in H_0^1(\Omega)$ such that

$$\mathcal{A}(u,\varphi) = (f,\varphi), \quad \text{for all} \quad \varphi \in H_0^1(\Omega). \tag{2.9}$$

This problem is uniquely solvable. Indeed, the bilinear form (2.8) is continuous since

$$\mathcal{A}(v,w) \le C\|v\|_1\|w\|_1, \tag{2.10}$$

where the constant C depends on the spectral properties of the coefficient matrix $\{a_{ij}\}$. Moreover, $\mathcal{A}(\cdot,\cdot)$ is $H_0^1(\Omega)$–coercive, i.e.

$$\mathcal{A}(v,v) \ge c(\Omega)\|v\|_1^2, \tag{2.11}$$

where the positive constant $c(\Omega)$ also depends on the spectral properties of $\{a_{ij}\}$. In fact, because of the positive definiteness of $\{a_{ij}\}$, there exists a positive constant c_0 such that

$$\mathcal{A}(v,v) \ge c_0|v|_1^2.$$

Due to the boundedness of Ω, there exists a positive constant $C(\Omega)$ such that

$$|u|_0 \le C(\Omega)|u|_1, \quad \text{for all} \quad u \in H_0^1(\Omega). \tag{2.12}$$

Hence, $|\cdot|_1$ introduces a norm on $H_0^1(\Omega)$, equivalent to $\|\cdot\|_1$ and (2.11) holds. We note that (2.12) is the well known Poincaré inequality. Observe also that (f,φ) is a bounded linear functional on $H_0^1(\Omega)$; i.e. there exists a positive constant $C(f)$ such that

$$|(f,\varphi)| \le C(f)\|\varphi\|_1, \quad \text{for all} \quad \varphi \in H_0^1(\Omega).$$

Thus, the existence and uniqueness of a solution to (2.9) follow from (2.10), (2.11), and the Riesz representation theorem (cf. [86]) stated below.

Theorem 2.4 (Riesz representation theorem) *Let X be a Hilbert space with an inner product $<\cdot,\cdot>$. Then for any bounded linear functional $\mathcal{F}(\cdot)$ on X there exists a unique element $y_\mathcal{F} \in X$ such that*

$$< y_\mathcal{F}, x >= \mathcal{F}(x), \quad \textit{for all} \quad x \in X.$$

Moreover, $\|\mathcal{F}\|_{X'} = \|y_\mathcal{F}\|_X$.

Remark 2.1 *It is clear that $\mathcal{A}(\cdot,\cdot)$ introduces an alternative inner product on $H_0^1(\Omega)$ and one can use $\mathcal{A}(\cdot,\cdot)$ or $|\cdot|_1$ as equivalent norms on $H_0^1(\Omega)$. This, however, changes the Hilbert space structure.*

Remark 2.2 *It is well known (cf. [37]) that the solution of* (2.9) *provides the minimum of the functional*

$$\mathcal{J}(v) = \frac{1}{2}\mathcal{A}(v,v) - (f,v)$$

in $H_0^1(\Omega)$. *In fact the weak formulation* (2.9) *is a general technique for solving many variational problems of interest.*

When the domain Ω is a convex polyhedron in $\mathbb{R}^n$, $n \leq 3$, and the coefficient matrix $\{a_{ij}\}$ consists of functions that are smooth enough on Ω, we shall refer to the following regularity result for the solution u of (2.9) (cf. [60]).

$$\|u\|_{1+\alpha} \leq C\|f\|_{-1+\alpha}, \tag{2.13}$$

for some $\alpha \in (0,1]$. The case of $\alpha = 1$ is known as full elliptic regularity.

2.2.2 Galerkin finite element discretization

Based on the considerations in the previous section, we define a Galerkin (also referred to as Ritz–Galerkin in the literature) approximation to u in (2.9) in a finite-dimensional space $S(\Omega) \subset H_0^1(\Omega)$.

We now specify a finite-dimensional space. To this effect, we partition Ω into triangles (or tetrahedra) $\{\tau_i^h\}$ in the usual way. Here h is the mesh parameter and is defined to be the maximal diameter of all such triangles. By definition, these triangles are closed sets. Unless explicitly stated, we assume that the triangulation is quasi-uniform. In the context of our considerations, quasi-uniform means that there exists a constant $c < 1$, independent of h, such that all triangles contain a ball of diameter ch. The collection of simplex vertices will be denoted by $\{x_i\}$. Let $S_h^0(\Omega)$ be the space of continuous piecewise linear (with respect to the triangulation) functions that vanish on $\partial\Omega$. Although the presentation of the main results and algorithms to be developed in this thesis will be based on considering piecewise linear functions only, most of them extend to higher-order elements without difficulty. We shall remark to indicate such possibilities when it is appropriate.

Thus, the finite element approximation of u is defined by the solution u_h of the following problem:

Find $u_h \in S_h^0(\Omega)$ *such that*

$$\mathcal{A}(u_h, \varphi) = (f, \varphi), \quad \text{for all} \quad \varphi \in S_h^0(\Omega). \tag{2.14}$$

By convention, nodal basis functions φ_i are set in $S_h^0(\Omega)$. Hence, every function $v \in S_h^0(\Omega)$ is represented by

$$v = \sum_{i=1}^{N} v_i \varphi_i,$$

where v_i are the appropriate weights and N is the total number of grid nodes.

We define a symmetric and positive definite linear operator $\mathbf{A} : S_h^0(\Omega) \mapsto S_h^0(\Omega)$ by

$$(\mathbf{A}u, v) = \mathcal{A}(u,v), \quad \text{for all} \quad u,\, v \in S_h^0(\Omega). \tag{2.15}$$

In the literature, $\{\mathcal{A}(\varphi_i, \varphi_j)\}$ is called a *stiffness* matrix, whereas the matrix $\mathcal{M} = \{(\varphi_i, \varphi_j)\}$ is the *mass* matrix. Let f_h be the $L^2(\Omega)$–projection of f into $S_h^0(\Omega)$. It is clear now that (2.14) admits the operator form

$$\mathbf{A}u_h = f_h. \tag{2.16}$$

The following estimates for the error $u_h - u$, which assume full elliptic regularity of u, are well known (cf. [37]).

$$\|u_h - u\|_1 \leq Ch\|u\|_2, \tag{2.17a}$$

$$\|u_h - u\|_0 \leq Ch^2\|u\|_2. \tag{2.17b}$$

In addition, the inverse inequality below holds for functions $v \in S_h^0(\Omega)$ (cf. [37]):

$$\|v\|_1 \leq Ch^{-1}\|v\|_0. \tag{2.18}$$

We remark that inverse inequalities connecting norms with fractional indices can be obtained by interpolation.

2.2.3 Mixed methods

A variety of physical phenomena can be modeled by a system of first-order partial differential equations in contrast to the second-order elliptic equation (2.6), considered in Section 2.2.1. In such models, a new variable $\mathbf{v}$ is introduced by

$$\mathbf{v} = A\nabla u \quad \text{in } \Omega, \tag{2.19a}$$

and is set to satisfy the so-called equilibrium relation

$$\nabla \cdot \mathbf{v} + f = 0 \quad \text{in } \Omega, \tag{2.19b}$$

with a boundary condition

$$u = 0 \quad \text{on } \partial\Omega. \tag{2.19c}$$

Here $A \equiv \{a_{ij}\}$, the coefficient matrix introduced in (2.7), is symmetric and uniformly positive definite on Ω. This is the natural setting for many physical problems whose derivations are based on conservation laws. Steady state fluid flow in porous media is a classical example of an elliptic problem written in the mixed form (2.19) (cf. Section 4.2.1). In this case A is the media permeability tensor, u is the fluid pressure and $\mathbf{v}$ is the fluid flux. Darcy's law which relates the fluid flux to the pressure is the analogue of (2.19a) whereas the mass conservation principle corresponds to (2.19b). We shall discuss in detail applications of the discretizations to be developed here to such problems in Chapter 4. Linear elasticity models involve a similar system of equations for the displacement u and stress fields $\mathbf{v}$. Hook's law, which in such applications relates the stress field to the linearized displacement by $\epsilon_{i,j} = 1/2(\partial u_i/\partial x_j + \partial u_j/\partial x_i)$, corresponds to (2.19a). In addition, the model problem (2.6) itself is often derived from (2.19) by eliminating $\mathbf{v}$. For example, the steady state heat equation is obtained from the first-order system above by eliminating the heat flux $\mathbf{v}$ and obtaining the model (2.6) for the temperature u. There are two observations which motivate further the mixed formulation. First, if solved, (2.19) would provide a direct solution for the two variables of interest. Second, the solution $\mathbf{v}$ is forced to satisfy the equilibrium condition (2.19b), which is very important in many physical applications.

To define the weak formulation of (2.19) we need a pair of Hilbert spaces H_1 and H_2. We denote the corresponding inner products by $(\cdot,\cdot)_{H_1}$ and $(\cdot,\cdot)_{H_2}$. In general these two spaces must be related so that $\nabla \cdot \boldsymbol{\varphi} \in H_2$, for all $\boldsymbol{\varphi} \in H_1$ which can be written in abstract form as

$$\nabla \cdot H_1 \subset H_2. \tag{2.20}$$

Remark 2.3 *Clearly, applying the divergence operator on $\boldsymbol{\varphi} \in H_1$ requires smoothness from $\boldsymbol{\varphi}$ in order for the result to be well defined. Hence, we have potentially abused the notation in the above definitions by not distinguishing explicitly between the classical divergence operator $\nabla\cdot$ and its generalization (meaning that the derivatives participating in $\nabla\cdot$ are taken to be the distributional ones) in cases where the functions in H_1 will not have classical derivatives. However, the implicit switching of the context in which differentiation is understood depending on the function to which it is applied will be the convention we adopt here and later in this thesis.*

Thus, the weak formulation of (2.19) in $H_1 \times H_2$ is given by:
Find $\{\mathbf{v}, u\} \in H_1 \times H_2$ such that

$$(A^{-1}\mathbf{v}, \boldsymbol{\varphi})_{H_1} + (u, \nabla \cdot \boldsymbol{\varphi})_{H_2} = 0, \qquad \text{for all} \quad \boldsymbol{\varphi} \in H_1, \tag{2.21a}$$

$$(\nabla \cdot \mathbf{v}, \psi)_{H_2} = (-f, \psi)_{H_2}, \quad \text{for all} \quad \psi \in H_2. \tag{2.21b}$$

A necessary and sufficient condition for existence and uniqueness of a solution to (2.21) is that the well known inf-sup condition (cf. [30, 85]) holds for the pair of spaces $H_1 \times H_2$, i.e.

$$\sup_{\boldsymbol{\varphi} \in H_1} \frac{(v, \nabla \cdot \boldsymbol{\varphi})^2_{H_2}}{(A^{-1}\boldsymbol{\varphi}, \boldsymbol{\varphi})_{H_1}} \geq c_0 \|v\|^2_{H_2}, \quad \text{for all} \quad v \in H_2, \tag{2.22}$$

for some positive constant c_0.

Therefore, the well-posed weak formulation of (2.19) is tied strongly to the spaces H_1 and H_2. A classical setting for the second-order problem under consideration is the pair $H_1 \times H_2 \equiv H(\mathrm{div};\,\Omega) \times L^2(\Omega)$, which guarantees unique solvability of (2.21) (cf. [30, 85]). The space $H(\mathrm{div};\,\Omega)$ is defined by

$$H(\mathrm{div};\,\Omega) = \left\{ \boldsymbol{\varphi} \in (L^2(\Omega))^n \,\middle|\, \nabla \cdot \boldsymbol{\varphi} \in L^2(\Omega) \right\} \tag{2.23}$$

with a norm given by

$$\|\boldsymbol{\varphi}\|^2_{H(\mathrm{div};\,\Omega)} = \|\boldsymbol{\varphi}\|^2_0 + \|\nabla \cdot \boldsymbol{\varphi}\|^2_0.$$

Raviart–Thomas mixed finite elements

In this section we consider an approximation of (2.21) in finite dimensional subspaces of $H(\mathrm{div};\,\Omega) \times L^2(\Omega)$ where the corresponding pair $\mathcal{V}_h \times \mathcal{W}_h$ to be defined below belongs to the Raviart–Thomas family of spaces [84] (the corresponding family of spaces in $\mathbb{R}^3$ was originally constructed by Nedelec [77]).

Using the simplicial (triangular or tetrahedral) triangulation introduced in Section 2.2.2, for each integer $k \geq 0$ we define

$$\hat{\mathcal{V}}_h^{(k)}(\tau) = (P_k(\tau))^n + \mathbf{x} P_k(\tau), \tag{2.24}$$

where τ is a finite element, $\mathbf{x} \in \mathbb{R}^n$, and $P_k(\tau)$ is a homogenious polynomial of degree k over τ. Also,

$$\mathcal{W}_h^{(k)}(\tau) = P_k(\tau). \tag{2.25}$$

These spaces are designed in such a way that for any $\boldsymbol{\varphi} \in \hat{\mathcal{V}}_h^{(k)}(\tau)$,

$$\begin{aligned} \nabla \cdot \boldsymbol{\varphi} &\in P_k(\tau), \\ \boldsymbol{\varphi} \cdot \mathbf{n}_{|\partial\tau} &\in P_k(\partial\tau), \end{aligned}$$

where $\mathbf{n}$ is the outward normal vector to the boundary $\partial\tau$. Moreover, the divergence operator is surjective from $\hat{\mathcal{V}}_h^{(k)}$ onto $P_k(\tau)$. Once these spaces are defined over each triangle, we set

$$\mathcal{V}_h^{(k)}(\Omega) = \left\{ \boldsymbol{\varphi} \in H(\mathrm{div};\,\Omega) \,\middle|\, \boldsymbol{\varphi}_{|\tau} \in \hat{\mathcal{V}}_h^{(k)}(\tau) \right\}. \tag{2.26}$$

We note that the degrees of freedom for the elements of $\mathcal{V}_h^{(k)}(\Omega)$ are chosen so that $\boldsymbol{\varphi} \cdot \mathbf{n}$ is continuous at the interfaces of elements (cf. [30]). In this thesis we shall consider only the case of $k = 0$ which is known as the *lowest order* Raviart–Thomas space. Thus, $\mathcal{V}_h(\Omega) \equiv \mathcal{V}_h^{(0)}(\Omega)$ consists of functions that are piecewise linear with respect to the triangulation with continuous normal components across the inter-element boundaries, whereas $\mathcal{W}_h(\Omega) \equiv \mathcal{W}_h^{(0)}$ consists of piecewise constant functions.

The discrete problem here is an indefinite system of linear equations given by:

Find $\{\mathbf{v}_h, u_h\} \in \mathcal{V}_h \times \mathcal{W}_h$ *such that*

$$(A^{-1}\mathbf{v}_h, \boldsymbol{\varphi}) + (u_h, \nabla \cdot \boldsymbol{\varphi}) = 0, \qquad \text{for all} \quad \boldsymbol{\varphi} \in \mathcal{V}_h, \tag{2.27a}$$

$$(\nabla \cdot \mathbf{v}_h, \psi) = (-f, \psi), \qquad \text{for all} \quad \psi \in \mathcal{W}_h. \tag{2.27b}$$

Here we have used $(\cdot, \cdot)$ to denote the $L^2(\Omega)$–inner product on $\mathcal{V}_h$ and $\mathcal{W}_h$ with a slight but nonconfusing abuse of notation for the former space.

Unlike (2.14), the well-posedness of (2.27) is not guaranteed unless $\mathcal{V}_h \times \mathcal{W}_h$ satisfies an inf-sup condition (2.22). The R–T spaces satisfy (2.22) with uniform constant c_0 independent of the discretization parameter h (cf. [30]).

Let us define linear operators by

$$\mathbf{A} : \mathcal{V}_h \mapsto \mathcal{V}_h, \quad (\mathbf{A}\boldsymbol{\varphi}, \boldsymbol{\xi}) = (A^{-1}\boldsymbol{\varphi}, \boldsymbol{\xi}), \quad \text{for all} \quad \boldsymbol{\varphi}, \boldsymbol{\xi} \in \mathcal{V}_h, \tag{2.28a}$$

$$\mathbf{B} : \mathcal{V}_h \mapsto \mathcal{W}_h, \quad (\mathbf{B}\boldsymbol{\varphi}, \phi) = (\nabla \cdot \boldsymbol{\varphi}, \phi), \quad \text{for all} \quad \boldsymbol{\varphi} \in \mathcal{V}_h,\ \phi \in \mathcal{W}_h, \tag{2.28b}$$

$$\mathbf{B}^T : \mathcal{W}_h \mapsto \mathcal{V}_h, \quad (\mathbf{B}^T\phi, \boldsymbol{\varphi}) = (\phi, \nabla \cdot \boldsymbol{\varphi}), \quad \text{for all} \quad \boldsymbol{\varphi} \in \mathcal{V}_h,\ \phi \in \mathcal{W}_h. \tag{2.28c}$$

It is clear now that (2.27) admits the operator form

$$\begin{pmatrix} \mathbf{A} & \mathbf{B}^T \\ \mathbf{B} & 0 \end{pmatrix} \begin{pmatrix} \mathbf{v}_h \\ u_h \end{pmatrix} = \begin{pmatrix} 0 \\ f_h \end{pmatrix}, \tag{2.29}$$

where f_h is the $L^2(\Omega)$–projection of f into $\mathcal{W}_h$.

The following estimate for the errors $u - u_h$ and $\mathbf{v} - \mathbf{v}_h$ is standard (cf. [30, 85]).

$$\|u - u_h\|_0 + \|\mathbf{v} - \mathbf{v}_h\|_{H(\mathrm{div};\,\Omega)} \leq Ch^k(|u|_k + |\mathbf{v}|_k + |\nabla \cdot \mathbf{v}|_k). \tag{2.30}$$

Lagrange multipliers

Frequently, the indefinite nature of (2.27) is a source of significant difficulties when numerical solution of this problem is attempted. In this section we introduce an alternative approach to discretizing (2.21) which is intended to result in better behaved systems of linear equations.

Consistent with the choice of the lowest order R–T spaces, we introduce the space Λ_h^0 of functions which are constants on each edge of the triangulation of Ω and zero on the edges on $\partial\Omega$. We define a bilinear form on $\mathcal{V}_h \times \Lambda_h^0$ by

$$< \mu_h, \mathbf{w}_h > = \sum_\tau \int_{\partial\tau} \mu_h \mathbf{w}_h \cdot \mathbf{n}\, d\sigma, \quad \text{for all} \quad \mu_h \in \Lambda_h^0,\ \mathbf{w}_h \in \mathcal{V}_h. \tag{2.31}$$

Let also,

$$\hat{\mathcal{V}}_h(\Omega) = \left\{ \boldsymbol{\varphi} \in (L^2(\Omega))^n \,\middle|\, \boldsymbol{\varphi}_{|\tau} \in \hat{\mathcal{V}}_h^{(0)}(\tau) \right\}. \tag{2.32}$$

Remark 2.4 *The main difference between $\hat{\mathcal{V}}_h(\Omega)$ and $\mathcal{V}_h(\Omega)$ is that the elements of the former space do not necessarily have continuous normal components across the element boundaries. This detail is important for the discretization defined in the theorem below (cf. [30]).*

Theorem 2.5 *Let $\{\mathbf{v}_h, u_h\}$ be the solution of (2.27). Then $\{\mathbf{v}_h, u_h, \lambda_h\}$ is the unique solution of the following problem: find $\{\mathbf{v}_h, u_h, \lambda_h\} \in \hat{\mathcal{V}}_h \times \mathcal{W}_h \times \Lambda_h^0$ such that*

$$(A^{-1}\mathbf{v}_h, \boldsymbol{\varphi}) + (u_h, \nabla \cdot \boldsymbol{\varphi}) = < \lambda_h, \nabla \cdot \boldsymbol{\varphi} >, \quad \textit{for all} \quad \boldsymbol{\varphi} \in \hat{\mathcal{V}}_h, \tag{2.33a}$$

$$(\nabla \cdot \mathbf{v}_h, \psi) = (-f, \psi), \quad \textit{for all} \quad \psi \in \mathcal{W}_h, \tag{2.33b}$$

$$< \mu_h, \nabla \cdot \boldsymbol{\varphi} > = 0, \quad \textit{for all} \quad \mu_h \in \Lambda_h^0. \tag{2.33c}$$

Remark 2.5 *An important information about the nature of the multipliers λ_h (also called Lagrange multipliers) can be deduced from the observation that if we take an inner product of equation (2.19a) with $\boldsymbol{\varphi} \in \hat{\mathcal{V}}_h$ and integrate by parts on every simplex τ we get*

$$(A^{-1}\mathbf{v}, \boldsymbol{\varphi})_\tau + (u, \nabla \cdot \boldsymbol{\varphi})_\tau = \int_{\partial\tau} u\mathbf{v} \cdot \mathbf{n}\, d\sigma. \tag{2.34}$$

Comparing (2.34) with (2.33a) shows that the multipliers behave like u on the inter-element boundaries.

The operator form of (2.33) is given by

$$\begin{pmatrix} \bar{\mathbf{A}} & \bar{\mathbf{B}}^T & \mathbf{C}^T \\ \bar{\mathbf{B}} & 0 & 0 \\ \mathbf{C} & 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{v}_h \\ u_h \\ \lambda_h \end{pmatrix} = \begin{pmatrix} 0 \\ f_h \\ 0 \end{pmatrix}, \tag{2.35}$$

where f_h is the $L^2(\Omega)$–projection of f into $\mathcal{W}_h$, the linear operator $\mathbf{C}^T$ is the adjoint of $\mathbf{C}$ defined by

$$\mathbf{C} : \hat{\mathcal{V}}_h \mapsto \Lambda_h^0, \quad < \mathbf{C}\boldsymbol{\varphi}, \lambda > = - < \boldsymbol{\varphi}, \lambda >, \quad \text{for all} \quad \boldsymbol{\varphi} \in \hat{\mathcal{V}}_h,\ \lambda \in \Lambda_h^0, \tag{2.36}$$

and $\bar{\mathbf{A}}$, $\bar{\mathbf{B}}$ and $\bar{\mathbf{B}}^T$ are defined similarly to (2.28) with the obvious changes of the domain and range space in view of Remark 2.4.

Although (2.35) looks even more complicated than (2.29), this system of equations is in fact easier to solve, due to the properties of $\hat{\mathcal{V}}_h(\Omega)$ and the nature of the multipliers λ_h which we emphasized in Remark 2.4 and Remark 2.5. The matrix of the operator $\bar{\mathbf{A}}$ is block diagonal, each block being a $(n+1)\times(n+1)$ matrix, corresponding to one finite element. Hence, the unknown $\mathbf{v}_h$ is easy to eliminate at the element level. Moreover, $\bar{\mathbf{B}}\bar{\mathbf{A}}^{-1}\bar{\mathbf{B}}^T$ is also block diagonal with blocks corresponding to single elements. Thus, a block Gaussian elimination procedure applied to (2.35) results in a linear system for the multipliers, which has symmetric and positive definite matrix (cf. [30, 85]).

2.3 Parabolic problems

In this section we define standard finite element approximations to parabolic problems. This will provide the basis for developing in the next section a new discretization technique for such equations.

Let us consider the model parabolic problem

$$u_t + \mathcal{L}u = f \qquad \text{in } \mathcal{Q} \equiv \Omega \times (0,T], \tag{2.37a}$$

$$u(x,t) = 0 \qquad \text{on } \partial\Omega, \text{ for all } t > 0, \tag{2.37b}$$

$$u(x,0) = u^0(x), \tag{2.37c}$$

where $\mathcal{L}$ is defined in (2.7) and $u_t \equiv \dfrac{\partial u}{\partial t}$. Following the approach from the previous section, we reformulate (2.37) in a weak form by

$$(u_t,\varphi) + \mathcal{A}(u,\varphi) = (f,\varphi), \quad \text{for all} \quad \varphi \in H_0^1(\Omega),\ t > 0, \tag{2.38a}$$

$$u(x,0) = u^0(x). \tag{2.38b}$$

Next, the above weak form is discretized both in space and time. For that purpose, we triangulate Ω as described in the previous section. Thus, the finite element discretizations described in Section 2.2.2 and Section 2.2.3 can be applied. Letting κ be the time step and u_h^m be the approximation in $S_h^0(\Omega)$ of $u(t)$ at $t = t_m = m\kappa$ (m – nonnegative integer), the time derivative u_t is replaced with a backward difference quotient

$$\bar{\partial}_t u_h^m = \frac{u_h^m - u_h^{m-1}}{\kappa},$$

which leads to backward Euler–Galerkin methods. The discrete problem is then written as:

Find $u_h^m \in S_h^0(\Omega)$ such that

$$(\bar{\partial}_t u_h^m,\varphi) + \mathcal{A}(u_h^m,\varphi) = (f^m,\varphi), \quad \text{for all} \quad \varphi \in S_h^0(\Omega), \tag{2.39a}$$

$$u_h(x,0) = u_h^0(x), \tag{2.39b}$$

if the finite element method from Section 2.2.2 is used in space or as :

Find $\mathbf{v}_h^m \in \mathcal{V}_h$ and $u_h^m \in \mathcal{W}_h$ such that

$$(A^{-1}\mathbf{v}_h^m,\boldsymbol{\varphi}) + (u_h^m, \nabla\cdot\boldsymbol{\varphi}) = 0, \qquad \text{for all} \quad \boldsymbol{\varphi} \in \mathcal{V}_h, \tag{2.40a}$$

$$(\nabla\cdot\mathbf{v}_h^m,\psi) - (\bar{\partial}_t u_h^m,\psi) = (-f^m,\psi), \quad \text{for all} \quad \psi \in \mathcal{W}_h, \tag{2.40b}$$

$$u_h(x,0) = u_h^0(x), \tag{2.40c}$$

if the mixed method from Section 2.2.3 is utilized in space (the equivalent multiplier form (2.33) can be used as well). u_h^0 in the above equations is some approximation to $u^0(x)$ in the space where u_h^m belongs.

We define an operator form of (2.39) by

$$\mathbf{A}_\kappa U^m = \bar{f}_h^m \tag{2.41a}$$

$$u_h(x,0) = u_h^0(x), \tag{2.41b}$$

where the operator $\mathbf{A}_\kappa$ is given by

$$(\mathbf{A}_\kappa \varphi, \phi) = \kappa^{-1}(\varphi, \phi) + \mathcal{A}(\varphi, \phi), \quad \text{for all} \quad \varphi,\ \phi \in S_h^0(\Omega) \tag{2.42}$$

and $\bar{f}_h^m$ is given by

$$(\bar{f}_h^m, \varphi) = (f^m, \varphi) + \kappa^{-1}(U^{m-1}, \varphi), \quad \text{for all} \quad \varphi \in S_h^0(\Omega). \tag{2.43}$$

A fundamental issue in the theory of fully discrete schemes for parabolic equations is their stability.

Definition 2.3 *A time-stepping method is called stable if*

$$\|u_h^m\|_p \le C_1 \|u_h^0\|_p + C_2 \max_{0 \le \ell \le m} \|f_h^\ell\|_q, \quad m \ge 0,$$

holds for some positive constants C_1 and C_2, independent of κ and h, and some norms $\|\cdot\|_p$ and $\|\cdot\|_q$.

Definition 2.4 *A time-stepping scheme is called* conditionally *stable if it is stable only for a range of time-steps κ, depending on h. A scheme is called* unconditionally *stable if it is stable for any κ, independent of h.*

An example of conditionally stable time-stepping scheme is the forward Euler–Galerkin method (cf. [90]), where for stability κ must satisfy the classical Courant–Friedrichs–Lax (CFL) condition

$$\kappa \le Ch^2, \tag{2.44}$$

for some constant $C > 0$ independent of h and κ.

It is a standard result (cf. [94]) that the backward Euler–Galerkin method (2.39) is unconditionally stable with error $\|u(t_m) - u_h^m\|_0$ bounded by

$$\|u(t_m) - u_h^m\|_0 \le \|u^0 - u_h^0\|_0 + Ch^r \left\{ \|u^0\|_r + \int_0^{t_m} \|u_t\|_r \, ds \right\} + \kappa \int_0^{t_m} \|u_{tt}\|_0 \, ds, \quad \text{for all} \quad m \ge 0. \tag{2.45}$$

A powerful tool for investigating the stability of time-stepping schemes is the *maximum principle*. In the remainder of this chapter we shall use a discrete variant of this principle which is defined below (cf. [90]).

Definition 2.5 *A mesh function $y_i^m = y(x_i, t_m)$ is a discrete real valued function, defined at the nodes (x_i, t_m) of the time-space grid in $\mathcal{Q}$.*

Let $\mathcal{D}$ be the matrix representation of a linear operator on $S_h^0(\Omega)$. Also let V be the vector representation of $v \in S_h^0(\Omega)$ with respect to the nodal basis of $S_h^0(\Omega)$. By convention, we shall use either V or v to denote the same element of $S_h^0(\Omega)$. Let us denote by $(\mathcal{D}V)_i$ the i-th element of the vector of real numbers $\mathcal{D}V$, for any $V \in S_h^0(\Omega)$. Clearly, $(\mathcal{D}V)_i$ can be written in the form

$$(\mathcal{D}V)_i = b(x_i, x_i)V_i - \sum_{x_j \ne x_i} b(x_i, x_j)V_j,$$

with some real weights $b(x_i, x_j)$.

Definition 2.6 *An operator stencil $ST(\mathcal{D})_i$ of the matrix $\mathcal{D}$ at a grid point $x_i \in \Omega$ is the set of grid points*

$$ST(\mathcal{D})_i = \{x_j \in \Omega \mid b(x_i, x_j) \ne 0\}.$$

Theorem 2.6 (Maximum principle) *Let V be a nonconstant mesh function on Ω. Also let the following inequalities hold for all $x_i \in \Omega \setminus \partial\Omega$:*

$$b(x_i, x_i) > 0, \tag{2.46a}$$

$$b(x_i, x_j) > 0, \quad \textit{for all} \quad x_j \in ST(\mathcal{D})_i, \tag{2.46b}$$

$$d(x_i) = b(x_i, x_i) - \sum_{x_j \in ST(\mathcal{D})_i} b(x_i, x_j) \geq 0. \tag{2.46c}$$

Moreover, if $(\mathcal{D}V)_i \leq 0$ ($(\mathcal{D}V)_i \geq 0$) for all $x_i \in \Omega \setminus \partial\Omega$, then V cannot attain its maximal positive (minimal negative) value at any interior node x_i of the triangulation of Ω.

It is an easy observation now that if a backward Euler–Galerkin discretization of (2.37) produces an operator $\mathbf{A}_\kappa$ with a corresponding matrix that complies with (2.46) the unconditional stability of such a scheme is guaranteed by Theorem 2.6.

An important preliminary step in defining discretizations that satisfy Theorem 2.6 is to use the *method of lumped masses* (cf. [94]). The idea here is to replace the mass matrix corresponding to (V, W) for any V and W in $S_h^0(\Omega)$ with a diagonal matrix obtained by applying an appropriate quadrature rule for evaluation of (φ_i, φ_j), where φ_i and φ_j are basis functions for $S_h^0(\Omega)$. Using such quadrature rules (cf. [94]), one gets

$$|(\varphi, \psi)_h - (\varphi, \psi)| \leq Ch^2 \|\varphi\|_1 \|\psi\|_1, \quad \text{for all} \quad \varphi, \psi \in S_h^0(\Omega).$$

which leads to an equivalent method that preserves the order of approximation. Here, $(\varphi, \psi)_h$ is the lumped form. In the remainder of this chapter we shall consider schemes with lumped masses only.

2.4 Special discretization techniques for parabolic problems

In this section we introduce a new technique for discretization of parabolic problems and provide stability and error analyses. It is motivated by the fact that parabolic partial differential equations are used to model a variety of time-dependent diffusive or convective-diffusive processes. The solution of these equations may develop highly localized properties both in space and in time. In many physical applications, such properties are due to stationary features of the domain Ω such as wells, cracks, obstacles, domain boundaries, etc., which are fixed in space. In many other cases they are moving in time, e.g., moving point loads, sharp fronts, etc. We already saw in the previous sections that the accuracy of the approximation depends on the discretization parameters h and κ. Thus, in typical large-scale applications, in order to bring the error within a desired tolerance, very fine grids may be needed. Because of the size of the corresponding numerical model, the local properties of the solution cannot be resolved using quasi-uniform grids even with the largest of today's supercomputers. Adaptive local grid refinement techniques are an attractive alternative for computing the local behavior of the solution within a given error tolerance while saving computational resources. The idea here is to first, introduce a global time-space discretization for the whole time-space region $\mathcal{Q}$; next, in some subdomains of Ω, chosen using either adaptive mesh refinement strategies or *a priori* information about potential rapid local temporal change of the solution, one introduces time steps and mesh sizes that are fractions of the global ones. In this way a composite time-space mesh is developed.

The most important problem that arises when local refinement is used is the construction of a stable and accurate time-stepping scheme. Difficulties occur at the interface between the subregions with different time steps. The available literature shows that these are the places that govern the stability and the accuracy of the whole scheme to a great extent. For example, the implicit schemes derived in [53] are based on a finite volume approximation of the balance equation. This approach produces schemes that are both unconditionally stable and locally conservative. However, this method does not allow linear interpolation in time along the interface which leads to a loss of one half in the order of the convergence rate. The schemes constructed in [39] use a combination of an implicit approximation in the interior of the subregions and an explicit treatment of the interfaces. This results in a computationally efficient but conditionally stable method. The schemes constructed in [50] are of backward Euler–Galerkin type and

are unconditionally stable but still do not allow linear interpolation in time along the interface, which again leads to a loss of one half in the order of the convergence rate.

The literature cited above indicates some time-step restrictions either for stability or accuracy when local time stepping is utilized. The scope of our considerations here is to provide a theoretical analysis of the stability and *a priori* error estimates of discretization schemes for parabolic problems when local refinement is applied in time and space. We shall investigate schemes with linear interpolation in time along the interface that preserve the unconditional stability and lead to more accurate approximation. Our analysis is based on ideas from the sequence of two papers [51, 52] where such schemes are considered in the finite difference paradigm.

2.4.1 Preliminaries

Our approach to defining a backward Euler–Galerkin method with local refinement is based on discretizations that comply with the maximum principle (cf. Theorem 2.6 above), in particular schemes with lumped masses. In order to keep the presentation short and terse, only a classical model problem will be analyzed in detail. We shall remark to indicate possible generalizations of our theory.

We begin with some standard assumptions about the properties of the coefficient matrix $\{a_{i,j}\}$ and the triangulation.

Assumption 2.1 *The coefficient matrix is diagonal and constant, i.e.*

$$\{a_{ij}\} \equiv diag\{a_1, \ldots, a_n\},$$

for some positive constants a_i, $i = 1, \ldots, n$.

Assumption 2.2 *The triangulation of* Ω *has no elements with obtuse angles.*

It is well known (cf. [94]) that Assumption 2.2 guarantees that the resulting stiffness matrix satisfies the hypothesis of the maximum principle theorem.

In the remainder of this chapter we shall also consider the special case of cubical domains in $\mathbb{R}^n$, i.e. $\Omega = [0,1]^n$. For such domains we define a triangulation by first subdividing Ω into uniform parallelepiped cells with faces parallel to the coordinate axes. The final triangulation is obtained by further splitting each cell into triangles (tetrahedra) so that no new grid nodes are introduced. The finite element space $S_h^0(\Omega)$ is then defined with respect to the tetrahedral discretization of Ω. Obviously, Assumption 2.2 holds for such triangulations.

In principle, there are few ways to construct such triangulations of cubical domains. The properties of some of them are rather useful for our purposes. Our guiding observation is that there exist triangulations of Ω which produce stiffness matrices that are the same as the matrices obtained from standard finite difference discretizations. To clarify the latter, let us enumerate the nodes in the grid by $x_{l_1 \ldots l_n}$ in some fashion, say lexicographical. Also let $y_{l_1 \ldots l_n}$ be a mesh function defined with respect to this grid. We define a *forward difference* operator Δ_j by

$$\Delta_j y_{l_1 \ldots l_n} = y_{l_1 \ldots l_{j+1} \ldots l_n} - y_{l_1 \ldots l_j \ldots l_n}$$

and the related divided forward difference operator by

$$\partial^{(j)} y_{l_1 \ldots l_n} = \frac{\Delta_j y_{l_1 \ldots l_n}}{h_j}.$$

Here h_j is the mesh size in the j–direction. Correspondingly, the *backward difference* operator $\bar{\Delta}_j$ is given by

$$\bar{\Delta}_j y_{l_1 \ldots l_n} = y_{l_1 \ldots l_j \ldots l_n} - y_{l_1 \ldots l_{j-1} \ldots l_n}$$

and the related divided backward difference by

$$\bar{\partial}^{(j)} y_{l_1 \ldots l_n} = \frac{\bar{\Delta}_j y_{l_1 \ldots l_n}}{h_j}.$$

We set $h = \max\{h_1, \ldots, h_n\}$ and $\hbar = h_1 \ldots h_n$.

Let us further define a discrete linear operator $\mathcal{L}_h : S_h^0(\Omega) \mapsto S_h^0(\Omega)$ by

$$\mathcal{L}_h V_\alpha = \sum_{i=1}^{n} a_i \partial^{(i)} \bar{\partial}^{(i)} V_\alpha, \quad \text{for all grid points } x_\alpha \in \Omega, \text{ all } V \in S_h^0(\Omega). \tag{2.47}$$

Here $\alpha = l_1 \ldots l_n$ is a multi-index such that x_α is a grid point in the discretization of Ω.

As we indicated above, provided that Assumption 2.1 holds, there exist triangulations of Ω such that

$$\sum_{\beta} V_\beta \mathcal{A}(\varphi_\beta, \varphi_\alpha) = -\hbar \mathcal{L}_h V_\alpha, \quad \text{for all} \quad V \in S_h^0(\Omega). \tag{2.48}$$

There are many examples where (2.48) is satisfied. For instance, (2.48) is true for a uniform discretization of the interval $[0,1]$. Another example is a rectangular uniform grid in $[0,1]^2$ where each rectangular cell is subdivided into two triangles by connecting the lower left corner with the upper right corner. A less trivial case is $[0,1]^3$. The triangulation considered in [93] guarantees that (2.48) holds. It is constructed on top of an existing uniform parallelepiped grid by first splitting each cell into two prisms and next subdividing them into tetrahedra by projecting the celestial diagonal on the faces. This discussion can be summarized in the following assumption.

Assumption 2.3 *$\Omega \equiv [0,1]^n$ is triangulated so that (2.48) holds.*

We conclude this section with definitions and results concerning the discrete operators introduced above, needed for the analysis. The discrete inner product on $S_h^0(\Omega)$ is defined by

$$\langle U, V \rangle = \hbar \sum_{\alpha} U_\alpha V_\alpha.$$

Correspondingly, the discrete $L^2(\Omega)$–norm is given by

$$\|V\|_{0,h} = \langle V, V \rangle^{1/2}.$$

The discrete $L^\infty(\Omega)$–norm for grid functions is defined by

$$\|V\|_{\infty,h} = \max_{\alpha} |V_\alpha|.$$

The following lemma establishes important norm equivalences in $S_h^0(\Omega)$.

Lemma 2.1 *$\mathcal{L}_h$ introduces an equivalent norm on $S_h^0(\Omega)$, i.e.*

$$c\|V\|_1^2 \leq -\langle \mathcal{L}_h V, V \rangle \leq C\|V\|_1^2, \quad \textit{for all} \quad V \in S_h^0(\Omega), \tag{2.49a}$$

holds with constants c and C independent of h. Moreover,

$$c\|V\|_0 \leq \|V\|_{0,h} \leq C\|V\|_0, \quad \textit{for all} \quad V \in S_h^0(\Omega). \tag{2.49b}$$

The proof of this lemma is a straightforward consequence from the local properties of functions in finite element spaces and shall be omitted.

Let us denote by $\mathcal{A}_h$ the stiffness matrix corresponding to $\mathbf{A}$, i.e.

$$\mathcal{A}_h \equiv \{\mathcal{A}(\varphi_i, \varphi_j)\}.$$

We define a discrete linear operator $\mathcal{T}_{\kappa,h} : S_h^0(\Omega) \mapsto S_h^0(\Omega)$ by

$$\mathcal{T}_{\kappa,h} V^m = \kappa^{-1} \hbar' V^m - \mathcal{A}_h V^m, \quad \text{for all} \quad V^m \in S_h^0(\Omega).$$

Here $\hbar'$ is the appropriate local (with respect to the grid nodes) positive weighting factor, determined by the mass lumping procedure.

It is clear now that if mass lumping is applied to (φ_i, φ_j), then the discrete problem (2.41) takes the algebraic form

$$\mathcal{T}_{\kappa,h} U^m = \bar{F}_h^m \tag{2.50a}$$

$$u_h(x,0) = u_h^0(x), \tag{2.50b}$$

where $\bar{F}_h^m = \hbar'(\kappa^{-1} V^{m-1} + F_h^m)$, and F_h^m is the vector of coefficients, representing the $L^2(\Omega)$–projection of f^m in (2.43) in the basis of $S_h^0(\Omega)$.

2.4.2 Meshes with local time stepping

In this section we define the composite grid where the locally refined discretization will be formulated. We begin with discretizations that use refinement in time only.

Let us introduce closed connected sets $\{\Omega_i\}_{i=1}^M$, which are subsets of Ω with boundaries aligned with the spatial parallelepiped discretization already defined and set

$$\Omega_0 = \Omega \ \setminus \ \bigcup_{i=1}^{M} \Omega_i.$$

We shall assume that $\Omega_0 \neq \emptyset$. In order to avoid unnecessary complications that contribute little to the generality of our considerations, we also assume that

$$\operatorname{dist}(\Omega_i, \Omega_j) \geq \ell h, \qquad \text{for} \quad i, j > 0,$$

where $\ell > 1$ is an integer. This means that there are no two neighbor nodes x_α and x_β such that $x_\alpha \in \Omega_i$ and $x_\beta \in \Omega_j$, for $i, j > 0$. Two nodes x_α and x_β are neighbors if the coefficient $b(x_\alpha, x_\beta)$ from the operator stencil at x_α is nonzero. The case of nested refinement can be treated in a similar manner but needs additional notation and special consideration.

Let ω be the set of the nodes of the initial discretization of Ω. Also let ω_i, $i = 1, \ldots, M$ be the set of all nodes in ω that belong to Ω_i. The nodes of the initial discretization of Ω which reside on $\partial\Omega_i$, $i > 0$, do not belong to Ω_0 (which is open relative to Ω). We set

$$\omega_o = \omega \ \setminus \ \bigcup_{i=1}^{M} \omega_i.$$

In each ω_i, $i = 0, \ldots, M$, let $\partial\omega_i$ be the subset of boundary nodes, i.e. all nodes which have at least one neighbor not in ω_i. It should be noted that $\partial\omega_i$ contains only nodes which do not reside on $\partial\Omega$ in case $\partial\Omega_i \cap \partial\Omega \neq \emptyset$.

A discrete time step κ_i is associated with each Ω_i such that, for positive integers m_i,

$$\kappa_0 = m_i \kappa_i, \quad \text{for all} \quad i = 1, \ldots, M. \tag{2.51}$$

Consequently, the discrete time levels t_i^j in $\Omega_i \times [0,T]$ are defined by $t_i^j = j\kappa_i$, $j = 1, 2, \ldots$. The set of grid points Θ_i in $\Omega_i \times [0,T]$ is then given by

$$\Theta_i = \bigcup_{\substack{x \in \omega_i \\ j = 0, 1, \ldots}} (x, j\kappa_i), \quad \text{for all} \quad j = 1, 2, \ldots, \quad i = 0, \ldots, M.$$

Finally, the set of all grid points in $\mathcal{Q}$ is given by

$$\Theta = \bigcup_{i=0}^{M} \Theta_i.$$

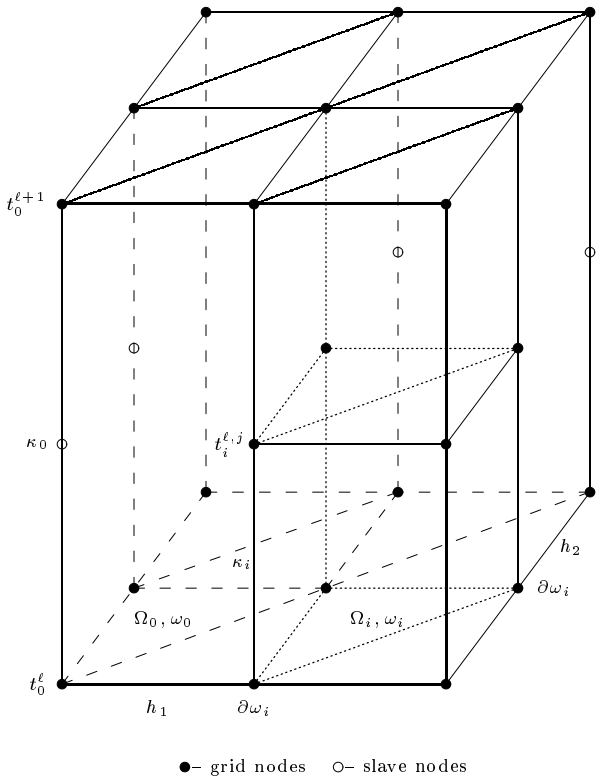


Figure 2.1: A grid with local time stepping in $\mathbb{R}^2 \times t$

Because of (2.51), the local refinement in time is associated with the subdomains Ω_i, $i = 1, \ldots, M$. Notice also that $(x, j\kappa_0)$ is a grid point in Θ for all $x \in \omega$ and all $j \geq 0$. Thus, it makes sense to introduce the notation Θ_i^ℓ, $\ell = 0, \ldots, m_i$, for the nodes in Θ_i between time levels t_0^ℓ and $t_0^{\ell+1}$, i.e.

$$\Theta_i^\ell = \bigcup_{\substack{x \in \omega_i \\ j=0}}^{m_i} (x, t_0^\ell + j\kappa_i) \;=\; \bigcup_{\substack{x \in \omega_i \\ j=0}}^{m_i} (x, t_i^{\ell,j}), \qquad t_i^{\ell,j} = t_0^\ell + j\kappa_i, \quad i = 0, \ldots, M.$$

Correspondingly, the boundary nodes of Θ_i^ℓ are defined by

$$\partial\Theta_i^\ell = \bigcup_{\substack{x \in \omega_i \\ j=0}}^{m_i} (x, t_i^{\ell,j}), \qquad x \in \partial\omega_i, \quad i = 0, \ldots, M.$$

Thus, the set of all grid nodes in the time slab $[t_0^\ell, t_0^{\ell+1}]$ is given by

$$\Theta^\ell = \bigcup_{i=0}^{M} \Theta_i^\ell.$$

An example of a locally refined grid is shown in Fig. 2.1.

2.4.3 The finite element discretization

Now we construct a finite dimensional space of functions for the backward Euler–Galerkin method with time steps varying in space. For each coarse time level t_0^ℓ let $S_h^0(\Omega)^\ell$ be the usual finite element space of functions that are piecewise linear with respect to a tetrahedral (triangle) triangulation of Ω, satisfying

Assumption 2.3. Because of the different time steps in some regions of Ω, we need to specify additional spaces associated with each Ω_i, $i > 0$. Let $\Omega_i^h \subset \Omega$ be an extension of Ω_i by one mesh size h in the interior of Ω. Notice that such an extension of Ω_i preserves the alignment of the boundary of Ω_i^h with the grid. Moreover, $\partial\Omega_i^h$ contains only nodes adjacent to $\partial\omega_i$ and nodes on $\partial\Omega$ in case $\partial\Omega_i \bigcap \partial\Omega \neq \emptyset$. For each intermediate time level $t_i^{\ell,j}$ let $S_h(\Omega_i^h)^{\ell,j}$ be the space of piecewise linear functions with respect to the triangulation of Ω_i^h with support contained in the closure of Ω_i^h whose traces on the boundary $\partial\Omega_i^h$ are determined by linear interpolation in time with respect to the formula

$$v_i(x)^{\ell,j} = \frac{j}{m_i} v(x)^{\ell+1} + \frac{m_i - j}{m_i} v(x)^{\ell}, \quad \text{for all} \quad 1 \leq j < m_i, \tag{2.52}$$

where $x \in \partial\Omega_i^h$, $v_i^{\ell,j} \in S_h(\Omega_i^h)^{\ell,j}$, $v^\ell \in S_h^0(\Omega)^\ell$, and $v^{\ell+1} \in S_h^0(\Omega)^{\ell+1}$. Clearly, if $x \in \partial\Omega_i^h$ then $(x, t_i^{\ell,j}) \notin \Theta_i^\ell$. The extra grid nodes on $\partial\Omega_i^h \times t_i^{\ell,j}$ needed to define the finite element approximation on the composite grid are denoted as *slave nodes* in Fig. 2.1.

Once we have identified approproiate finite element spaces, we can switch to operator notation for presenting the algebraic problem similarly to (2.41). We note that because of Assumption 2.1 and Assumption 2.3, the discrete operator here is equivalent to a finite difference operator. The only special regions are the interfaces of the refined regions. To define the operator action at the points of the interfaces, we use an interpolation in time according to (2.52) (cf. [52]). The algebraic problem for advancing from t_0^ℓ to $t_0^{\ell+1}$ in Θ_i^ℓ, $i = 0, 1, \ldots, M$, $j = 1, \ldots, m_i$, is given by

$$\mathcal{T}_{\kappa_0,h} U_\alpha^{\ell+1} = \hbar' \kappa_0^{-1} U_\alpha^\ell + F_\alpha^{\ell+1}, \qquad \text{in } \Theta_0^\ell, \tag{2.53a}$$

$$\mathcal{T}_{\kappa_i,h} U_\alpha^{\ell,j} = \hbar' \kappa_i^{-1} U_\alpha^{\ell,j-1} + F_\alpha^{\ell,j}, \quad \text{in } \Theta_i^\ell \text{ for } i = 1, \ldots, M, \tag{2.53b}$$

$$U(x_\alpha, t) = 0, \qquad \text{for all } x_\alpha \in \partial\Omega,\ t > 0. \tag{2.53c}$$

Let us denote by $U^{[\ell,\ell+1]}$ the vector of all unknowns in (2.53). Similarly, let $F^{[\ell,\ell+1]}$ be the corresponding right hand side vector. Let also $G^{[\ell,\ell+1]}(U^\ell)$ be the contribution of U^ℓ to the right hand side of (2.53) due to interpolation according to (2.52). Then the above composite-grid problem can be written in the form

$$\mathcal{T}^{[\ell,\ell+1]} U^{[\ell,\ell+1]} = G^{[\ell,\ell+1]}(U^\ell) + F^{[\ell,\ell+1]}, \quad \text{in } \Theta^\ell, \tag{2.54a}$$

$$U(x_\alpha, t) = 0, \quad \text{for all } x_\alpha \in \partial\Omega,\ t > 0, \tag{2.54b}$$

where $\mathcal{T}^{[\ell,\ell+1]}$ is the corresponding composite-grid operator.

2.4.4 Stability and error analysis

In this section we investigate the stability and approximation error of the discrete problem (2.53). Our analysis is based on the discrete form of the maximum principle, provided by Theorem 2.6, combined with norm equivalences in finite element function spaces provided by the Sobolev imbedding theorem. We derive error estimates in the discrete maximum norm. Such a technique is standard in the literature (cf. [90]). These estimates properly take into account the local properties of the solution u but usually require higher regularity of u than the one required in the estimate (2.45). We remark that Assumption 2.1 and Assumption 2.3 guarantee the existence of regular enough solutions.

We first consider the stability of time-stepping scheme defined in (2.54).

Theorem 2.7 *Let Assumption 2.2 hold. Then the time-stepping method* (2.54) *is unconditionally stable.*

Proof: It is straightforward to check that due to the linear interpolation in (2.52) along the interfaces of the refined regions, the stencil of the composite-grid operator $\mathcal{T}^{[\ell,\ell+1]}$ satisfies the hypothesis of Theorem 2.6 on Θ^ℓ. In addition, $U^{[\ell,\ell+1]}$ vanishes on $\partial\Omega$ for all $t \in [t^\ell, t^{\ell+1}]$. Thus, by the maximum principle,

$$\begin{aligned} \max_{x_\alpha \in \omega} |U^\ell(x_\alpha)| &\leq \max |U^{[\ell,\ell+1]}| \\ &\leq \max_{x_\alpha \in \omega} |U^{\ell-1}(x_\alpha)| + C \max |F^{[\ell,\ell+1]}|. \end{aligned}$$

The constant C above is independent of κ_i, $i = 1, \ldots, M$, and h. Summing the last inequality over ℓ proves stability. □

Remark 2.6 *Since only Assumption 2.2 and standard assumptions for symmetry and positive definiteness of the coefficient matrix $\{a_{ij}\}$ (need not be diagonal) are needed in the proof of Theorem 2.7, this result guarantees stability of the time-stepping method* (2.54) *when domains with general geometry are considered. In addition, it is possible to generalize our argument to include equations with convective and reaction terms (cf. [52]).*

We now turn to the error analysis of the approximation defined by (2.54). An estimation of the local truncation error shall be used; due to this, we shall restrict our attention only to parallelepiped domains in $\mathbb{R}^n$. In addition, it is required that Assumption 2.1 and Assumption 2.3 hold. Let us define the error of the approximation to the solution of (2.53) by

$$\begin{aligned} e(x_\alpha, t_0^\ell) &= u(x_\alpha, t_0^\ell) - U^\ell(x_\alpha), \qquad x_\alpha \in \omega_0, \\ e(x_\alpha, t_i^{\ell,j}) &= u(x_\alpha, t_i^{\ell,j}) - U^{\ell,j}(x_\alpha), \quad x_\alpha \in \omega_i,\ i = 1, \ldots, M. \end{aligned}$$

According to our assumptions, the stiffness matrix $\mathcal{A}_h$ reduces to the finite difference operator $\mathcal{L}_h$. Hence, using a Taylor series expansion, it is straightforward to check that

$$\mathcal{T}_{\kappa_0,h} e(x_\alpha, t_0^{\ell+1}) = \hbar' \kappa_0^{-1} e(x_\alpha, t_0^\ell) + g_0(x_\alpha, t_0^{\ell+1})(\kappa_0 + h^2), \quad \text{in } \Theta_0^\ell, \tag{2.55a}$$

$$\mathcal{T}_{\kappa_i,h} e(x_\alpha, t_i^{\ell,j}) = \hbar' \kappa_i^{-1} e(x_\alpha, t_i^{\ell,j-1}) + g_i(x_\alpha, t_i^{\ell,j})(\kappa_i + h^2), \quad \text{in } \Theta_i^\ell \backslash \partial\Theta_i^\ell, \ \ 1 \le i \le M, \tag{2.55b}$$

$$\mathcal{T}_{\kappa_i,h} e(x_\alpha, t_i^{\ell,j}) = \hbar' \kappa_i^{-1} e(x_\alpha, t_i^{\ell,j-1}) + g_i(x_\alpha, t_i^{\ell,j}) \left(\kappa_i + h + \frac{\kappa_0^2}{h^2} \right), \quad \text{on } \partial\Theta_i^\ell \ 1 \le i \le M, \tag{2.55c}$$

$$e(x_\alpha, t) = 0, \quad \text{on } \partial\Omega,\ t \ge 0, \tag{2.55d}$$

where $g_i(x_\alpha, t)$, $0 \le i \le M$, are bounded grid functions which depend on the time derivatives of $u(x,t)$ up to third order, evaluated at the point (x_α, t).

We shall provide an estimate for the error $e(x_\alpha, t^\ell)$ that takes into account the special presentation of the local truncation error in (2.55). For this purpose we introduce two types of auxiliary functions $\psi_i(x)$ and $\xi_i(x)$ in $S_h^0(\Omega)$, one pair for each subdomain Ω_i. The functions $\{\psi_i(x)\}_{i=0}^M$ are solutions to the problems

$$-\mathcal{L}_h \psi_i(x_\alpha) = \chi_i(x_\alpha), \quad x_\alpha \in \omega, \tag{2.56a}$$

$$\psi_i(x_\alpha) = 0, \qquad x_\alpha \in \partial\omega, \tag{2.56b}$$

where $\chi_i(x)$ is the discrete characteristic function of $\omega_i \backslash \partial\omega_i$ and $\chi_0(x)$ is the discrete characteristic grid function of ω_0. The functions $\{\xi_i(x)\}_{i=1}^M$ are solutions to the problems

$$-\mathcal{L}_h \xi_i(x_\alpha) = \delta_i(x_\alpha), \quad x_\alpha \in \omega, \tag{2.57a}$$

$$\xi_i(x_\alpha) = 0, \qquad x_\alpha \in \partial\omega, \tag{2.57b}$$

where $\delta_i(x)$ is the discrete characteristic function of $\partial\omega_i$.

The following lemma provides bounds for $\psi_i(x)$ and $\xi_i(x)$, defined by (2.56) and (2.57).

Lemma 2.2 *The solutions $\{\psi_i(x)\}_{i=0}^M$ and $\{\xi_i(x)\}_{i=1}^M$ to* (2.56) *and* (2.57)*, respectively, exist and are nonnegative. In addition, the following estimates hold:*

$$\|\psi_i\|_\infty \leq \begin{cases} C, & in \quad \mathbb{R}^1, \\ C|\log h|^{1/2}, & in \quad \mathbb{R}^2, \\ Ch^{-1/2}, & in \quad \mathbb{R}^3, \end{cases} \tag{2.58}$$

and

$$\|\xi_i\|_\infty \leq \begin{cases} Ch, & in \quad \mathbb{R}^1, \\ Ch|\log h|^{1/2}, & in \quad \mathbb{R}^2, \\ Ch^{1/2}, & in \quad \mathbb{R}^3. \end{cases} \tag{2.59}$$

Proof: Because of (2.49a), the solutions to (2.56) and (2.57) exist. They are nonnegative since the right hand sides and the boundary conditions are nonnegative and the stencil of the operator $\mathcal{L}_h$ complies with the requirements of the maximum principle.

Taking an inner product of both sides of (2.56a) with ψ_i and using (2.49a) yields

$$\|\psi_i\|_1^2 \leq C\|\chi_i\|_0\|\psi_i\|_0.$$

Hence,

$$\|\psi_i\|_1 \leq C_1\|\chi_i\|_0 \leq C.$$

Applying Theorem 2.3, we obtain (cf. [76])

$$\|\psi_i\|_\infty \leq D(h,n)\|\psi_i\|_1,$$

where

$$D(h,n) \quad = \quad \begin{cases} C, & \text{in } \mathbb{R}^1, \\ C|\log h|^{1/2}, & \text{in } \mathbb{R}^2, \\ Ch^{-1/2}, & \text{in } \mathbb{R}^3. \end{cases} \tag{2.60}$$

This proves (2.58).

To get a similar estimate for $\xi_i(x)$, we consider first a subdomain with a simple boundary in a domain $\Omega \subset \mathbb{R}^2$ (cf. Fig. 2.2). As indicated in Fig. 2.2, the boundary of the i-th subdomain consists of the pieces $\partial\omega_i^{(j)}$, $j = 1, \ldots, 4$, i.e.

$$\partial\omega_i = \bigcup_{j=1}^{4} \partial\omega_i^{(j)}.$$

With each boundary region $\partial\omega_i^{(j)}$ we associate a grid function $\gamma_i^{(j)}$ such that

$$\gamma_i^{(j)} = \begin{cases} 1, & \text{in } \left(\omega_i \cup \Lambda_i^{(j)}\right) \setminus \partial\Omega \\ 0, & \text{elsewhere in } \Omega, \end{cases} \quad , \qquad j = 1,2,3,4.$$

The regions $\Lambda_i^{(j)}$ correspond to the rectangular area enclosed by the end points of j-th piece of $\partial\omega_i$ and and $\partial\Omega$ (cf. Fig. 2.2). Then, the grid representation of δ_i is given by

$$\delta_i = h_1\bar{\partial}^{(1)}\gamma_i^{(1)} + h_2\bar{\partial}^{(2)}\gamma_i^{(2)} - h_1\partial^{(1)}\gamma_i^{(3)} - h_2\partial^{(2)}\gamma_i^{(4)}.$$

Therefore, taking an inner product of (2.57a) with $\xi_i(x)$, using summation by parts and (2.49a), we get

$$\begin{aligned} \|\xi_i\|_1^2 &\leq C\,\langle\delta_i, \xi_i\rangle \\ &= C\left(-\left\langle h_1\gamma_i^{(1)}, \partial^{(1)}\xi_i\right\rangle - \left\langle h_2\gamma_i^{(2)}, \partial^{(2)}\xi_i\right\rangle + \left\langle h_1\gamma_i^{(3)}, \bar{\partial}^{(1)}\xi_i\right\rangle + \left\langle h_2\gamma_i^{(4)}, \bar{\partial}^{(2)}\xi_i\right\rangle\right) \\ &\leq Ch\|\xi_i\|_1. \end{aligned}$$

Clearly, the same argument holds for subdomains with more complex boundaries in $\mathbb{R}^n$, $n = 2, 3$, provided that the number of times their boundaries intersect the spatial axes is bounded independent of h. Hence, by Theorem 2.3, (2.59) holds. □

The next theorem establishes error estimates for the locally refined discretization defined above.

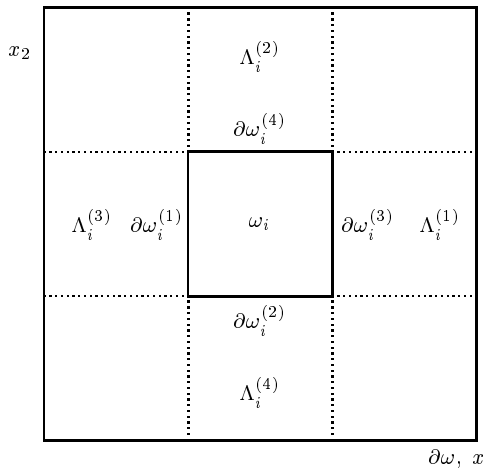


Figure 2.2: Auxiliary functions in a simple 2D case

Theorem 2.8 *Let the solution $u(x,t)$ to (2.37) be a suitably smooth function. Also let Assumption 2.1 and Assumption 2.3 hold. Then, the error of the discrete problem (2.53) satisfies:*

$$\|e\|_{\infty,h} \leq D(h,n) \sum_{i=0}^{M} \left\{ C_i(\kappa_i + h^2) + I_i \left(h\kappa_i + \frac{\kappa_0^2}{h} \right) \right\}, \tag{2.61}$$

with constants C_i and I_i independent of the discretization parameters κ_i and h, and $D(h,n)$ given by (2.60).

Proof: In view of the representation of the local truncation error (2.55), let us define

$$\eta(x) = \sum_{i=0}^{M} \psi_i(x) C_i(\kappa_i + h^2) + \sum_{i=1}^{M} \xi_i(x) I_i \left(\kappa_i + h + \frac{\kappa_0^2}{h^2} \right),$$

where

$$C_i = \max_{(x_\alpha,t)\in\Theta_i^\ell\setminus\partial\Theta_i^\ell} |g_i(x_\alpha,t)| \quad \text{and} \quad I_i = \max_{(x_\alpha,t)\in\partial\Theta_i^\ell} |g_i(x_\alpha,t)|.$$

By induction over ℓ, it is easy to observe that

$$\hbar'\bar{\partial}_{t_0} \left(\eta(x_\alpha) - e(x_\alpha, t_0^{\ell+1}) \right) - \mathcal{L}_h \left(\eta(x_\alpha) - e(x_\alpha, t_0^{\ell+1}) \right) \geq 0, \qquad \text{in } \Theta_0^\ell,$$

and

$$\hbar'\bar{\partial}_{t_i} \left(\eta(x_\alpha) - e(x_\alpha, t_i^{\ell,j}) \right) - \mathcal{L}_h \left(\eta(x_\alpha) - e(x_\alpha, t_i^{\ell,j}) \right) \geq 0, \qquad \text{in } \Theta_i^\ell,$$

for $i = 1, \ldots, M$. Here $\bar{\partial}_{t_i}$ denotes the backward difference in time with respect to the time-step κ_i.

In addition,

$$\left(\eta(x_\alpha) - e(x_\alpha, t) \right) \Big|_{\partial\Omega} \geq 0, \quad \text{for all} \quad t \geq 0.$$

Hence, by Theorem 2.6, we get

$$e(x_\alpha, t_i^{\ell,m_i}) \leq \eta(x_\alpha), \qquad \text{in } \Theta_i^\ell \quad \text{for } i = 0, \ldots, M.$$

Repeating the same argument for $\eta(x_\alpha) + e(x_\alpha, t)$ yields

$$\left| e(x_\alpha, t_i^{\ell,m_i}) \right| \leq \eta(x_\alpha), \qquad \text{in } \Theta_i^\ell \quad \text{for } i = 0, \ldots, M.$$

Combining the last inequality with the result of Lemma 2.2 proves (2.61). □

Remark 2.7 *The structure of the error estimate in* (2.61) *shows that the scheme properly takes into account the local characteristics of the solution, thus providing a better error control over the time-space region $\mathcal{Q}$. A constant interpolation in time along the interfaces of the refined regions can be applied too, but this will lead to error estimates much worse than* (2.61). *It is also possible to derive estimates for the error in the $L^2(\Omega)$-norm using the fact that in $S_h^0(\Omega)$ the norm $\|\cdot\|_{\infty,h}$ dominates $\|\cdot\|_{0,h}$.*

Remark 2.8 *According to* (2.61), *in the case of one- or two-dimensional problems for $\kappa_0 = h$ our scheme is of optimal or almost optimal order. In the case of three dimensional problems, one can improve the error estimate* (2.61) *using a discrete full elliptic regularity assumption on $\mathcal{L}_h$ (cf. [5, 63]).*

Remark 2.9 *The technique used in the proof of Theorem 2.8 allows the analysis of the temporal part to be separated from the analysis of the stationary part, thus expanding the scope of our method to a variety of different problems. We shall use this fact in the next section to investigate schemes with refinement in time and space.*

Remark 2.10 *Our approach generalizes to more complex parabolic problems with variable but suitably smooth coefficients and elliptic part given by*

$$\mathcal{L}u = \nabla \cdot (A\nabla u) + \mathbf{r}(x,t) \cdot \nabla u - c(x,t)u,$$

with

$$\begin{aligned} A &= diag(a_1(x,t) \dots a_n(x,t)), \quad 0 < a^{-1} \le a_i(x,t) \le a, \\ \mathbf{r}(x,t) &= (r_1(x,t) \dots r_n(x,t))^T, \qquad c(x,t) \ge 0. \end{aligned}$$

We refer to [52] where a detailed consideration of such problems can be found.

Remark 2.11 *It is clear from the proof of Theorem 2.8 that the locations of the refined regions Ω_i need not be fixed for all t in the interval $[0,T]$. In fact, at every time level t_0^ℓ the regions where local time stepping is performed or the degree of refinement may change. Thus, adaptive grid refinement based on* a posteriori *error estimation can be easily incorporated in our method to help prevent highly varying local phenomena from crossing the interfaces of the refined regions which is a potential source of larger approximation error.*

2.4.5 Composite grids with refinement in time and space

In this section we extend our analysis to schemes with local refinement in time and space. Since all of the ideas developed in Section 2.4.4 carry over with a minor modification we shall only sketch the main arguments.

In terms of the notations used in the previous sections, the refinement in space is to be introduced in the subdomains Ω_i, $i > 0$, for the solution of (2.37). This constitutes space-time subregions with locally refined grids in time and space. In many possible applications (cf. [48]), such schemes are more interesting because the space and time local refinement techniques result in very efficient numerical approximations within a given error tolerance. Generally speaking, the utilization of local refinement in space along with local refinement in time is a much more difficult problem. The difficulties here greatly depend on the dimension of $\mathbb{R}^n$ and the treatment of the interface nodes. For example, in $\mathbb{R}^1$, the local refinement in space does not involve any difficulties. Moreover, the arguments used in the previous section can be applied directly here, resulting in an estimate similar to (2.61).

In higher spatial dimensions, local refinement in space can be introduced similarly to the approach in Section 2.4.2. The two-dimensional example in Fig. 2.3 shows that in order to define our scheme at the interface points, we use linear interpolation in space and time. As noted above, this interpolation produces schemes that comply with the maximum principle. Thus, the values at the T-slave nodes are obtained by linear interpolation in time between the corresponding nodes in Θ. The values at the S-slave nodes are obtained by linear interpolation in space between the corresponding spatial positions

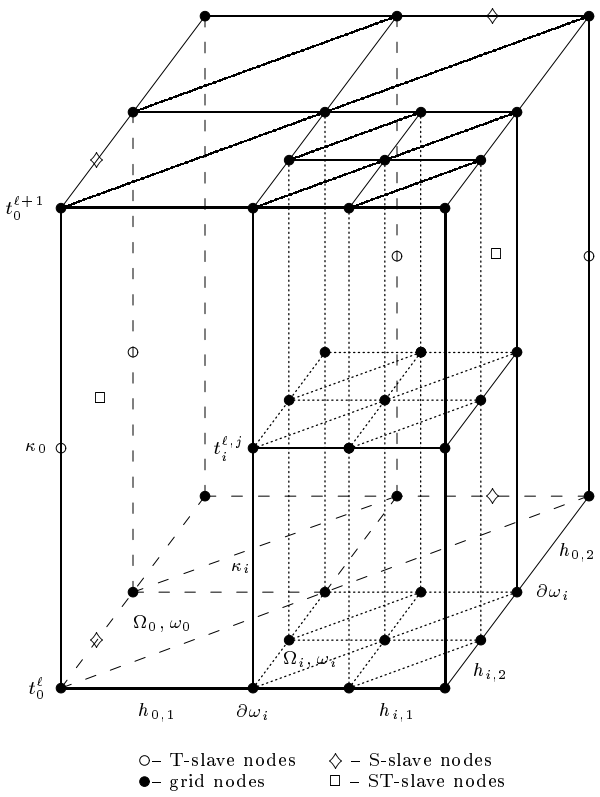


Figure 2.3: A fragment of a grid refined in space and time

in Θ using the spatial analog of (2.52). Further, the values at ST-slave nodes are obtained by linear interpolation in time between the corresponding S-slave nodes.

It has been proven by many authors (cf. [31, 64, 72, 74]) that the corresponding spatial grid operator $\tilde{\mathcal{L}}_h$ is $H_0^1(\Omega)$–coercive. As we already observed in Remark 2.9, the analysis of the spatial and temporal parts of the composite grid operator can be separated, which allows us to combine these coercivity results with the estimates for the local truncation error. Hence, using again a Taylor series expansion yields

$$\mathcal{T}_{\kappa_0,h} e(x_\alpha, t_0^{\ell+1}) = \hbar' \kappa_0^{-1} e(x_\alpha, t_0^{\ell}) + g_0(x_\alpha, t_0^{\ell+1})(\kappa_0 + h_0^2), \quad \text{in } \Theta_0^\ell, \tag{2.62a}$$

$$\mathcal{T}_{\kappa_i,h} e(x_\alpha, t_i^{\ell,j}) = \hbar' \kappa_i^{-1} e(x_\alpha, t_i^{\ell,j-1}) + g_i(x_\alpha, t_i^{\ell,j})(\kappa_i + h_i^2), \quad \text{in } \Theta_i^\ell \backslash \partial\Theta_i^\ell, \ \ 1 \le i \le M, \tag{2.62b}$$

$$\mathcal{T}_{\kappa_i,h} e(x_\alpha, t_i^{\ell,j}) = \hbar' \kappa_i^{-1} e(x_\alpha, t_i^{\ell,j-1}) + g_i(x_\alpha, t_i^{\ell,j}) \left(\kappa_i + h + \frac{\kappa_0^2 + h_0^2}{h_0^2}\right), \quad \text{on } \partial\Theta_i^\ell \ 1 \le i \le M, \tag{2.62c}$$

$$e(x_\alpha, t) = 0, \quad \text{on } \partial\Omega, \ t \ge 0, \tag{2.62d}$$

where $h_i = \max\limits_{j=1,\dots,n} h_j^{(i)}$ is the largest space discretization parameter associated with the i-th refined subdomain Ω_i.

Defining auxiliary functions by (2.56) and (2.57), it is easy to see that (because of the coercivity of $\tilde{\mathcal{L}}_h$) the estimates of Lemma 2.2 hold. Hence, constructing the function

$$\eta(x) = \sum_{i=0}^{M} \psi_i(x) C_i(\kappa_i + h_i^2) + \sum_{i=1}^{M} \xi_i(x) I_i \left(\kappa_i + h_0 + \frac{\kappa_0^2 + c(n) h_0^2}{h_0^2}\right),$$

where $c(1) = 0$, $c(2) = c(3) = 1$, yields the following theorem:

Theorem 2.9 *Let the solution $u(x,t)$ to (2.37) be a suitably smooth function. Also let Assumption 2.3 hold. Then, (2.53) defined on the time and space refined grid is stable, and the following estimates for the error hold:*

$$\|e\|_{\infty,h} \leq D(h,n) \sum_{i=0}^{M} \left\{ C_i(\kappa_i + h_i^2) + I_i \left(h_0\kappa_i + \frac{\kappa_0^2 + c(n)h_0^2}{h_0} \right) \right\}, \tag{2.63}$$

with constants C_i and I_i independent of the discretization parameters κ_i and h_i, and $D(h,n)$ given by (2.60).

Remark 2.12 *The effect of the spatial interpolation for $n = 2, 3$ is clearly visible in (2.63). Thus, keeping the interfaces at a reasonable distance from the regions with strong local behavior will help make the interfaces invisible for the discretization in terms of the overall error. It is also worth mentioning that one may consider space-time regions with local refinement where the boundaries of the subregions refined in space do not necessarily coincide with the boundaries of the subregions refined in time. Our analysis covers such cases as well.*

Remark 2.13 *It is possible to apply the local refinement ideas developed above to mixed finite element discretizations. The guiding observation here is that mixed approximations with the lowest order Raviart–Thomas spaces on rectangular meshes in $\mathbb{R}^n$ can be reduced to cell-centered finite difference approximation with a $(2n+1)$-point stencil which provide the same accuracy. This is achieved using special quadrature rules for integration [87, 97]. The analysis of locally refined cell-centered finite difference schemes is a straightforward extension of our method. The refined scheme however is not guaranteed to be conservative along the interfaces between the refined and unrefined regions.*

2.4.6 Numerical investigation of discretizations with local refinement

There are two main goals we want to achieve with the presentation of numerical experiments involving discretizations with local refinement. The first goal in testing the properties of the proposed schemes is to understand their behavior in terms of stability and accuracy. We focus our attention primarily on the effects of local time stepping in order to investigate the influence of the interfaces and the intermediate time steps in the refined regions on the stability and the convergence properties. The second goal is to experiment with discretizations that are not covered by the theory but which are very attractive in terms of accuracy. We shall consider locally refined schemes based on the Crank–Nicholson time-stepping procedure (cf. [90]).

Investigation of the backward Euler discretizations

As the structure of the error in (2.61) and (2.63) suggests, the most delicate places, which determine to a great extent the accuracy of the scheme, are the interfaces of the refined regions. From this point of view, one can easily conclude that if the solution does not change much near the interfaces, the contribution of the interfacial terms to the total error of the scheme should be negligible. On the other hand, if the solution changes substantially near the interfaces, the interfacial error will govern the total approximation error. This is the basis for setting up the experiments described below*.

First, we experimentally assess the properties of the constant and linear interpolations in time for 1D problems. The differential problem solved is (2.37) and the coefficients and the triangulation comply with Assumptions 2.1–2.3. In this case we use local refinement in space as well, which does not introduce any additional interfacial error—there is no interpolation in space. The model problem we started with is the heat equation with constant coefficients. The following function is used as an exact solution:

$$u(x,t) = \exp(20t^{-2}) \exp(-37x^2 + 66x - 30), \tag{2.64}$$

Table 2.1: Backward Euler with $\kappa_0 = h_0^2$

	Constant		Linear	
h_0^{-1}	Max-norm	Reduction	Max-norm	Reduction
20	1.716e-3		6.912e-7	
40	0.885e-3	1.94	1.938e-7	3.57
80	0.450e-3	1.96	0.499e-7	3.88
160	0.226e-3	1.99	0.125e-7	3.97

Table 2.2: Backward Euler with $\kappa_0 = h_0^{3/2}$

	Constant		Linear	
h_0^{-1}	Max-norm	Reduction	Max-norm	Reduction
16	1.004e-2		3.390e-4	
64	0.465e-2	2.15	0.486e-4	6.97
256	0.190e-2	2.44	0.063e-4	7.62
1024	0.073e-2	2.58	0.008e-4	7.82

which represents a bump with maximum around $x = 0.75$. In the interval $[0, 1/2]$, this function is close to zero, changing negligibly in time. On the contrary, in the interval $[1/2, 1]$, it changes substantially in time. It is therefore useful for simulating real problems with local behavior.

The first series of experiments of the refined region is $(0.3, 1)$. The refinement in time uses the factors of 4, 6, etc. In this case, the scheme behaves as a finite difference scheme on a regular grid with error $O(\kappa_1 + h_1^2)$, where κ_1 and h_1 are the discretization parameters in the refined region.

The next, more interesting set of experiments is when the refined region is $(0.75, 1)$, where the solution changes substantially in time. In practice, one does not use local refinement with interfaces crossing the very place where the local phenomena is observed. However, this is a way to test the properties of the scheme in the so-called "worst case". In other words, this will resemble the case when the local process approaches the boundaries of the refined region, which could happen in many applications. Moreover, as (2.61) and (2.63) suggest, the behavior of the scheme in this case is governed by the interfacial terms of the error, which is a good test to distinguish the properties of the different interpolation in time used at the interface. In the following, h_0 and κ_0 denote the discretization parameters of the coarse region. In Tables 2.1, 2.2, and 2.3 the results from the experiments with different relations between κ_0 and h_0 are presented. In all experiments, the ratio κ_0/κ_1 is held equal to 4.

The results of the experiments are shown on the basis of the comparison of two interpolations in time—piecewise constant and piecewise linear. These results show the fact that the treatment of the

Table 2.3: Backward Euler with $\kappa_0 = h_0$

	Constant		Linear	
h_0^{-1}	Max-norm	Reduction	Max-norm	Reduction
64	1.351e-2		4.244e-4	
128	1.081e-2	1.25	2.148e-4	1.97
256	0.831e-2	1.30	1.077e-4	1.99
512	0.622e-2	1.34	0.539e-4	2.00
1024	0.457e-2	1.36	0.269e-4	2.00

interface is of crucial importance for the convergence of the scheme. They also indicate the advantages of linear interpolation. In fact, keeping $\kappa_0 = h_0^2$ results in a second-order accurate scheme when linear interpolation is used (see Table 2.1). This ratio between κ_0 and h_0 is very close to the CFL condition and for this reason is not satisfactory for implicit time stepping. For this reason we have experimented with $\kappa_0 = h_0^{3/2}$, which is much less restricitve than the CFL condition. The experimental results are shown in Table 2.2. According to the theoretical estimates (2.61) and (2.63), reducing h_0 four times should result in an error reduction of a factor of two when constant interpolation is used and a corresponding factor of eight when linear interpolation is used. Thus, a scheme with a linear interpolation in time has the same accuracy as a regular backward Euler scheme even if $\kappa_0 = h_0^{3/2}$. To avoid confusion we should emphasize that these tests show the behavior of the scheme in the worst case. In principle, the locally refined discretizations capture the local behavior of the solution very acurately and are much more efficient than regular schemes. The results from our final experiment when $\kappa_0 = h_0$ are shown in Table 2.3. In this case we should obesrve $O(h_0)$ error for the scheme with linear interpolation, i.e. the same as the error of a regular scheme. We have to point out that in the case of $h_0 = \kappa_0$ shown in Table 2.3, the scheme with constant interpolation in time performs better than expected, because the theory predicts $O(1)$ error in such a case. In fact, we have made experiments with other exact solutions, for instance $\sin(2\pi x)\sin(2\pi t)$, where the asymptotic behavior is consistent with the theory. Experimental results with ratio κ_0/κ_1 equal to 8, 10, 16 show the same asymptotic behavior with smaller maximum norms of the error.

We might point out that all error estimates are in maximum norms, which are known to be much more demanding than many other norms, e.g. $L^2(\Omega)$. However, for truly local solutions, the schemes developed in Section 2.4.3 and Section 2.4.5 prove to be very efficient.

Numerical experiments with Crank–Nicholson type discretizations

Our goal now is to understand the behavior of discretizations with local refinement in time when the Crank–Nicholson time stepping is used (cf. [90]). From an implementation point of view, such schemes are minor modifications of backward Euler schemes. However, for classical schemes, Crank–Nicholson time stepping results in second order accuracy of the time discretization, which is a very attractive feature. We have combined such discretizations with the local time-stepping approach described above. Obviously, the resulting scheme does not comply with the requirements of the maximum principle and therefore the approach to the analysis from Section 2.4.4 cannot be used here.

We have tested the properties of the various Crank–Nicholson schemes for 1D problems. The differential problem solved is (2.37) and the coefficients and the triangulation comply with Assumptions 2.1–2.3. We have performed experiments[†] with two exact solutions: the function in (2.64) and the function given by

$$u(x,t) = \sin(2\pi x)\sin(2\pi t). \tag{2.65}$$

Clearly, (2.65) is a global function in space and time, i.e. does not exhibit local behavior in $[0,\,1] \times t$.

In view of the higher accuracy of the Crank–Nicholson schemes, we have experimented with two types of interpolation in time along the interfaces, namely linear and quadratic. To test the stability of the resulting schemes, we have performed tests with up to 5120 coarse time steps with various ratios between the coarse and fine time steps. The schemes showed unconditional stability in all test cases.

More interesting are the experiments designed to test the accuracy of the new schemes obtained. We try to determine experimentally the lowest degree α, such that if κ_0 is equal to h^α, the accuracy of the resulting scheme is still $O(h_0^2)$ in the worst case.

We begin with a series of experiments based on the exact solution given by (2.65). There is one refined region located in the interval $[3/4,\,1]$. The results reported in Tables 2.4, 2.5, and 2.6 show errors measured in three different norms: the maximum norm in space and time, computed by taking the maximum of the error at the coarse time levels in space and taking the maximum of these in time (denoted as $L^\infty(L^\infty(\Omega))$–norm); the standard discrete $L^\infty(L^2(\Omega))$–norm, i.e. discrete $L^2(\Omega)$–norm in

[†]Portions reprinted from [51] with kind permission of Elsevier Science – NL, Sara Burgerhartstraat 25, 1055 KV Amsterdam, The Netherlands.

Table 2.4: Crank–Nicholson with linear interpolation and $\kappa_0 = h_0^{3/2}$

h_0^{-1}	$L^\infty(L^\infty(\Omega))$	Reduction	$L^\infty(L^2(\Omega))$	Reduction	$L^\infty(H^1(\Omega))$	Reduction
16	4.9854e-5		8.4884e-6		1.1074e-5	
64	3.5393e-6	14.08	3.0151e-7	28.15	4.7779e-7	23.17
256	2.2828e-7	15.50	9.7245e-9	31.00	2.3499e-8	20.33
1024	1.4379e-8	15.87	3.0629e-10	31.74	1.3408e-9	17.52

Table 2.5: Crank–Nicholson with quadratic interpolation and $\kappa_0 = h_0$

h_0^{-1}	$L^\infty(L^\infty(\Omega))$	Reduction	$L^\infty(L^2(\Omega))$	Reduction	$L^\infty(H^1(\Omega))$	Reduction
100	4.3493e-6		2.9150e-7		1.0700e-6	
200	1.0933e-6	3.97	5.1699e-8	5.63	2.5499e-7	4.19
400	2.7408e-7	3.98	9.1533e-9	5.64	6.2110e-8	4.10
800	6.8615e-8	8.99	1.6193e-9	5.65	1.5318e-8	4.05

space at the coarse time levels and the maximum of these in time; and finally the discrete $L^\infty(H^1(\Omega))$ computed according to [53].

The experimental results for a scheme with linear interpolation in time along the refined interface and $\kappa_0 = h_0^{3/2}$ are shown in Table 2.4. The ratio κ_0/κ_1 is equal to 4. No refinement in space is utilized.

As can be seen in Table 2.4, the asymptotic accuracy of the scheme is $O(h_0^2)$ in both $L^\infty(L^\infty(\Omega))$– and $L^\infty(H^1(\Omega))$–norms and $O(h_0^{5/2})$ in the $L^\infty(L^2(\Omega))$–norm. This suggests that the asymtotic error behavior of this scheme is governed by $O(\kappa_0^2 + h_0^2 + \kappa_0^\beta/h_0)$ with $\beta \geq 2$. The indication of the possibility of superconvergence properties is very interesting as well. Obviously, keeping $\kappa_0 = h_0^{3/2}$ results in overall accuracy $O(h_0^2)$.

The next experiment is with a Crank–Nicholson scheme with quadratic interpolation along the interface and $\kappa_0 = h_0$. The results are shown in Table 2.5. Again, an $O(h_0^2)$ behavior of the error is observed in the $L^\infty(L^\infty(\Omega))$– and $L^\infty(H^1(\Omega))$–norms and the presence of superconvergence is indicated in the $L^\infty(L^2(\Omega))$–norm. The results from this experiment suggest that the asymptotic error behavior is governed by $O(\kappa^2 + h_0^2 + \kappa^\beta/h_0)$ with $\beta \geq 3$.

The behavior of a scheme with linear interpolation in time and $\kappa_0 = h_0$ is different. The results from such an experiment are shown in Table 2.6. Apparently, the degree β in the term κ^β/h_0 in the hypothesis for the error bound in the case of linear interpolation in time along the interface cannot be greater than 2.

The next series of experiments is designed to reveal the importance of the location of the interface on the overall accuracy of the discretization. We present four different cases involving schemes with linear and quadratic interpolation in time along the interface. The exact solution was the function in (2.64). The interface locations are chosen to be 0.3 and 0.75. The changes of the exact solution with respect to the time variable around $x = 0.3$ are negligible. On the other hand, this function changes significantly

Table 2.6: Crank–Nicholson with linear interpolation and $\kappa_0 = h_0$

h_0^{-1}	$L^\infty(L^\infty(\Omega))$	Reduction	$L^\infty(L^2(\Omega))$	Reduction	$L^\infty(H^1(\Omega))$	Reduction
100	3.2483e-5		1.4640e-6		5.8456e-6	
200	1.6031e-5	2.02	4.9457e-7	2.96	2.6443e-6	2.21
400	7.9602e-6	2.01	1.7073e-7	2.89	1.2524e-6	2.11
800	3.9658e-6	2.00	5.9630e-8	2.86	6.0873e-7	2.05

Table 2.7: Crank–Nicholson with linear interpolation and $\kappa_0 = h_0$. Interface at $x = 0.3$

h_0^{-1}	$L^\infty(L^\infty(\Omega))$	Reduction	$L^\infty(L^2(\Omega))$	Reduction	$L^\infty(H^1(\Omega))$	Reduction
100	2.0170e-3		7.0188e-5		1.8882e-4	
200	5.0376e-4	4.00	1.2397e-5	5.66	3.2768e-5	5.76
400	1.2591e-4	4.00	2.1910e-6	5.65	5.7526e-6	5.69
800	3.1479e-5	3.99	3.8731e-7	5.65	1.0195e-6	5.64

Table 2.8: Crank–Nicholson with quadratic interpolation and $\kappa_0 = h_0$. Interface at $x = 0.3$

h_0^{-1}	$L^\infty(L^\infty(\Omega))$	Reduction	$L^\infty(L^2(\Omega))$	Reduction	$L^\infty(H^1(\Omega))$	Reduction
100	2.0169e-3		7.0187e-5		1.8881e-4	
200	5.0376e-4	4.00	1.2396e-5	5.66	3.2768e-5	5.76
400	1.2590e-4	4.00	2.1910e-6	5.65	5.7519e-6	5.69
800	3.1478e-5	3.99	3.8730e-7	5.65	1.0192e-6	5.64

in time in the region around $x = 0.75$. The effects of these differences are seen clearly in Tables 2.7–2.10.

The experimental results when the interface is located at $x = 0.3$ are shown in Tables 2.7 and 2.8. Obviously, when the interface is located in an area of negligible changes in time of the solution, both the linear and quadratic perform equally well resulting in error behavior $O(h_0^2)$.

The behavior of the schemes changes drastically when the interface is located at $x = 0.75$. The corresponding results are shown in Tables 2.9 and 2.10. The quadratic interpolation in time helps preserve the overall accuracy to $O(h_0^2)$ whereas the linear interpolation is insufficient for this and results in error $O(h_0)$. Even though the quadratic interpolation in higher dimensions would require much more memory to store the data needed for calculating the interpolant, the resulting scheme is much more accurate.

We pointed out above that maximum principle arguments cannot be used for the analysis of the Crank–Nicholson schemes with local refinement. In fact, no estimates of the error of these schemes in the natural norms mentioned above is known to the author of this thesis. Based on the exhausting experimentation with the Crank–Nicholson scheme we conclude this section with two conjectures concerning the stability and the error of such schemes.

Conjecture 2.1 *Crank–Nicholson schemes with local refinement in time and space are unconditionally stable when linear or quadratic interpolation in time along the interfaces are used.*

Conjecture 2.2 *If the solution of the corresponding differential problem is suitably smooth, the error e*

Table 2.9: Crank–Nicholson with linear interpolation and $\kappa_0 = h_0$. Interface at $x = 0.75$

h_0^{-1}	$L^\infty(L^\infty(\Omega))$	Reduction	$L^\infty(L^2(\Omega))$	Reduction	$L^\infty(H^1(\Omega))$	Reduction
100	1.3446e-2		6.2969e-4		2.5889e-3	
200	7.2318e-3	1.85	2.3026e-4	2.73	1.2977e-3	1.99
400	3.7461e-3	1.93	8.2897e-5	2.77	6.4467e-4	2.01
800	1.9050e-3	1.96	2.9582e-5	2.80	3.2169e-4	2.00

Table 2.10: Crank–Nicholson with quadratic interpolation and $\kappa_0 = h_0$. Interface at $x = 0.75$

h_0^{-1}	$L^\infty(L^\infty(\Omega))$	Reduction	$L^\infty(L^2(\Omega))$	Reduction	$L^\infty(H^1(\Omega))$	Reduction
100	2.6230e-3		1.0586e-4		3.8062e-4	
200	6.6375e-4	3.95	1.9184e-5	5.51	9.1295e-5	4.16
400	1.6697e-4	3.97	3.4366e-6	5.58	2.2366e-5	4.08
800	4.1886e-5	3.98	6.1183e-7	5.61	5.5410e-6	4.03

of the schemes with linear interpolation in time along the interfaces can be bounded by

$$\|e\| \leq C \left(\kappa_0^2 + h_0^2 + \frac{\kappa_0^2}{h_0} \right).$$

Correspondingly, if quadratic interpolation in time is applied then the error e can by bounded by

$$\|e\| \leq C \left(\kappa_0^2 + h_0^2 + \frac{\kappa_0^3}{h_0} \right).$$

The constant C above is independent of the discretization parameters and $\|\cdot\|$ is some appropriate norm.

Chapter 3

Iterative methods for second-order problems

The development of efficient iterative methods for solving systems of linear equations arising from finite element discretizations of second-order partial differential equations has been a very active area of mathematical research over the last few decades. Even though the advantages of iterative algorithms were seen by Gauss almost two centuries ago, their dominance over the classical direct methods was established only recently. Today, the success of the finite element method depends to a large extent on the existence of fast iterative techniques for solving the corresponding discrete problems.

In Chapter 2 we considered two major types of finite element discretizations, namely Galerkin and mixed, which lead to quite different discrete systems. Typically, Galerkin discretizations result in symmetric and positive definite but ill-conditioned systems. The main emphasis in this case is naturally given to development of efficient preconditioners. On the other hand, mixed discretizations lead to indefinite systems which are much more difficult for iterative solution. Both new iterative methods and efficient preconditioners for such systems are needed in order to achieve the desired effectiveness.

This chapter is central for the dissertation. Here new, very efficient iterative algorithms and preconditioners for the iterative solution of linear systems arising from the discretizations considered in Chapter 2 are developed and analyzed. The chapter is organized as follows. First, we shall consider some basic facts from the theory of iterative methods. Next, in Section 3.2 we construct and analyze new nonoverlapping domain decomposition preconditioners. We show that the new methods exhibit very good condition numbers and provide robust and efficient preconditioners for Galerkin and mixed discretizations of elliptic and parabolic equations as well as discretizations with local refinement. In Section 3.3 we consider the class of inexact Uzawa algorithms for solving saddle point problems. In Section 3.3.1 we provide new analysis of inexact variants of the Uzawa algorithm that are much more effective than the classical Uzawa algorithm. Most of the theory developed in this chapter is a result of a joint research with Bramble and Pasciak [23, 24].

3.1 Preconditioned iterative methods

Let us consider the problem of finding the solution to the system of equations

$$\mathbf{A}x = f, \tag{3.1}$$

where $\mathbf{A}$ is a linear, symmetric and positive definite (SPD) operator on a finite dimensional real space S with inner product $(\cdot,\cdot)$ and dimension N, with f given and x unknown. We shall consider this problem in the context of the operators $\mathbf{A}$ induced by bilinear forms defined on finite element spaces (cf. Chapter 2, (2.16) and (2.41)); i.e. there exists a symmetric and positive definite bilinear form $\mathcal{A}(\cdot,\cdot)$ on $S \times S$ such that

$$(\mathbf{A}\varphi, \psi) = \mathcal{A}(\varphi, \psi), \quad \text{for all} \quad \varphi,\, \psi \in S.$$

It is well known from linear algebra considerations that such operators $\mathbf{A}$ have real and positive eigenvalues $\{\lambda_i\}_{i=1}^N$ and a corresponding complete orthonormal set of eigenvectors $\{\varphi_i\}_{i=1}^N$. Thus, every element $v \in S$ has a unique representation in the basis $\{\varphi_i\}_{i=1}^N$ given by

$$v = \sum_{i=1}^{N} (v, \varphi_i)\varphi_i.$$

Let $\mathbf{B}$ be another linear SPD operator on S. Given initial guess $x_0 \in S$, we define the basic iterative method for solving (3.1) by

$$x_{i+1} = x_i - \mathbf{B}(\mathbf{A}x_i - f). \tag{3.2}$$

This iteration is linear because both $\mathbf{A}$ and $\mathbf{B}$ are linear operators. It is an easy observation that the solution x to (3.1) is a fixed point for (3.2).

In order to motivate the discussion and introduce appropriate terminology we begin with the simple case of $\mathbf{B} = \tau\mathbf{I}$, where τ is an appropriately chosen real number. As we shall see below, the choice of τ is crucial for the convergence of this method. It is instructive to look at the error $e_i = x - x_i$. It satisfies the equation

$$e_i = e_{i-1} - \tau\mathbf{A}e_{i-1} = (\mathbf{I} - \tau\mathbf{A})^i e_0. \tag{3.3}$$

Obviously, $e_i \in S$ and hence,

$$(\mathbf{I} - \tau\mathbf{A})^i e_0 = \sum_{j=1}^{N} (e_0, \varphi_j)(1 - \tau\lambda_j)^i \varphi_j.$$

Using the fact that $\{\varphi_i\}_{i=1}^N$ is an orthonormal basis for S, we obtain

$$\begin{aligned} \|(\mathbf{I} - \tau\mathbf{A})^i e_0\|^2 &= \sum_{j=1}^{N} (e_0, \varphi_j)^2 (1 - \tau\lambda_j)^{2i} \\ &\leq \max_{\lambda_j \in \sigma(\mathbf{A})} |1 - \tau\lambda_j|^{2i} \|e_0\|, \end{aligned} \tag{3.4}$$

where $\|\cdot\|^2 = (\cdot, \cdot)$ and $\sigma(\mathbf{A})$ is the spectrum of $\mathbf{A}$. From (3.4) we obtain

$$\|(\mathbf{I} - \tau\mathbf{A})\| \leq \max_{\lambda_j \in \sigma(\mathbf{A})} |1 - \tau\lambda_j|.$$

Clearly, for convergence we must have $\lambda_1 \leq \tau \leq \lambda_N$ with the best choice being $\tau = 2/(\lambda_1 + \lambda_N)$.

Definition 3.1 *The condition number $K(\mathbf{A})$ of the SPD linear operator $\mathbf{A}$ is given by*

$$K(\mathbf{A}) = \frac{\lambda_N}{\lambda_1}.$$

Obviously $K(\mathbf{A}) \geq 1$. Moreover, for the best choice of τ we have

$$\rho = \max_{\lambda_j \in \sigma(\mathbf{A})} |1 - \tau\lambda_j| = \frac{K(\mathbf{A}) - 1}{K(\mathbf{A}) + 1}.$$

This relationship between the spectral radius ρ of $(\mathbf{I} - \tau\mathbf{A})$ and $K(\mathbf{A})$ plays an essential role in designing efficient iterative methods for solving systems of equations arising from Galerkin discretizations of elliptic and parabolic equations. It is well known (cf. [11]) that the bilinear form $\mathcal{A}(\cdot, \cdot)$ defining the finite element solution $u_h \in S_h^0(\Omega)$ by (2.14) satisfies

$$ch^2(\varphi, \varphi)_\Omega \leq \mathcal{A}(\varphi, \varphi) \leq C(\varphi, \varphi)_\Omega, \quad \text{for all} \quad \varphi \in S_h^0(\Omega), \tag{3.5}$$

where $(\cdot,\cdot)_\Omega$ is the standard $L^2(\Omega)$ inner product. In view of (2.15), (3.5) implies $K(\mathbf{A}) = O(h^{-2})$, where $\mathbf{A}$ is the operator induced by $\mathcal{A}(\cdot,\cdot)$. Because of (2.17), it is evident that finite element discretizations eventually lead to very ill-conditioned operators $\mathbf{A}$ which in turn means very slowly convergent iteration. The latter comes from the fact that in general the approximation error $u_h - u$ is guaranteed to be small when h is small. This discussion motivates a more general iteration than the one with $\mathbf{B} = \tau\mathbf{I}$. Now let $\mathbf{B} : S \mapsto S$ be a linear operator defined by

$$(\mathbf{B}\varphi_i, \varphi_j) = \mathcal{B}(\varphi_i, \varphi_j), \quad \text{for all} \quad \varphi_i,\, \varphi_j \in S_h^0(\Omega), \tag{3.6}$$

where $\mathcal{B}(\cdot,\cdot)$ is some symmetric and positive definite bilinear form on $S \times S$. Let us define a new inner product in S by

$$[u, v] = (\mathbf{A}u, v), \quad \text{for all} \quad u,\, v \in S.$$

Then the operator $\mathbf{B}^{-1}\mathbf{A}$ is symmetric and positive definite with respect to $[\cdot,\cdot]$. Moreover, it has positive eigenvalues $\mu_1 \le \mu_2 \le \ldots \mu_N$ and a corresponding complete set of orthonormal vectors $\{\psi_i\}_{i=1}^N$. Considering now the iteration

$$x_{i+1} = x_i - \tau\mathbf{B}^{-1}(\mathbf{A}x_i - f), \tag{3.7}$$

in a very similar fashion as before, with the appropriate choice of τ, we obtain the spectral radius ρ of $(\mathbf{I} - \mathbf{B}^{-1}\mathbf{A})$ given by

$$\rho = \frac{K(\mathbf{B}^{-1}\mathbf{A}) - 1}{K(\mathbf{B}^{-1}\mathbf{A}) + 1}.$$

Clearly, if $\mathbf{B} = \mathbf{A}$ then $\rho = 0$ and the iteration (3.2) converges in one step. This, however, is not a good choice for $\mathbf{B}$ since in our context $\mathbf{A}$ is very large, and inverting it directly is prohibitively expensive. Ideally, we would like to have $\mathbf{B}$ "close" to $\mathbf{A}$, but the evaluation of $\mathbf{B}^{-1}$ should be proportional to the evaluation of the action of $\mathbf{A}$. In the context of our considerations we give the following definition.

Definition 3.2 *The symmetric and positive definite bilinear form $\mathcal{B}(\cdot,\cdot)$ is a* good *preconditioner for $\mathcal{A}(\cdot,\cdot)$ if it satisfies the following two basic requirements. First, the solution W of*

$$\mathcal{B}(W, \varphi) = (g, \varphi) \quad \textit{for all} \quad \varphi \in S, \tag{3.8}$$

with $g \in S$ given, should be much easier to compute than the solution of

$$\mathcal{A}(W, \varphi) = (g, \varphi), \quad \textit{for all} \quad \varphi \in S.$$

Second, the two forms should be equivalent in the sense that

$$\gamma_1\mathcal{B}(V, V) \le \mathcal{A}(V, V) \le \gamma_2\mathcal{B}(V, V) \quad \textit{for all} \quad V \in S, \tag{3.9}$$

for some positive constants γ_1 and γ_2 with γ_2/γ_1 not too large.

Notice that this definition implies that $K(\mathbf{B}^{-1}\mathbf{A})$ is small and the action of $\mathbf{B}^{-1}$ is easy to compute.

Another important iterative method for solving the linear system of equations (3.1) is the *preconditioned conjugate gradient* (PCG). This is a nonlinear algorithm which may be described briefly as follows. The i-th PCG iterate is determined by means of projections onto the Krylov subspace V_i generated by the operator $\mathbf{B}^{-1}\mathbf{A}$ and the starting guess x_0 (cf. (3.94)). We refer the reader to [11] where an excellent presentation of this method can be found. Here we shall only point out some important facts about PCG. If no round-off errors are present in the calculations, PCG obtains the solution to (3.1) in N steps. This, however, is not the important property of the method that makes it useful. What is really important is that if $x_0 = 0$ then the error satisfies

$$\mathcal{A}(\mathbf{B}^{-1}\mathbf{A}(x - x_i), (x - x_i)) \le 2\rho^{2i}\mathcal{A}(x, x),$$

where

$$\rho = \frac{K(\mathbf{B}^{-1}\mathbf{A})^{1/2} - 1}{K(\mathbf{B}^{-1}\mathbf{A})^{1/2} + 1}.$$

Thus, when N is very large, the error may be acceptably small for some $i < \|\| < N$. Even though PCG is by its nature a direct method for solving (3.1), its real power is as an iterative algorithm. Moreover, the convergence rate of PCG is better than the rate of the linear iteration (3.7) since

$$\frac{K(\mathbf{B}^{-1}\mathbf{A})^{1/2} - 1}{K(\mathbf{B}^{-1}\mathbf{A})^{1/2} + 1} \leq \frac{K(\mathbf{B}^{-1}\mathbf{A}) - 1}{K(\mathbf{B}^{-1}\mathbf{A}) + 1}.$$

We note that PCG is a parameter-free algorithm in contrast to the linear iteration (3.7).

It is clear now that the construction of good preconditioners is an important element of the development of efficient iterative techniques for solving linear SPD systems arising from Galerkin discretizations of second-order problems.

3.2 Domain decomposition preconditioners

Developing preconditioners by way of domain decomposition has become a classical technique in numerical analysis. The first domain decomposition algorithm applied to the solution of partial differential equations was suggested by Schwarz [91]. Recently, such algorithms have become increasingly popular because they take full advantage of modern parallel computing technology.

There are two basic approaches to the development of domain decomposition preconditioners. The first is the so-called nonoverlapping approach and is characterized by the need to solve subproblems on disjoint subdomains. Early work was applicable to domains partitioned into subdomains without internal crosspoints [19], [9], [40]. To handle the case of crosspoints, Bramble, Pasciak and Schatz introduced in [18] algorithms involving a coarse grid problem and provided analytic techniques for estimating the conditioning of the domain decomposition boundary preconditioner, a central issue in the subject. Various extensions of these ideas were provided in [43] including a Neumann-Dirichlet checkerboard-like preconditioner. Subsequently, these techniques were extended to problems in three dimensions in [22] and [41]. A critical ingredient in the three-dimensional algorithms was a coarse grid problem involving the solution averages developed in [20]. Related work is contained in [38], [78], [92].

The papers [19], [18], [20], [21], and [22] developed domain decomposition preconditioners for the original discrete system. The alternative approach, to reduce to an iteration involving only the unknowns on the boundary, was taken in [9], [38], [92] and [28]. The difference in the two techniques is important in that for the first, it is at least feasible to consider replacing the subproblem solves by preconditioning sweeps.

The second approach for developing domain decomposition preconditioners involves the solution of subproblems on overlapping subdomains. For such methods it is always possible to replace the subproblem solution with a preconditioning evaluation [25]. However, in parallel implementations, the amount of inter-processor communication is proportional to the amount of overlap. These methods lose some efficiency as the overlap becomes smaller [44]. Theoretically, they are much worse in the case when there are jumps in coefficients (cf. Remark 3.4 below). In contrast, the convergence estimates for correctly designed nonoverlapping domain decomposition algorithms are the same as those for smooth coefficients as long as the jumps align with subdomain boundaries.

Thus, it is natural to investigate the effect of inexact solves on nonoverlapping domain decomposition algorithms. Early computational results showing that inexact nonoverlapping algorithms can perform well were reported in [59]. References to other experimental work can be found in [42]. Analysis and numerical experiments with inexact algorithms of Neumann–Dirichlet and Dirichlet types under the additional assumption of high accuracy of the inexact solves were given in [10] and [62]. Their analysis suggests that the inexact preconditioners do not, in general, preserve the asymptotic condition number behavior of the corresponding exact method, even when the forms providing the inexact interior solves are uniformly equivalent to the original.

In this section, we construct and analyze new inexact nonoverlapping domain decomposition preconditioners which are variations of the exact algorithm considered in [20]. The algorithms are developed based only on the assumption that the interior solves are provided by uniform preconditioning forms. The inexact methods exhibit the same asymptotic condition number growth as the one in [20] and are

much more efficient computationally. The new preconditioners are alternatives to and in many applications less restrictive than the preconditioners in [10] and [62]. The convergence estimates developed here are independent of jumps of the operator coefficients across subdomain boundaries.

3.2.1 Preliminaries

In this section we construct a decomposition of the domain Ω in which the model elliptic problem (2.6) is posed. Correspondingly, we define appropriate finite element spaces and introduce appropriate notation.

We consider the Galerkin finite element discretization defined in Section 2.2.2. To define a decomposition of Ω, by convention, any union of elements τ_j^h in a given triangulation will be called a mesh subdomain. For the purposes of the domain decomposition theory, Ω is assumed partitioned into n_d mesh subdomains $\{\Omega_k\}_{k=1}^{n_d}$. The notation Ω_k will be used for the set of all points of a subdomain including the boundary $\partial\Omega_k$.

We now define finite element spaces. Let $S_h^0(\Omega)$ be the space of continuous piecewise linear functions from Section 2.2.2. Correspondingly, $S_h^0(\Omega_k)$ will be the space of functions whose support is contained in Ω_k and hence vanish on $\partial\Omega_k$. $S_h(\Omega_k)$ will consist of restrictions to Ω_k of functions in $S_h^0(\Omega)$. Let Γ denote $\bigcup_k \partial\Omega_k$ and let $S_h(\Gamma)$ and $S_h(\partial\Omega_k)$ be the spaces of functions that are restrictions to Γ and $\partial\Omega_k$ of functions in $S_h^0(\Omega)$. We consider piecewise linear functions for convenience since the results and algorithms to be developed extend to higher-order elements without difficulty.

The following additional notation will be used. Let the $L^2(\partial\Omega_k)$–inner product be denoted by

$$\langle u, v\rangle_{\partial\Omega_k} = \int_{\partial\Omega_k} uv \; ds$$

and the corresponding norm by

$$|v|_{\partial\Omega_k} = \langle v, v\rangle_{\partial\Omega_k}^{1/2}.$$

On $S_h(\partial\Omega_k)$, the discrete inner product and norm are defined by

$$\langle u, v\rangle_{\partial\Omega_k,h} = h^{n-1} \sum_{x_i \in \partial\Omega_k} u(x_i)v(x_i)$$

and

$$|v|_{\partial\Omega_k,h} = \langle v, v\rangle_{\partial\Omega_k,h}^{1/2}.$$

We remind the reader that x_i is used to denote the grid points in the discretization of Ω.

Finally, $\mathcal{D}_k(\cdot,\cdot)$ denotes the Dirichlet inner product on Ω_k defined by

$$\mathcal{D}_k(v, w) = \sum_{i=1}^{n} \int_\Omega \partial_i v \partial_i w \; dx, \quad \text{for all} \quad v,\, w \in H^1(\Omega_k). \tag{3.10}$$

The development of a method for efficient iterative solution of (2.14) is the subject of our considerations in this section. In particular, using the decomposition of Ω described above, we shall define a bilinear form $\mathcal{B}(\cdot,\cdot)$ on $S_h^0(\Omega) \times S_h^0(\Omega)$ which is a *good* preconditioner for $\mathcal{A}(\cdot,\cdot)$.

The classical nonoverlapping domain decomposition preconditioners are easily understood from the matrix point of view. In this case, one orders the unknowns so that the stiffness matrix corresponding to $\mathcal{A}(\cdot,\cdot)$ can be written in a block form as

$$\begin{pmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{pmatrix}.$$

Here $\mathbf{A}_{22}$ corresponds to the nodes on Γ and $\mathbf{A}_{11}$ to the remaining nodes. With this ordering, the form corresponding to a typical domain decomposition preconditioner (e.g., [18],[20],[21], [22]) has a stiffness matrix of the form

$$\hat{\mathbf{A}} = \begin{pmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{Z} \end{pmatrix},$$

where $\mathbf{Z} = \mathbf{B}_{22} + \mathbf{A}_{21}\mathbf{A}_{11}^{-1}\mathbf{A}_{12}$ and $\mathbf{B}_{22}$ is the domain decomposition boundary preconditioning matrix. Inverting $\hat{\mathbf{A}}$ is a three step block Gaussian elimination procedure.

The classical inexact method is defined by replacing $\mathbf{A}_{11}$ with $\mathbf{B}_{11}$ where $\mathbf{B}_{11}$ is another symmetric and positive definite matrix. This defines a new preconditioning operator $\mathbf{B}$ given by

$$\mathbf{B} = \begin{pmatrix} \mathbf{B}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \tilde{\mathbf{Z}} \end{pmatrix}.$$

Here $\tilde{\mathbf{Z}}$ is given by $\tilde{\mathbf{Z}} = \mathbf{B}_{22} + \mathbf{A}_{21}\mathbf{B}_{11}^{-1}\mathbf{A}_{12}$.

Generally, the inexact algorithm may not converge as well as the exact version. Even if one takes $\mathbf{B}_{22}$ to be the Schur complement, $\mathbf{A}_{22} - \mathbf{A}_{21}\mathbf{B}_{11}^{-1}\mathbf{A}_{12}$, the inexact preconditioner may perform poorly unless the difference between the two matrices $\mathbf{B}_{11}$ and $\mathbf{A}_{11}$ is sufficiently small in an appropriate sense (see Theorem 3.2).

3.2.2 A nonoverlapping inexact domain decomposition preconditioner and its analysis

We now construct an inexact nonoverlapping domain decomposition preconditioner and prove an estimate for the condition number of the preconditioned system. We also show that our preconditioner is of additive Schwarz type with appropriately defined subspace decomposition.

The preconditioner

To define our domain decomposition preconditioner, we will need boundary extension operators. For each k, let us define linear extension operators $\mathcal{E}_k : S_h(\partial\Omega_k) \to S_h(\Omega_k)$ by

$$\mathcal{E}_k\phi(x_i) = \begin{cases} \phi(x_i) & \text{for } x_i \in \partial\Omega_k, \\ 0 & \text{for } x_i \in \Omega_k \setminus \partial\Omega_k. \end{cases}$$

We remind the reader that the functions in the finite element spaces defined above are fully determined by their values at the grid nodes and thus it is sufficient to define the extensions $\mathcal{E}_k$ at the nodal points x_i. Also, $\mathcal{E}_k$ can be viewed as a linear operator $S_h^0(\Omega) \to S_h^0(\Omega)$ with a trivial modification of the above definition, namely

$$\mathcal{E}_k\phi(x_i) = \begin{cases} \phi(x_i) & \text{for } x_i \in \partial\Omega_k, \\ 0 & \text{for } x_i \in \Omega \setminus \partial\Omega_k. \end{cases}$$

We shall use $\mathcal{E}_k$ in both contexts since it will be easy to determine which is the right one from the functions $\mathcal{E}_k$ is applied to.

Similarly, let $\mathcal{E} : S_h^0(\Omega) \mapsto S_h^0(\Omega)$ be defined by

$$\mathcal{E}\phi(x_i) = \begin{cases} \phi(x_i) & \text{for } x_i \in \Gamma, \\ 0 & \text{for } x_i \in \Omega \setminus \Gamma. \end{cases} \tag{3.11}$$

For each k, let $\mathcal{B}_k(\cdot,\cdot)$ be a bilinear form on $S_h^0(\Omega_k) \times S_h^0(\Omega_k)$ which is uniformly equivalent to $\mathcal{A}_k(\cdot,\cdot)$. By this we mean that for each k there are constants c_k and C_k with C_k/c_k bounded independently of h and d such that

$$c_k\mathcal{B}_k(V,V) \le \mathcal{A}_k(V,V) \le C_k\mathcal{B}_k(V,V), \quad \text{for all} \quad V \in S_h^0(\Omega_k). \tag{3.12}$$

The preconditioning form is given by

$$\begin{aligned} \mathcal{B}(U,V) = &\sum_{k=1}^{n_d} \mathcal{B}_k(U - \bar{U}_k - \mathcal{E}_k(U - \bar{U}_k), V - \bar{V}_k - \mathcal{E}_k(V - \bar{V}_k)) \\ &+ h^{-1}\sum_{k=1}^{n_d} \tilde{a}_k\langle U - \bar{U}_k, V - \bar{V}_k\rangle_{\partial\Omega_k,h}. \end{aligned} \tag{3.13}$$

Here, $\bar{U}_k$ denotes the discrete mean value of U on $\partial\Omega_k$, i.e.,

$$\bar{U}_k \equiv \frac{\langle U, 1\rangle_{\partial\Omega_k,h}}{\langle 1, 1\rangle_{\partial\Omega_k,h}}.$$

In (3.13), $\tilde{a}_k,\ k = 1, \ldots, n_d$ are parameters. For example, if $\tilde{a}_k$ is taken to be the smallest eigenvalue of $\{a_{i,j}\}$ at some point $x \in \Omega_k$ then

$$C_k^{-1}\tilde{a}_k\mathcal{D}_k(v,v) \le \mathcal{A}_k(v,v) \le C_k\tilde{a}_k\mathcal{D}_k(v,v), \quad \text{for all} \quad v \in S_h(\Omega_k), \tag{3.14}$$

where C_k depends only on the local variation of the coefficients $\{a_{ij}\}$ on the subdomain Ω_k. Consequently, we will assume that (3.14) holds with C_k/c_k bounded independently of d, h, and k.

Analysis of the preconditioning form $\mathcal{B}(\cdot,\cdot)$

Let us introduce some standard assumptions about the domain Ω, the subdomain splitting and the associated finite element spaces which are needed for the analysis. We remind the reader that unless explicitly indicated, we shall use c and C to denote generic positive constants independent of discretization parameters such as h, d, and subdomain index k. The actual values of these constants will not necessarily be the same in any two instances.

Assumption 3.1 *The collection $\{\Omega_k\}$ is quasi-uniform of size d.*

Assumption 3.2 *For every subdomain Ω_k, the inequality*

$$|u|^2_{\partial\Omega_k} \le C\{\epsilon^{-1}\|u\|^2_{\Omega_k} + \epsilon\mathcal{D}_k(u,u)\}, \tag{3.15}$$

holds for any ϵ in $(0, d]$.

Assumption 3.3 *A Poincaré inequality of the form*

$$\|v\|^2_{\Omega_k} \le Cd^2\mathcal{D}_k(v,v) \tag{3.16}$$

holds for functions v with zero mean value on Ω_k.

Remark 3.1 *Assumption 3.2 and Assumption 3.3 are satisfied for the vast majority of problems that occur in practice. For example, if all Ω_k are uniformly star-shaped with respect to a point then* (3.15) *holds. By definition, a domain Ω_k has the star-shape property if there is a point $\hat{x}_k \in \Omega_k$ and a constant $c_k > 0$ such that $(x - \hat{x}_k)\cdot\mathbf{n}(x) \ge c_k d$ for all $x \in \partial\Omega_k$. The uniform property here means that $c_k \ge c$ for some constant c not depending on d, k or h. Here $\mathbf{n}(x)$ denotes the outward unit normal vector to $\partial\Omega_k$ at point x. In addition,* (3.16) *holds for subdomains with uniformly Lipschitz continuous boundaries which is the case of polyhedral star-shaped domains in $\mathbb{R}^n$.*

Because of the mesh quasi-uniformity, the norm equivalence

$$c\,|v|^2_{\partial\Omega_k} \le |v|^2_{\partial\Omega_k,h} \le C\,|v|^2_{\partial\Omega_k} \tag{3.17}$$

holds for function $v \in S_h(\partial\Omega_k)$.

The following lemma will be used in the derivation of our results.

Lemma 3.1 *If $v \in S_h(\Omega_k)$ and vanishes at all interior nodes of Ω_k then*

$$\mathcal{D}_k(v,v) \le Ch^{-1}\,|v|^2_{\partial\Omega_k,h}\,. \tag{3.18}$$

This lemma is obvious from the local properties of the functions in finite element spaces and we shall omit its proof.

The following theorem establishes bounds for the asymptotic behavior of the preconditioner $\mathcal{B}(\cdot,\cdot)$.

Theorem 3.1 *Let $\mathcal{A}(\cdot,\cdot)$ and $\mathcal{B}(\cdot,\cdot)$ be given by (2.8) and (3.13), respectively. Then there exist positive constants c and C not depending on d or h such that*

$$c\mathcal{A}(V,V) \le \mathcal{B}(V,V) \le C\frac{d}{h}\mathcal{A}(V,V), \tag{3.19}$$

for all $V \in S_h^0(\Omega)$.

Proof: Because of the uniform equivalence of $\mathcal{A}_k(\cdot,\cdot)$ and $\mathcal{B}_k(\cdot,\cdot)$ according to (3.12), it suffices to prove the theorem for a preconditioner $\mathcal{BB}_k(\cdot,\cdot)$ defined by

$$\begin{aligned}\mathcal{B}(U,V) = &\sum_{k=1}^{n_d} \mathcal{A}_k(U-\bar{U}_k-\mathcal{E}_k(U-\bar{U}_k), V-\bar{V}_k-\mathcal{E}_k(V-\bar{V}_k)) \\ &+ h^{-1}\sum_{k=1}^{n_d} \tilde{a}_k \langle U-\bar{U}_k, V-\bar{V}_k\rangle_{\partial\Omega_k,h}.\end{aligned} \tag{3.20}$$

We first prove the left inequality in (3.19). A straightforward calculation that uses the arithmetic-geometric mean inequality shows that for any constant α we have

$$\begin{aligned}\frac{1}{2}\mathcal{A}_k(V,V) &= \frac{1}{2}\mathcal{A}_k(V-\alpha, V-\alpha) \\ &\le \mathcal{A}_k(V-\alpha-\mathcal{E}_k(V-\alpha), V-\alpha-\mathcal{E}_k(V-\alpha)) \\ &\quad + \mathcal{A}_k(\mathcal{E}_k(V-\alpha), \mathcal{E}_k(V-\alpha)).\end{aligned} \tag{3.21}$$

The left inequality in (3.19) is a simple consequence of Lemma 3.18, (3.14), (3.21), and the definition of $\mathcal{E}_k$.

In order to prove the right inequality, we apply the arithmetic-geometric mean inequality to the terms in the first sum in the definition of $\mathcal{B}(\cdot,\cdot)$ and get

$$\begin{aligned}\mathcal{B}(V,V) \le &2\mathcal{A}(V,V) + 2\sum_{k=1}^{n_d} \mathcal{A}_k(\mathcal{E}_k(V-\bar{V}_k), \mathcal{E}_k(V-\bar{V}_k)) \\ &+ h^{-1}\sum_{k=1}^{n_d} \tilde{a}_k \langle V-\bar{V}_k, V-\bar{V}_k\rangle_{\partial\Omega_k,h}.\end{aligned} \tag{3.22}$$

By (3.14) and Lemma 3.18, we obtain

$$\mathcal{B}(V,V) \le 2\mathcal{A}(V,V) + Ch^{-1}\sum_{k=1}^{n_d} \tilde{a}_k \langle V-\bar{V}_k, V-\bar{V}_k\rangle_{\partial\Omega_k,h}. \tag{3.23}$$

Let $\bar{\bar{V}}_k$ be the mean value of V on Ω_k. Using the definition of $\bar{V}_k$ and the Cauchy–Schwarz inequality yields

$$\begin{aligned}\langle V-\bar{V}_k, V-\bar{V}_k\rangle_{\partial\Omega_k,h} &= \langle V-\bar{V}_k, V-\bar{\bar{V}}_k\rangle_{\partial\Omega_k,h} \\ &\le \left|V-\bar{V}_k\right|_{\partial\Omega_k,h} \left|V-\bar{\bar{V}}_k\right|_{\partial\Omega_k,h}.\end{aligned}$$

Thus,

$$\langle V-\bar{V}_k, V-\bar{V}_k\rangle_{\partial\Omega_k,h} \le \langle V-\bar{\bar{V}}_k, V-\bar{\bar{V}}_k\rangle_{\partial\Omega_k,h}.$$

We combine the above inequality with (3.17) and obtain

$$\left|V-\bar{V}_k\right|^2_{\partial\Omega_k,h} \le C\left|V-\bar{\bar{V}}_k\right|^2_{\partial\Omega_k}.$$

Applying (3.15) with $\epsilon = d$ and (3.16) to the right hand side of the last inequality gives

$$\left|V - \bar{V}_k\right|^2_{\partial\Omega_k,h} \leq Cd\mathcal{A}_k(V,V). \tag{3.24}$$

Using this estimate in (3.23) proves the right inequality in (3.19). □

Remark 3.2 *The preconditioning form $\mathcal{B}(\cdot,\cdot)$ defined above is not uniformly equivalent to $\mathcal{A}(\cdot,\cdot)$. Nevertheless, its preconditioning effect is very close to that of a uniform preconditioner for many practical problems, particularly in three spatial dimensions. The number of subdomains often equals the number of processors in a parallel implementation and it is now feasible to keep d on the order of $h^{1/2}$. Applying a conjugate gradient method preconditioned by $\mathcal{B}(\cdot,\cdot)$ for solving* (2.14) *would result in a conditioning proportional to $h^{-1/4}$. In $\mathbb{R}^3$, if Ω is the unit cube, $h = 10^{-2}$ corresponds to a very large computational problem whereas $10^{1/2} \approx 3.2$. Also, it is well known that classical overlapping domain decomposition algorithms with small overlap exhibit the same condition number growth but in contrast to our method the overlapping preconditioners are adversely sensitive to large jumps in the operator coefficients (see Remark 3.4 below).*

Remark 3.3 *The constants c and C in Theorem 3.1 depend on the local (with respect to the subdomains) behavior of the operator and the preconditioner. Clearly, one of the most influential factors on the local properties of $\mathcal{A}(\cdot,\cdot)$ and $\mathcal{B}(\cdot,\cdot)$ is the coefficient matrix $\{a_{i,j}\}|_{\Omega_k}$. In fact, the constants C_k in* (3.14) *depend on the local lower and upper bounds for the eigenvalues of $\{a_{i,j}\}|_{\Omega_k}$ and in general so do the constants c_k and C_k in* (3.12). *Therefore, in applications to problems with large jumps in the coefficients, it is desirable, if possible, to align the subdomain boundaries with the locations of the jumps. In this case the preconditioner* (3.13) *will be independent of these jumps.*

Remark 3.4 *The utilization of the averages $\bar{U}_k$ plays the role of a coarse problem especially designed to take into account cases with interior subdomains and also applications with large jumps in the operator coefficients, provided that the locations of the jumps are aligned with the subdomain boundaries. To illustrate that the role of the averages in overcoming difficulties coming from large jumps of the coefficients is essential, we consider a conventional additive Schwarz preconditioner with minimal overlap [44]. The asymptotic condition number bound provided in [44] is the same as that of our theorem in the case of smooth coefficients. However, because of the deterioration in the approximation and boundedness properties of the weighted L^2 projection into the coarse subspace [29], the condition number of the preconditioned system for the minimal overlap algorithm when $n = 3$ can only be bounded by $(d/h)^2$.*

Our preconditioner is very economical computationally. In fact, it allows the use of efficient subdomain preconditioners such as one multigrid V-cycle (cf. [11]). The use of the simple extension $\mathcal{E}$ also results in enhanced efficiency. We shall discuss the computational aspects of this algorithm in Section 3.2.6.

An additive Schwarz reformulation of the domain decomposition algorithm

A very important observation for the subsequent analysis is that the preconditioner $\mathcal{B}(\cdot,\cdot)$ can be viewed as an additive subspace correction method (cf. [26] and [98]) with judiciously chosen subspaces. Let the linear operator $\tilde{\mathcal{E}} : S_h^0(\Omega) \mapsto S_h^0(\Omega)$ be defined by

$$\tilde{\mathcal{E}}V = \mathcal{E}V + \sum_{k=1}^{n_d} (\bar{V}_k - \mathcal{E}_k\bar{V}_k).$$

Furthermore, define

$$\hat{S}_h^0(\Omega) = \left\{ v \in S_h^0(\Omega) \,|\, v = 0 \quad \text{on}\,\Gamma \right\}$$

and

$$S_\Gamma(\Omega) = \{\tilde{\mathcal{E}}v \mid v \in S_h^0(\Omega)\}.$$

Thus $\hat{S}_h^0(\Omega)$ and $S_\Gamma(\Omega)$ provide a direct sum decomposition of $S_h^0(\Omega)$.

The additive Schwarz preconditioner applied to $g \in S_h^0(\Omega)$ based on the above two spaces results in a function $Y = Y_0 + Y_\Gamma$ where $Y_0 \in \hat{S}_h^0(\Omega)$ satisfies

$$\mathcal{B}_0(Y_0, \phi) = (g, \phi), \text{ for all } \phi \in \hat{S}_h^0(\Omega) \tag{3.25}$$

and $Y_\Gamma \in S_\Gamma(\Omega)$ satisfies

$$\mathcal{B}_\Gamma(Y_\Gamma, \phi) = (g, \phi), \text{ for all } \phi \in S_\Gamma(\Omega). \tag{3.26}$$

Here $\mathcal{B}_0(\cdot,\cdot)$ and $\mathcal{B}_\Gamma(\cdot,\cdot)$ are symmetric and positive definite bilinear forms.

We shall see that the preconditioner in (3.13) is equivalent to the additive Schwarz method above when

$$\mathcal{B}_0(\varphi, \phi) = \sum_{k=1}^{n_d} \mathcal{B}_k(\varphi, \phi) \tag{3.27}$$

and

$$\mathcal{B}_\Gamma(\varphi, \phi) = h^{-1} \sum_{k=1}^{n_d} \tilde{a}_k \langle \varphi - \bar{\varphi}_k, \phi - \bar{\phi}_k \rangle_{\partial\Omega_k, h}. \tag{3.28}$$

Let W be the solution of (3.8). Then

$$\mathcal{B}(W, \varphi) = \mathcal{B}_k(W^{(k)}, \varphi) = (g, \varphi)_\Omega, \quad \text{for all} \quad \varphi \in S_h^0(\Omega_k), \tag{3.29}$$

where $W^{(k)}k \equiv W - \bar{W}_k - \mathcal{E}_k(W - \bar{W}_k)$. The function Y_0 satisfying (3.25) is given by

$$Y_0 = W - \tilde{\mathcal{E}} W \quad \text{on} \quad \Omega_k.$$

The form given by (3.28) depends only on the boundary values of φ and ϕ. Also, the function Y_Γ solving (3.26) equals the solution W on Γ. From the definition of $\tilde{\mathcal{E}}$,

$$Y_\Gamma = \tilde{\mathcal{E}} W = \mathcal{E} W + \sum_{k=1}^{n_d} (\bar{W}_k - \mathcal{E}_k \bar{W}_k).$$

Thus, the solution W of (3.8) is the result of the additive Schwarz algorithm with subspace decomposition given by $\hat{S}_h^0(\Omega)$ and $S_\Gamma(\Omega)$, with forms defined by (3.27) and (3.28). More details concerning this reasoning can be found in Section 3.2.6

3.2.3 Application to parabolic problems

Our preconditioning approach can be extended to more general bilinear forms of the type

$$\mathcal{A}(v, w) = \delta \sum_{i,j=1}^{n} \int_\Omega a_{ij} \partial_i v \partial_j w \, dx + (bv, w)_\Omega.$$

Here δ is a small parameter and b is a bounded nonnegative function on Ω. Such forms arise from implicit time-stepping numerical approximations of parabolic problems (cf. Section 2.3). In such settings δ is related to the time step and is usually small. We shall consider the case when $ch^2 \le \delta \le Cd^2$.

We define a preconditioner $\mathcal{B}(\cdot,\cdot)$ by

$$\mathcal{B}(v, w) = \sum_{k=1}^{n_d} \mathcal{B}_k(v - \mathcal{E}_k v, w - \mathcal{E}_k w) + \frac{\delta}{h} \sum_{k=1}^{n_d} \langle w, v \rangle_{\partial\Omega_k, h}, \tag{3.30}$$

where $\mathcal{B}_k(\cdot,\cdot)$ are the subdomain preconditioning forms satisfying (3.12) with C_k/c_k bounded independently of h and d. Note that the above form no longer includes the average values on the subdomain

boundaries and thus does not require the solution of an average value problem (cf. Section 3.2.6), which is essentially a special coarse grid problem.

It is easy to see that

$$\begin{aligned} \mathcal{A}(v,v) &\le 2[\mathcal{A}(v-\mathcal{E}v, v-\mathcal{E}v) + \mathcal{A}(\mathcal{E}v,\mathcal{E}v)] \\ &\le C\left\{\sum_{k=1}^{n_d}\mathcal{B}_k(v-\mathcal{E}_k v, v-\mathcal{E}_k v) + (h+\frac{\delta}{h})\sum_{k=1}^{n_d}\langle v,v\rangle_{\partial\Omega_k,h}\right\} \\ &\le C\mathcal{B}(v,v). \end{aligned} \tag{3.31}$$

Moreover, applying (3.15) gives

$$\frac{\delta}{h}\langle v,v\rangle_{\partial\Omega_k,h} \le C\frac{\delta}{h}\left(\frac{1}{\epsilon}(v,v)_{\Omega_k} + \epsilon\mathcal{D}_k(v,v)\right).$$

Choosing $\epsilon = \max(\delta^{1/2}, d)$ in the last inequality yields

$$\frac{\delta}{h}\langle v,v\rangle_{\partial\Omega_k,h} \le C\frac{\delta^{1/2}}{h}\mathcal{A}_k(v,v). \tag{3.32}$$

Using (3.32) for each k as in the proof of Theorem 3.1, we obtain

$$\mathcal{B}(v,v) \le C\frac{\delta^{1/2}}{h}\mathcal{A}(v,v). \tag{3.33}$$

Combining (3.31) and (3.33) shows that

$$c\mathcal{A}(v,v) \le \mathcal{B}(v,v) \le C\frac{\delta^{1/2}}{h}\mathcal{A}(v,v) \quad \text{for all} \quad v \in S_h^0(\Omega). \tag{3.34}$$

The resulting condition number depends on δ in a natural way. Smaller time steps correspond to better conditioning. Obviously, the preconditioner would be uniform if $\delta = h^2$ but such time stepping is too restrictive for the vast majority of applications. On the other hand, $\delta = h$ corresponds to a very reasonable time stepping scheme whose condition number is governed by $h^{-1/2}$. Again, although not uniform, such rate of growth is often acceptable in practice for reasons already mentioned.

3.2.4 Applications to parabolic problems with local refinement

In this section we address the problem for iterative solution of linear systems coming from discretizations with local refinement. We already discussed the importance of using composite grids in Section 2.4. We now turn to the question of how to solve the resulting system, namely (2.54) involving the composite-grid operator $\mathcal{T}^{[\ell,\ell+1]}$.

There are several difficulties associated with $\mathcal{T}^{[\ell,\ell+1]}$. First of all, it is nonsymmetric but has a positive definite symmetric part (cf. [50]). Problem (2.54) is solvable but is much more difficult than the standard backward Euler–Galerkin system (2.41). The matrix of $\mathcal{T}^{[\ell,\ell+1]}$ has a complicated structure which results in very involved implementations.

There are several algorithms proposed in the literature for solving composite–grid problems. The basic algorithms in this field originated from the pioneering work of Bramble, Pasciak, and Schatz in the theory of domain decomposition methods (cf. [18, 20, 21, 22]). One of the first algorithms for elliptic problems on locally-refined grids was suggested by Bramble, Ewing, Pasciak, and Schatz [12]. Related approaches are also available in the work of McCormick [72], McCormick and Thomas [74], and Ewing, Lazarov, and Vassilevski [54]. Recently, the approach of [12] was extended to parabolic problems on locally-refined grids in time and space by Ewing, Lazarov, Pasciak, and Vassilevski [50]. Due to the available good understanding of how to iteratively solve (2.54), we shall limit our considerations here to observe that the domain decomposition preconditioners defined in (3.13) and (3.48) can be used to efficiently precondition Algorithm 5.1 in [50].

Let us split the unknowns in $U^{[\ell,\ell+1]}$ from (2.54) into two categories. Let U_1 be the part of the solution for $t = t_0^{\ell+1}$ that vanishes on all spatial coarse-grid nodes in the interior of the refined regions. Obviously, if no refinement in space is utilized then U_1 vanishes in the interior of the refined regions. In other words, U_1 is the coarse-grid part of $U^{[\ell,\ell+1]}$ that vanishes in the interior of the regions where refinement is introduced. Also let U_2 be the remaining part, i.e.

$$U^{[\ell,\ell+1]} = U_1 + U_2.$$

To simplify the notation we shall use $\mathcal{T}$ instead of $\mathcal{T}^{[\ell,\ell+1]}$ in the remaining part of this section. Then the matrix that corresponds to (2.54) can be written

$$\begin{pmatrix} \mathcal{T}_{11} & \mathcal{T}_{12} \\ \mathcal{T}_{21} & \mathcal{T}_{22} \end{pmatrix} \begin{pmatrix} U_1 \\ U_2 \end{pmatrix} = \begin{pmatrix} F_1 \\ F_2 \end{pmatrix}, \tag{3.35}$$

where F_1 and F_2 are the corresponding splitting of the right hand side of (2.54).

Performing block Gaussian elimination in (3.35) gives

$$\begin{pmatrix} \mathcal{T}_{11} - \mathcal{T}_{12}\mathcal{T}_{22}^{-1}\mathcal{T}_{21} & 0 \\ \mathcal{T}_{21} & \mathcal{T}_{22} \end{pmatrix} \begin{pmatrix} U_1 \\ U_2 \end{pmatrix} = \begin{pmatrix} F_1 - \mathcal{T}_{12}\mathcal{T}_{22}^{-1}F_2 \\ F_2 \end{pmatrix}.$$

Following Algorithm 5.1 in [50], we see that the first step is to compute $\mathcal{T}_{12}\mathcal{T}_{22}^{-1}F_2$ which amounts to performing local time stepping in each region Ω_i where refinement in time is introduced with zero boundary conditions and time step κ_i, from time level t_i^1 to $t_i^{m_i}$ (essentially in the time slab between two consecutive coarse time levels). The second step of this algorithm corresponds to performing a preconditioned iteration with the system $\mathcal{T}_{11} - \mathcal{T}_{12}\mathcal{T}_{22}^{-1}\mathcal{T}_{21}$ for solving for U_1. The third and final step is back-solving for U_2 once U_1 is known.

It is pointed in [50] that in order to precondition the second step of the above algorithm it is enough to solve a system which results from a backward Euler discretization of the entire domain Ω with time-step κ_0, i.e a system of the type (2.41) with the coarsest time step. Clearly, the utilization of the preconditioner (3.30) will provide an effective way of solving the latter system.

3.2.5 Application to mixed discretizations

In this section we consider in some detail the application of the domain decomposition preconditioner (3.13) to mixed finite elements with Lagrange multipliers for problem (2.19) (cf. Section 2.2.3). We show that these preconditioners are efficient for such problems as well. It will become clear that standard nonconforming elements and even finite differences can be treated similarly. The technique for extending our results to such problems is based on an application of a method developed in [38] and for this reason we shall only sketch the idea.

Mixed methods with Lagrange multipliers for second-order elliptic problems are defined in Section 2.2.3. As is observed there, the elimination of the original variables of the mixed method results in a symmetric and positive definite bilinear form, corresponding to the multiplier system. To define a domain decomposition preconditioner we use the triangulation of Ω introduced in Section 2.2.2 and the nonconforming finite element space Λ_h^0 corresponding to Lagrange multipliers defined with respect to the triangulation (cf. Section 2.2.3). Ω is decomposed into subdomains as described in Section 3.2.1, and corresponding subspaces of Λ_h^0 with respect to this splitting are introduced in a very similar way. The main difference here is that the nodal values for the multipliers are not specified on the element vertices. For instance, for the lowest order Raviart–Thomas–Nedelec spaces the multiplier degrees of freedom are given in the middle of the sides of the triangles or at the centers of the tetrahedra faces.

The discrete problem related to (2.19) by Theorem 2.5 can be written as follows.

Find $U \in \Lambda_h^0$ such that

$$\mathcal{G}(U, \varphi) = (f, \varphi)_\Omega \quad \text{for all} \quad \varphi \in \Lambda_h^0, \tag{3.36}$$

where $\mathcal{G}(\cdot,\cdot)$ is the corresponding bilinear form on $\Lambda_h^0 \times \Lambda_h^0$. The properties of this form are well understood. In particular, it is known (cf. [38]) that there exist positive constants c and C independent

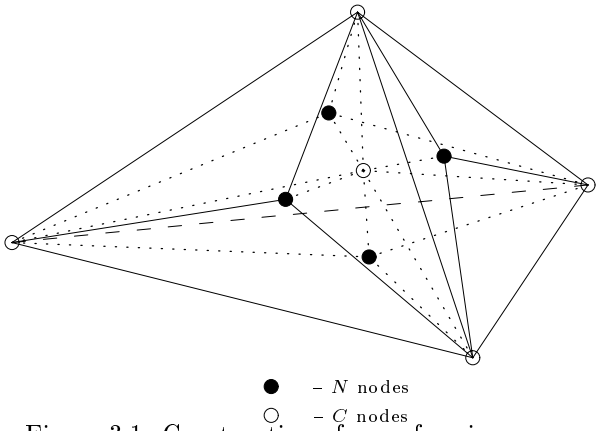


Figure 3.1: Construction of a conforming space

of the mesh size h such that

$$c\mathcal{G}(V,V) \leq \sum_{\tau} \hat{a}_{\tau} |\tau|^{1-2/n} \sum_{x_i, x_j \in \tau} (V(x_i) - V(x_j))^2 \leq C\mathcal{G}(V,V), \tag{3.37}$$

for every $V \in \Lambda_h^0$. Here x_i denotes the location of the i-th degree of freedom of V, $n = 2$ or 3, $\hat{a}_{\tau}$ is a constant over each triangle τ that depends on the operator coefficients $\{a_{ij}\}$ but is independent of h, and $|\tau|$ is the measure of τ.

Let us define a preconditioner for $\mathcal{G}(\cdot,\cdot)$ as in (3.13). We argue as in the proof of Theorem 3.1 in order to establish a similar result. Because of (3.37), it is easy to check that if a function $V \in \Lambda_h^0$ vanishes at all interior nodes of Ω_k then

$$\mathcal{G}_k(V,V) \leq Ch^{-1} |V|_{\partial\Omega_k, h}. \tag{3.38}$$

Here $\mathcal{G}_k(\cdot,\cdot)$ is the restriction of $\mathcal{G}(\cdot,\cdot)$ to Ω_k. Clearly, the left inequality in (3.19) follows immediately from the arithmetic-geometric mean inequality and (3.38).

In order to get the right inequality, it suffices to show that

$$h^{-1} \sum_{k=1}^{n_d} \tilde{a}_k \langle U - \bar{U}_k, U - \bar{U}_k \rangle_{\partial\Omega_k, h} \leq C \frac{d}{h} \mathcal{G}(U,U). \tag{3.39}$$

To this effect we apply a standard argument of equivalence between conforming and nonconforming spaces [38]. First, we note that a conforming space that is isomorphic to $\Lambda_h^0(\Omega)$ is constructed. The idea behind this construction is shown in Fig. 3.1. Each tetrahedron is split into twelve tetrahedra as indicated in Fig. 3.1. The nodes referred to as N nodes are the original nodes from the nonconforming discretization whereas C nodes are the ones added to define the conforming space. For the example in Fig. 3.1, $C - nodes$ are the vertices and the mass center of the tetrahedron. Since the original triangulation is quasi-uniform, the new triangulation is quasi-uniform too. If V_k is the restriction of $V \in \Lambda_h^0$, given in terms of its nodal values, to the k-th subdomain, we define a function $\hat{V}_k$ in the corresponding conforming space as follows:

$$\hat{V}_k(x_i) = \begin{cases} V_k(x_i), \text{ if } x_i \text{ is a } N \text{ node;} \\ \text{the average of the values of } V_k \text{ at all adjacent } N \text{ nodes,} \\ \qquad \text{if } x_i \text{ is a } C \text{ node in the interior of } \Omega_k; \\ \text{the average of all adjacent } N \text{ nodes on } \partial\Omega_k, \\ \qquad \text{if } x_i \in \partial\Omega_k \setminus \partial\Omega; \\ \text{the interpolant of } V_k \text{ in the nonconforming space,} \\ \qquad \text{if } x_i \text{ is a mass center of a tetrahedron;} \\ \text{the average of all adjacent } N \text{ nodes on } \partial\Omega, \\ \qquad \text{if } x_i \in \partial\Omega. \end{cases}$$

Let $\hat{V}_k$ be the conforming functions that correspond to V_k as a result of this construction. Then (cf. [38]),

$$c\mathcal{G}_k(V_k, V_k) \leq \mathcal{D}_k(\hat{V}_k, \hat{V}_k) \leq C\mathcal{G}_k(V_k, V_k). \tag{3.40}$$

The constants of equivalence c and C are controlled because the nonconforming triangulation is quasi-uniform. In addition, the nonconforming nodes are a subset of all nodes that are used to define the conforming space. Hence, with some abuse of notation,

$$\langle V_k, V_k\rangle_{\partial\Omega_k,h} \leq C\langle \hat{V}_k, \hat{V}_k\rangle_{\partial\Omega_k,h}, \quad \text{for each} \quad k \text{ and } V \in \Lambda_h^0. \tag{3.41}$$

Thus, (3.39) can be deduced easily by combining (3.24), (3.40), and (3.41). Therefore, the equivalent of (3.19) holds for this preconditioner too. Preconditioners for classical nonconforming problems as well as finite difference schemes can be constructed and analyzed in a similar manner.

We conclude this section by observing that the implementation of the preconditioner is based on the nonconforming discretization only and follows Algorithm 3.1.

3.2.6 Computational aspects of the preconditioning problem

In this section we provide an algorithm for inverting the preconditioning form $\mathcal{B}(\cdot,\cdot)$. It consists of two main steps: a solution of the approximate subdomain problems and an inversion of the boundary form. As we shall see, these steps are independent and can be carried out in parallel.

We outline an algorithm for inverting $\mathcal{B}(\cdot,\cdot)$. For φ in $S_h^0(\Omega_k)$, $\bar{\varphi}_k = 0$ and thus $\mathcal{E}_k\varphi \equiv 0$ for every k. Consequently,

$$\langle W - \bar{W}_k, \varphi - \bar{\varphi}_k\rangle_{\partial\Omega_k,h} \equiv 0 \quad \text{for all} \quad k.$$

Thus, (3.8) and (3.29) imply

$$\mathcal{B}(W, \varphi) = \mathcal{B}_k(W^{(k)}, \varphi) = (g, \varphi)_\Omega, \quad \text{for all} \quad \varphi \in S_h^0(\Omega_k). \tag{3.42}$$

Clearly, the computation of $W^{(k)} \in S_h^0(\Omega)$ reduces to the solution of subdomain problems which can be performed in parallel.

The second major step for inverting $\mathcal{B}(\cdot,\cdot)$ involves the inversion of a boundary form as we shall now describe. For $\psi \in S_h^0(\Omega)$ set $\varphi = \mathcal{E}\psi$. Notice that $\varphi = \mathcal{E}_k\psi$ on Ω_k, $(\overline{\mathcal{E}\psi})_k = \bar{\psi}_k$, and $\mathcal{E}_k^2\psi = \mathcal{E}_k\psi$. For this choice of φ, (3.8) becomes

$$\begin{aligned}\mathcal{B}(W, \varphi) = &\sum_{k=1}^{n_d} \mathcal{B}_k(W - \bar{W}_k - \mathcal{E}_k(W - \bar{W}_k), \mathcal{E}_k\bar{\psi}_k - \bar{\psi}_k) \\ &+ h^{-1}\sum_{k=1}^{n_d} \tilde{a}_k\langle W - \bar{W}_k, \psi\rangle_{\partial\Omega_k,h} = (g, \mathcal{E}\psi)_\Omega.\end{aligned} \tag{3.43}$$

Here we have also used the fact the $W - \bar{W}_k$ has zero mean value on $\partial\Omega_k$ and therefore is orthogonal to the constants with respect to the boundary inner product.

Since $\mathcal{E}_k\bar{\psi}_k - \bar{\psi}_k$ vanishes on $\partial\Omega_k$,

$$\sum_{k=1}^{n_d} \mathcal{B}_k(W - \bar{W}_k - \mathcal{E}_k(W - \bar{W}_k), \mathcal{E}_k\bar{\psi}_k - \bar{\psi}_k) = \sum_{k=1}^{n_d} (g, \mathcal{E}_k\bar{\psi}_k - \bar{\psi}_k)_{\Omega_k}$$

and hence

$$h^{-1}\sum_{k=1}^{n_d} \tilde{a}_k\langle W - \bar{W}_k, \psi\rangle_{\partial\Omega_k,h} = (g, \mathcal{E}\psi)_\Omega - \sum_{k=1}^{n_d} (g, \mathcal{E}_k\bar{\psi}_k - \bar{\psi}_k)_{\Omega_k}. \tag{3.44}$$

Notice that because of the explicit extensions used in the definition of $\mathcal{B}(\cdot,\cdot)$, the setup of the right hand side in (3.44) involves minimal computational cost. Clearly, this step is independent of the previous

one and thus the procedure for inverting $\mathcal{B}(\cdot,\cdot)$ decouples into two independent tasks. Once W on the interior boundaries is known then the assembly of the solution in Ω is easy. The implementation of the solution procedure for (3.44) includes two main steps. First, one determines the averages $\bar{W}_k$ and then the values of W on the interior boundary. We note that the latter step is trivial once the averages are known because of the diagonal matrix that corresponds to the discrete L^2 inner product on the subdomain boundaries. The algorithm for solving the problem for the averages was developed originally in [20] and shall not be included here.

The above discussion can be summarized in the following algorithm.

Algorithm 3.1 *Solve the preconditioning problem* (3.8) *by*

1. *Compute the solution* $W^{(k)}$ *of* (3.42) *for each* k.
2. *Compute the trace of* W *on* Γ *from* (3.44).
3. *Set the final solution to* (3.8) *by*

$$W = \mathcal{E}W + \sum_{k=1}^{n_d} (W^{(k)} + \bar{W}_k - \mathcal{E}_k \bar{W}_k).$$

3.2.7 Alternative inexact additive preconditioners

We now consider a classical technique for developing nonoverlapping domain decomposition preconditioners. The behavior of such methods has been investigated in the case when the boundary form is uniformly equivalent to the corresponding Schur complement subsystem [10], [62]. Here, we show that this method also reduces to an additive Schwarz preconditioner. In addition, we show that the inexact solve technique combined with the boundary form discussed earlier provides an effective preconditioner. Indeed, our results are much better than what would be expected from the analysis of [10], [62].

The inexact algorithm as a two level additive Schwarz procedure

We now show that the inexact preconditioners correspond to additive Schwarz methods. The first subspace in this decomposition is $\hat{S}_h^0(\Omega)$. Let $\mathcal{B}_0(\cdot,\cdot)$ be the form on $\hat{S}_h^0(\Omega) \times \hat{S}_h^0(\Omega)$ with stiffness matrix $\mathbf{B}_{11}$. The second subspace is given by

$$\begin{aligned} \hat{S}_h(\Gamma) = \Big\{ & \mathcal{E}\varphi + \varphi_0 \,|\, \varphi \in S_h^0(\Omega); \\ & \mathcal{B}_0(\varphi_0, \phi) = -\mathcal{A}(\mathcal{E}\varphi, \phi), \text{ for all } \phi \in \hat{S}_h^0(\Omega) \Big\}. \end{aligned} \tag{3.45}$$

Clearly, the functions in $\hat{S}_h(\Gamma)$ are completely determined by their traces on Γ. Let $\mathcal{B}_\Gamma(\cdot,\cdot)$ be the form on $\hat{S}_h(\Gamma) \times \hat{S}_h(\Gamma)$ with stiffness matrix $\mathbf{B}_{22}$. $\mathcal{B}_\Gamma(u,v)$ depends only on the boundary nodal values of u and v and thus naturally extends to $S_h^0(\Omega) \times S_h^0(\Omega)$.

Clearly, $\hat{S}_h^0(\Omega)$ and $\hat{S}_h(\Gamma)$ provide a direct sum decomposition of $S_h^0(\Omega)$. This decomposition is tied strongly to the bilinear form $\mathcal{B}_0(\cdot,\cdot)$. In particular, if $\mathcal{B}_0(\cdot,\cdot) \equiv \mathcal{A}(\cdot,\cdot)$ on $\hat{S}_h^0(\Omega) \times \hat{S}_h^0(\Omega)$ then the space $\hat{S}_h(\Gamma)$ consists of discrete harmonic functions and the decomposition is $\mathcal{A}(\cdot,\cdot)$–orthogonal. In general, the decomposition is not $\mathcal{A}(\cdot,\cdot)$–orthogonal.

Conditioning estimates for the inexact algorithms

The preconditioner defined by (3.2.1) can be restated as an operator $\mathbf{B} : S_h^0(\Omega) \mapsto S_h^0(\Omega)$. In fact, it is a straightforward exercise to check that it corresponds to the preconditioning operator defined in the following algorithm.

Algorithm 3.2 *Given* $g \in S_h^0(\Omega)$ *we define* $\mathbf{B}^{-1}g = U$ *where* U *is computed as follows:*

1. *Compute* $U_0 \in \hat{S}_h^0(\Omega)$ *by solving*

$$\mathcal{B}_0(U_0, \varphi) = (g, \varphi) \quad \textit{for all} \quad \varphi \in \hat{S}_h^0(\Omega). \tag{3.46}$$

2. *Compute the trace* U_Γ *on* Γ *by solving*

$$\mathcal{B}_\Gamma(U_\Gamma, \mathcal{E}\phi) = (g, \mathcal{E}\phi) - \mathcal{A}(U_0, \mathcal{E}\phi) \quad \textit{for all} \quad \phi \in \hat{S}_h(\Gamma).$$

3. *Compute* $U_{\Gamma 0}$ *by solving*

$$\mathcal{B}_0(U_{\Gamma 0}, \varphi) = -\mathcal{A}(\mathcal{E}U_\Gamma, \varphi) \quad \textit{for all} \quad \varphi \in \hat{S}_h^0(\Omega).$$

4. *Set* $U = U_0 + \mathcal{E}U_\Gamma + U_{\Gamma 0}$.

Although the above algorithm appears as a multiplicative procedure, we shall now demonstrate that it is equivalent to an additive Schwarz method. It is easy to see that the problem solved in Step 2 of Algorithm 3.2 is independent of U_0. Indeed, for any $\phi \in \hat{S}_h(\Gamma)$, we decompose $\phi = \mathcal{E}\phi + \phi_0$ as in (3.45) and observe

$$-\mathcal{A}(\mathcal{E}\phi, U_0) = \mathcal{B}(\phi_0, U_0) = (g, \phi_0).$$

Thus, Steps 2 and 3 of the above algorithm reduce to finding $U_\Gamma \in \hat{S}_h(\Gamma)$ such that

$$\mathcal{B}_\Gamma(U_\Gamma, \phi) = (g, \phi) \quad \text{for all} \quad \phi \in \hat{S}_h(\Gamma). \tag{3.47}$$

Hence, $\mathbf{B}^{-1}g = U = U_0 + U_\Gamma$ where U_0 and U_Γ satisfy (3.46) and (3.47) respectively, i.e., Algorithm 3.2 is an implementation of an additive Schwarz procedure.

Notice that Algorithm 3.2 avoids the need of knowing explicitly a basis for the space $\hat{S}_h(\Gamma)$ which could be either a computationally expensive problem or a significant complication of the overall algorithm. Obviously this procedure provides inexact variants of the methods given in [18], [20], [21], and [22].

It follows that the preconditioning form $\mathcal{B}(\cdot, \cdot)$ corresponding to the operator defined in Algorithm 3.2 is given by

$$\mathcal{B}(V, V) = \mathcal{B}_0(V_0, V_0) + \mathcal{B}_\Gamma(V_\Gamma, V_\Gamma). \tag{3.48}$$

Here $V = V_0 + V_\Gamma$ with $V_0 \in \hat{S}_h^0(\Omega)$ and $V_\Gamma \in \hat{S}_h(\Gamma)$.

In the remainder of this section we analyze the above preconditioner by providing bounds for (3.48). We take

$$\mathcal{B}_0(u, v) = \sum_{k=1}^{n_d} \mathcal{B}_k(u, v)$$

where $\mathcal{B}_k(\cdot, \cdot)$ is defined as in Section 3.2.2 (with C_k/c_k in (3.12) bounded independently of h, k, and d).

The first theorem in this section was given by Börgers [10] and Haase at al. [62] and provides a result when $\mathbf{B}_{22}$ is uniformly equivalent to the Schur complement $\mathbf{A}_{22} - \mathbf{A}_{21}\mathbf{A}_{11}^{-1}\mathbf{A}_{12}$. This is the same as assuming that the quadratic form $\mathcal{B}_\Gamma(\cdot, \cdot)$ is equivalent to the boundary form with diagonal

$$\inf_{\phi \in \hat{S}_h^0(\Omega)} A(u + \phi, u + \phi), \quad \text{for all} \quad u \in \hat{S}_h(\Gamma). \tag{3.49}$$

Theorem 3.2 *Let* $\mathcal{A}(\cdot, \cdot)$ *be given by* (2.8) *and* $\mathcal{B}(\cdot, \cdot)$ *by* (3.48). *Assume that the quadratic form* $\mathcal{B}_\Gamma(\cdot, \cdot)$ *is uniformly equivalent to the quadratic from induced by* (3.49). *In addition, let* γ *be the smallest positive constant such that*

$$|\mathcal{A}(\varphi, \varphi) - \mathcal{B}(\varphi, \varphi)| \leq \gamma \mathcal{A}(\varphi, \varphi) \quad \textit{for all} \quad \varphi \in \hat{S}_h^0(\Omega). \tag{3.50}$$

Then

$$c\left(\frac{\gamma^2}{h}\right)^{-1} \mathcal{A}(U, U) \leq \mathcal{B}(U, U) \leq C\frac{\gamma^2}{h}\mathcal{A}(U, U)$$

holds for all $U \in S_h^0(\Omega)$ *with constants* c *and* C *independent of* d *and* h.

Remark 3.5 *Condition* (3.50) *requires that* $\mathcal{B}_0(\cdot,\cdot)$ *should be a good approximation to* $\mathcal{A}(\cdot,\cdot)$ *for the preconditioner* (3.48) *to be efficient. The result of the theorem shows that if* (3.50) *holds with* γ *on the order of* $h^{1/2}$ *then the preconditioner* $\mathcal{B}(\cdot,\cdot)$ *is uniform. However, the development of a form* $\mathcal{B}_0(\cdot,\cdot)$ *satisfying* (3.50) *usually involves significant additional computational work since* γ *must tend to zero as* h *becomes small. Alternatively keeping* γ *fixed independent of* h *may result in a rather ill-conditioned method when* h *is small. However, there are examples of reasonably accurate preconditioners* $\mathcal{B}_0(\cdot,\cdot)$*, e.g. multigrid V- or W-cycles, which appear to perform well when* h *is not very small (cf. [10]) due to the fact that the corresponding* γ*'s are comparable to* $h^{1/2}$*.*

The main result of this section is given in the next theorem. It is for the case when

$$\mathcal{B}_\Gamma(u,v) = h^{-1}\sum_{k=1}^{n_d} \tilde{a}_k \langle u-\bar{u}_k, v-\bar{v}_k\rangle_{\partial\Omega_k,h}, \quad \text{for all} \quad u,\, v \in \hat{S}_h(\Gamma). \tag{3.51}$$

Theorem 3.3 *Let* $\mathcal{A}(\cdot,\cdot)$ *be given by* (2.8)*,* $\mathcal{B}(\cdot,\cdot)$ *be given by* (3.48)*, and* $\mathcal{B}_\Gamma(\cdot,\cdot)$ *defined by* (3.51)*. Then*

$$c\mathcal{A}(U,U) \le \mathcal{B}(U,U) \le C\frac{d}{h}\mathcal{A}(U,U) \tag{3.52}$$

holds for all $U \in S_h^0(\Omega)$ *with constants* c *and* C *independent of* d *and* h*.*

Remark 3.6 *The result of Theorem 3.3 shows that introducing inexact solves in the interior of the subdomains does not deteriorate the overall preconditioning effect of the corresponding exact method analyzed in [20]. As we have pointed out in Remark 3.2, the adverse effect of* h *approaching zero on the condition number can be compensated for easily by adjusting the parameter* d*. This balance is an alternative to* (3.50) *and could be a better choice when* h *is small relative to* γ*. In fact, the utilization of the bilinear form* (3.51) *leads to computationally efficient algorithms, unconstrained by accuracy conditions like* (3.50)*. The differences in the preconditioning effect of the inexact (Algorithm 3.2) and exact (cf. [20]) methods are negligible. However, the savings of computational time are significant in favor of Algorithm 3.2.*

We conclude this section with the proof of Theorem 3.3.

Proof: [of Theorem 3.3] Because of (3.48), the technique for establishing (3.52) is similar to the one used in the proof of Theorem 3.1.

Let $U_\Gamma = \mathcal{E}U_\Gamma + U_{\Gamma 0}$ as in (3.45) and write $U = U_0 + U_\Gamma$. The first inequality in (3.52) follows from the arithmetic-geometric mean inequality and the assumptions (3.12) on $\{\mathcal{B}_k(\cdot,\cdot)\}_{k=1}^{n_d}$. Indeed, we have

$$\begin{aligned} \mathcal{A}(U,U) &= \mathcal{A}(U_0+U_\Gamma, U_0+U_\Gamma) \\ &\le C\left(\mathcal{B}_0(U_0,U_0) + \mathcal{B}_0(U_{\Gamma 0},U_{\Gamma 0}) + \mathcal{A}(\mathcal{E}U_\Gamma,\mathcal{E}U_\Gamma)\right). \end{aligned} \tag{3.53}$$

If follows from the definition of $U_{\Gamma 0}$ that

$$\mathcal{B}_0(U_{\Gamma 0},U_{\Gamma 0}) \le C\mathcal{A}(\mathcal{E}U_\Gamma,\mathcal{E}U_\Gamma). \tag{3.54}$$

Using (3.54) together with (3.18) and (3.14) in (3.53) yields

$$\mathcal{A}(U,U) \le C\mathcal{B}(U,U).$$

To prove the right hand inequality in (3.52), we again use the decomposition of U. Thus,

$$\begin{aligned} \mathcal{B}_0(U_0,U_0) &\le C\mathcal{A}(U-U_\Gamma, U-U_\Gamma) \le C\left(\mathcal{A}(U,U) + \mathcal{A}(\mathcal{E}U_\Gamma,\mathcal{E}U_\Gamma)\right) \\ &\le C\left(\mathcal{A}(U,U) + \mathcal{B}_\Gamma(U_\Gamma,U_\Gamma)\right). \end{aligned} \tag{3.55}$$

Hence, we need to estimate $\mathcal{B}_\Gamma(U_\Gamma,U_\Gamma)$ from above by $\mathcal{A}(U,U)$. Applying the reasoning used to show (3.24) in (3.55) gives the desired bound. □

Table 3.1: Condition numbers with the inexact preconditioner (3.13)

h	$d = 1/3$	$d = 1/6$
1/12	21.46	8.12
1/24	55.70	23.20
1/48	131.19	59.33

3.2.8 Numerical investigation of the nonoverlapping domain decomposition algorithms

In this section we present numerical experiments involving the nonoverlapping domain decomposition preconditioners developed in Section 3.2.2 and Section 3.2.7. We report results obtained from experiments with Algorithm 3.1 and Algorithm 3.2 with boundary form given by (3.51). We tested two main aspects of these preconditioners, namely the computational efficiency of the method in terms of the condition numbers obtained, and the independence of the jumps in the operator coefficients $\{a_{ij}\}$. Comparisons between the inexact algorithms and the corresponding exact methods are included as well.

The numerical results presented in this section are applied to

$$\mathcal{L} = -\nabla \cdot a\nabla, \tag{3.56}$$

where a is a piecewise constant function in Ω and constant on each subdomain. In all experiments Ω is the unit cube in three spatial dimensions. The subdomains are obtained by subdividing Ω into regions by slicing it parallel to the coordinate axes. Here we shall consider only cases where the unit cube is split into m^3 equal sub-cubes, which implies $d = 1/m$. In the examples below, $S_h^0(\Omega)$ is the space of piecewise linear functions with respect to a uniform mesh of size h. Also, the action of one multigrid V-cycle is used as an inexact solver in the interior of the subdomains. In general, a sequence of coarser spaces is needed for the definition of a multigrid algorithm. In the simplified setting of our examples, these spaces are defined with respect to coarser discretizations of Ω_k, obtained by doubling the mesh size. The result of such a procedure is a sequence of nested meshes and spaces. The multigrid algorithm is variational and based on a trilinear finite element approximation. A nested sequence of approximation subspaces is defined by successively doubling the mesh size. For computational efficiency, the fine grid form is defined by numerical quadrature utilizing a quadrature which gives rise to a seven point operator. The operators on the coarser grids are twenty seven point and determined variationally from the fine gird operator. The analysis of variational multigrid procedures based on a fine grid operator defined by numerical quadrature can be found in [13]. Pointwise forward and backward Gauss–Seidel sweeps are used as pre- and post-smoothing iterations respectively. On the coarsest level we apply five pairs of forward and backward Gauss–Seidel sweeps. Obviously, if we have only one degree of freedom on the coarsest level, then this is equivalent to an exact solve on that level. This results in a symmetric and positive definite operator whose action provides an inexact interior solve. It is well known that the corresponding $\mathcal{B}_k(\cdot,\cdot)$ satisfies (3.12) with uniform constants c_k and C_k for each k. Also, the evaluation of the action of this operator is proportional to the number of grid points on the mesh used for the discretization of Ω_k.

The first experiment we report is intended to confirm numerically the d/h-like behavior of the condition number K, established in Theorem 3.19. We consider the model problem (2.6) with $\mathcal{L} \equiv -\Delta$. The results are presented in Table 3.1. Notice that according to our theory if d/h is fixed, the resulting condition number K should also be a constant. Such a behavior is clearly visible in the experimental results in Table 3.1.

The second experiment we report illustrates that the preconditioner defined in (3.13) is independent of large jumps in the operator coefficients. The data in Table 3.2 represent experimental results where Ω is split into $4 \times 4 \times 4$ subdomains. The coefficient a in (3.56) is defined as follows: $a_{222} = a_{333} = 10^5$; a is a constant in the interval [0.1, 21.1] for the remaining subdomains. Here a_{ijk} is the operator coefficient in the subdomain with integer coordinates i, j, k. The largest jump in the operator coefficient between two neighboring subdomains in this case is 10^6. For comparison, we have included the corresponding

Table 3.2: Condition numbers with the inexact preconditioner (3.13); $d = 1/4$

h	jumping a	$a \equiv 1$
1/12	15.71	13.87
1/24	42.94	39.79
1/48	106.76	95.38

Table 3.3: Comparison of the inexact and the exact methods; $d = 1/3$

h	K_{exact}	K-Algorithm 3.1	K-Algorithm 3.2
1/6	6.27	6.73	6.27
1/12	15.23	21.96	15.40
1/24	32.55	57.01	33.83
1/48	66.12	130.88	70.76

condition numbers for the case when $a \equiv 1$ in Ω. Clearly, the results in Table 3.2 are in good agreement with Remark 3.3.

The final numerical example which we present here is a comparison of the performance of the inexact preconditioners (3.13) and (3.48) with $\mathcal{B}_\Gamma(\cdot,\cdot)$ given by (3.51), and the exact method analyzed in [20]. The piecewise constant coefficient a in this case is defined according to the data for μ in Example 3 in [20]. We note that the condition numbers for the exact method reported in Table 3.3 are better than the ones reported in Table 4.5 in [20] due to the different scaling of the boundary form (cf. Remark 2.5, [20]). The data in Table 4.5, [20] are obtained when the boundary form is scaled by d^{-1} whereas the results in Table 3.3 are obtained when the boundary form is scaled by h^{-1}. Clearly, the exact preconditioner and the inexact method implemented by Algorithm 3.2 exhibit almost the same condition numbers, which is in good agreement with Remark 3.6. Although the condition numbers reported for these two methods are better that those for Algorithm 3.1, one application of the inexact preconditioner (3.13) requires substantially less computer time thus resulting in a computationally more efficient algorithm. For example, for mesh sizes between 1/12 and 1/48, the inexact preconditioner was more than 4.5 times faster to evaluate than the exact method which utilized the FFT method for the interior subdomain solves. The comparison was made on a SUN Sparc 20/502 workstation. Thus, for the grid with $h = 1/48$, we would have more than 3 times reduction of the computing time if a preconditioned conjugate gradient with the inexact preconditioner were applied for the reduction of the initial error by a factor of 10^6, in contrast to the same method with the exact preconditioner. A similar comparison between Algorithm 3.1 and Algorithm 3.2 suggests that Algorithm 3.1 is about 25 percent more efficient.

3.3 Iterative methods for saddle point problems

The second major part of this chapter is devoted to the theory of iterative methods for saddle point problems*. Let H_1 and H_2 be finite dimensional Hilbert spaces with inner products which we shall denote by $(\cdot,\cdot)$. There is no ambiguity even though we use the same notation for the inner products on both of these spaces since the particular inner product will be identified by the type of functions appearing. We consider the abstract saddle point problem:

$$\begin{pmatrix} \mathbf{A} & \mathbf{B}^T \\ \mathbf{B} & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \end{pmatrix} = \begin{pmatrix} F \\ G \end{pmatrix}, \tag{3.57}$$

*Portions of [23] reprinted with permission from the SIAM Journal on Numerical Analysis. Copyright by SIAM, Philadelphia, Pennsylvania. All rights reserved.

where $F \in H_1$ and $G \in H_2$ are given and $X \in H_1$ and $Y \in H_2$ are the unknowns. Here $\mathbf{A}$: $H_1 \mapsto H_1$ is assumed to be a linear, symmetric, and positive definite operator. In addition, the linear map $\mathbf{B}^T$: $H_2 \mapsto H_1$ is the adjoint of $\mathbf{B}$: $H_1 \mapsto H_2$. It is well known (cf. [4]) that this block matrix is indefinite and has both positive and negative eigenvalues. In general, it is not even invertible unless additional conditions on the spaces H_1 and H_2 are imposed (cf. (2.22) and (3.60)). Because of this the iterative solution of (3.57) is a very delicate problem and special methods should be developed.

Applying block elimination to (3.57) yields

$$\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T Y = \mathbf{B}\mathbf{A}^{-1}F - G. \tag{3.58}$$

Clearly, $\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T$ is symmetric and nonnegative, and setting $W = \mathbf{B}^T Y$ we obtain

$$(\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T Y, Y) = (\mathbf{A}^{-1/2}W, \mathbf{A}^{-1/2}W) = (Z, Z),$$

where $Z = \mathbf{A}^{-1/2}W$. Thus,

$$(Z, Z) = \sup_{V \in H_1} \frac{(Z, V)^2}{(V, V)} = \sup_{U \in H_1} \frac{(Z, U)^2}{(U, U)},$$

where $U = \mathbf{A}^{1/2}V$. Hence,

$$(\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T Y, Y) = \sup_{U \in H_1} \frac{(Y, \mathbf{B}U)^2}{(\mathbf{A}U, U)}. \tag{3.59}$$

Consequently, a necessary and sufficient condition for the unique solvability of (3.57) is that the Ladyzhenskaya–Babuška–Brezzi condition holds, i.e.

$$\sup_{U \in H_1} \frac{(V, \mathbf{B}U)^2}{(\mathbf{A}U, U)} \geq c_0 \|V\|^2 \quad \text{for all } V \in H_2\,, \tag{3.60}$$

for some positive number c_0. Here $\|\cdot\|$ denotes the norm in the space H_2 (or H_1) corresponding to the inner product $(\cdot, \cdot)$.

One could iteratively solve (3.58) for Y by conjugate gradient (or preconditioned conjugate gradient) iteration [55]. Then X is obtained by $X = \mathbf{A}^{-1}(F - \mathbf{B}^T Y)$. The Uzawa method [3] (Algorithm 3.3 below) is a particular implementation of a linear iterative method for solving (3.58). One common problem with the methods just described is that they require the evaluation of the action of the operator $\mathbf{A}^{-1}$ in each step of the iteration. For many applications, this operation is expensive and is also implemented as an iteration. The inexact Uzawa methods (Algortihm 3.5) replace the exact inverse in the Uzawa algorithm by an "incomplete" or "approximate" evaluation of $\mathbf{A}^{-1}$. These algorithms are defined in Sections 3.3.1 and 3.3.3. They were also studied in [46].

There are other general iterative techniques for solving saddle point problems of the form of (3.57), e.g., [6], [14], [16], [89]. In [14], a preconditioner for $\mathbf{A}$ is introduced and the system (3.57) is reformulated as a well conditioned symmetric and positive definite algebraic system which may be solved efficiently by applying the conjugate gradient algorithm. In [89], the authors consider the convergence properties when the minimal residual algorithm is applied to a more direct preconditioned reformulation of (3.57). Both of the above-mentioned techniques incorporate preconditioning and avoid the inversion of $\mathbf{A}$. Other interesting methods for solving (3.57) that also do not require the action of $\mathbf{A}^{-1}$ can be found in [6] and [16].

There is also a variety of application-specific techniques that depend strongly on the particular approximation spaces, geometry of the domain, etc. In the case of the mixed approximation of second-order problems, those include domain decomposition techniques [58], a reduction technique involving the use of additional Lagrange multipliers [38], as well as an indefinite preconditioner [49].

The inexact Uzawa algorithms are of interest because they are simple and have minimal computer memory requirements. This could be important in large scale scientific applications implemented for today's computing architectures. In addition, an Uzawa algorithm implemented as a double iteration can be transformed trivially into an inexact Uzawa algorithm. It is not surprising that the inexact Uzawa methods are widely used in the engineering community.

Here we present new estimates for the inexact Uzawa algorithm both in the linear and nonlinear case. In the former case, the evaluation of $\mathbf{A}^{-1}$ is replaced by the inverse of a linear preconditioner. Theorem 3.4 shows that the resulting algorithm always converges and gives bounds on the rate of convergence provided that the preconditioner is properly scaled. The inexact Uzawa algorithm in the nonlinear case replaces the evaluation of $\mathbf{A}^{-1}$ by some approximate nonlinear process. To avoid confusion we note that a nonlinear algorithm in this paper means a nonlinear iteration for solving the linear problem (3.57). Theorem 3.5 shows that the resulting algorithm converges provided that the nonlinear approximation to $\mathbf{A}^{-1}$ is suitably accurate. More restrictive results for variants of the inexact Uzawa algorithms have already appeared in the literature (cf. [46, 83]).

3.3.1 The abstract inexact Uzawa algorithm

The inexact Uzawa method when linear preconditioners are used is motivated by first considering the Uzawa iteration [3], which can be defined as follows.

Algorithm 3.3 (Uzawa) *For $X_0 \in H_1$ and $Y_0 \in H_2$ given, the sequence $\{(X_i, Y_i)\}$ is defined, for $i = 1, 2, \ldots,$ by*

$$\begin{aligned} X_{i+1} &= X_i + \mathbf{A}^{-1}\left(F - (\mathbf{A}X_i + \mathbf{B}^T Y_i)\right), \\ Y_{i+1} &= Y_i + \tau(\mathbf{B}X_{i+1} - G), \end{aligned} \tag{3.61}$$

with τ a given real number.

Let $E_i^Y = Y - Y_i$ be the iteration error generated by the above method. It is easy to show that

$$E_{i+1}^Y = (\mathbf{I} - \tau\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T)E_i^Y.$$

Let c_1 denote the largest eigenvalue of $\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T$. Then, Y_i converges to Y if τ is chosen such that

$$\rho = \max(1 - c_0\tau, c_1\tau - 1) < 1.$$

In this case, X_i and Y_i converge respectively to X and Y with a rate of convergence per step bounded by ρ.

One problem with the above method is that it may converge slowly if $\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T$ is not well conditioned. Thus, it is natural to introduce a preconditioner $\mathbf{Q}_B : H_2 \mapsto H_2$. We assume that $\mathbf{Q}_B$ is linear, symmetric and positive definite and define the preconditioned Uzawa algorithm as follows.

Algorithm 3.4 (Preconditioned Uzawa) *For $X_0 \in H_1$ and $Y_0 \in H_2$ given, the sequence $\{(X_i, Y_i)\}$ is defined, for $i = 1, 2, \ldots,$ by*

$$\begin{aligned} X_{i+1} &= X_i + \mathbf{A}^{-1}\left(F - (\mathbf{A}X_i + \mathbf{B}^T Y_i)\right), \\ Y_{i+1} &= Y_i + \mathbf{Q}_B^{-1}(\mathbf{B}X_{i+1} - G). \end{aligned} \tag{3.62}$$

For convenience of notation, we have absorbed the parameter τ into the preconditioner $\mathbf{Q}_B$. Accordingly, we assume that $\mathbf{Q}_B$ is scaled so that

$$(\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T W, W) \le (\mathbf{Q}_B W, W) \quad \text{for all} \quad W \in H_2. \tag{3.63}$$

Note that since $\mathbf{Q}_B$ is positive definite, it follows that

$$(1 - \gamma)(\mathbf{Q}_B W, W) \le (\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T W, W) \quad \text{for all} \quad W \in H_2, \tag{3.64}$$

holds for some γ in the interval $[0, 1)$. In practice, effective preconditioners satisfy (3.64) with γ bounded away from one.

If $E_i^Y = Y - Y_i$ where Y_i is generated by (3.62) then

$$E_{i+1}^Y = (\mathbf{I} - \mathbf{Q}_B^{-1}\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T)E_i^Y.$$

Clearly, $\mathbf{Q}_B^{-1}\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T$ is symmetric with respect to the inner product

$$< V, W >= (\mathbf{Q}_B V, W) \quad \text{for all} \quad V, W \in H_2.$$

Let $\|\cdot\|_{Q_B}$ denote the corresponding norm

$$\|W\|_{Q_B} =< W, W >^{1/2} .$$

Then by (3.63) and (3.64),

$$\left\|E_i^Y\right\|_{Q_B}^2 \leq \gamma^i \left\|E_0^Y\right\|_{Q_B}^2 .$$

Here and in the rest of the thesis, for a symmetric and positive definite linear operator $\mathbf{L}$ on H_j, $j = 1, 2$, $\|\cdot\|_L$ will denote the norm $(\mathbf{L}\cdot, \cdot)^{1/2}$.

One problem with the above algorithms is that they require the computation of the action of the operator $\mathbf{A}^{-1}$ at each step of the iteration. For many of the applications, this is an expensive operation which is also done iteratively. This leads to a two level iteration, an inner iteration for computing the action of $\mathbf{A}^{-1}$ coupled with the outer Uzawa iteration (3.61) or (3.62). The inexact Uzawa method replaces the action of $\mathbf{A}^{-1}$ by a preconditioner. A preconditioner $\mathbf{Q}_A$ is a linear operator $\mathbf{Q}_A : H_1 \mapsto H_1$ which is symmetric and positive definite. In practice, good preconditioners are relatively cheap to invert (cf. Definition 3.2). The inexact Uzawa algorithm is then given as follows (this algorithm was also studied in [46]).

Algorithm 3.5 (Inexact Uzawa) *For $X_0 \in H_1$ and $Y_0 \in H_2$ given, the sequence $\{(X_i, Y_i)\}$ is defined, for $i = 1, 2, \ldots,$ by*

$$\begin{aligned} X_{i+1} &= X_i + \mathbf{Q}_A^{-1}\left(F - \left(\mathbf{A}X_i + \mathbf{B}^T Y_i\right)\right), \\ Y_{i+1} &= Y_i + \mathbf{Q}_B^{-1}(\mathbf{B}X_{i+1} - G). \end{aligned} \tag{3.65}$$

One step of the inexact Uzawa algorithm involves an evaluation of each of the operators, $\mathbf{A}$, $\mathbf{B}$, $\mathbf{B}^T$, $\mathbf{Q}_A^{-1}$ and $\mathbf{Q}_B^{-1}$.

3.3.2 Analysis of the inexact Uzawa algorithm

Let us now investigate the stability and convergence rate of the inexact Uzawa algorithm defined above. The main theorem will show that the inexact Uzawa algorithm will always converge provided that the preconditioners are properly scaled. By this we mean that (3.63) holds and that

$$(\mathbf{A}W, W) < (\mathbf{Q}_A W, W) \tag{3.66}$$

for all $W \in H_1$ with $W \neq 0$. The strict inequality above will be replaced by

$$(\mathbf{A}W, W) \leq (\mathbf{Q}_A W, W) \quad \text{for all} \quad W \in H_1, \tag{3.67}$$

in a subsequent corollary.

Bounds for the rates of iterative convergence will be provided in terms of two natural parameters. The first parameter has already been defined and is the convergence factor γ (see (3.64)) for the preconditioned Uzawa algorithm. The second parameter is the rate δ at which the preconditioned iteration

$$U_{i+1} = U_i + \mathbf{Q}_A^{-1}(W - \mathbf{A}U_i)$$

converges to the solution of

$$\mathbf{A}U = W.$$

If $E_i = U - U_i$ then

$$E_{i+1} = (\mathbf{I} - \mathbf{Q}_A^{-1}\mathbf{A})E_i.$$

Clearly $\mathbf{Q}_A^{-1}\mathbf{A}$ is a symmetric operator with respect to the inner product $(\mathbf{Q}_A\cdot,\cdot)$ and hence the convergence rate δ is the largest eigenvalue of $\mathbf{I}-\mathbf{Q}_A^{-1}\mathbf{A}$. Alternatively, δ is the smallest number for which the inequality

$$(1-\delta)(\mathbf{Q}_A W, W) \le (\mathbf{A}W, W) \quad \text{for all} \quad W \in H_1 \tag{3.68}$$

is satisfied. It will sometimes be convenient to rewrite (3.68) as

$$((\mathbf{Q}_A - \mathbf{A})W, W) \le \delta(\mathbf{Q}_A W, W) \quad \text{for all} \quad W \in H_1. \tag{3.69}$$

The first convergence estimate will be provided in terms of a norm on $H_1 \times H_2$ which we shall now define. Consider the bilinear form on $H_1 \times H_2$ given by

$$\left[\begin{pmatrix} U \\ V \end{pmatrix}, \begin{pmatrix} R \\ S \end{pmatrix}\right] = ((\mathbf{Q}_A - \mathbf{A})U, R) + (\mathbf{Q}_B V, S). \tag{3.70}$$

By (3.66), $[\cdot,\cdot]$ generates a norm on $H_1 \times H_2$ which we shall denote by

$$|\!|\!| T |\!|\!| = [T,T]^{1/2}, \quad \text{for all } T \in H_1 \times H_2.$$

We can now state the main result of this section.

Theorem 3.4 *Assume that (3.63) and (3.66) hold and that γ and δ satisfy (3.64) and (3.68), respectively. Let $\{X,Y\}$ be the solution pair for (3.57), $\{X_i,Y_i\}$ be defined by the inexact Uzawa algorithm and set*

$$e_i = \begin{pmatrix} X - X_i \\ Y - Y_i \end{pmatrix}.$$

Then, for $i = 1, 2, \ldots,$

$$|\!|\!| e_i |\!|\!| \le \rho^i |\!|\!| e_0 |\!|\!|, \tag{3.71}$$

where

$$\rho = \frac{\gamma(1-\delta) + \sqrt{\gamma^2(1-\delta)^2 + 4\delta}}{2}. \tag{3.72}$$

Remark 3.7 *It is elementary to see that*

$$\rho \le 1 - \frac{1}{2}(1-\gamma)(1-\delta).$$

Thus the inexact Uzawa method converges if (3.63) and (3.66) hold. As expected, the convergence rate deteriorates as either γ or δ approach one. In addition, if δ tends to zero (thus, $\mathbf{Q}_A$ tends to $\mathbf{A}$ and the norm $|\!|\!|\cdot|\!|\!|$ tends to $\|\cdot\|_{Q_B}$) then ρ (defined by (3.72)) tends to γ, the convergence rate of the preconditioned Uzawa algorithm. In the limit, one recovers the convergence results of Algorithm 3.4.

Proof:[Theorem 3.4] We first derive a relationship between the errors e_{i+1} and e_i. The components of the corresponding errors are denoted by $E_i^X = X - X_i$ and $E_i^Y = Y - Y_i$. From (3.57) and (3.65) we see that the errors satisfy the recurrence

$$\begin{aligned} E_{i+1}^X &= \left(\mathbf{I} - \mathbf{Q}_A^{-1}\mathbf{A}\right) E_i^X - \mathbf{Q}_A^{-1}\mathbf{B}^T E_i^Y, \\ E_{i+1}^Y &= E_i^Y + \mathbf{Q}_B^{-1}\mathbf{B} E_{i+1}^X. \end{aligned} \tag{3.73}$$

Replacing E_{i+1}^X in the second equation with its expression from the first gives

$$\begin{pmatrix} E_{i+1}^X \\ E_{i+1}^Y \end{pmatrix} = \begin{pmatrix} (\mathbf{I} - \mathbf{Q}_A^{-1}\mathbf{A}) & -\mathbf{Q}_A^{-1}\mathbf{B}^T \\ \mathbf{Q}_B^{-1}\mathbf{B}\left(\mathbf{I} - \mathbf{Q}_A^{-1}\mathbf{A}\right) & (\mathbf{I} - \mathbf{Q}_B^{-1}\mathbf{B}\mathbf{Q}_A^{-1}\mathbf{B}^T) \end{pmatrix} \begin{pmatrix} E_i^X \\ E_i^Y \end{pmatrix}$$

(3.74)

$$\equiv \mathcal{M} \begin{pmatrix} E_i^X \\ E_i^Y \end{pmatrix}.$$

This can be rewritten as

$$e_{i+1} = \mathcal{M} e_i. \tag{3.75}$$

The proof of the theorem will be complete if we can show that the operator norm

$$[\![\mathcal{M}]\!] = \sup_{x \in H_1 \times H_2} \frac{[\![\mathcal{M}x]\!]}{[\![x]\!]}$$

is bounded by ρ given by (3.72).

The operator $\mathcal{M}$ can be written in the form

$$\mathcal{M} = \begin{pmatrix} -\mathbf{I} & 0 \\ 0 & \mathbf{I} \end{pmatrix} \begin{pmatrix} -(\mathbf{I} - \mathbf{Q}_A^{-1}\mathbf{A}) & \mathbf{Q}_A^{-1}\mathbf{B}^T \\ \mathbf{Q}_B^{-1}\mathbf{B}\left(\mathbf{I} - \mathbf{Q}_A^{-1}\mathbf{A}\right) & (\mathbf{I} - \mathbf{Q}_B^{-1}\mathbf{B}\mathbf{Q}_A^{-1}\mathbf{B}^T) \end{pmatrix}$$

$$\equiv \mathcal{E}\mathcal{M}_1.$$

It is straightforward to check that both $\mathcal{E}$ and $\mathcal{M}_1$ are symmetric in the $[\cdot,\cdot]$-inner product. Let $\mathcal{M}^*$ denote the adjoint of $\mathcal{M}$ with respect to $[\cdot,\cdot]$. Then we have

$$\mathcal{M}^* = (\mathcal{E}\mathcal{M}_1)^* = \mathcal{M}_1\mathcal{E}$$

and

$$\mathcal{M}^*\mathcal{M} = \mathcal{M}_1\mathcal{E}^2\mathcal{M}_1 = \mathcal{M}_1^2.$$

Consequently,

$$[\![\mathcal{M}]\!]^2 = \sup_{x \in H_1 \times H_2} \frac{[\mathcal{M}x, \mathcal{M}x]}{[x,x]} = \sup_{x \in H_1 \times H_2} \frac{[\mathcal{M}^*\mathcal{M}x, x]}{[x,x]}$$

$$= \sup_{x \in H_1 \times H_2} \frac{[\mathcal{M}_1^2 x, x]}{[x,x]} = \sup_{\lambda_i \in \sigma(\mathcal{M}_1)} |\lambda_i|^2.$$

Therefore, to estimate the norm of $\mathcal{M}$, it suffices to bound the spectrum $\sigma(\mathcal{M}_1)$ of $\mathcal{M}_1$. Since $\mathcal{M}_1$ is symmetric with respect to the $[\cdot,\cdot]$ inner product, its eigenvalues are real. We shall bound the positive and negative eigenvalues of $\mathcal{M}_1$ separately.

We first provide a bound for the positive eigenvalues of $\mathcal{M}_1$. The operator $\mathbf{I} - \mathbf{Q}_A^{-1}\mathbf{A}$ is symmetric with respect to the inner product $((\mathbf{Q}_A - \mathbf{A})\cdot,\cdot)$. Moreover, it follows from (3.66) that it is positive definite and its positive square root is well defined. Let

$$\mathcal{D} = \begin{pmatrix} \delta^{-1/2}(\mathbf{I} - \mathbf{Q}_A^{-1}\mathbf{A})^{1/2} & 0 \\ 0 & \mathbf{I} \end{pmatrix}.$$

It follows from (3.66) that $\mathcal{D}$ is invertible and from (3.68) that

$$|||\mathcal{D}x||| \leq |||x||| \quad \text{for all} \quad x \in H_1 \times H_2. \tag{3.76}$$

Let $\mathcal{N} = \mathcal{D}^{-1}\mathcal{M}_1\mathcal{D}^{-1}$. Then

$$\mathcal{N} = \begin{pmatrix} -\delta\mathbf{I} & \delta^{1/2}\mathbf{L} \\ \delta^{1/2}\mathbf{L}^* & (\mathbf{I} - \mathbf{L}^*\mathbf{L}) \end{pmatrix} \tag{3.77}$$

where $\mathbf{L} = (\mathbf{I} - \mathbf{Q}_A^{-1}\mathbf{A})^{-1/2}\mathbf{Q}_A^{-1}\mathbf{B}^T$ and $\mathbf{L}^* = \mathbf{Q}_B^{-1}\mathbf{B}(\mathbf{I} - \mathbf{Q}_A^{-1}\mathbf{A})^{1/2}$.

The largest eigenvalue λ_m of $\mathcal{M}_1$ satisfies

$$\lambda_m = \sup_{x \in H_1 \times H_2} \frac{[\mathcal{M}_1 x, x]}{[x, x]} = \sup_{x \in H_1 \times H_2} \frac{[\mathcal{N}\mathcal{D}x, \mathcal{D}x]}{[x, x]}$$

$$= \sup_{x \in H_1 \times H_2} \frac{[\mathcal{N}\mathcal{D}x, \mathcal{D}x]\,[\mathcal{D}x, \mathcal{D}x]}{[\mathcal{D}x, \mathcal{D}x]\,[x, x]} \leq \sup_{y \in H_1 \times H_2} \frac{[\mathcal{N}y, y]}{[y, y]}.$$

We used (3.76) for the last inequality above. Since both $\mathcal{D}$ and $\mathcal{M}_1$ are symmetric with respect to $[\cdot, \cdot]$, it follows that $\mathcal{N}$ is also. Consequently, λ_m is bounded by the largest eigenvalue of $\mathcal{N}$.

Let λ be a nonnegative eigenvalue of $\mathcal{N}$ with corresponding eigenvector $\{\psi_1, \psi_2\}$, i.e.,

$$\begin{aligned} -\delta\psi_1 + \delta^{1/2}\mathbf{L}\psi_2 &= \lambda\psi_1, \\ \delta^{1/2}\mathbf{L}^*\psi_1 + (\mathbf{I} - \mathbf{L}^*\mathbf{L})\psi_2 &= \lambda\psi_2. \end{aligned} \tag{3.78}$$

Eliminating ψ_1 in the above equations gives

$$-\lambda\mathbf{L}^*\mathbf{L}\psi_2 = (\lambda + \delta)(\lambda - 1)\psi_2$$

and hence

$$-\lambda < \mathbf{L}^*\mathbf{L}\psi_2, \psi_2 >= (\lambda + \delta)(\lambda - 1) < \psi_2, \psi_2 > . \tag{3.79}$$

By (3.68) and (3.64), it follows that

$$\begin{aligned} < \mathbf{L}^*\mathbf{L}\psi_2, \psi_2 > &= (\mathbf{B}\mathbf{Q}_A^{-1}\mathbf{B}^T\psi_2, \psi_2) \geq (1 - \delta)(\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T\psi_2, \psi_2) \\ &\geq (1 - \delta)(1 - \gamma) < \psi_2, \psi_2 > . \end{aligned} \tag{3.80}$$

Since $\delta > 0$ and λ is nonnegative, we see from the first equation in (3.78) that if $\psi_2 = 0$ then $\psi_1 = 0$. Consequently, ψ_2 is not equal to zero. Thus, from (3.79) and (3.80), we get

$$\lambda^2 - \lambda(1 - \delta)\gamma - \delta \leq 0$$

from which it follows that $\lambda \leq \rho$ where ρ is given by (3.72). This provides the desired bound for the positive eigenvalues of $\mathcal{M}_1$.

We next estimate the negative eigenvalues of $\mathcal{M}_1$. Let λ be a negative eigenvalue of $\mathcal{M}_1$ with corresponding eigenvector (ψ_1, ψ_2), i.e.,

$$\begin{aligned} -\left(\mathbf{I} - \mathbf{Q}_A^{-1}\mathbf{A}\right)\psi_1 + \mathbf{Q}_A^{-1}\mathbf{B}^T\psi_2 &= \lambda\psi_1, \\ \mathbf{Q}_B^{-1}\mathbf{B}\left(\mathbf{I} - \mathbf{Q}_A^{-1}\mathbf{A}\right)\psi_1 + \left(\mathbf{I} - \mathbf{Q}_B^{-1}\mathbf{B}\mathbf{Q}_A^{-1}\mathbf{B}^T\right)\psi_2 &= \lambda\psi_2. \end{aligned} \tag{3.81}$$

The first equation in (3.81) together with (3.64) imply that if $\psi_1 = 0$ then $\psi_2 = 0$. Consequently, any eigenvector must have a nonzero component ψ_1.

Multiplying the first equation of (3.81) by $\mathbf{Q}_B^{-1}\mathbf{B}$ from the left and adding it to the second one yields

$$(1-\lambda)\psi_2 = \lambda \mathbf{Q}_B^{-1}\mathbf{B}\psi_1. \tag{3.82}$$

Substituting (3.82) into the first equation of (3.81) and taking an inner product with $\mathbf{Q}_A\psi_1$ gives

$$-((1-\lambda)((1+\lambda)\mathbf{Q}_A - \mathbf{A})\psi_1, \psi_1) + \lambda(\mathbf{Q}_B^{-1}\mathbf{B}\psi_1, \mathbf{B}\psi_1) = 0,$$

which we rewrite as

$$\lambda(\mathbf{Q}_B^{-1}\mathbf{B}\psi_1, \mathbf{B}\psi_1) = ((1-\lambda^2)(\mathbf{Q}_A\psi_1, \psi_1) - (1-\lambda)(\mathbf{A}\psi_1, \psi_1). \tag{3.83}$$

For any $V \in H_1$,

$$\begin{aligned}(\mathbf{Q}_B^{-1}\mathbf{B}V, \mathbf{B}V) &= \sup_{W\in H_2} \frac{(V, \mathbf{B}^T W)^2}{(\mathbf{Q}_B W, W)} = \sup_{W\in H_2} \frac{(\mathbf{A}^{1/2}V, \mathbf{A}^{-1/2}\mathbf{B}^T W)^2}{(\mathbf{Q}_B W, W)} \\ &\le \sup_{W\in H_2} \frac{(\mathbf{A}V, V)(\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T W, W)}{(\mathbf{Q}_B W, W)} \le (\mathbf{A}V, V).\end{aligned} \tag{3.84}$$

For the last inequality above we used (3.63). Applying (3.84) to the left hand side of (3.83) and (3.69) on the right hand side of (3.83) gives

$$\lambda(\mathbf{A}\psi_1, \psi_1) \le (\delta - \lambda^2)(\mathbf{Q}_A\psi_1, \psi_1) + \lambda(\mathbf{A}\psi_1, \psi_1)$$

or

$$0 \le (\delta - \lambda^2)(\mathbf{Q}_A\psi_1, \psi_1).$$

This implies that $\lambda \ge -\sqrt{\delta}$ since ψ_1 is nonzero. It is elementary to check that $\sqrt{\delta} \le \rho$ where ρ is defined by (3.72). This completes the proof of the theorem. □

The proof of Theorem 3.4 depended on (3.66) so that the inner product $[\cdot,\cdot]$ induced a norm. The next result shows that the inexact Uzawa method converges even when only (3.67) is assumed. It also provides an estimate for the error $E_i^X = X - X_i$ in a more natural norm.

Corollary 3.1 *Assume that (3.63) and (3.67) hold and that γ and δ satisfy (3.64) and (3.68), respectively. Let $\{X, Y\}$ be the solution pair for (3.57), let $\{X_i, Y_i\}$ be defined by the inexact Uzawa algorithm, and set $E_i^X = X - X_i$ and $E_i^Y = Y - Y_i$. Then*

$$(\mathbf{Q}_B E_i^Y, E_i^Y)^{1/2} \le \rho^i |\!|\!| e_0 |\!|\!| \tag{3.85}$$

where ρ is given by (3.72). In addition,

$$(\mathbf{A}E_i^X, E_i^X)^{1/2} \le \rho^{i-1} |\!|\!| e_0 |\!|\!|. \tag{3.86}$$

The above inequalities hold for $i = 1, 2, \ldots$.

Proof: Taking the $(\cdot,\cdot)$-inner product of the first equation of (3.73) with $\mathbf{Q}_A E_{i+1}^X$, applying the Schwarz inequality, and (3.63) gives

$$\begin{aligned}(\mathbf{Q}_A E_{i+1}^X, E_{i+1}^X) &= ((\mathbf{Q}_A - \mathbf{A})E_i^X, E_{i+1}^X) - (\mathbf{B}^T E_i^Y, E_{i+1}^X) \\ &\le ((\mathbf{Q}_A - \mathbf{A})E_i^X, E_i^X)^{1/2}((\mathbf{Q}_A - \mathbf{A})E_{i+1}^X, E_{i+1}^X)^{1/2} \\ &\quad + (\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T E_i^Y, E_i^Y)^{1/2}(\mathbf{A}E_{i+1}^X, E_{i+1}^X)^{1/2} \\ &\le (((\mathbf{Q}_A - \mathbf{A})E_i^X, E_i^X) + \left\| E_i^Y \right\|_{Q_B}^2)^{1/2}(\mathbf{Q}_A E_{i+1}^X, E_{i+1}^X)^{1/2}.\end{aligned}$$

Here we have used the elementary inequality $(ab+cd)^2 \le (a^2+c^2)(b^2+d^2)$, for any real numbers a, b, c, and d. Thus, applying (3.67) gives

$$(\mathbf{A}E^X_{i+1}, E^X_{i+1}) \le (\mathbf{Q}_A E^X_{i+1}, E^X_{i+1}) \le [\![e_i]\!]^2. \tag{3.87}$$

Let $\mathbf{Q}_{A,\epsilon} = \epsilon\mathbf{I} + \mathbf{Q}_A$ for $0 < \epsilon < 1-\delta$. Then (3.66) holds for $\mathbf{Q}_{A,\epsilon}$ and by (3.68),

$$(1-\delta_\epsilon)(\mathbf{Q}_{A,\epsilon}W, W) \le (\mathbf{A}W, W) \quad \text{for all} \quad W \in H_1 \tag{3.88}$$

for $\delta_\epsilon = \delta + \epsilon$. Fix $(X_0, Y_0) \in H_1 \times H_2$ and consider the sequence of iterates $\{X_{\epsilon,i}, X_{\epsilon,i}\}$ generated by the inexact Uzawa algorithm which replaces $\mathbf{Q}_A$ in the first equation of (3.65) by $\mathbf{Q}_{A,\epsilon}$. Applying Theorem 3.4 gives that the error

$$e_{\epsilon,i} = \begin{pmatrix} X - X_{\epsilon,i} \\ Y - X_{\epsilon,i} \end{pmatrix}$$

satisfies

$$[\![e_{\epsilon,i}]\!]_\epsilon \le \rho^i_\epsilon [\![e_{\epsilon,0}]\!]_\epsilon \tag{3.89}$$

where $[\![\cdot]\!]_\epsilon = [\cdot,\cdot]^{1/2}_\epsilon$,

$$\left[\begin{pmatrix} U \\ V \end{pmatrix}, \begin{pmatrix} R \\ S \end{pmatrix}\right]_\epsilon = ((\mathbf{Q}_{A,\epsilon} - \mathbf{A})U, R) + (\mathbf{Q}_B V, S),$$

and

$$\rho_\epsilon = \frac{\gamma(1-\delta_\epsilon) + \sqrt{\gamma^2(1-\delta_\epsilon)^2 + 4\delta_\epsilon}}{2}.$$

Clearly,

$$\left\| E^X_{\epsilon,i} \right\|_{Q_B} \le [\![e_{\epsilon,i}]\!]_\epsilon. \tag{3.90}$$

Inequality (3.85) results from combining (3.89) and (3.90) and taking the limit as ϵ tends to zero.

In a similar manner we prove (3.86). Taking the limit in (3.89) as ϵ tends to zero gives

$$[\![e_{i-1}]\!] \le \rho^{i-1} [\![e_0]\!]. \tag{3.91}$$

Combining (3.87) and (3.91) gives (3.86) and completes the proof of the corollary. □

Remark 3.8 *More restrictive convergence results (in these norms) were obtained by Queck [83]. He proved a convergence result which required stronger conditions with respect to the scaling of* $\mathbf{Q}_A$ *and* $\mathbf{Q}_B$*. In particular, there are cases which fail to satisfy the hypothesis of the theory of [83], yet convergence is guaranteed by the corollary above. In addition, there are many cases when the convergence estimates given above are substantially better than those of [83].*

3.3.3 Analysis of the nonlinear inexact Uzawa algorithm

As was pointed out in Section 3.3.1, the Uzawa algorithm is often implemented as a two level iterative process, an inner iteration for computing $\mathbf{A}^{-1}$ coupled with the outer Uzawa iteration (3.61) or (3.62). In this section we investigate the stability and convergence rate of an abstract inexact Uzawa algorithm where the computation of the action of $\mathbf{A}^{-1}$ is replaced with that of an approximation to $\mathbf{A}^{-1}$ which results from applying a nonlinear iterative process for inverting $\mathbf{A}$. Two examples of such approximations come from defining the approximate inverse by a preconditioned conjugate gradient (PCG) iteration or the operator which results from the application of a multigrid cycling algorithm with a nonlinear smoother [73].

The nonlinear approximate inverse is described as a map $\Psi : H_1 \mapsto H_1$. For $\phi \in H_1$, $\Psi(\phi)$ is an "approximation" to the solution ξ of

$$\mathbf{A}\xi = \phi. \tag{3.92}$$

We shall assume that our approximation satisfies

$$\|\Psi(\phi) - \mathbf{A}^{-1}\phi\|_A \leq \delta \|\phi\|_{A^{-1}} \quad \text{for all } \phi \in H_1 \tag{3.93}$$

for some $\delta < 1$. As will be seen below, (3.93) is a reasonable assumption which is satisfied by the approximate inverse associated with PCG. It also can be shown (see [73, 15]) that (3.93) holds under reasonable assumptions for approximate inverses defined by one sweep of a multigrid algorithm with conjugate gradient smoothing.

Perhaps the most natural example of a nonlinear approximate inverse is defined in terms of the PCG [65, 81, 11]. Let $\mathbf{Q}_A$ be a symmetric and positive definite operator on H_1 and consider applying n steps of the conjugate gradient algorithm preconditioned by $\mathbf{Q}_A$ to solve the problem (3.92) with a zero starting iterate. We define $\Psi(\phi) = \xi_n$ where ξ_n is the resulting approximation to ξ. PCG provides the best approximation (with respect to the norm corresponding to the $(A\cdot, \cdot)$-inner product) to the solution ξ in the n-th Krylov subspace $\mathcal{V}_n$ given by

$$\mathcal{V}_n \;=\; \text{span}\left\{\phi, \mathbf{Q}_A^{-1}\mathbf{A}\phi, \ldots, (\mathbf{Q}_A^{-1}\mathbf{A})^{n-1}\phi\right\}. \tag{3.94}$$

It is well known (cf. [11]) that this implies

$$\|\xi_n - \mathbf{A}^{-1}\phi\|_A \leq \delta \|\phi\|_{A^{-1}} \quad \text{for all } \phi \in H_1\ , \tag{3.95}$$

where

$$\delta = \delta_n \leq \frac{1}{\cosh(n \cosh^{-1}\eta)}.$$

Here $\eta = (K(\mathbf{Q}_A^{-1}\mathbf{A}) + 1)/(K(\mathbf{Q}_A^{-1}\mathbf{A}) - 1)$ and $K(\mathbf{Q}_A^{-1}\mathbf{A})$ is the condition number of $\mathbf{Q}_A^{-1}\mathbf{A}$. Note that δ_n is a decreasing function of n and δ_1 is less than one. Thus, (3.93) holds in the PCG example. In fact,

$$\delta_n \leq 2\left(\frac{K(\mathbf{Q}_A^{-1}\mathbf{A})^{1/2} - 1}{K(\mathbf{Q}_A^{-1}\mathbf{A})^{1/2} + 1}\right)^n.$$

Since δ_n tends to zero as n tends to infinity, it is possible to make δ_n as small as we want by taking a suitably large number PCG iterations.

The variant of the inexact Uzawa algorithm we investigate in this section is defined as follows.

Algorithm 3.6 (Nonlinear Uzawa) *For $X_0 \in H_1$ and $Y_0 \in H_2$ given, the sequence $\{(X_i, Y_i)\}$ is defined, for $i = 1, 2, \ldots$, by*

$$\begin{aligned} X_{i+1} &= X_i + \Psi\left(F - \left(\mathbf{A}X_i + \mathbf{B}^T Y_i\right)\right), \\ Y_{i+1} &= Y_i + \mathbf{Q}_B^{-1}(\mathbf{B}X_{i+1} - G). \end{aligned} \tag{3.96}$$

Clearly, (3.96) reduces to the preconditioned Uzawa algorithm (3.62) if $\Psi(f) = A^{-1}f$ for all $f \in H_1$, and (3.96) reduces to the inexact Uzawa algorithm if Ψ is a linear operator $\mathbf{Q}_A^{-1}$.

We provide bounds for the rate of convergence for the above algorithm in terms of two parameters, the convergence factor γ for the preconditioned Uzawa algorithm defined in (3.64) and the parameter δ of (3.93). The next result is a sufficient condition on δ for convergence of the nonlinear Uzawa algorithm and provides bounds for the resulting rate of convergence.

Theorem 3.5 *Assume that (3.63) and (3.93) hold and that γ satisfies (3.64). Let $\{X, Y\}$ be the solution pair for (3.57) and $\{X_i, Y_i\}$ be defined by the nonlinear Uzawa algorithm (3.96). Then X_i and Y_i converge to X and Y, respectively, if*

$$\delta < \frac{1-\gamma}{3-\gamma}. \tag{3.97}$$

In this case the following inequalities hold:

$$\begin{aligned}\frac{\delta}{1+\delta}(\mathbf{A}E_i^X, E_i^X) + (\mathbf{Q}_B E_i^Y, E_i^Y) \\ \leq \rho^{2i}\left(\frac{\delta}{1+\delta}(\mathbf{A}E_0^X, E_0^X) + (\mathbf{Q}_B E_0^Y, E_0^Y)\right)\end{aligned} \tag{3.98}$$

and

$$(\mathbf{A}E_i^X, E_i^X) \leq (1+\delta)(1+2\delta)\rho^{2i-2}\left(\frac{\delta}{1+\delta}(\mathbf{A}E_0^X, E_0^X) + (\mathbf{Q}_B E_0^Y, E_0^Y)\right) \tag{3.99}$$

where

$$\rho = \frac{2\delta + \gamma + \sqrt{(2\delta+\gamma)^2 + 4\delta(1-\gamma)}}{2}. \tag{3.100}$$

Remark 3.9 *The result of Theorem 3.5 is somewhat weaker than the results obtained in Section 3.3.2 for the linear case due to the threshold condition (3.97) on δ. In the case of PCG, it is possible to take sufficiently many iterations n so that (3.97) holds for any fixed γ and $K(\mathbf{Q}_A^{-1}\mathbf{A})$. In applications involving partial differential equations, γ and $K(\mathbf{Q}_A^{-1}\mathbf{A})$ may depend on the discretization parameter h. If, however, $K(\mathbf{Q}_A^{-1}\mathbf{A})$ can be bounded and γ can be bounded away from 1 independently of h then by Theorem 3.5, a fixed number (independent of h) of iterations of PCG are sufficient to guarantee convergence of the nonlinear Uzawa algorithm.*

Remark 3.10 *An analysis of (3.96) is given in [45] and [46] in the case of applications to Stokes problems. The sufficient condition for convergence derived there is that the iterate X_{i+1} satisfies*

$$\|F - \mathbf{B}^T Y_i - \mathbf{A}X_{i+1}\| \leq \tau \|\mathbf{B}X_i - G\|_{Q_A^{-1}}, \tag{3.101}$$

where τ is independent of the mesh size. The above norms are not natural for procedures such as PCG and multigrid with nonlinear smoothing. PCG does not give rise to monotone error behavior in the norm $\|\cdot\|$ even though convergence is guaranteed by the canonical bound (3.95),

$$\|F - \mathbf{B}^T Y_i - \mathbf{A}X_{i+1}\|_{A^{-1}} \leq \delta \|F - \mathbf{B}^T Y_i - \mathbf{A}X_i\|_{A^{-1}}$$

and equivalence of norms in finite dimensional spaces. Such norm equivalences depend on the mesh parameter h. A second problem with the requirement (3.101) is that the norm to the right-hand side converges to zero as X_i converges to the solution X. This means that even though τ is fixed independent of h, considerably more iterations of PCG may be required to satisfy (3.101) as the approximate solution converges. In contrast, our result implies that a fixed number of iterations (independent of h) of PCG will guarantee convergence, provided that the preconditioners $\mathbf{Q}_A$ and $\mathbf{Q}_B$ are uniform.

Proof:[Theorem 3.5] We start by deriving norm inequalities involving the errors E_i^X and E_i^Y. As in (3.73),

$$\begin{aligned}E_{i+1}^X &= E_i^X - \Psi\left(\mathbf{A}E_i^X + \mathbf{B}^T E_i^Y\right), \\ E_{i+1}^Y &= E_i^Y + \mathbf{Q}_B^{-1}\mathbf{B}E_{i+1}^X.\end{aligned} \tag{3.102}$$

The first equation above can be rewritten

$$E_{i+1}^X = (\mathbf{A}^{-1} - \Psi)\left(\mathbf{A}E_i^X + \mathbf{B}^T E_i^Y\right) - \mathbf{A}^{-1}\mathbf{B}^T E_i^Y. \tag{3.103}$$

It follows from the triangle inequality, (3.93) and (3.63) that

$$\begin{aligned} \left\|E_{i+1}^X\right\|_A &\leq \delta\left(\left\|E_i^X\right\|_A + (\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T E_i^Y, E_i^Y)^{1/2}\right) \\ &\quad + (\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T E_i^Y, E_i^Y)^{1/2} \\ &\leq \delta\left\|E_i^X\right\|_A + (1+\delta)\left\|E_i^Y\right\|_{Q_B}. \end{aligned} \tag{3.104}$$

Using (3.103) in the second equation of (3.102), we obtain

$$E_{i+1}^Y = \left(\mathbf{I} - \mathbf{Q}_B^{-1}\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T\right) E_i^Y + \mathbf{Q}_B^{-1}\mathbf{B}\left(\mathbf{A}^{-1} - \Psi\right)\left(\mathbf{A}E_i^X + \mathbf{B}^T E_i^Y\right).$$

Since $\mathbf{Q}_B^{-1}\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T$ is a symmetric operator in the $<\cdot,\cdot>$-inner product, it follows from (3.64) that

$$\left\|\left(\mathbf{I} - \mathbf{Q}_B^{-1}\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T\right) E_i^Y\right\|_{Q_B} \leq \gamma\left\|E_i^Y\right\|_{Q_B}.$$

Thus, by the triangle inequality, (3.63), (3.84) and (3.93),

$$\begin{aligned} \left\|E_{i+1}^Y\right\|_{Q_B} &\leq \gamma\left\|E_i^Y\right\|_{Q_B} + \left\|\mathbf{Q}_B^{-1}\mathbf{B}\left(\mathbf{A}^{-1} - \Psi\right)\left(\mathbf{A}E_i^X + \mathbf{B}^T E_i^Y\right)\right\|_{Q_B} \\ &\leq \gamma\left\|E_i^Y\right\|_{Q_B} + \left\|\left(\mathbf{A}^{-1} - \Psi\right)\left(\mathbf{A}E_i^X + \mathbf{B}^T E_i^Y\right)\right\|_A \\ &\leq \gamma\left\|E_i^Y\right\|_{Q_B} + \delta\left(\left\|E_i^X\right\|_A + (\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T E_i^Y, E_i^Y)^{1/2}\right) \\ &\leq (\gamma+\delta)\left\|E_i^Y\right\|_{Q_B} + \delta\left\|E_i^X\right\|_A. \end{aligned} \tag{3.105}$$

Let us adopt the notation

$$\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} \leq \begin{pmatrix} x_2 \\ y_2 \end{pmatrix}$$

for vectors of nonnegative numbers x_1, x_2, y_1, y_2 if $x_1 \leq x_2$ & $y_1 \leq y_2$. Repeated application of (3.104) and (3.105) gives

$$\begin{pmatrix} \left\|E_i^X\right\|_A \\ \left\|E_i^Y\right\|_{Q_B} \end{pmatrix} \leq M^i \begin{pmatrix} \left\|E_0^X\right\|_A \\ \left\|E_0^Y\right\|_{Q_B} \end{pmatrix} \tag{3.106}$$

where M is given by

$$M = \begin{pmatrix} \delta & 1+\delta \\ \delta & \gamma+\delta \end{pmatrix}.$$

We consider two-dimensional Euclidean space with the inner product

$$\left\lfloor \begin{pmatrix} x_1 \\ y_1 \end{pmatrix}, \begin{pmatrix} x_2 \\ y_2 \end{pmatrix} \right\rfloor \equiv \frac{\delta}{1+\delta} x_1 x_2 + y_1 y_2.$$

A trivial computation shows that M is symmetric with respect to the $\lfloor\cdot,\cdot\rfloor$-inner product. It follows from (3.106) that

$$\begin{aligned} \frac{\delta}{1+\delta}(\mathbf{A}E_i^X, E_i^X) + (\mathbf{Q}_B E_i^Y, E_i^Y) &= \left\lfloor \begin{pmatrix} \left\|E_i^X\right\|_A \\ \left\|E_i^Y\right\|_{Q_B} \end{pmatrix}, \begin{pmatrix} \left\|E_i^X\right\|_A \\ \left\|E_i^Y\right\|_{Q_B} \end{pmatrix} \right\rfloor \\ &\leq \left\lfloor M^i \begin{pmatrix} \left\|E_0^X\right\|_A \\ \left\|E_0^Y\right\|_{Q_B} \end{pmatrix}, M^i \begin{pmatrix} \left\|E_0^X\right\|_A \\ \left\|E_0^Y\right\|_{Q_B} \end{pmatrix} \right\rfloor \\ &\leq \rho^{2i}\left(\frac{\delta}{1+\delta}(\mathbf{A}E_0^X, E_0^X) + (\mathbf{Q}_B E_0^Y, E_0^Y)\right) \end{aligned}$$

where ρ is the norm of the matrix M with respect to the $\lfloor\cdot,\cdot\rfloor$-inner product. Since M is symmetric in this inner product, its norm is bounded by its spectral radius. The eigenvalues of M are the roots of

$$\lambda^2 - (2\delta + \gamma)\lambda - \delta(1-\gamma) = 0.$$

It is elementary to see that the spectral radius of M is equal to its positive eigenvalue which is given by (3.100).

Examining the expression for ρ given by (3.100) we see that ρ is an increasing function of δ for any fixed $\gamma \in [0,1]$. Moreover, $\rho = 1$ for

$$\delta = \frac{1-\gamma}{3-\gamma}.$$

This completes the proof of the (3.98).

To prove (3.99) we apply the arithmetic-geometric mean inequality to (3.104) and get for any positive η,

$$\left\| E_i^X \right\|_A^2 \leq (1+\eta)\delta^2 \left\| E_{i-1}^X \right\|_A^2 + (1+\eta^{-1})(1+\delta)^2 \left\| E_{i-1}^Y \right\|_{Q_B}^2 .$$

Inequality (3.99) follows taking $\eta = 1+1/\delta$ and applying (3.98). This completes the proof of the theorem. □

3.3.4 Applications to mixed finite element discretizations of elliptic problems

Clearly, the inexact Uzawa algorithms can be used for solving the indefinite system (2.27) arising from mixed finite element discretizations of the differential problem (2.19). In fact, this method was perhaps the first known approach to solving such problems (cf. [30]). For such applications, it is relatively easy to construct preconditioners $\mathbf{Q}_A$ while the development of a suitable operator $\mathbf{Q}_B$ is more difficult.

It was pointed out in Section 2.2.3 that the Raviart–Thomas spaces satisfy (3.60) with uniform constant c_0, independent of the discretization parameter h. It is also evident from (2.27) that the operator $\mathbf{A}$ corresponds to a mass matrix in the space $\mathcal{V}_h(\Omega)$ and is either well conditioned or easily preconditioned by a diagonal matrix. In fact, if the tensor A in (2.27) has smooth coefficients, then $\mathbf{Q}_A$ can be chosen to be $\tau\mathbf{I}$, where τ is an appropriately chosen real number. In this case the preconditioner $\mathbf{Q}_A$ reduces to simple scaling according to (3.67). If A has coefficients with jumps in Ω, then appropriately defined diagonal matrix $\mathbf{Q}_A$ provides a very good preconditioner.

On the other hand, the operator $\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T$ is not uniformly well conditioned. In fact, it exhibits a condition number growth like $O(h^{-2})$ and should be preconditioned in order to get an efficient algorithm of type (3.62) or (3.65). It is well known that $\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T$ behaves like a discretization of a second-order operator. In practice, it can be preconditioned by cell-centered techniques [87], multigrid [27], or incomplete Choleski factorization of $\mathbf{B}\mathbf{B}^T$ [89, 88].

3.3.5 Numerical investigation of inexact Uzawa algorithms

Our goal here is to experiment with Algorithm 3.5 in order to assess its efficiency. In Section 3.3.4 we observed that an effective preconditioner for the Schur complement of the mixed system is needed for the efficient performance of the inexact algorithm in the case of second-order elliptic problems. On the other hand the same type of preconditioner is needed to apply the nonoverlapping preconditioners developed in this chapter to the solution of the same problem as shown in Section 3.2.5. The approach from Section 3.2.5 allows the use of the preconditioned conjugate gradient for the iterative solution of the problem, which is a much faster method than the linear process provided by Algorithm 3.5. Because of this, we shall concentrate on much more difficult applications where Algorithm 3.5 provides an effective way of solving the corresponding problems. In particular, we shall consider applications to the steady state Stokes equation, which is a difficult problem. The inexact Uzawa algorithm appears to be one of the most efficient methods for solving it (cf. [30, 56, 57, 46, 45]). Moreover, developing a method for this problem may potentially lead to a new technique for solving the Navier–Stokes equation, which is a rather challenging mathematical problem.

The Stokes problem

Here we consider an application of the theory developed in the previous sections to solving indefinite systems of linear equations arising from finite element approximations of the Stokes equations. For simplicity we restrict our discussion to the following model problem:

Find $\mathbf{u}$ *and* p *such that*

$$-\Delta \mathbf{u} - \nabla p = \mathbf{g} \quad \text{in } \Omega, \tag{3.107a}$$

$$\nabla \cdot \mathbf{u} = f \quad \text{in } \Omega, \tag{3.107b}$$

$$\mathbf{u} = 0 \quad \text{on } \partial\Omega, \tag{3.107c}$$

$$\int_\Omega p(x)\ dx = 0, \tag{3.107d}$$

where Ω is the unit cube in $\mathbb{R}^2$, Δ is the componentwise Laplace operator, $\mathbf{u}$ is a vector valued function representing the velocity, and the pressure p is a scalar function.

Let $L_0^2(\Omega)$ be the set of functions in $L^2(\Omega)$ with zero mean value on Ω and $H^1(\Omega)$ denote the Sobolev space of order one on Ω. The space $H_0^1(\Omega)$ consists of those functions in Ω whose traces vanish on $\partial\Omega$. Also, $(H_0^1(\Omega))^2$ will denote the product space consisting of vector valued functions with each vector component in $H_0^1(\Omega)$.

In order to derive the weak formulation of (3.107) we multiply the first two equations of (3.107) by functions in $(H_0^1(\Omega))^2$ and $L_0^2(\Omega)$ respectively and integrate over Ω to get

$$D(\mathbf{u}, \mathbf{v}) + (p, \nabla \cdot \mathbf{v}) = (\mathbf{g}, \mathbf{v}), \quad \text{for all} \quad \mathbf{v} \in (H_0^1(\Omega))^2, \tag{3.108a}$$

$$(\nabla \cdot \mathbf{u}, q) = (f, q)\ , \quad \text{for all} \quad q \in L_0^2(\Omega). \tag{3.108b}$$

Here $(\cdot, \cdot)$ is the $L^2(\Omega)$ inner product and $D(\cdot, \cdot)$ denotes the vector Dirichlet form for vector functions on Ω defined by

$$D(\mathbf{v}, \mathbf{w}) = \sum_{i=1}^{2} \int_\Omega \nabla v_i \cdot \nabla w_i\ dx.$$

We next identify approximation subspaces of $(H_0^1(\Omega))^2$ and $L_0^2(\Omega)$. The discussion here is very closely related to the examples given in [17] and [14] where additional comments and other applications can be found. We partition Ω into $2n \times 2n$ square shaped elements, where n is a positive integer and define $h = 1/2n$. Let $x_i = ih$ and $y_j = jh$ for $i, j = 1, \ldots, 2n$. Each of the square elements is further partitioned into two triangles by connecting the lower right corner to the upper left corner. Let S_h be the space of functions that vanish on $\partial\Omega$ and are continuous and piecewise linear with respect to the triangulation thus defined. We set $H_1 \equiv S_h \times S_h \subset (H_0^1(\Omega))^2$. The choice of H_2 is motivated by the observation [68] that the space $\tilde{H}_2$ of functions that are piecewise constant with respect to the square elements together with H_1 as defined above form an unstable pair of approximation spaces. This means that the functions from $H_1 \times \tilde{H}_2$ do not satisfy (3.60) with a constant c_0 independent of the discretization parameter h. To overcome this problem, one may consider a smaller space defined as follows. Let η_{kl} for $k, l = 1, \ldots, 2n$ be the function that is 1 on the square element $[x_{k-1}, x_k] \times [y_{l-1}, y_l]$ and vanishes elsewhere. Define $\phi_{ij} \in \tilde{H}_2$ for $i, j = 1, \ldots, n$ by

$$\phi_{ij} = \eta_{2i-1,2j-1} - \eta_{2i,2j-1} - \eta_{2i-1,2j} + \eta_{2i,2j}$$

(see Figure 3.2). The space H_2 is then defined by

$$H_2 \equiv \left\{ W \in \tilde{H}_2 \ :\ (W, \phi_{ij}) = 0 \text{ for } i, j = 1, \ldots, n \right\}.$$

The pair $H_1 \times H_2$ now satisfies (3.60) with a constant c_0 independent of h [68]. Moreover, the exclusion of the functions $\phi_{i,j}$ does not change the order of approximation for the space since H_2 still contains the piecewise constant functions of size $2h$.

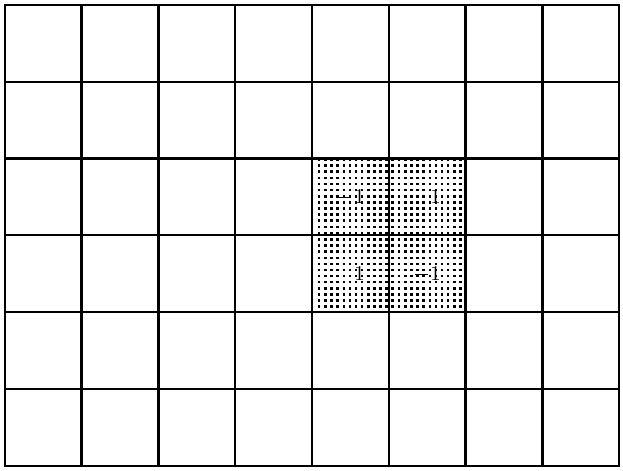


Figure 3.2: The square mesh used for $\tilde{H}_2$; the support (shaded) and values for a typical ϕ_{ij}

The approximation to the solution of (3.108) is defined as the unique pair $(X, Y) \in H_1 \times H_2$ satisfying

$$D(X,V) + (Y, \nabla \cdot V) = (\mathbf{g}, V), \quad \text{for all} \quad V \in H_1, \tag{3.109a}$$

$$(\nabla \cdot X, W) = (f, W), \quad \text{for all} \quad W \in H_2. \tag{3.109b}$$

Obviously, (3.109) is a system of linear equations whose unique solvability is guaranteed by (3.60).

The system (3.109) can be reformulated in terms of operators as follows. Let

$$\begin{aligned} \mathbf{A} : H_1 \mapsto H_1, \quad (\mathbf{A}U, V) &= D(U,V), \quad \text{for all} \quad U, V \in H_1, \\ \mathbf{B} : H_1 \mapsto H_2, \quad (\mathbf{B}U, W) &= (\nabla \cdot U, W), \quad \text{for all} \quad U \in H_1, W \in H_2, \\ \mathbf{B}^T : H_2 \mapsto H_1, \quad (\mathbf{B}^T W, V) &= (W, \nabla \cdot V), \quad \text{for all} \quad V \in H_1, W \in H_2. \end{aligned}$$

It follows that the solution (X, Y) of (3.109) satisfies (3.57) with F equal to the $L^2(\Omega)$ projection of f into H_2 and G equal to the $(L^2(\Omega))^2$ projection of $\mathbf{g}$ into H_1.

It is straightforward to check that (3.63) holds for $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{B}^T$ as above. Moreover, it follows from (3.60) that (3.64) holds with γ independent of the mesh size h.

Remark 3.11 *It appears from the definition of the above operators that one has to invert Gram matrices in order to evaluate the action of* $\mathbf{A}$, $\mathbf{B}^T$ *and* $\mathbf{B}$ *on vectors from the corresponding spaces. In practice, the* H_1 *Gram matrix inversion is avoided by suitable definition of the preconditioner* $\mathbf{Q}_A$. *For the purpose of computation, the evaluation of* $\mathbf{Q}_A^{-1} f$ *for* $f \in H_1$ *is defined as a process which acts on the inner product data* (f, ψ_i) *where* $\{\psi_i\}$ *is the basis for* H_1. *Moreover, from the definition of the Uzawa-like algorithms in the previous sections, it is clear that every occurrence of* $\mathbf{A}$ *or* $\mathbf{B}^T$ *is followed by an evaluation of* $\mathbf{Q}_A^{-1}$. *Thus the inversion of the Gram matrix is avoided since the data for the computation of* $\mathbf{Q}_A^{-1}$, $((\mathbf{B}^T Q, \psi_i)$ *and* $(\mathbf{A}V, \psi_i))$, *can be computed by applying simple sparse matrices. In the case of this special choice of* H_2, *it is possible to compute the operator* $\mathbf{B}$ *in an economical way (see Remark 5 of [14]) and we can take* $\mathbf{Q}_B$ *to be the identity. For more general spaces* H_2, *the inversion of Gram matrices can be avoided by introducing a preconditioner* $\mathbf{Q}_B$ *whose inverse is implemented acting on inner product data as in the* H_1 *case above.*

We still need to provide preconditioners for $\mathbf{A}$. However, $\mathbf{A}$ consists of two copies of the operator which results from a standard finite element discretization of Dirichlet's problem. There has been an intensive effort focused on the development and analysis of preconditioners for such problems. For the examples in Section 3.3.5, we will use a preconditioning operator which results from a V-cycle variational multigrid algorithm. Such a preconditioner is known to be scaled so that both (3.67) hold and (3.64) holds with γ bounded away from 1 independently of the mesh parameter h.

Table 3.4: Comparison of **UMG**and **UEX**algorithms

h	**UMG**		**UEX**	
	# iterations	time (secs)	# iterations	time (secs)
1/8	53	0.39	54	1.56
1/16	48	1.09	55	7.25
1/32	52	4.74	55	30.59
1/64	55	20.66	55	127.32
1/128	57	90.18	55	556.62

Numerical examples

In this section we present the results from numerical experiments with Algorithm 3.5 applied to the solution of (3.107) with $\Omega \equiv (0,1)^2$, $\mathbf{g} = 0$ and $f = 0$. Clearly, its exact solution is zero for both pressure and velocity.

We compare the performance of Algorithm 3.5 with that of the exact method provided by Algorithm 3.3. We also compare Algorithm 3.5 with the algorithm introduced in [14] which uses a SPD reformulation of (3.107) and takes advantage of the acceleration provided by the conjugate gradient algorithm. To organize the comparisons we always start the iterations with an arbitrary but fixed initial iterate. The performance of all of the iterative methods considered is a function of the error and thus, iterating for a problem with a zero solution and a nonzero starting guess is equivalent to solving a related problem with a nonzero solution and a zero initial guess. We use the discretization described in Section 3.3.5.

We report results when efficient preconditioners are used as well as results obtained when very weak preconditioners are utilized. The reason for experimenting with the latter case is that in many engineering applications good preconditioners are not readily available and thus we have to assess the performance of the inexact methods in such cases.

The algorithms involved in the tests we present are given below:

UEX : The exact Algorithm 3.3 with $\tau = 1$. The inverse of $\mathbf{A}$ is computed by a PCG method with a uniform preconditioner provided by one multigrid V-cycle;

UID : The algorithm (3.65) with $\mathbf{Q}_A = \bar{\lambda}_{max}\mathbf{I}$ and $\mathbf{Q}_B = \mathbf{I}$. Here $\bar{\lambda}_{max}$ is an upper bound for the largest eigenvalue of $\mathbf{A}$;

USTD : The algorithm (3.96) with $\mathbf{Q}_B = \mathbf{I}$ and Ψ defined by one step of the steepest descent method (SDM) applied to approximate the action of $\mathbf{A}^{-1}$;

BPID : The preconditioned conjugate gradient algorithm for saddle point problems given in [14] with $\mathbf{Q}_A = \bar{\lambda}_{min}\mathbf{I}$, where $\bar{\lambda}_{min}$ is a lower bound for the smallest eigenvalue of $\mathbf{A}$ and $\mathbf{Q}_B = \mathbf{I}$. Notice that the scaling required by Theorem 1 of [14] is in the opposite direction of (3.66);

UMG : The algorithm (3.65) with $\mathbf{Q}_B = \mathbf{I}$ and $\mathbf{Q}_A^{-1}$ being the action of multigrid;

BPMG : The algorithm from [14] with the $\mathbf{A}$ block preconditioned by $.5\mathbf{Q}_A^{-1}$ and $\mathbf{Q}_B = \mathbf{I}$.

The first test is intended to compare the performance of the exact **UEX** and the inexact **UMG** algorithms. The experimental results shown in Table 3.4 represent the time (in seconds) taken by these two algorithms to reduce the initial error down to 10^{-6}. The test ran on a Sun Sparc 20 workstation. The advantage of the inexact method **UMG** is clearly seen. It is more than five times more efficient than the exact algorithm **UEX**.

Table 3.5: Errors in **UID**, **USTD** and **BPID** by (3.110)

h	200 iterations		
	UID	**USTD**†	**BPID**
1/8	4.2×10^{-3}	5.1×10^{-6}	‡6.5×10^{-12}
1/16	0.4	5.8×10^{-2}	2.9×10^{-10}
1/32	1.5	0.2	1.1×10^{-4}
1/64	2.7	4.5	2.0×10^{-2}

† *one SDM step per inexact Uzawa iteration*;
‡ *for 109* **BPID** *iterations.*

Table 3.6: Errors in **UID** and **USTD** by (3.110)

h	2000 iterations	
	UID	**USTD**†
1/8	0	2.0×10^{-23}
1/16	3.7×10^{-6}	3.9×10^{-16}
1/32	2.5×10^{-2}	2.1×10^{-4}
1/64	1.5	8.7×10^{-2}

† *one SDM step per inexact Uzawa iteration.*

In Table 3.5 we give results for three algorithms using $\mathbf{Q}_A$ equal to an appropriate multiple of the identity. The reported error values represent the relative error norm after i iterations computed by

$$Error_i = \left(\frac{D(E_i^X, E_i^X) + \left\| E_i^Y \right\|^2}{D(E_0^X, E_0^X) + \left\| E_0^Y \right\|^2} \right)^{1/2} . \tag{3.110}$$

Clearly, this is not the norm which appears in the theory and one cannot expect the errors to behave in a monotone way. This explains the increase in the reported error for **UID** when $h = 1/32$ and $h = 1/64$. That the **USTD** method appears convergent for $h \leq 32$ is surprising since (3.97) is not satisfied for these applications. The **BPID** method converges considerably faster in these examples since the saddle point method of [14] is known to give a rate of convergence which exhibits square root acceleration in cases when poor preconditioners are employed. As expected, all methods deteriorate due to lack of preconditioning as the mesh size is decreased.

In order to establish experimentally the convergence of **UID** and **USTD**, we ran these two algorithms for 2000 iterations. The results are shown in Table 3.6. Even though improved convergence is observed in all cases when compared to Table 3.5, the **UID** algorithm still appears unstable for $h = 1/64$. We ran **UID** for 10000 iterations and observed an error of .0048. Although convergent, the inexact Uzawa method with such a poor preconditioner converges too slowly to be of practical use.

The above results may at first appear to contradict the validity of the theory of the inexact Uzawa algorithms. The reason that the methods appear divergent at a relatively low numbers of iterations is that the theorems guarantee monotonicity of the errors in norms which are different from those used in (3.110). Our next experiment was designed to illustrate the monotone convergence of **UID** and **BPID** predicted by Theorem 3.4 and Theorem 1 in [14]. Accordingly, we measured the errors in the norms appearing in the theorems. In the case of **UID**, we use

$$\text{Error}_i = \left(\frac{\bar{\lambda}_{max} \left\| E_i^X \right\|^2 - D(E_i^X, E_i^X) + \left\| E_i^Y \right\|^2}{\bar{\lambda}_{max} \left\| E_0^X \right\|^2 - D(E_0^X, E_0^X) + \left\| E_0^Y \right\|^2} \right)^{1/2} . \tag{3.111}$$

Table 3.7: Errors in **UID** and **BPID** by (3.110) and (3.112)

h	200 iterations	
	UID	**BPID**
1/8	4.29×10^{-3}	‡2.1×10^{-12}
1/16	0.18	3.1×10^{-10}
1/32	0.52	1.1×10^{-4}
1/64	0.77	2.0×10^{-2}

‡ *for 109* **BPID** *iterations.*

Table 3.8: Errors in **UMG** and **BPMG** by (3.110)

h	40 iterations	
	UMG†	**BPMG**†
1/8	1.6×10^{-5}	1.0×10^{-11}
1/16	9.4×10^{-7}	6.9×10^{-9}
1/32	1.6×10^{-6}	1.3×10^{-8}
1/64	2.2×10^{-6}	4.5×10^{-9}

† *one multigrid V–cycle per iteration.*

In the case of **BPID**, we use

$$\text{Error}_i = \left(\frac{D(E_i^X, E_i^X) - \bar{\lambda}_{min} \left\| E_i^X \right\|^2 + \left\| E_i^Y \right\|^2}{D(E_0^X, E_0^X) - \bar{\lambda}_{min} \left\| E_0^X \right\|^2 + \left\| E_0^Y \right\|^2} \right)^{1/2}. \tag{3.112}$$

The convergence results in these norms are reported in Table 3.7. Note that all of the reported errors are less than one. We made additional runs at lower numbers of iterations. All runs reflected the monotone error behavior in these norms as guaranteed by the theory.

The last experiment given in this section is intended to illustrate the performance of the algorithms when effective preconditioners are used, namely **UMG** and **BPID**. In this case, we define $\mathbf{Q}_A^{-1}$ to be the operator which corresponds to one V-cycle sweep of variational multigrid with point Gauss-Seidel smoothing. The order of points in the Gauss-Seidel iteration was reversed in pre- and post-smoothing. Note that $\mathbf{Q}_A$ automatically satisfies (3.67) and satisfies (3.68) with δ independent of h. Table 3.8 contains the error reductions for this example. The effect of applying a better preconditioner $\mathbf{Q}_A$ is clearly seen when we compare the results from **UID** (Tables 3.5 and 3.6) with those from **UMG**. Notice that the **UMG** data in Table 3.8 show little, if any, deterioration as the mesh size becomes small.

In all of the reported results, the reformulation method of [14] shows faster convergence. Nevertheless, the inexact Uzawa methods are of interest since they are robust, simple to implement, cheaper computationally, have minimal memory requirements, and avoid the necessity of computing inner products. In addition, the inexact Uzawa algorithms are more efficient when other discretization methods for (3.107) are applied (cf. [45]). These properties make the inexact Uzawa methods attractive in certain applications.

Chapter 4

Multiphase fluid flow in porous media

This chapter is devoted to the application of the theory developed in Chapters 2 and 3 to the important real-life problem of modeling fluid flow in porous media. Flow of underground water has been studied by hydrologists and soil scientists in connection with applications to both civil and agricultural engineering. The reservoir modeling of multiphase and multicomponent flows has been used in the petroleum industry for production and recovery of hydrocarbons. In addition, various problems of flows in porous media are related to the design and evaluation of remediation technologies and water quality control.

During the last few decades geologists and petroleum engineers have become increasingly involved in modeling and computer simulation of flows in underground reservoirs. These efforts have led to the development of a wide range of mathematical models for saturated single-phase flow, saturated/unsaturated two-phase flow, and multiphase flow. In general, these are systems of nonlinear partial differential equations of convection-diffusion-reaction type. The formulation of the differential model is usually based on the mass conservation principle enhanced with appropriate constitutive relations.

In some practical situations, the system of equations can be simplified substantially. For example, incompressible fluid flow in a fully saturated reservoir is adequately described by a single elliptic equation for the pressure. This model has been successfully used in underground hydrology in the past century. However, driven by the need for design of advanced technologies for production and recovery of oil and gas, the petroleum industry has developed and implemented complex multiphase multicomponent flow models (cf. [2]).

Environmental protection applications represent a class of practical problems closely related to the oil applications in terms of the physics involved. Both areas require a good description of the geological structure of the reservoir for reliable results; similar fluids may be involved in the simulation; in both cases various length and time scales are present at which the processes occur. However, there are some specific features of groundwater modeling that make such problems rather difficult. For example, different pressure regimes may occur here as opposed to those of a typical oil recovery application. Also, the variety of simulated species is larger, and the needed accuracy is often very high (especially for the concentration of the pollutants). Thus, sophisticated mathematical models and accurate numerical techniques should be combined for obtaining reliable results.

In this chapter we consider a variety of groundwater flow models that have been used in computer simulation for study and design of remediation and clean-up technologies. Even though most of the models considered here extend without substantial difficulties to three-phase underground flows, we shall restrict ourselves to the equations of saturated and unsaturated flows of air and water phases. We shall also discuss the important question of the choice of the approximation method for the corresponding mathematical problem. In numerical simulation of fluid reservoirs (aquifer or oil) there are two important practical requirements: the method should conserve mass locally and should produce accurate velocities (fluxes) even for highly heterogeneous media with large variations of physical properties. This is the reason why the finite volume method with harmonic averaging of the coefficients has been very popular and successful in computer simulation of flows in porous media. However, when the

problem requires accurate description of the various geological formations in the reservoir, more general techniques based on the finite element approximation are needed. The mixed finite element method considered in Chapter 2 has all these properties. Since its introduction by Raviart and Thomas [84] and its implementation by Ewing and Wheeler [55] for flow problems, it has become a standard way of deriving high-order conservative approximations. It should be noted that the lowest-order mixed method implemented on rectangles (or parallelepipeds) with certain numerical integration produces cell-centered finite differences with harmonic averaging (cf. Remark 2.13). The locally refined discretizations from Chapter 2 can be combined with these discretizations in order to improve the efficiency and the accuracy of the resulting numerical approximation. The iterative techniques from Chapter 3 can be employed for the efficient solution of the corresponding discrete problems. This allows us to apply the new methods to the solution of sophisticated groundwater flow models in order to improve the quality of the simulation results.

The chapter is structured as follows. In Section 4.1 we introduce the fundamental concepts used in the derivation of groundwater flow models. Next, in Section 4.2 four particular flow models are considered with special emphasis given to the fractional flow model. In Section 4.3 we consider the important issues of boundary conditions setup for the fractional flow model as well as the incorporation of wells into the model. Finally, in Section 4.4 we discuss the development of a sophisticated flow simulator and present results from an interesting simulation.

4.1 Fundamentals of fluid flow in porous media

In this section we introduce the physical principles on which the models of fluid flow in porous media are based. In an attempt to keep the volume of this dissertation within reasonable limits we shall use terminology from fluid dynamics and geology without precise definition. The reader is referred to the classical book of Bear [7] for a comprehensive consideration of many concepts concerning fluid flow in porous media.

Typically, there are various flow conditions that occur underground. Below the water table the only fluid flowing is water and, correspondingly, saturated flow conditions are observed. Above the water table is the zone of unsaturated flow conditions characterized by the presence of water and air phases. Thus, a coupled water and air differential flow model is needed for dealing with interesting phenomena occurring in the unsaturated zone, such as soil venting (cf. [8]), soil gas exchange with the atmosphere, rainfall-runoff estimation, and hazardous waste disposal.

There are several assumptions made for the derivation of the models to be described. We begin with the outline of the most fundamental ones (cf. [8]).

1. **There are three phases in the system:** solid, liquid and gas phases. The solid phase is assumed to be immobile and consolidated. The gas and liquid phases are assumed to be fully mobile, but immiscible.

2. **Single time and length scales are assumed**. The processes are considered on the time scale of a single infiltration event and so evaporation and consolidation are assumed negligible.

3. **The media is isothermal**. Thus, heat effects are assumed negligible.

4. **The flow of the two fluids is independent of the presence of chemicals dissolved in either phase**. Thus, the flow equations can be considered separately from the contaminant transport equations which are beyond the scope of this thesis.

As we mentioned already, the saturated zone can be modeled by a single-phase equation. On the other hand the unsaturated zone requires a multiphase system by definition, due to the presence of air and water fluid phases. A macroscopic description of the latter system uses solid and fluid properties defined over the porous media continuum, and is based on conservation laws for each phase. To close the system of equations obtained, media- and material-dependent constitutive relations are added. Typical examples of constitutive relations used in the unsaturated zone are the capillary pressure-saturation relation and the relative permeability-saturation relation. As a result the governing equations for the single-phase saturated or the two-phase unsaturated flow are obtained.

4.1.1 The conservation of mass principle

The mass balance equation for each fluid phase can be written as (cf. [7])

$$\frac{\partial(\phi\rho_\alpha S_\alpha)}{\partial t} + \nabla\cdot(\rho_\alpha \mathbf{u}_\alpha) = F_\alpha, \tag{4.1}$$

where S_α is the saturation, ρ_α is the density, ϕ is the porosity, $\mathbf{u}_\alpha$ is the volumetric flux of phase α, and F_α is the source term. The index α refers to the air (a) and water (w) phase, respectively.

The mass balance given in (4.1) states that the change of mass in a control volume (described by the first term in the left-hand side) and the divergence of the mass flux in that volume (described by the second term) are compensated by the mass supplied/removed by the sources/sinks on the right-hand side.

An alternative form of (4.1) can be obtained by defining a volumetric fluid content of phase α by

$$\theta_\alpha = \phi S_\alpha.$$

Thus, the mass balance law becomes

$$\frac{\partial(\rho_\alpha\theta_\alpha)}{\partial t} + \nabla\cdot(\rho_\alpha \mathbf{u}_\alpha) = F_\alpha. \tag{4.2}$$

Obviously, neither (4.1) nor its alternative formulation (4.2) are solvable by itself. Additional equations of fluid motion must be supplied for that purpose (cf. [7, 66]).

4.1.2 Darcy's law

The momentum balance is the fundamental principle used in fluid dynamics to close the system of equations that describes the fluid flow in porous media (cf. [7, 66]). In the case of Newtonian fluids this principle reduces to the well known Navier-Stokes equation (cf. [61]). Such models represent adequate treatment of the underlying physics. However, they bring in enormous difficulties due not only to the complicated mathematical nature of the corresponding differential equations but also because they require solving the equations at the microscopic pore level. Therefore, it is impossible to use such models to handle reservoirs of realistic sizes for two basic reasons. First, it is not possible to obtain description of the media properties of such reservoirs at the microscopic level. Second, since a numerical approximation is the only way to solve the corresponding equation, no computer of the present time (or likely in the near future) can handle the corresponding discrete model. Therefore, to reach beyond the scope of small laboratory experiments, an alternative set of equations that model the flow at the macroscopic level should be used.

Darcy's law provides the needed alternative. It replaces the momentum equation for each fluid phase by an empirical relation that links the individual phase pressures to the corresponding fluxes:

$$\mathbf{u}_\alpha = -\frac{\mathbf{K}k_{r\alpha}}{\mu_\alpha}(\nabla p_\alpha - \rho_\alpha \mathbf{g}), \quad \alpha = a,\, w, \tag{4.3}$$

where $\mathbf{K}$ is the absolute permeability tensor of the medium, $k_{r\alpha}$ is the relative permeability of phase α, μ_α is the dynamic fluid viscosity of α, p_α is the corresponding fluid pressure, and $\mathbf{g}$ is the acceleration vector due to gravity.

Although Darcy's law was discovered as an empirical relation, there are several examples of analytical derivation of (4.3) from the momentum balance equations under certain additional assumptions on the fluid flow available in the literature (cf. [7, 2]). These assumptions can be summarized as follows: the rock is chemically inert, the fluid is Newtonian, shear stresses and fluid inertia are negligible, and the momentum exchange with the rock is in the form of Stokes drag. Darcy's law provides a quite accurate characterization of the flux $\mathbf{u}_\alpha$ in terms of the phase pressure when laminar flows in a medium with relatively low permeability are considered (cf. [7]). These are the typical flow conditions which occur underground at low Reynolds numbers.

Within the groundwater literature, the pressure normally is scaled by the gravity potential function. This allows the definition of a pressure head h_α (or the pressure in an equivalent water column height) given by

$$h_\alpha = \frac{p_\alpha}{\rho_{0w} g},$$

where ρ_{0w} is the density of water at standard temperature and pressure, and g is the magnitude of the acceleration due to gravity.

Often the nonlinear conductivity tensor $\mathbf{K}_\alpha$ is used in hydrology. It is defined by

$$\mathbf{K}_\alpha = \frac{\rho_{0w} g \mathbf{K} k_{r\alpha}}{\mu_\alpha} = \mathbf{K}_{\alpha_s} k_{r\alpha},$$

where $\mathbf{K}_{\alpha_s}$ represents the conductivity when the porous medium is saturated with fluid α. Then (4.3) can be written as

$$\mathbf{u}_\alpha = -\mathbf{K}_\alpha \left(\nabla h_\alpha - \frac{\rho_\alpha}{\rho_{0w}} \mathbf{i}_z \right), \tag{4.4}$$

where $\mathbf{i}_z$ is the unit vector oriented in the direction of the gravity force in an orthogonal cartesian coordinate system.

4.1.3 Constitutive relations

A set of constitutive relations is needed to close the system of equations (4.1) and (4.3). Most frequently these represent attempts to fit experimental data by an empirical relation. Such relations can be formulated in a variety of ways depending the choice of independent variables. Perhaps the most natural choice of independent variables are the separate phase pressures and saturations (cf. [8]). In this section we consider three constitutive relations: capillary pressure-saturation, relative permeability-saturation, and density-pressure.

A capillary pressure-saturation relation

It is well known that fluid saturation is a function of the difference of the two phase pressures in the system. The pressure difference is called *capillary pressure* and is defined by

$$p_c = p_a - p_w.$$

Correspondingly,

$$h_c = h_a - h_w.$$

These relations involve effects of hysteresis (cf. [7]) which are important phenomena that influence fluid behavior in the unsaturated zone and should be taken into account. However, in this section we shall only indicate some basic features of the capillary pressure-saturation relation, and for this reason we shall consider a simple but useful characterization that ignores hysteresis effects. Experimental data corresponding to a water-air system is shown in Fig. 4.1. One of the most commonly used fitting function forms is that of van Genuchten [96]. It is given by

$$\theta(h_c) = \frac{\theta_{ws} - \theta_{wr}}{|1 + (\delta h_c)^n|^{1-1/n}} + \theta_{wr}. \tag{4.5}$$

Here θ_{ws}, θ_{wr}, δ, and n are fitting parameters. We note that the air fluid content can be obtained easily from the water content through the assumption that the fluids fill the volume, i.e.

$$\theta_w + \theta_a = \phi,$$

or equivalently

$$S_w + S_a = 1.$$

The relative permeability-saturation relation

Another constitutive relation is the relative permeability as a function of the saturation. This function is usually empirically determined. The common approach is to use fitting parameters from the capillary pressure-saturation relation. Examples of such function forms (cf. [8]) are given by

$$k_{rw}(\theta_w) = \theta_e^{1/2}\left(1-(1-\theta_e^{1/m})^m\right)^2,$$

where $m = 1 - 1/n$ and

$$\theta_e = \frac{\theta_w - \theta_{wr}}{\theta_{ws} - \theta_{wr}}.$$

Correspondingly,

$$k_{ra}(\theta_w) = (1-\theta_e)^{1/2}(1-\theta_e^{1/m})^{2m}.$$

The graphs of $k_{rw}(\theta_w)$ and $k_{ra}(\theta_w)$ are illustrated in Fig. 4.1.

The density-pressure relation

The last constitutive relation we need to specify is the one between the fluid density and the corresponding pressure. Because of the particular fluids considered, these constitutive equations can be simplified without affecting the adequateness of the resulting model. The water is nearly incompressible, in particular when compared to the air. Moreover, we already assumed that the media is isothermal and so none of the phase densities will be affected by temperature changes. As a result we can consider a system of two fluids in which only the air phase is compressible with density changing linearly with respect to the air pressure, i.e.

$$\rho_a = \rho_{0a}\left(1+\frac{h_a}{h_{0a}}\right),$$

where ρ_{0a} is the density of air at pressure h_{0a}.

4.2 Mathematical models of fluid flow in porous media

In this section we introduce four flow models that reflect different levels of sophistication with respect to the modeled physical phenomena. Correspondingly, we shall discuss techniques for discretization and numerical solution for each of them. Consistent with the considerations in the previous chapters, we shall assume that the physical domain Ω where the fluid flow is modeled has a polyhedral shape.

4.2.1 A saturated flow model

The simplest and the most popular model is that of a fully saturated, incompressible porous media. In this case the water (or the liquid) phase occupies the whole pore space and the flow is due to the nonuniform pressure distribution. The mathematical formulation of a steady state flow is based on the mass balance equation (4.1) and Darcy's law (4.3):

$$\nabla\cdot(\rho_w\mathbf{u}_w) = F_w, \qquad \text{in}\,\Omega, \tag{4.6a}$$

$$\mathbf{u}_w = -\frac{\mathbf{K}}{\mu_\mathbf{w}}(\nabla p_w - \rho_w\mathbf{g}), \quad \text{in}\,\Omega, \tag{4.6b}$$

with an appropriate standard boundary conditions on the boundary $\partial\Omega$. Examples of those are Dirichlet, Neumann, and Robin for p_w and $\mathbf{u}_w$ on $\partial\Omega$. The assumption about the incompressibility of water reduces (4.6) to a linear elliptic equation of the form (2.19). Alternatively, one can eliminate the flux $\mathbf{u}_w$ to obtain the elliptic problem (2.6). As we pointed out in Chapter 2, the mathematical properties of these simple models are well understood. Clearly, both Galerkin and mixed discretization can be applied here and

the iterative techniques considered in Chapter 3 are guaranteed to work very efficiently. Temporal effects can be accommodated easily in (4.6) by introducing a term $\eta\dfrac{\partial p_w}{\partial t}$ which leads to a model very similar to (2.37). The function η here is used to model certain heat compressibility or storativity effects. The discretization and iterative solutions of the time-dependent problem are also considered in detail in the previous chapters.

4.2.2 A two-pressure equation formulation

The mass balance statement (4.1) for each fluid phase together with Darcy's law (4.3) and the constitutive relations from Section 4.1.3 define a coupled system of equations that models two-phase flow in porous media. It is written as

$$\frac{\partial(\phi\rho_\alpha S_\alpha)}{\partial t} + \nabla\cdot(\rho_\alpha \mathbf{u}_\alpha) = F_\alpha, \qquad \text{in}\,\Omega, \tag{4.7a}$$

$$\mathbf{u}_\alpha = -\frac{\mathbf{K}k_{r\alpha}}{\mu_\alpha}(\nabla p_\alpha - \rho_\alpha \mathbf{g}), \quad \text{in}\,\Omega, \tag{4.7b}$$

where $\alpha = a, w$. A corresponding set of independent boundary conditions for each phase equation together with the constitutive relations complete this model.

One principle difficulty in solving (4.7) is in the linearization procedure used for solving these nonlinear equations. In general, there are two widely used techniques: Picard and Newton-Raphson. Even though a definite assessment of the advantages of these two approaches is not available, it appears that the Picard method is very robust and reliable but perhaps somewhat slow.

Another difficulty lies in the coupling between the air and water equations. A standard approach used in the petroleum industry is to decouple them by solving implicitly for one of them and explicitly for the other (IMPES). The IMPES schemes work well when the coupling between phases is very weak. They produce poor results when the coupling is strong. In addition, they result in very inefficient time-stepping procedures and should be avoided. A much better approach is to use a fully implicit scheme in both phases combined with a Picard iteration for resolving the nonlinearities of the coupling (cf. [8]).

The equations can be discretized using mixed methods to produce mass conservative approximations. On the other hand, the corresponding fluxes can be eliminated and backward Euler-Galerkin discretizations can be applied. Binning (cf. [8]) has observed that in the latter case, the mass lumping technique for treating the time derivative term is essential for preventing oscillations in the numerical results. Even though the derivation of this model is based on the mass conservation principle, applying Galerkin discretization is not guaranteed to produce a locally conservative scheme in contrast to the mixed method. Also, if the fluxes have to be computed to provide coupling with a transport equation, the mixed method is far superior than the Galerkin technique, especially on nonorthogonal grids. At each time level, the linearized discrete problem for each phase equation can be solved efficiently using the methods developed in Chapter 3.

The formulation of the two-phase flow model (4.7), however, exhibits certain deficiencies that must be taken into account when assessing the robustness of the model in various applications. Equations (4.7) are applied in both the saturated and the unsaturated zones. Examining the relative permeability curves in Fig. 4.1, it is easy to see that the air equation has a very small elliptic term near the water table and degenerates in the saturated zone. Because of this, the model defined in (4.7) behaves badly as a mathematical model, and no reliable flow simulators can be built using it.

4.2.3 Richards equation

A very useful model for the unsaturated/saturated zones which is closely related to the two-pressure model (4.7) can be obtained by eliminating the air equation. This is based on the assumption that the air phase remains at atmospheric pressure. Such an assumption is reasonable in many cases because the mobility of air is much larger than that of water. We should note that such an assumption does not imply that the air phase is stagnant but just the opposite, i.e. the air has a very high mobility. Often, the resulting model is referred to as an implicit two-phase model.

Assuming that the air pressure is a known constant, (4.7) can be rewritten as

$$\frac{\rho_w}{\rho_{0w}}\left[\frac{\partial\theta_w}{\partial t}+S_s\frac{\theta_w}{\phi}\frac{\partial h_w}{\partial t}\right]-\nabla\cdot\mathbf{K}_w\left(\nabla h_w-\frac{\rho_w}{\rho_{0w}}\nabla\mathbf{i}_z\right)=\frac{1}{\rho_{0w}}F_w,\quad \text{in}\,\Omega. \tag{4.8}$$

Here S_s is the specific storativity of water. If a coupling with contaminant transport equation is considered, a term $\frac{\theta_w}{\rho_{0w}}\frac{\partial\rho_w}{\partial c}\frac{\partial c}{\partial t}$ describing the rate of change of density of water with respect to the concentration c of the contaminant should be added to the left-hand side of (4.8). A typical way of linearizing the coupling with a transport equation is to treat the term $\frac{\theta_w}{\rho_{0w}}\frac{\partial\rho_w}{\partial c}\frac{\partial c}{\partial t}$ as a forcing term by providing a constitutive relation $\rho_w=\rho_w(c)$.

We can easily rewrite the Richards equation (4.8) in a mixed form using the alternative form of the Darcy's law (4.4) to define a flux of water $\mathbf{u}_w$.

It should be noted that (4.8), or its mixed form, are typical parabolic equations that model the flow in the saturated and unsaturated zones. It can be discretized and solved after linearization very similarly to the approach described in the previous section.

4.2.4 A fractional flow model

There are important practical cases when the Richards equation (4.8) is insufficient to adequately describe the flow process. Examples are vapor extraction systems or soil venting in which there is a substantial dynamic interaction between the two phases and the contaminant can be transported both in the air and water phases (cf. [8, 32, 36]). Another example where the air phase has to be solved explicitly is the presence of injection wells which pump fluid into the porous media at high pressure. In such situations the coupled nonlinear system for the air-water complex must be considered. On the other hand, we already pointed out in Section 4.2.2 that a model better than (4.7) has to be devised for reliable simulation.

Here, we present a fractional flow formulation of the two-phase fluid flow model (4.7). This approach results in a mathematical problem which is well behaved when solved numerically. The fractional flow formulation involves a global pressure p and total velocity $\mathbf{u}$. This provides a two-phase water (w) and air (a) flow model which is described by the following equations (cf. [34]):

$$C(p,S_w)\frac{\partial p}{\partial t}+\nabla\cdot\mathbf{u}=f(p,S_w),\qquad \text{in}\,\Omega, \tag{4.9a}$$

$$\mathbf{u}=-\mathbf{K}\lambda(\nabla p-\mathbf{G}_\lambda),\quad \text{in}\,\Omega, \tag{4.9b}$$

and

$$\frac{\partial(\phi\rho_w S_w)}{\partial t}+\nabla\cdot\rho_w(f_w\mathbf{u}-\mathbf{K}\lambda_a f_w\delta\rho\mathbf{g}-\mathbf{D}(S_w)\cdot\nabla S_w)=F_w,\ \text{in}\,\Omega. \tag{4.9c}$$

The variables participating in this model are defined as follows. Equation (4.9a) represents a mass balance statement for the total fluid mass whereas (4.9b) is a generalized Darcy's law. The global pressure p and total velocity $\mathbf{u}$ are defined by (cf. [34]):

$$p=\frac{1}{2}(p_w+p_a)+\frac{1}{2}\int_{S_c}^{S_w}\frac{\lambda_a-\lambda_w}{\lambda}\frac{dp_c}{d\xi}d\xi, \tag{4.10}$$

and

$$\mathbf{u}=\mathbf{u}_w+\mathbf{u}_a. \tag{4.11}$$

The coefficient $C(p,S_w)$ in (4.9a) is given by

$$C(p,S_w)=\frac{\phi S_w}{\rho_a}\frac{d\rho_a}{dt}. \tag{4.12}$$

The right hand side term $f(p, S_w)$ of (4.9a) is defined by

$$f(p,S_w) = \frac{1}{\rho_a}(F_a - \mathbf{u}_a \cdot \nabla\rho_a - \phi S_a \frac{\partial \rho_a}{\partial t}) + \frac{1}{\rho_w}(F_w - \mathbf{u}_w \cdot \nabla\rho_w - \phi S_w \frac{\partial \rho_w}{\partial t}), \tag{4.13}$$

where $\lambda = \lambda_w + \lambda_a$ is the total mobility, and $\lambda_\alpha = \dfrac{k_{r\alpha}}{\mu_\alpha}$, $\alpha = w, a$, is the mobility for water and air, where $k_{r\alpha}$ is the relative permeability. The capillary pressure p_c is given by $p_c = p_a - p_w$. The gravity forces $\mathbf{G}_\lambda$ and capillary diffusion term $\mathbf{D}(S)$ are expressed as

$$\mathbf{G}_\lambda = \frac{\lambda_w\rho_w + \lambda_a\rho_a}{\lambda}\,\mathbf{g} \qquad \text{and} \qquad \mathbf{D}(S) = -\mathbf{K}\lambda_a f_w \frac{dp_c}{dS}.$$

The phase velocities for water and air, which are needed in transport calculations, are given by:

$$\mathbf{u}_w = f_w\mathbf{u} + \mathbf{K}\lambda_a f_w \nabla p_c - \mathbf{K}\lambda_a f_w \delta\rho\mathbf{g}, \tag{4.14a}$$

$$\mathbf{u}_a = f_a\mathbf{u} - \mathbf{K}\lambda_w f_a \nabla p_c + \mathbf{K}\lambda_w f_a \delta\rho\mathbf{g}, \tag{4.14b}$$

where $f_\alpha = \lambda_\alpha/\lambda$, $\alpha = w, a$, and $\delta\rho = \rho_a - \rho_w$. To complete the model, we assume the constitutive relations between capillary pressure and saturation, between relative permeabilities and saturation, and between fluid density and pressure discussed in Section 4.1.3. Notice that the phase velocity for air is given by (4.14b) even if the Richards approximation is used.

The boundary conditions are an important element of the above model. Standard types of boundary conditions, namely Dirichlet, Neumann, and Robin, may be defined for the pressure-saturation formulation of the flow model (4.9). Let the boundary $\partial\Omega$ be partitioned into nonoverlapping parts $\partial\Omega^i$, $i = 1, 2$. Then boundary conditions for (4.9a)-(4.9b) may be given by a combination of the following expressions:

$$p = p_{\partial\Omega}(x,t), \qquad x \in \partial\Omega^1, \tag{4.15a}$$

$$\mathbf{u}\cdot\boldsymbol{\nu} + b(x,t,S_w)p = G_{\partial\Omega}(x,t,S_w), \quad x \in \partial\Omega^2, \tag{4.15b}$$

where $\boldsymbol{\nu}$ is the outward normal vector to the corresponding boundary part and $p_{\partial\Omega}(x,t)$, $b(x,t,S_w)$, and $G_{\partial\Omega}(x,t,S_w)$ are given functions. Obviously the boundary conditions (4.15) constitute a well posed problem from a mathematical point of view. However, in applications such types of boundary conditions may not be available. This comes from the fact that the physical boundary conditions are specified with respect to the separate phase equations (4.7). Thus, boundary conditions for the fractional flow model must be derived using the separate phase boundary conditions. Several difficulties are encountered here and because of this we shall discuss this issue in more detail in Section 4.3.1.

A corresponding set of boundary conditions for the saturation equation (4.9c) must be specified. From a physical point of view, a Dirichlet condition for the saturation imposed on the boundary $\partial\Omega$ makes very good sense. Of course, boundary conditions of the form (4.15) make perfect mathematical sense here, but their physical meaning is not necessarily well defined.

Since the term $\mathbf{K}\lambda$ does not vanish in the saturated and unsaturated zones, the model (4.9) is better behaved mathematically. Moreover, existence and uniqueness of solutions to (4.9) has been established for the incompressible case (cf. [34]). The compressible case, however, remains still an open problem. To discretize (4.9) we recall that three main factors play an important role in selecting our discretization strategy: (a) the mass conservation expressed by the differential equations; (b) the geometry of the domain; (c) nonlinearities in the model and their linearization. There are nonlinearities on multiple levels in the coupled flow model (4.9): within each of the equations and between the pressure and saturation equations. One way of solving this system is to first linearize each equation by lagging in time the setup of the coefficients in order to get an initial guess. After that a Picard or Newton-Raphson iteration can be applied in order to resolve the nonlinearities. Such an approach has been used successfully in [8] in the case of unsaturated flows. Other approaches to linearization are discussed in [34, 47].

The mathematical nature of the saturation (4.9c) and pressure (4.9a)-(4.9b) equations are different and specific methods for their approximation should be considered. Typically, the saturation equations are convection dominated and thus special care should be taken in their discretization. Also, the diffusion terms there are small but important and cannot be neglected. On the other hand, the pressure equation has a strong elliptic part and this fact should influence the choice of the discretization method.

Based on these observations, two types of finite element approximations can be used: the standard conforming Galerkin method and the mixed method. Advantages of the former are its simplicity, its smaller number of unknowns, and the availability of efficient methods for solving the resulting system of linear equations (cf. Chapters 2 and 3). These features are particularly important for three-dimensional problems. Combined with upstream weighting (cf. [90]), Godunov type approximations, or Riemann solvers (cf. [69]), such discretizations can be used for the saturation equations (for a combination with the mixed method see [67]). The characteristic methods (cf. [47]) are very important here because they symmetrize the discrete systems and thus allow efficient iterative solution. Such results are reported in [8]. An enhanced performance results from the use of logically rectangular grids.

Among the disadvantages of the conforming discretizations are the lack of local mass conservation of the numerical model and some difficulties in computing the phase velocities needed in the transport and saturation equations. Clearly, accurate velocities are needed in the saturation equation (4.9c). The straightforward numerical differentiation is far from justifiable in problems formulated in highly heterogeneous medium with complex geometry. On the other hand, the mixed finite element method offers an attractive alternative, as we already observed in Chapter 2. Besides that, efficient iterative methods for solving the corresponding discrete systems are developed in Chapter 3. Because of all this, a mixed discretization of the pressure equations (4.9a)–(4.9b) can be applied.

4.3 Fractional flow models with special features

We touched on some of the difficulties related to the boundary conditions of the fractional flow model in Section 4.2.4. In this section we shall consider in some detail the issues about setting up realistic boundary conditions for this model and ways to incorporate wells into it.

4.3.1 Boundary conditions

The difficulties coming from the boundary conditions for the fractional flow model arise from the fact that the separate phase equations (4.7) come with an independent set of boundary conditions. In addition, the independent variables in (4.9) are the total pressure p and the total flux $\mathbf{u}$, neither of which is a physical quantity but rather an abstract mathematical variable. Consequently, in most applications measuring values for these variables is virtually impossible. Thus, the most practical way for setting up boundary conditions for this model is to derive them from the separate phase boundary conditions.

Let us denote by $\partial\Omega^{i,\alpha}$, $i = 1, 2$, the nonoverlapping regions on which Dirichlet and Robin boundary conditions are assigned for phase α correspondingly. Here we assume that the pure Neumann conditions are a special case of Robin conditions. Clearly, there are four possibilities such that different types of conditions are assigned for the phases over a particular region. Some of these combinations represent important cases used to model interesting physical phenomena. An example is *ponding* of water on the surface of the ground (cf. [33]). With a heavy rain or flooding, if the infiltration of water in the soil occurs at a limited rate, a layer of water builds upon the surface of the ground. This is shown in Fig. 4.2 (a). To model this situation, one observes that because of the water layer, no air can escape from the ground and so $\mathbf{u}_a \cdot \boldsymbol{\nu} = 0$. On the other hand, since the thickness of the water layer d is known, the water pressure at the surface of the ground can be computed by $p_w = \rho_w g d$. Thus, we get a combination of different types of boundary conditions for the phases. Similarly, when the water layer becomes thin enough due to dry weather conditions, the air trapped underground, which is at a higher pressure, breaks through the water film (see Fig. 4.2 (b)). At this particular instance the air pressure on the surface is equal to atmospheric, i.e. $p_a = p_{\text{atmos}}$. Also, the water film is very thin, so we can set the water flux infiltrating the ground to the infiltration rate for the specific soil type. Therefore, the condition $\mathbf{u}_w \cdot \boldsymbol{\nu} = \mathbf{G}$, with $\mathbf{G}$ given, can be used here. Again, we obtain a combination of different boundary conditions for the phases.

Attempts to derive boundary conditions for the total variables for all possible combinations of conditions lead to strongly coupled and nonlinear types of boundary conditions that involve the total flux, total pressure, the saturation of water and its gradient on the boundary $\partial\Omega$. In addition, we already mentioned that only Dirichlet conditions for the saturation equation have well defined physical meaning.

In order to overcome these difficulties we have devised an alternative approach. The key observation here is that if Dirichlet boundary conditions are specified for both phases on a given subregion of $\partial\Omega$, then we can formulate Dirichlet boundary conditions for the total pressure p and for the saturation of water S_w. Indeed, given p_a and p_w, we get $p_c = p_a - p_w$. Given p_c, we compute h_c and in turn S_w, since the relation $h_c = h_c(S_w)$ is one-to-one in the interval $[0,1]$ (cf. Fig. 4.1). Then, the total pressure p in (4.10) is computable. Therefore, we obtain Dirichlet data for p and S_w.

This reasoning is used to devise an iterative scheme for imposing all possible combinations of boundary conditions. Suppose that we want to model the ponding condition, i.e. $\mathbf{u}_a \cdot \boldsymbol{\nu}$ and p_w are given. Let p_a^0 be an initial guess for the air pressure. Using the above scheme, we setup p^0 and S_w^0. Using these Dirichlet conditions, the model is solved to get $\mathbf{u}^0$ and p^0. Next, we compute $\mathbf{u}_a^0 \cdot \boldsymbol{\nu}$ by (4.14b). From here, a correction $\hat{p}_a^0 = \boldsymbol{\nu} \cdot (\mathbf{u}_a - \mathbf{u}_a^0)$ and the next iterate $p_a^1 = p_a^0 + \beta \hat{p}_a^0$ are computed, where β is an appropriately chosen iteration parameter. The convergence properties of such an iterative scheme for imposing boundary conditions are very good in the case of mixed discretizations of model elliptic or parabolic problems (cf. [80]). It turns out that the iteration matrix for this iteration is symmetric and positive definite and thus, the preconditioned conjugate gradient can be used to accelerate the convergence. The corresponding result for the fractional flow model (4.9) has not yet been established analytically but preliminary experiments suggest that the above procedure is reliable. Moreover, it is based on the right physical background and results in physical Dirichlet conditions for the saturation equation. In fact, using this iterative scheme for the boundary conditions makes the equation (4.9c) invisible for the setup of the model in terms of the boundary data and simplifies considerably the implementation of the model.

4.3.2 Wells

Another important aspect of the fractional flow model is the adoption of adequate well models. The wells often play an essential role in determining the groundwater flow and for this reason their nature as generators of certain types of flow behavior must be understood very well. A widespread consensus is that the most common type of extraction/injection well used in field applications is one consisting of a screened subsurface region from which fluid is being extracted or injected at a known pump rate. Since the inside of the screened region does not contain porous media, the flow there is determined by the Navier-Stokes equations. A good formulation of the flow model requires coupling Navier-Stokes flow to the Darcy flow outside. However, such a coupling is a very challenging mathematical problem and for this reason various simpler models have been proposed. Perhaps the simplest way of simulating a partially (or fully) penetrating well is to treat the well surface Γ_{well} as an additional boundary where the prescribed pumping rate is distributed in some fashion. One problem, though, is that in general the correct flux distribution is not known. If the flux is assumed to be distributed uniformly, then one can consider this to be a "constant flux" well model. Alternatively, the flux can be distributed linearly with respect to the depth z. Another popular assumption is that the hydraulic pressure head along the well is constant but unknown. This is often referred to as a "constant head" well model. Mathematically, these models can be given by

$$\int_{\Gamma_{well}} \mathbf{u} \cdot \boldsymbol{\nu} = q(t), \quad \text{and} \quad p - \rho g z = \text{Const}(t), \tag{4.16}$$

where $t > 0$, z is the vertical spatial direction pointing downward, $\boldsymbol{\nu}$ is the outward normal vector to the well surface, g is the acceleration constant due to gravity, $q(t)$ is the pumping rate, and $\text{Const}(t)$ is the unknown pressure value that may change in time. The constant head model results in variable extraction rates on the well surface. We note that the use of the global pressure and flux are fully justifiable in this case since the flux through the well is actually the total flux. In addition, near or inside the well bore, the pressures of both phases are practically the same.

In practice, such well models can be implemented within a mixed finite element discretization by choosing a column of grid cells to constitute the well. This, however, may result in "wells" of enormous diameter if the grid is not fine enough. Another difficulty is that the wells generate important flow behavior which must be resolved very accurately. Thus, the techniques for local refinement, which we considered in Section 2.4, are crucial for the implementation of accurate well models. Typically, the grid and the time-step are refined around each well location to provide high accuracy and efficiency of the numerical approximation.

There are other well models that are based on analytical solutions of simple one dimensional steady state flow problems (cf. [82]). However, the assumptions needed to justify such models do not hold even in simple multidimensional simulations. Because of this, we shall not consider such approaches.

4.4 A multiphase flow simulator

In this section we discuss briefly the development of a large scale numerical simulator of multiphase fluid flow in porous media. The author of this dissertation was one of the main code developers of the Partnership in Computational Sciences (PICS) where he worked on the flow module (cf. [79]). PICS is an initiative sponsored by the U.S. Department of Energy. The goal of this project is to develop a state-of-the-art computer simulator of groundwater flow and contaminant transport (GCT).

GCT simulates the flow and reactive transport of subsurface fluids through a heterogeneous porous medium of irregular geometry. This simulator is designed to run on a massively parallel, distributed memory computers as well as on conventional serial machines. The flow module is a major part of the PICS code development effort. The flow module evolved from a model based on the Richards equation (4.8) to a model that solves the two-phase equations (4.7). A Galerkin discretization is used for the earlier versions of GCT. A mixed discretization is used for the pressure equations of the two-phase model whereas the saturation equation is discretized by an upstream weighted Galerkin method. A detailed description of this code can be found in [35].

The approach taken in defining the triangulation of the computational domain is based on first introducing an underlying logically rectangular grid. Such grids offer perhaps the most economical way to maintain a simple data structure and to build finite element approximations with a minimal number of unknowns. It essentially simplifies many coding issues and yet allows complex geometries to be handled. In fact, the computational grid can be as complex as any reasonable union of logically rectangular structures including toroidal or L-shaped domains.

The logically rectangular grid is used for the Galerkin finite element method. To define the mixed method, each grid cell is further split into five tetrahedra. When the lowest order Raviart-Thomas spaces are used, one pressure and four velocity unknowns are attached to every tetrahedron in the grid. It is clear that the numerical solution of such models requires extensive memory and CPU resources. Only supercomputers appear to be capable of solving these numerical model in reasonable time. The distributed memory architectures such as Intel's Paragon are quite convenient in view of the domain decomposition approaches to the solution of the discrete problems.

A domain decomposition approach is used in order to utilize these machines. The original computational domain is decomposed into a set of logically rectangular structures each of which is attached to a single processor. Then a corresponding parallel algorithm for solving the problem is applied.

The system for remote procedure calls, developed at the Brookhaven National Laboratory (cf. [71]), has been used for the parallelization of the flow code. This system allows the development of parallel codes in a style that is very close to the common serial style of writing numerical codes, and for this reason considerably reduces the complexity of the software for distributed architectures. In addition, the resulting software is less dependent on the vendor supplied primitives for parallel processing, which is important for portability. Another interesting feature of this system is that it combines the exchange of data with a specification of a method for processing it on the remote processor. This leads to a more enhanced software environment for developing parallel programs.

We conclude this chapter with computational results from a simulation of a groundwater flow and contaminant transport problem. The geological data for the simulation was provided by Michael Celia (Princeton). The purpose of this simulation is to study the contamination process of a large site polluted

by a very dense contaminant penetrating into the aquifer. The chemical waste is dumped into an open field storage pit on the surface of the aquifer. The horizontal dimensions of the site are 4700 by 5000 feet. The maximum depth is approximately 80 feet. The simulated site has a relatively homogeneous soil type (clay). Its bottom is an impermeable bedrock with complicated geometry as is indicated in Fig. 4.3. The bedrock elevations are provided through field measurements. There is an underground water flow that determines to a great extent the contaminant movement. In the simulation it is setup by appropriate boundary conditions. The pressure distribution near the surface is shown in Fig. 4.4. It indicates flow which is diagonally oriented with respect to the domain. The contaminant can be dissolved in water at normal (for the rain) temperature. Thus, the rain water dissolves the pollutant and results in a very dense liquid which moves through the unsaturated zone to the saturated zone of an almost homogeneous porous medium. There are no wells in this simulation and the processes occur close to the surface of the ground. Therefore, the unsaturated/saturated flow is adequately described by the Richards equation (4.8). The site is discretized by a large grid with 137×129×11 grid nodes.

There are only natural forces involved in the simulation: groundwater flow, gravity, diffusion. Because of this, the processes develop very slowly in time. The simulation spans a period of 20 years discretized by an implicit time-stepping with each time step equal to 5 days. Also, the contaminant is very dense which makes the term $\rho_w \frac{\partial \rho_w}{\partial c} \frac{\partial c}{\partial t}$ in Richards equation (4.8) act as a strong forcing term in addition to the existing gravity forces. The interaction between these natural forces is very important in determining the spreading of the contamination. The effects of the interaction between the groundwater flow and the diffusion on the shape and main direction of movement of contaminant iso-surfaces are clearly seen in Fig. 4.5 and Fig. 4.6. The water flow is the predominant factor here, which is reflected by the shape of the 0.5% iso-surface.

Since the contaminant is dense, in the presence of gravity and relatively small diffusion and advection, it moves predominantly downwards. This is very similar to a situation when honey is poured into a glass of cold water. The honey moves straight down until it reaches the bottom and then spreads around in the bottom area. Such a behavior is observed in Fig. 4.7, where a vertical slice along the Y-axis is shown. The slice is located exactly at the end of the source area. High concentrations are observed near the top and the bottom. In the middle, relatively low concentration levels are observed due to the "pushing" effect of the groundwater flow. The profile of the contaminant spreading near the bottom is very clearly seen in Fig. 4.7. The geometry of the impermeable bedrock plays an interesting role in determining the direction of contaminant movement. The balance between the gravity, the diffusion, and the advection combined with the geometry of the bottom shows that contaminant movement in a direction opposite to the main flow direction is possible. Such effects are indicated in Fig. 4.7 near the bottom where a contaminant movement in the direction towards the corner of the domain can be seen. We observe high concentrations there even though the advection "pushes" the pollutant in the opposite direction. The simulation is essential in understanding why the contaminant flowed upstream but "down" the bedrock slope. Similar results are observed in Fig. 4.8 and Fig. 4.9 where slices along the X-axis near both ends of the source region are shown. The movement of contaminant near the bottom follows closely the geometry of the bedrock. This is seen particularly well in Fig. 4.9.

The computational results fully agree with the underlying physical principles and are in a good agreement with experimental results obtained by measuring concentrations in real sites with similar conditions.

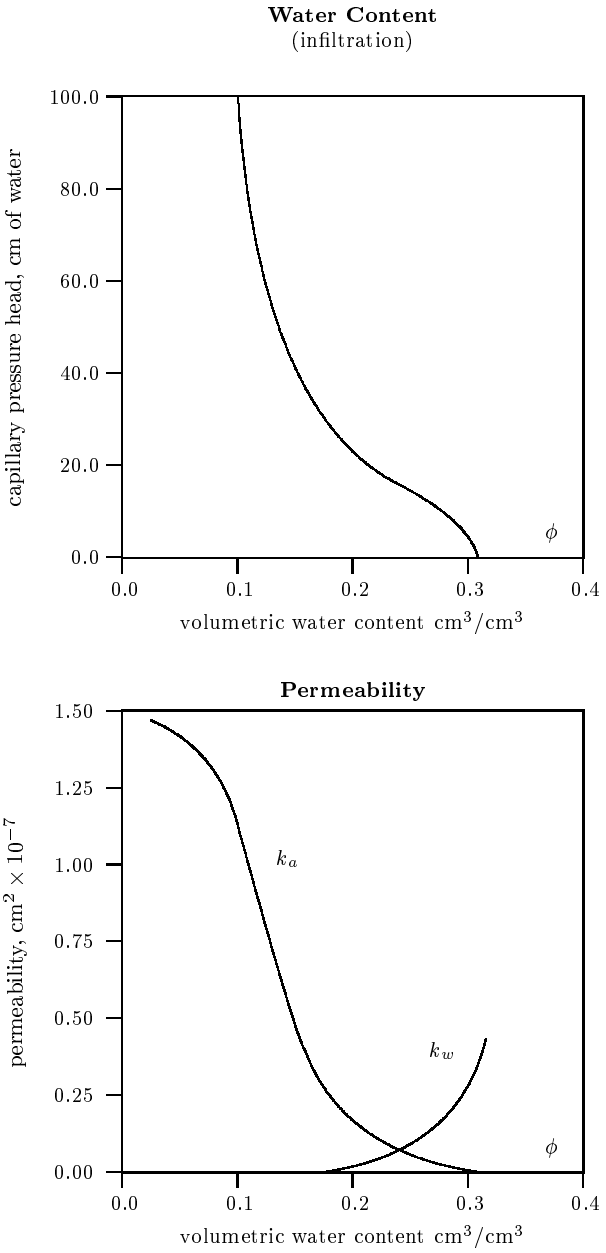


Figure 4.1: Capillary pressure and relative permeability as functions of saturation for the experimental data of Touma and Vauclin [95]

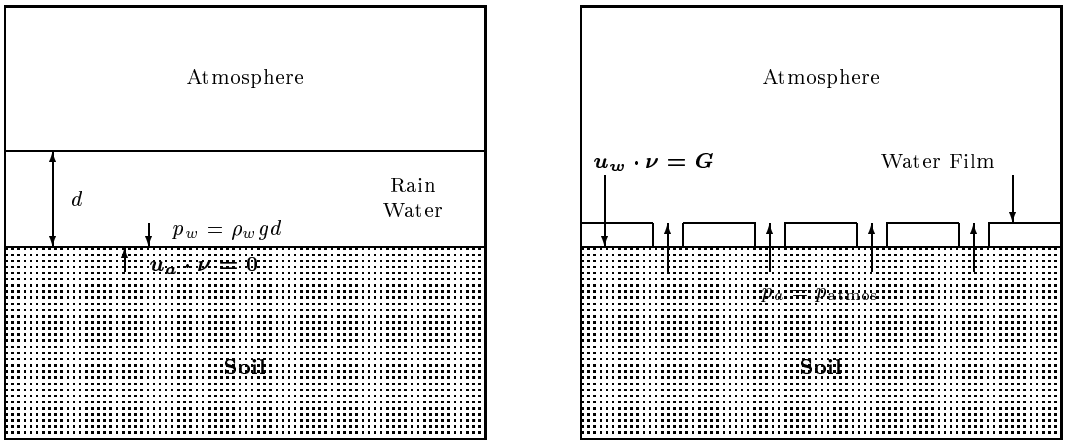


Figure 4.2: Ponding

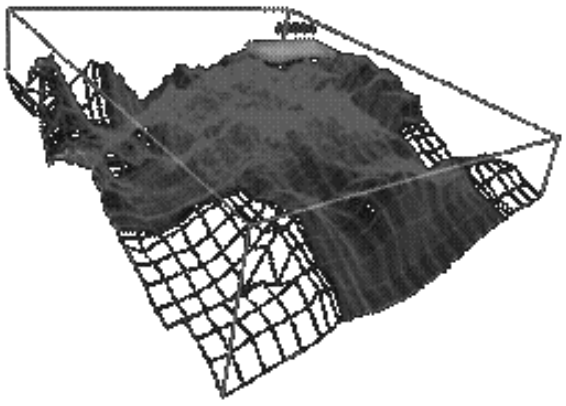


Figure 4.3: 3-D simulation: Initial condition of iso-surface 0.5%

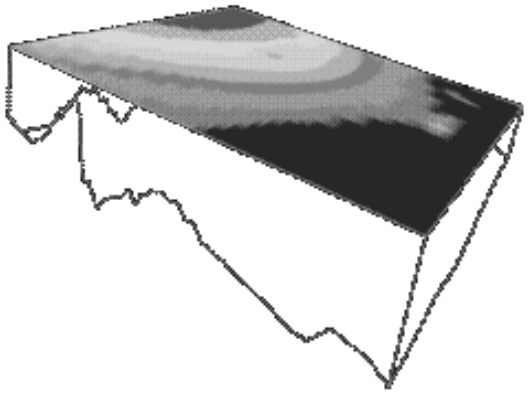


Figure 4.4: 3-D simulation: Pressure distribution near the surface

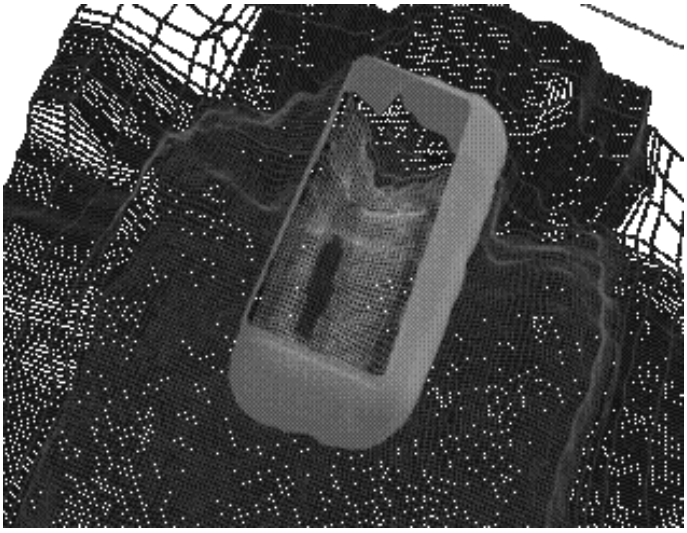


Figure 4.5: 3-D simulation: Iso-surface 0.5% after 2 years

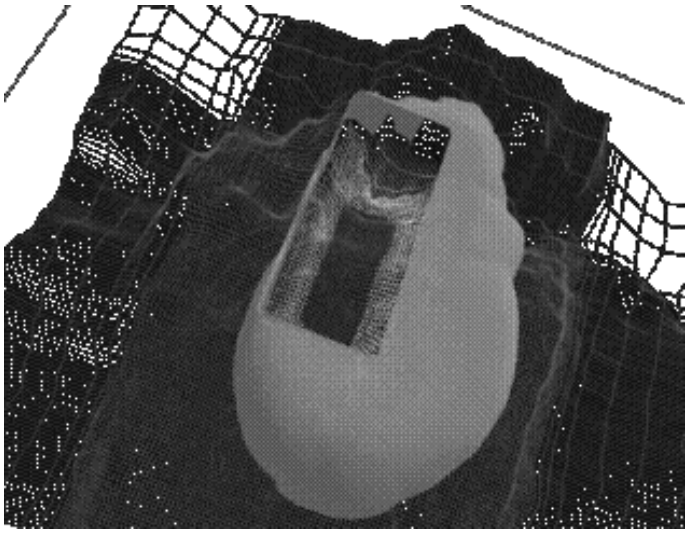


Figure 4.6: 3-D simulation: Iso-surface 0.5% after 20 years

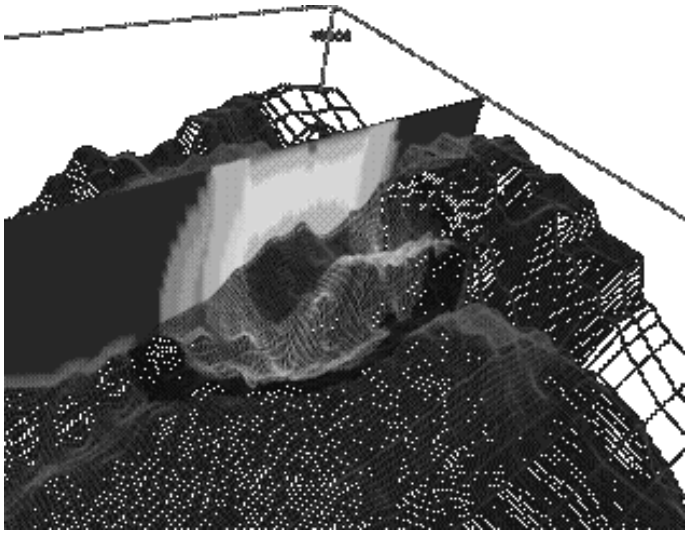


Figure 4.7: 3-D simulation: Vertical slice along the Y axis

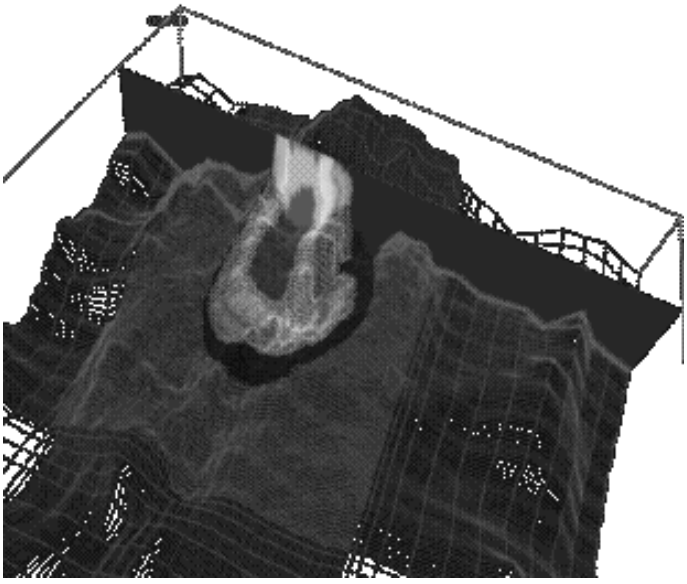


Figure 4.8: 3-D simulation: Vertical slice along the X axis (small X)

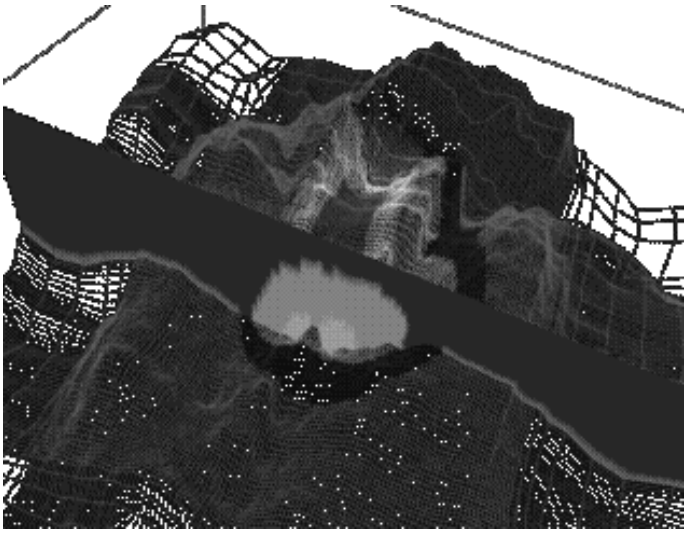


Figure 4.9: 3-D simulation: Vertical slice along the X axis (large X)

Chapter 5

Conclusions

The theory developed in this dissertation combined with the considered applications represent an advanced methodology for discretization and efficient solution of difficult real-life problems. Progress is made in several directions, both theoretical and computational.

In addition to the existing standard finite element discretizations, the new discretization techniques from Section 2.4 which utilize refinement in time and space are very useful in groundwater flow applications for they allow the interesting local flow behavior to be resolved very accurately. The analysis of the backward Euler schemes is sharp and the theoretical results are supported by numerical experiments. These results are better than similar results available in the literature which exhibit either stability or accuracy deficiencies. The numerical experiments reported in Section 2.4.6 suggest that this local time-stepping technique can be extended successfully to Crank–Nicholson type schemes. A different approach to the analysis of such schemes needs to be developed. Even though the author is aware of recent progress in this direction from a private communication with Joseph Pasciak, the question of obtaining error estimates in the natural norms for parabolic problems is still open. Another challenge that has to be addressed in the near future is the efficient implementation of the refinement algorithms, in particular on distributed parallel computers. The implicit time stepping combined with local refinement in time and space typically results in enormous computational problems, which can be solved efficiently only if adequate data structure and algorithms for manipulating it are constructed. The development of such computational technology would allow adaptivity of the refinement and would result in far more effective numerical approximations. In view of the targeted three dimensional flow applications, this is very important.

In Chapter 2 we saw that a large class of finite element discretizations lead to, or can be reduced to, symmetric and positive definite problems. We also observed that the development of efficient preconditioners applied to the iterative solution of such problems is the key for efficiency. The nonoverlapping domain decomposition preconditioners developed in Section 3.2 are very effective and versatile. These algorithms require minimal computational resources and exhibit attractive preconditioning effects, independent of jumps of the operator coefficients across the subdomain boundaries. Even though these preconditioners are not uniform, they are efficient and very practical for implementation. For these reasons they are quite useful for problems that typically arise in reservoir simulations. To further improve the theoretical bounds for the asymptotic rate of growth of the condition number, the possibility of incorporating Chebychev iteration for calculating better boundary extensions should be explored.

The inexact Uzawa algorithms considered in Section 3.3 are simple but efficient methods for solving saddle point problems. The abstract approaches to the analysis taken in Section 3.3.1 and Section 3.3.3 result in a general theory that can be applied to a variety of saddle point problems. The new result established for the linear inexact algorithm (cf. Theorem 3.4) relies on minimal assumptions and is very strong. The sufficient condition for convergence of the nonlinear algorithm (cf. Theorem 3.5) relies on the assumption that the approximation to $\mathbf{A}^{-1}$ is accurate enough. Both results are a significant improvement of the existing theory available in the literature (cf. Remark 3.8 and Remark 3.10). The computational results reported in Section 3.3.5 suggest that the requirement for accuracy in the nonlinear algorithm is not a necessary condition and that a further improvement of the result of Theorem 3.5 is

possible. The generality of the theory developed in Section 3.3 is not limited to the saddle point problems considered in this dissertation. In fact, a preliminary research shows that it is possible to extend the technique for analyzing the inexact Uzawa algorithms for the abstract problem (3.57) to the analysis of such algorithms applied to more complex saddle point systems where the operator $\mathbf{A}$ is nonsymmetric and indefinite. Such problems arise, for example, from linearizations and discretizations of Navier–Stokes equations.

We have demonstrated that the new discretization and iterative algorithms developed in this dissertation can be applied successfully to the solution of groundwater flow problems. The Richards equation (cf. Section 4.2.3) provides a very good implicit two-phase flow model for a wide class of environmental problems. The fractional flow model avoids the difficulties associated with the separate pressure two-phase model and results in a well behaved mathematical problem which is more suitable for numerical solution. The mixed finite element discretization is favorable for such applications because of the local mass conservation and accurate velocities this method provides. The Lagrange multiplier reformulation of the mixed system (cf. Section 2.2.3) and the nonoverlapping domain decomposition preconditioners are essential for improving the efficiency of the iterative solution. Alternatively, the inexact Uzawa algorithms can be used for solving the mixed saddle point system.

In many reservoir or remediation process simulations, wells play an important role in forcing the flow of fluids underground. In such cases it is essential to calculate the flow near wells very accurately. The local refinement techniques developed in this dissertation are specially designed for that purpose. The boundary conditions are another important element of any flow model. Many interesting phenomena can be modeled by defining appropriate boundary conditions for the model. The iterative method for imposing boundary conditions on the fractional flow model, considered in Section 4.3.1, has many advantages. It is based on the physical boundary conditions of the separate phases and allows a variety of different cases to be handled in a uniform way. This method is guaranteed to work well in the case of model problems, even though such analysis is not included in the dissertation. Additional theoretical investigation of this approach is needed in the case of the fractional flow model.

A sophisticated flow simulator is built and the results from the computer simulation reported in Chapter 4 are very good. In order to further develop this simulator, research in several directions should be continued. A comprehensive study of the nonlinearities in the fractional flow model is needed in order to better understand the mathematical properties of these equations and improve the iterative methods for resolving them. This will make the computer simulation more efficient and robust. The coupling of the iteration for resolving the nonlinearities coming from the operator coefficients with the iteration for the boundary conditions should be investigated in detail. More advanced well models are needed as well as methods for discretization and iterative solution of the corresponding systems of equations. The possibilities of coupling the Navier–Stokes flow in the well bore with the Darcy flow outside should be investigated both theoretically and computationally. Finally, an extension of the two-phase fractional flow model to a three-phase total pressure flow model needs to be developed. This would allow the modeling of even more complicated practical problems.

Bibliography

[1] R. Adams, *Sobolev spaces*, Academic Press, New York, 1975.

[2] M. Allen, G. Behie, and G. Trangenstein, *Multiphase flows in porous media*, Lecture Notes in Engineering (S. Brebia and S. Orszag, eds.), vol. 34, Springer-Verlag, New York, 1992.

[3] K. Arrow, L. Hurwicz, and H. Uzawa, *Studies in nonlinear programming*, Stanford University Press, Stanford, CA, 1958.

[4] O. Axelsson, *Iterative solution methods*, Cambridge University Press, New York, 1994.

[5] N. S. Bahvalov, *On the convergence of a relaxation method with natural constraints on the elliptic operator*, USSR Comp. Math. and Math. Phys., **6** (1966), 101–135.

[6] R. Bank, B. Welfert, and H. Yserentant, *A class of iterative methods for solving saddle point problems*, Numer. Math., **56** (1990), 645–666.

[7] J. Bear, *Dynamics of fluids in porous media*, Dover Publications, New York, 1988.

[8] P. Binning, *Modelling unsaturated zone flow and contaminant transport in the air and water phases*, PhD thesis, Princeton University, Princeton, NJ, 1994.

[9] P. Björstad and O. Widlund, *Iterative methods for the solution of elliptic problems on regions partitioned into substructures*, SIAM J. Numer. Anal., **23** (1986), 1097–1120.

[10] C. Börgers, *The Neumann–Dirichlet domain decomposition method with inexact solvers on the subdomains*, Numer. Math., **55** (1989), 132–136.

[11] J. Bramble, *Multigrid methods*, Pitman Research Notes in Mathematics Series, vol. 294, Longman Scientific & Technical, London, 1993.

[12] J. Bramble, R. Ewing, J. Pasciak, and A. Schatz, *A preconditioning technique for the efficient solution of problems with local grid refinement*, Comp. Meth. Appl. Mech. Engrg., **67** (1988), 149–159.

[13] J. Bramble, C. Goldstein, and J. Pasciak, *Analysis of V–cycle multigrid algorithms for forms defined by numerical quadrature*, SIAM J. Sci. Comp., **15** (1994), 566–576.

[14] J. Bramble and J. Pasciak, *A preconditioning technique for indefinite systems resulting from mixed approximations of elliptic problems*, Math. Comp., **50** (1988), 1–18.

[15] ——, *Conjugate gradient smoothers for multigrid algorithms*, Dept. of Mathematics, Texas A&M University, 1990. Unpublished notes.

[16] ——, *A domain decompositon technique for Stokes problems*, App. Num. Math., **6** (1990), 251–261.

[17] ——, *Iterative techniques for time dependent Stokes problems*, Tech. Rep. BNL-49970, Brookhaven National Laboratory, Upton, NY, 1994.

[18] J. Bramble, J. Pasciak, and A. Schatz, *The construction of preconditioners for elliptic problems by substructuring, I*, Math. Comp., **47** (1986), 103–134.

[19] ——, *An iterative method for elliptic problems on regions partitioned into substructures*, Math. Comp., **46** (1986), 361–369.

[20] ——, *The construction of preconditioners for elliptic problems by substructuring, II*, Math. Comp., **49** (1987), 1–16.

[21] ——, *The construction of preconditioners for elliptic problems by substructuring, III*, Math. Comp., **51** (1988), 415–430.

[22] ——, *The construction of preconditioners for elliptic problems by substructuring, IV*, Math. Comp., **53** (1989), 1–24.

[23] J. Bramble, J. Pasciak, and A. Vassilev, *Analysis of the inexact Uzawa algorithm for saddle point problems*, SIAM J. Numer. Anal. To appear.

[24] ——, *Non–overlapping domain decomposition algorithms with inexact subdomain solves*, Tech. Rep. ISC-95-08-MATH, Institute for Scientific Computation, Texas A&M University, College Station, TX, 1995.

[25] J. Bramble, J. Pasciak, J. Wang, and J. Xu, *Convergence estimates for product iterative methods with applications to domain decomposition*, Math. Comp., **57** (1991), 1–21.

[26] J. Bramble, J. Pasciak, and J. Xu, *Parallel multilevel preconditioners*, Math. Comp., **55** (1990), 1–22.

[27] ——, *The analysis of multigrid algorithms with non-nested spaces or non-inherited quadratic forms*, Math. Comp., **56** (1991), 1–34.

[28] ——, *A multilevel preconditioner for domain decomposition boundary systems*, Proceedings of the 10'th International Conference on Computational Methods in Applied Sciences and Engineering, Nova Sciences, New York, 1991.

[29] J. Bramble and J. Xu, *Some estimates for weighted L^2 projections*, Math. Comp., **56** (1991), 463–476.

[30] F. Brezzi and M. Fortin, *Mixed and hybrid finite element methods*, Springer-Verlag, New York, 1991.

[31] Z. Cai, J. Mandel, and S. McCormick, *The finite volume element method for diffusion equations on general triangulations*, SIAM J. Numer. Anal., **28** (1991), 392–403.

[32] M. Celia and P. Binning, *Two-phase unsaturated flow: One dimensional simulation and air phase velocities*, Water Resources Research, **28** (1992), 2819–2828.

[33] M. Celia, L. Ferrand, D. Rudolph, P. Binning, and H. Rajaram, *Unsaturated zone hydrology: modeling, monitoring, and remediation.* Lecture Notes, Book 1, Dept. of Civil Engineering, Princeton University, Princeton, NJ, 1993.

[34] G. Chavent and J. Jaffre, *Mathematical models and finite elements for resevoir simulation*, Elsevier Science Publishers B.V., Amsterdam, 1986.

[35] H. Chen, R. Ewing, S. Maliassov, I. Mishev, J. Pasciak, and A. Vassilev, *The TAMU two–phase flow simulator: Programmer's guide*, Tech. Rep. ISC-96-01-MATH, Institute for Scientific Computation, Texas A&M University, College Station, TX, 1996.

[36] H. Cho and P. Jaffe, *The volatilization of organic compounds in unsaturated porous media during infiltration*, J. Contaminant Hydrology, **6** (1990), 387–410.

[37] P. Ciarlet, *The finite element method for elliptic problems*, North-Holland, New York, 1978.

[38] L. Cowsar, J. Mandel, and M. Wheeler, *Balancing domain decomposition for mixed finite elements*, Math. Comp., **64** (1995), 989–1015.

[39] C. Dawson, Q. Du, and T. Dupont, *A finite difference domain decomposition algorithm for numerical solution of the heat equation*, Math. Comp., **57** (1991), 63–71.

[40] M. Dryja, *A capacitance matrix method for the Dirichlet problem on a plygonal region*, Numer. Math., **39** (1982), 51–64.

[41] ——, *A method of domain decomposition for three-dimensional finite element elliptic problems*, First International Symposium on Domain Decomposition Methods for Partial Differential Equations (R. Glowinski, G. Golub, G. Meurant, and J. Périaux, eds.), SIAM, Philadelphia, PA, 1988, pp. 43–61.

[42] M. Dryja, B. Smith, and O. Widlund, *Schwarz analysis of iterative substructuring algorithms for elliptic problems in three dimensions*, SIAM J. Numer. Anal., **31** (1994), 1662–1694.

[43] M. Dryja and O. Widlund, *Additive Schwarz methods for elliptic finite element problems in three dimensions*, Tech. Rep. 570, Courant Institute of Mathematical Sciences, New York, NY, 1991.

[44] M. Dryja and O. Widlund, *Domain decomposition algorithms with small overlap*, SIAM J. Sci. Comp., **15** (1994), 604–620.

[45] H. Elman, *Multigrid and Krylov subspace methods for the discrete Stokes equations*, Tech. Rep. CS-TR-3302, Dept. of Computer Science, University of Maryland, College Park, MD, 1994.

[46] H. Elman and G. Golub, *Inexact and preconditioned Uzawa algorithms for saddle point problems*, SIAM J. Numer. Anal., **6** (1994), 1645–1661.

[47] M. Espedal and R. Ewing, *Characteristic Petrov–Galerkin saturation methods for two-phase immiscible flow*, Comp. Meth. Appl. Mech. Engrg., **64** (1987), 113–135.

[48] R. Ewing, B. Boyett, D. Babu, and R. Heinemann, *Efficient use of locally refined grids for multiphase reservoir simulation*, Proceedings of the Tenth SPE Symposium on Reservoir Simulation, SPE, Houston, TX, 1989, pp. 55–70.

[49] R. Ewing, R. Lazarov, P. Lu, and P. Vassilevski, *Preconditioning indefinite systems arising from mixed finite element discretization of second-order elliptic problems*, Preconditioned Conjugate Gradient Methods (O. Axelsson and L. Kolotilina, eds.), Lecture Notes in Mathematics, vol. 1457, Springer-Verlag, Berlin, 1990, pp. 280–343.

[50] R. Ewing, R. Lazarov, J. Pasciak, and P. Vassilevski, *Domain decomposition type iterative techniques for parabolic problems on locally refined grids*, SIAM J. Numer. Anal., **30** (1993), 1537–1557.

[51] R. Ewing, R. Lazarov, and A. Vassilev, *Adaptive techniques for time–dependent problems*, Comp. Meth. Appl. Mech. Engrg., **101** (1992), 113–126.

[52] ——, *Finite difference scheme for parabolic problems on composite grids with refinement in time and space*, SIAM J. Numer. Anal., **31** (1994), 1605–1622.

[53] R. Ewing, R. Lazarov, and P. Vassilevski, *Finite difference schemes on grids with local refinement in time and space for parabolic problems. I: Derivation, stability, and error analysis*, Computing, **45** (1990), 193–215.

[54] ——, *Local refinement techniques for elliptic problems on cell-centered grids, II*, Num. Lin. Alg. Appl., **1** (1994), 337–368.

[55] R. Ewing and M. Wheeler, *Computational aspects of mixed finite element methods*, Numerical Methods for Scientific Computing (R. Steelman, ed.), North-Holland, Amsterdam, 1983, pp. 163–172.

[56] M. Fortin and R. Glowinski, *Augmented lagrangian methods: Applications to the numerical solution of boundary–value problems*, North–Holland, New York, 1983.

[57] R. Glowinski, *Numerical methods for nonlinear variational problems*, Springer–Verlag, New York, 1984.

[58] R. Glowinski and M. Wheeler, *Domain decomposition and mixed methods for elliptic problems*, First International Symposium on Domain Decomposition Methods for Partial Differential Equations (R. Glowinski, G. Golub, G. Meurant, and J. Periaux, eds.), SIAM, Philadelphia, PA, 1988, pp. 144–172.

[59] R. Gonzalez and M. Wheeler, *Domain decomposition for elliptic partial differential equations with neumann boundary conditions*, Parallel Comput., **5** (1987), 257–263.

[60] P. Grisvard, *Elliptic problems in nonsmooth domains*, Pitman, Boston, 1985.

[61] M. Gurtin, *An introduction to continuum mechanics*, Academic Press, New York, 1981.

[62] G. Haase, U. Langer, and A. Meyer, *The approximate Dirichlet domain decomposition method. Part I: An algebraic approach*, Computing, **47** (1991), 137–151.

[63] W. Hackbusch, *On the regularity of difference schemes*, Arkiv för Matematik, **19** (1981), 71–95.

[64] B. Heinrich, *Finite difference methods on irregular networks*, Akademie–Verlag, Berlin, 1987.

[65] M. Hestenes and E. Stiefel, *Methods of conjugate gradients for solving linear systems*, J. Res. Nat. Bur. Stand., **49** (1952), 409–436.

[66] D. Hittel, *Fundamentals of soil physics*, Academic Press, New York, 1980.

[67] J. Jaffre, *Flux calculations at the interface between two rock types for two-phase flow in porous media*, Tech. Rep. 2075, Institut National de Recherche en Informatique et en Automatique, Rennes Cedex, France, 1994.

[68] C. Johnson and J. Pitkäranta, *Analysis of some mixed finite element methods related to reduced integration*, Math. Comp., **38** (1982), 375–400.

[69] R. Le Veque, *Numerical methods for conservation laws*, Birkhäuster Verlag, Berlin, 1990.

[70] J. Lions and J. Peetre, *Sur une classe d'espaces d'interpolation*, Institut des Hautes Etudes Scientifique, Publ. Math., **19** (1964), 5–68.

[71] R. Marr, J. Pasciak, and R. Peierls, *IPX – Preemptive remote procedure execution for concurrent applications*, Dept. of Applied Sciences, Brookhaven National Laboratory, Upton, NY, 1994.

[72] S. McCormick, *The fast adaptive composite grid (FAC) methods: Theory for the variational case*, Computing Suppl., **5** (1984), 115–121.

[73] S. McCormick, *Multigrid methods for variational problems: General theory for the V-cycle*, SIAM J. Numer. Anal., **22** (1985), 634–643.

[74] S. McCormick and J. Thomas, *The fast adaptive composite grid (FAC) method for elliptic equations*, Math. Comp., **46** (1986), 439–456.

[75] N. Meyers and J. Serrin, $W = H$, Proc. Natl. Acad. Sci. USA, **51** (1964), 1055–1056.

[76] Y. Mokin, *A mesh analogue of the embedding theorem for type W classes*, USSR Comp. Math. and Math. Phys., **11** (1971), 1–16.

[77] J. Nedelec, *Elements finis mixtes incompressibles pour l'equation de Stokes dans* R^3., Numer. Math., **39** (1982), 97–112.

[78] S. Nepomnyaschikh, *Application of domain decomposition to elliptic problems with discontinuous coefficients*, Fourth International Symposium on Domain Decomposition Methods for Partial Differential Equations (R. Glowinski, Y. Kuznetzov, G. Meurant, and J. Périaux, eds.), SIAM, Philadelphia, PA, 1991, pp. 242–251.

[79] Partnership in Computational Science Consortium, *User's Guide to GCT: The Groundwater Contaminant Transport Simulator*, Department of Energy, Washington, DC. In preparation.

[80] J. Pasciak and A. Vassilev, *Neumann to Dirichlet maps for mixed discretizations of Lagrange type*, Dept. of Mathematics, Texas A&M University, 1995. Unpublished notes.

[81] W. Patterson, 3rd, *Iterative methods for the solution of a linear operator equation in Hilbert space - a survey*, Lecture Notes in Mathematics, vol. 394, Springer-Verlag, New York, 1974.

[82] D. Peaceman, *Interpretation of well–block pressures in numerical reservoir simulation with nonsquare grid blocks and anisotropic permeability*, SPE J., **6** (1983), 531–543.

[83] W. Queck, *The convergence factor of preconditioned algorithms of the Arrow-Hurwicz type*, SIAM J. Numer. Anal., **26** (1989), 1016–1030.

[84] P. Raviart and J. Thomas, *A mixed finite element method for 2-nd order elliptic problems*, Mathematical Aspects of Finite Element Methods (I. Galligani and E. Magenes, eds.), Lecture Notes in Mathematics, vol. 606, Springer-Verlag, New York, 1977, pp. 292–315.

[85] J. Roberts and J. Thomas, *Mixed and hybrid methods*, Handbook of Numerical Analysis (P. Ciarlet and J. Lions, eds.), vol. 2, Elsevier Science Publishers B.V., North–Holland, 1991, pp. 524–639.

[86] H. Royden, *Real analysis*, Macmillan, New York, 1988.

[87] T. Russell and M. Wheeler, *Finite element and finite difference methods for continuous flows in porous media*, The Mathematics of Reservoir Simulation (R. Ewing, ed.), SIAM, Philadelphia, PA, 1983, pp. 35–106.

[88] T. Rusten and R. Winter, *Substructure preconditioners for elliptic saddle point problems*, Math. Comp, **60** (1993), 23–48.

[89] T. Rusten and R. Winther, *A preconditioned iterative method for saddle point problems*, SIAM J. Matrix Anal. Appl., **13** (1992), 887–904.

[90] A. Samarskii, *An introduction to the theory of difference schemes*, Nauka, Moscow, 1971. In Russian.

[91] H. Schwarz, *Ueber einige abbildungsaufgaben*, J. Reine Angew. Math, **70** (1869), 105–120. Copublished with Ges. Math. Abh., **2** (1890), 65–83.

[92] B. Smith, *Domain decomposition algorithms for the partial differential equations of linear elasticity*, PhD thesis, Courant Institute of Mathematical Sciences, New York, NY, 1990.

[93] N. Strelkov, *Simplicial extensions of mesh functions and their application to the solution of problems of mathematical physics*, USSR Comp. Math. and Math. Phys., **11** (1971), 190–204.

[94] V. Thomée, *Galerkin finite element methods for parabolic problems*, Lecture Notes in Mathematics (A. Dold and B. Eckmann, eds.), vol. 1054, Springer-Verlag, New York, 1984.

[95] J. Touma and M. Vauclin, *Experimental and numerical analysis of two–phase infiltration in a partially saturated soil*, Transport in Porous Media, **1** (1986), 27–55.

[96] M. van Genuchten, *A closed form equation for predicting the hydraulic conductivity in soils*, Soil Sci. Soc. Am. J., **44** (1980), 892–898.

[97] A. Weiser and M. Wheeler, *On convergence of block–centered finite differences for elliptic problems*, SIAM J. Numer. Anal., **25** (1988), 351–375.

[98] J. Xu, *Iterative methods by space decomposition and subspace correction*, SIAM Review, **34** (1992), 581–613.

[99] D. Young, *Iterative methods for solving partial differential equations of elliptic type*, PhD thesis, Harvard University, Cambridge, MA, 1950.

Appendix A

SOCIETY for INDUSTRIAL and APPLIED MATHEMATICS
3600 UNIVERSITY CITY SCIENCE CENTER PHILADELPHIA, PA 19104-2688 (215) 382-9800

siam@siam.org
Fax: (215) 386-7999

January 12, 1996

Apostol Vassilev
Institute for Scientific Computation
Department of Mathematics
Texas A&M University, M.S. 3404
College Station TX 77843-3404

Dear Mr. Vassilev:

You have SIAM's permission to use parts of the articles "Finite Difference Scheme for Parabolic Problems on Composite Grids with Refinement in Time and Space" and "Analysis of the Inexact Uzawa Algorithm for Saddlepoint Problems" in your thesis, provided you obtain permission from the authors and include the footnotes below:

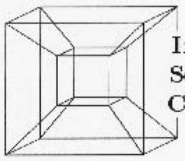
> Portions of "Finite Difference Scheme for Parabolic Problems" reprinted with permission from the SIAM Journal on Numerical Analysis, volume 31, issue 6, pp. 1605-1622. Copyright 1994 by the Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania. All rights reserved.
>
> Portions of "Analysis of the Inexact Uzawa Algorithm for Saddlepoint Problems" reprinted with permission from the SIAM Journal on Numerical Analysis, to appear.

SIAM acknowledges that your thesis will be distributed upon request by University Microfilms, Inc.

Sincerely,

Mary Rose Muccie
Editorial Director
muccie@siam.org

Appendix B



Institute for
Scientific
Computation

Texas A&M University
The Institute for Scientific Computation
633 Blocker Building, Mail Stop 3404
College Station, Texas 77843–3404
U.S.A.

January 25, 1996

Mrs. Linda Versteeg
Rights & Permissions
Publishing Support & Services Department
P.O. Box 521
1000 AM Amsterdam
The Netherlands

RE: Permission to use material from the following paper:
R.E. Ewing, R.D. Lazarov, and A.T. Vassilev, Adaptive techniques for time-dependent problems, *CMAME* **101** (1992), 113–126.

Dear Mrs. Versteeg,

I received your letter of January 17, 1996, granting me permission to reproduce material from the referenced publication. As you advise me therein, I should reapply for permission in case my thesis is publised commercially. University Microfilms Inc. has the right to sell copies of dissertations upon request. Please, consider this as my reapplication for permission under these circumstances.

Should you grant me permission to use a portion of my paper in CMAME, please fax a copy of it to (409) 845–5827 as soon as possible. My defence is scheduled for February 19, 1996 and I would like to have this matter resolved by that time.

Thank you very much for your cooperation.

Sincerely,

Apostol Vassilev

Apostol Vassilev

Phone: (409) 845–7652 e-mail: apostol@isc.tamu.edu Fax: (409) 845–5827

Apostel Vassilev
per fax 409.845.5827

ELSEVIER SCIENCE B.V.

Amsterdam Publishing Division

Sara Burgerhartstraat 25
1055 KV Amsterdam
The Netherlands

P.O. Box 521
1000 AM Amsterdam
The Netherlands

Tel (+31) 20 485 2932
Fax (+31) 20 485 2729

Direct Line: (20) 4852 751
Direct Fax : (20) 4852 722

Amsterdam, january 1996

We hereby grant you permission to reprint to material specified in your letter (see recto) for the purpose you have indicated therein, at no charge provided that:

1. The material to be used has appeared in our publication without credit or acknowledgement to another source.
2. Suitable acknowledgement to the source is given as follows:

   For Books: "Reprinted from (Author/Title), (Copyright Year), (Page Nos), with kind permission from Elsevier Science - NL, Sara Burgerhartstraat 25, 1055 KV Amsterdam, The Netherlands"

   For Journals: "Reprinted from (Journal), (Volume)(Issue)(Author), (Title of Article), (Page Nos), (Copyright Year) with kind permission of Elsevier Science - NL, Sara Burgerhartstraat 25, 1055 KV Amsterdam, The Netherlands".
3. Reproduction of this material is confined to the purpose for which permission is hereby given.
4. This permissions is for non-exclusive English rights only. For other languages, please reapply separately for each one required.
5. Permission EXCLUDES use in an electronic form. Should you have a specific project in mind, please reapply for permission.

Yours sincerely,
ELSEVIER SCIENCE - NL
Publishing Support & Services Department

Mrs. Linda Versteeg
Rights and Permissions

Imprints
Elsevier
Pergamon
North-Holland
Excerpta Medica

Bankers: Hollandsche Bank-Unie
Rotterdam 62.30.60.493
HR Amsterdam 158992

Vita

Apostol Todorov Vassilev was born to the family of Todor and Tanya Vassilev on June 1, 1964. Being highly educated themselves, a civil engineer and a dentist, his parents helped their only son get a solid education and find his way in life. From 1978 to 1982 Apostol attended the Blagoevgrad Mathematics Gymnasium, where he first studied mathematics in a systematic way. He became interested in number theory. In April 1979 Apostol's essay "On the small Fermat theorem" won a special award at the Spring Conference of the Bulgarian Math Society. He graduated with honors in May 1982. Computers had already captured Apostol's interest and he decided to pursue a degree in engineering. He passed the admittance exam of Sofia Tech University with a perfect score. After serving his mandatory duty in the army for two years, Apostol studied electronics and computer science at Sofia Tech and specialized in digital image processing. In March 1985 he met Mariana and they were married in November 1988. Their first child Victoria was born in April 1989. Apostol graduated with a M.S. degree in Electronics in July 1989 and in November 1989 he was invited to make a presentation of his thesis work at the International Student Conference in Košice, Czechoslovakia. The same year Apostol started working at the Bulgarian Academy of Sciences. There he became interested in numerical analysis and decided to pursue a doctoral degree in this field. In April 1990 his second daughter Teodora was born. In August 1991 Apostol received a research assistantship at the University of Wyoming. He studied mathematics there until May 1992 when he transferred to Texas A&M University. He received a Ph.D. in Mathematics in May 1996 and was invited to make a presentation at the Ninth International Conference on Domain Decomposition in June 1996 in Hardanger, Norway. To contact Apostol, please write to: A. Vassilev, St. Lisichkov 7, Blagoevgrad 2700, BULGARIA.