M. J. Holst

The Poisson-Boltzmann Equation

Analysis and Multilevel Numerical Solution

Abstract

We consider the numerical solution of the Poisson-Boltzmann equation (PBE), a three-dimensional second order nonlinear elliptic partial differential equation arising in biophysics. This problem has several interesting features impacting numerical algorithms, including discontinuous coefficients representing material interfaces, rapid nonlinearities, and three spatial dimensions. Similar equations occur in various applications, including nuclear physics, semiconductor physics, population genetics, astrophysics, and combustion. In this work, we study the PBE, discretizations, and develop multilevel-based methods for approximating the solutions of these types of equations.

We first outline the physical model and derive the PBE, which describes the electrostatic potential of a large complex biomolecule lying in a solvent. We next study the theoretical properties of the linearized and nonlinear PBE using standard function space methods; since this equation has not been previously studied theoretically, we provide existence and uniqueness proofs in both the linearized and nonlinear cases. We also analyze box-method discretizations of the PBE, establishing several properties of the discrete equations which are produced. In particular, we show that the discrete nonlinear problem is well-posed.

We study and develop linear multilevel methods for interface problems, based on algebraic enforcement of Galerkin or variational conditions, and on coefficient averaging procedures. Using a stencil calculus, we show that in certain simplified cases the two approaches are equivalent, with different averaging procedures corresponding to different prolongation operators. We extend these methods to the nonlinear case through global inexact-Newton iteration, and derive necessary and sufficient descent conditions for the inexact-Newton direction, resulting in extremely efficient yet robust global methods for nonlinear problems. After reviewing some of the classical and modern multilevel convergence theories, we construct a theory for analyzing general products and sums of operators, based on recent ideas from the finite element multilevel and domain decomposition communities. The theory is then used to develop an algebraic Schwarz framework for analyzing our Galerkin-based multilevel methods.

Numerical results are presented for several test problems, including a nonlinear PBE calculation of the electrostatic potential of Superoxide Dismutase, an enzyme which has recently been linked to Lou Gehrig's disease. We present a collection of performance statistics and benchmarks for the linear and nonlinear methods on a number of sequential and parallel computers, and discuss the software developed in the course of the research.

Preface

In this work we consider nonlinear elliptic equations of the form:

$$-\nabla \cdot (\bar{\mathbf{a}}(\mathbf{x})\nabla u(\mathbf{x})) + b(\mathbf{x}, u(\mathbf{x})) = f(\mathbf{x}) \text{ in } \Omega \subset \mathbb{R}^d,$$

as well as the special case of linear equations. The tensor $\bar{\mathbf{a}}(\mathbf{x})$ as well as the scalar functions $b(\mathbf{x}, u(\mathbf{x}))$ and $f(\mathbf{x})$ may be only piecewise continuous, and in fact may jump by orders of magnitude across internal boundaries or *interfaces* in the domain. Problems of this type, referred to as *interface problems*, occur (on their own, or as the equilibrium form of a governing diffusion equation) in various applications, including flows in porous media, nuclear physics, semiconductor physics, population genetics, astrophysics, combustion, and biophysics, to name only a few.

As research in numerical analysis, our aim is to develop efficient and accurate numerical methods for approximating the solutions of these types of equations, and to understand the various complexity and convergence properties of these methods. As scientists attempt to solve larger and larger problems, it becomes important to develop nearly optimal or optimal complexity algorithms (in this case, algorithms which scale linearly with the number of unknown quantities). Our approach is to employ certain types of *multilevel* or *multigrid* iterative methods, which can be shown (both theoretically and numerically) to be optimal in this sense in many situations.

While the methods we consider are generally applicable to the class of nonlinear interface problems above, we will focus our efforts on some nonlinear elliptic equations which arise in a particular application from biophysics: the Poisson-Boltzmann equation (PBE) in its various forms, including a linearization. This equation, a three-dimensional second order nonlinear elliptic partial differential equation, has several interesting features impacting numerical algorithms, including discontinuous coefficients representing material interfaces, rapid nonlinearities, three spatial dimensions, and infinite domain. In this work, we study the PBE, discretizations, and develop multilevel-based methods for approximating the solutions of these types of equations.

Before we give a preview of the work, let us diverge for a moment to construct an appropriate frame of reference. The field of *numerical analysis*, or *computational mathematics*, is concerned with several of the steps required to solve a scientific problem of current interest using modern computers:

(1) Statement of the physical problem as a governing system of equations
(2) Analysis of the properties of the governing system
(3) Construction and analysis of an approximating system
(4) Development and analysis of numerical methods for the approximating system
(5) Efficient implementation of the numerical methods on sequential or parallel computers
(6) Analysis of the results.

The degree of concern for a particular step depends on the particular area of specialization. Steps 1 and 6 clearly must involve the physical scientist. Steps 2 through 4 are in the realm of applied mathematics and numerical analysis, whereas in this era of complex parallel computers, Step 5 requires the computer scientist. While our emphasis here is Steps 3 through 5, we must also be concerned somewhat with Step 2; this is necessary for the following reasons. For the analysis of multilevel iterative methods, as opposed to other iterative methods which often can be analyzed from purely algebraic properties of the discrete equations, one must have a good understanding of the mathematical properties of the continuous equations involved (i.e., how smooth or *regular* the solutions are), and an understanding of the finite element method for discretization, which itself requires knowledge of properties of the continuous problem.

Overview

We now give a brief overview of the material, which is presented in three parts. *Note: At the beginning of each chapter, we state clearly the purpose of the chapter, and provide a short overview of the material to follow.*

Part I (Chapters 1 and 2) begins in Chapter 1 with the Poisson-Boltzmann equation, which arises in the *Debye-Hückel theory* of macromolecule electrostatics. The unknown function $u(\mathbf{x})$ in the equation represents the electrostatic potential generated by a macromolecule lying in an ionic solvent. Since the fundamental forces in molecular systems are electrostatic in origin, calculation of the potential using Poisson-Boltzmann equation is useful for several applications in biophysics, and in particular the electrostatic forces needed for molecular and Brownian dynamics simulation can be computed from the potential. Some of the properties of the Poisson-Boltzmann equation make it a formidable problem, for both analytical and numerical techniques. To motivate the work, we provide a thorough discussion of the Poisson-Boltzmann equation, including derivation from a few basic assumptions, discussions of special case solutions, as well as common (analytical) approximation techniques.

In Chapter 2, we study the theoretical properties of the linearized and nonlinear PBE using standard function space methods. Solutions to general elliptic equations can be explicitly constructed in only very ideal situations (which is of course the main reason we are interested in numerical methods), and it is therefore important to have some knowledge of the existence and uniqueness theory for the equations involved, even if it is nonconstructive. This is especially true in the nonlinear case, where even small changes in a coefficient function or boundary data can be sufficient to cause bifurcations in the solution of a formerly uniquely solvable problem; it is important to know when the problem is well-posed.[1] Since the Poisson-Boltzmann equation does not appear to have been previously studied in detail theoretically, we provide existence and uniqueness proofs in both the linearized and nonlinear cases. We also analyze box-method discretizations of the PBE, establishing several properties of the discrete equations which are produced. In particular, we show that the discrete nonlinear problem is well-posed.

In Part II (Chapters 3, 4, and 5), we study and develop multilevel methods for linear and nonlinear elliptic problems. In Chapter 3, we provide a detailed overview and analysis of linear multilevel methods and conjugate gradient accelerations. We study special methods for interface problems based on algebraic enforcement of Galerkin or variational conditions, and on coefficient averaging procedures. Using a stencil calculus, we show that in certain simplified cases the two approaches are equivalent, with different averaging procedures corresponding to different prolongation operators. In Chapter 4, we develop methods for nonlinear problems based on a nonlinear multilevel method, and on linear multilevel methods combined with a globally convergent damped-inexact-Newton method. We derive a necessary and sufficient descent condition for the inexact-Newton direction, enabling the development of extremely efficient yet robust damped-inexact-Newton-multilevel methods. In Chapter 5, we outline the fundamental ideas of modern multilevel convergence theory, and we adapt and apply some of the more recent results from the finite element multilevel literature to the Galerkin-based methods. In particular, we develop a fully algebraic theory for bounding the norms and condition numbers of products and sums of operators, based on recent results appearing in the finite element multilevel and domain decomposition literature. This theory is then used to develop an algebraic Schwarz theory framework for analyzing our Galerkin-based multilevel methods.

The motivation for considering multilevel methods for the discretized Poisson-Boltzmann equation is their observed optimal or near optimal behavior for a wide range of elliptic problems. In certain situations which will be explained in more detail later, classical multigrid methods can be shown to possess a *contraction*

[1] The term "well-posedness" as used here refers to three questions: existence of a solution, uniqueness of a solution, and continuous dependence of the solution on the data of the problem.

property of the form:

$$\|u - u^{n+1}\|_A \leq \delta_J \|u - u^n\|_A, \qquad \delta_J = 1 - \frac{1}{C(\alpha, J)} < 1,$$

where J is the number of levels in the multilevel algorithm, and α is a *regularity parameter*, which indicates how "smooth" the solution to the underlying elliptic problem is in a certain mathematical sense. The norm $\| \cdot \|_A$ above denotes the energy norm, and the other quantities appearing above are the true solution to the discrete problem u, and the successive iterates u^n and u^{n+1} produced by the method to approximate u, starting with some initial u^0.

In the context of a problem containing N pieces of data, a solution method for the problem can be considered to be of *optimal order* if the complexity of the method is $O(N)$. If δ_J can be shown to be independent of the number of levels J in a multilevel method, then often each iteration of the method can be constructed to have a complexity of $O(N)$ for N mesh points. In addition, if the contraction number δ_J can be shown to be independent of the mesh size, then the complexity of the resulting algorithm which produces a result with error on the order of truncation error can be shown to be $O(N \ln N)$. When a "nested iteration" technique is employed to provide an improved initial approximation, it can be shown that this cost improves to the optimal $O(N)$. Similar properties can often be shown analytically or numerically for nonlinear multilevel iterative methods. These types of complexity properties become especially important when large three-dimensional problems are considered (extremely large N), and the importance of multilevel methods becomes clear when one notes that no other class of methods has been shown to demonstrate this optimal complexity behavior for such a broad range of problems.

Part III (Chapters 6, 7, and 8) consists of several detailed numerical studies of the multilevel methods developed earlier. In particular, we present experiments with several linear (Chapter 6) and nonlinear (Chapter 7) test problems, provide detailed numerical studies of the complexity and convergence properties of the methods as a function of the problem and method parameters, and apply the most promising of the multilevel methods to the linearized and nonlinear Poisson-Boltzmann equations. We attempt to determine numerically as completely as possible the convergence and complexity properties of these multilevel methods for the Poisson-Boltzmann equation. We demonstrate the utility of our methods and software in Chapter 7 by including a nonlinear PBE calculation of the electrostatic potential of Superoxide Dismutase, an enzyme which has recently been linked to Lou Gehrig's disease. Chapter 8, the final chapter, consists of a collection of performance statistics and benchmarks for the linear and nonlinear solvers on a number of sequential and parallel computers.

In Appendix A we provide the details of computing the Galerkin coarse matrix entries (outlined in Chapter 3) directly and by using MAPLE and MATHEMATICA. In Appendix B we discuss the software developed in the course of the research.

Before we begin, we wish to make a final comment. The material we present is somewhat broad, and as a result the manuscript is long. Although we have tried to eliminate as many errors as possible, there are bound to be many more. We will also have certainly left out many references that should have been included. Making our apologies in advance, we would appreciate hearing from the reader when errors and/or omissions are found.

Acknowledgments

This is a summary of work done over a six year period while a member of the Numerical Computing Group (NCG) in the Department of Computer Science at the University of Illinois, Urbana-Champaign. NCG includes Professors Tom Kerkhoven, Faisal Saied, Paul Saylor, and Bob Skeel. I wish to thank each for being on my dissertation committee, along with Professor Shankar Subramaniam of the Beckman Institute at the University of Illinois, who was the fifth committee member.

I especially wish to express my gratitude to Professor Saied, who supervised my dissertation work. Professor Saied gave me tremendous freedom in pursuing this topic, and was always willing to listen to my ideas. His genuine excitement about and interest in numerical analysis and science in general is quite infectious, and working with him closely over the last three years not only helped to turn my work into something interesting, but also something thoroughly enjoyable to me. I wish to thank him for both his help and friendship during my stay in Urbana.

Professor Kerkhoven, Professor Saylor, and Professor Skeel have each been helpful at various times over the last six years. Professor Saylor has always been supportive and encouraging, and his generosity and kindness helped make life as a graduate student much less difficult. Professor Kerkhoven's expertise in semiconductor modeling and the mathematics of elliptic equations was very helpful, especially in the last year, and I hope to continue to learn from him in the future. Professor Skeel, who funded most of my Ph.D. studies, introduced me to the Poisson-Boltzmann problem, and has been an inspiration to me during my studies in Urbana.

Several scientists from outside NCG contributed to this work. In particular, Professor Subramaniam has been very helpful with his knowledge of biophysics and the Poisson-Boltzmann equation. Dr. Richard Kozack, also at the Beckman Institute, was always prepared to answer my questions about biophysics, whether at the office or while playing pool. Dr. Anthony Nicholls at Columbia University took great pains to help me throughout the project, which included an invitation to visit the lab of Dr. Barry Honig at Columbia for a few days in the Summer of 1992; I wish to thank Anthony for his help and advice during the last two years. Dr. Stefan Vandewalle, during our first few months as postdocs together at Caltech, read the material on algebraic Schwarz theory quite carefully, suggesting simplifications in some of the proofs and also establishing an improved bound for Lemma 5.26.

I was given the opportunity to spend the summers of 1991 and 1992 at Lawrence Livermore National Laboratory, which enabled me to learn from many quality people, including Steve Ashby, Chuck Baldwin, Milo Dorr, Rob Falgout, Anthony Skjellum, and Bert Still. Steve took a strong interest in me throughout my graduate studies, and was kind enough to include me in many of his projects. Milo, Tony, Rob, Bert, and Chuck provided a stimulating environment in which I was always challenged. Rob was kind enough to teach me the stencil calculus he developed in his thesis, which was very helpful to me this last year. Interactions (at LLNL and elsewhere) with Tom Manteuffel, Beth Ong, Randy Bank, Jan Mandel, Joel Dendy, and Paul Newton at various times were also extremely helpful.

Franklin Bodine of the Beckman Institute at the University of Illinois helped produce the 3D visualizations of SOD appearing here. The Chemistry Department at the University of Houston provided access to the University of Houston Brownian Dynamics (UHBD) Program, developed in the laboratory of Dr.

J. Andrew McCammon. The vectorizable plane-diagonal-wise ordered incomplete Cholesky factorization preconditioners, used for comparisons with some of the linear multilevel solvers, were obtained from the NETLIB numerical software library with the help of Dr. Henk van der Vorst.

The National Center for Supercomputer Applications, located at the University of Illinois at Urbana-Champaign, provided access to the Cray Y-MP through support by the National Science Foundation. Access to the Convex C240 was provided by the Computing Services Organization, also at the University of Illinois at Urbana-Champaign. CRL/DCS provided support for the SUN workstations on which this work was written, using LaTeXalong with Xdvi and GhostView postscript previewers. All plots appearing here were generated by MATLAB 4.0 and XFIG. This work was supported in part by Department of Energy Grant No. DOE DE-FG02-91ER25099. The Center Library, with its large number of holdings, was indispensable to the work presented here, and The Center Espresso Bar and Sound System provided espresso and stimulating Stan Getz music at all hours.

Steve Lee and Dan Okunbor shared the ups and downs of graduate student life with me over the last five years; I wish to thank them for many insightful discussions about numerical analysis, coffee, and Nigeria. I am leaving three "families" at the same time as I depart from Urbana: my French friends Vincent, Vero, and their daughter Zoé; Faisal, his wife Anne-Marie, and daughters Laila and Ayesha; and Paul, his wife Cindy, and sons Jack and Gerard. Each of these three families took me in and looked after me over the last few years, and I will always remember their kindness. In addition, my "DCL" family, consisting of Zigrida Arbatsky, Ruth Aydt, Erna Amerman, James Bordner, Becky Green, Michael Ham, Marsha Penrose, Kathy Johnson, Bob Warren, and Michelle Wilson, kept me sane while I was working on this material, providing both help and friendship.

I would like to acknowledge the continuous support and encouragement of my family and Mai Nguyen. I can never repay the kindness and support they have given me during my life, and especially over the last six years. My parents Dale and Shirley, my brothers Greg and Brian, and especially my brother Jon and Mai, were without exception always willing to talk to me on the phone at any time day or night if I had a problem. They encouraged me to pursue my interests, and were always there to offer help or advice.

Finally, to Mai, my friend, colleague, and loved one, I thank you for supporting me during my graduate studies, and for putting up with all of the plane trips, the phone calls, and the seemingly endless separations. We can now look forward to a wonderful life together.

For my family: Dale, Shirley, Greg, Brian, Jon, and Mai.

You must concentrate upon and consecrate yourself wholly to each day,
as though a fire were raging in your hair. –*zen saying*

Contents

Notation

The notation we employ is fairly standard. Generally, points and functions are written in lower case, e.g. u, with boldface $\mathbf{u}$ and overbars $\bar{\mathbf{u}}$ representing vector and matrix functions, respectively. Sometimes it will be more standard to represent vectors and operators in finite-dimensional spaces by subscripts such as u_k and A_k, respectively. Operators are written in uppercase, with caligraphic type for the special case of differential operators. Spaces are represented with the standard symbols in uppercase, or in caligraphic type for arbitrary Hilbert spaces.

Sets, points, vectors, and tensors

$\mathbb{R}^d$	Euclidean d-space.
$\mathbf{x}$	A point $\mathbf{x} \in \mathbb{R}^d, \mathbf{x} = (x_1, \ldots, x_d)$, where $x_i \in \mathbb{R}$.
$\lvert\mathbf{x}\rvert$	The norm in $\mathbb{R}^d$, $(\sum_{i=1}^d x_i^2)^{1/2}$.
$\Omega \subset \mathbb{R}^d$	Ω is a bounded subset of $\mathbb{R}^d$.
$\Omega_1 \subset\subset \Omega$	The closure of Ω_1 is contained in Ω, or $\bar{\Omega}_1 \subset \Omega$.
$\partial\Omega$ or Γ	The *boundary* of $\Omega \subset \mathbb{R}^d$.
$u(\mathbf{x})$	Scalar functions (zero order tensors), mapping $\Omega \mapsto \mathbb{R}$.
$\mathbf{u}(\mathbf{x})$	Vector functions (first order tensors), mapping $\Omega \mapsto \mathbb{R}^d$.
$\bar{\mathbf{u}}(\mathbf{x})$	Matrix functions (second order tensors), mapping $\Omega \mapsto \mathbf{L}(\mathbb{R}^d, \mathbb{R}^d)$.
$\mathbf{u} \cdot \mathbf{v}$	The dot (scalar) product of two vectors, $\sum_{i=1}^d u_i v_i$.
$\mathbf{uv}$	The dyadic (tensor) product of two vectors, $(\mathbf{uv})_{ij} = u_i v_j$.
$\bar{\mathbf{u}} \cdot \mathbf{v}$	The tensor-vector product, $(\bar{\mathbf{u}} \cdot \mathbf{v})_i = \sum_{j=1}^d u_{ij} v_j$.
$\mathbf{v} \cdot \bar{\mathbf{u}}$	The vector-tensor product, $(\mathbf{v} \cdot \bar{\mathbf{u}})_i = \sum_{j=1}^d v_j u_{ji}$.
$u_{ij} v_j$	Einstein summation convention for repeated indices, $\sum_{j=1}^d u_{ij} v_j$.
supp u	The *support* of u on Ω; the closure $\bar{\Omega}_1$ of the set $\Omega_1 \subset \Omega$ on which $u \neq 0$.
supp $u \subset \Omega$	u has *compact support in* Ω.
$\mathrm{dist}(\mathbf{x}, \Omega)$	The *distance* between a point $\mathbf{x}$ and a set Ω, $\inf_{\mathbf{y}\in\Omega} \lvert\mathbf{x} - \mathbf{y}\rvert$.
$\mathrm{int}(\Omega)$	The set of *interior* points of Ω.
$\mathrm{meas}(\Omega)$	Lebesgue measure or volume of the set Ω.

Differentiation and integration

α	A multi-index; the d-tuple $\alpha = (\alpha_1, \ldots, \alpha_d)$, α_i a nonnegative integer.
$\lvert\alpha\rvert$	$\sum_{i=1}^{d} \alpha_i$.
$D^\alpha u$	$\partial^{\lvert\alpha\rvert} u / (\partial x_1^{\alpha_1} \cdots \partial x_d^{\alpha_d})$.
$D_i u$	$\partial u / \partial x_i$.
∇	$(D_1, \ldots, D_d)$.
grad u	The gradient of a scalar function, $(\text{grad } u)_i = (\nabla u)_i = D_i u$.
grad $\mathbf{u}$	The gradient of a vector function, $(\text{grad } \mathbf{u})_{ij} = (\nabla \mathbf{u})_{ij} = D_i u_j$.
div $\mathbf{u}$	The divergence of a vector function, div $\mathbf{u} = \nabla \cdot \mathbf{u} = \sum_{i=1}^{d} D_i u_i$.
u'	The gradient vector of a scalar function, $u' = (\nabla u)^T$.
$\mathbf{u}'$	The Jacobian matrix of a vector function, $\mathbf{u}' = (\nabla \mathbf{u})^T$.
$\int_\Omega u \; d\mathbf{x}$	Volume integration of a function $u(\mathbf{x})$ over a set Ω.
$\oint_{\partial\Omega} u \; ds$	Surface integration of a function $u(\mathbf{x})$ over the boundary $\partial\Omega$ of Ω.

Spaces

$\mathcal{H}$, $\mathcal{H}^*$	Hilbert space and its associated dual space.
$\mathbf{L}(\mathcal{H}_1, \mathcal{H}_2)$	Space of linear operators mapping $\mathcal{H}_1$ into $\mathcal{H}_2$.
$\mathbf{L}(\mathcal{H}_1 \times \mathcal{H}_2, \mathbb{R})$	Space of bilinear forms mapping $\mathcal{H}_1 \times \mathcal{H}_2$ into $\mathbb{R}$.
$\mathcal{H}_k$, $\mathcal{M}_k$, $\mathcal{U}_k$	Subspaces of a Hilbert space.
$C^k(\Omega)$, $C_0^k(\Omega)$	Spaces of k-times continuously differentiable functions.
$L^p(\Omega)$	Lebesgue space of p-th power integrable functions.
$W^k(\Omega)$	Space of k-times weakly differentiable functions.
$W^{k,p}(\Omega)$, $W_0^{k,p}(\Omega)$	Sobolev spaces associated with $W^k(\Omega)$ and $L^p(\Omega)$.
$H^k(\Omega)$, $H_0^k(\Omega)$	Sobolev spaces $W^{k,2}(\Omega)$ and $W_0^{k,2}(\Omega)$.
$H^{-k}(\Omega)$	Dual space of $H^k(\Omega)$.

Norms

$(\cdot,\cdot)_{\mathcal{H}}$, $\lVert\cdot\rVert_{\mathcal{H}}$, $\lVert\cdot\rVert_{\mathcal{H}^*}$	Inner-product and norm in $\mathcal{H}$, and the norm in $\mathcal{H}^*$.
$(\cdot,\cdot)_k$, $\lVert\cdot\rVert_k$	Inner-product and norm in $\mathcal{H}_k$, $\mathcal{M}_k$, or $\mathcal{U}_k$.
$(\cdot,\cdot)_A$, $\lVert\cdot\rVert_A$	A-inner-product and A-norm defined by $A(\cdot,\cdot) = (A\cdot,\cdot)$.
$\lVert\cdot\rVert_{L^p(\Omega)}$	Norm in $L^p(\Omega)$.
$(\cdot,\cdot)_{L^2(\Omega)}$, $\lVert\cdot\rVert_{L^2(\Omega)}$	Inner-product and norm in $L^2(\Omega)$.
$\lVert\cdot\rVert_{W^{k,p}(\Omega)}$, $\lvert\cdot\rvert_{W^{k,p}(\Omega)}$	Norm and semi-norm in $W^{k,p}(\Omega)$ and $W_0^{k,p}(\Omega)$.
$(\cdot,\cdot)_{H^k(\Omega)}$, $\lVert\cdot\rVert_{H^k(\Omega)}$, $\lvert\cdot\rvert_{H^k(\Omega)}$	Inner-product, norm, semi-norm in $H^k(\Omega)$ and $H_0^k(\Omega)$.
$(\cdot,\cdot)$, $\lVert\cdot\rVert = (\cdot,\cdot)^{1/2}$	Generic inner-product and its induced norm.

Functions and operators

u, u_k	Elements of a Hilbert space $\mathcal{H}$.
B, B_k	Linear operators mapping $\mathcal{H}_1$ into $\mathcal{H}_2$.
B^T, B^*	(Hilbert) adjoint and (Hilbert) A-adjoint of a linear operator B.
$F(\cdot)$, $F_k(\cdot)$	Linear or nonlinear functions or functionals on $\mathcal{H}$.
$F'(\cdot)$, $F_k'(\cdot)$	G(Gauteux)- or F(Frechet)-derivative of $F(\cdot)$ and $F_k(\cdot)$.
$A(\cdot,\cdot)$, $A_k(\cdot,\cdot)$	Bilinear forms mapping $\mathcal{H}_1 \times \mathcal{H}_2$ into $\mathbb{R}$.
$\mathcal{L}$, $\mathcal{N}(\cdot)$	Linear and nonlinear differential operators on $\mathcal{H}$.
$\lambda_i(B)$, $\sigma(B)$, $\rho(B)$	Eigenvalue, point spectrum, and spectral radius of operator B.

Multilevel methods

$\mathcal{H}_1 \subset \mathcal{H}_2 \subset \cdots \subset \mathcal{H}_J \equiv \mathcal{H}$	Nested sequence of Hilbert spaces.
$(\cdot,\cdot)_k$, $\|\cdot\|_k$	Inner-product and induced norm in $\mathcal{H}_k$.
$A_k u_k = f_k$	Discrete linear equations in $\mathcal{H}_k$.
C_k	Two-level operator in $\mathcal{H}_k$.
B_k	Multilevel operator in $\mathcal{H}_k$.
R_k	Smoothing operator in $\mathcal{H}_k$.
$Q_{k;k-1}$	Orthogonal projector from $\mathcal{H}_k$ onto $I_{k-1}^k \mathcal{H}_{k-1}$.
$P_{k;k-1}$	A-orthogonal projector from $\mathcal{H}_k$ onto $I_{k-1}^k \mathcal{H}_{k-1}$.
$E_k = I - B_k A_k$	Multilevel error propagator at level k.
I_{k-1}^k	Prolongation operator from $\mathcal{H}_{k-1}$ to $\mathcal{H}_k$.
I_k^{k-1}	Restriction operator from $\mathcal{H}_k$ to $\mathcal{H}_{k-1}$.
$I_k = I_{J-1}^J I_{J-2}^{J-1} \cdots I_{k+1}^{k+2} I_k^{k+1}$	Composite prolongation operator from $\mathcal{H}_k$ to $\mathcal{H}$.
I_k^T	(Variational) composite restriction operator from $\mathcal{H}$ to $\mathcal{H}_k$.

Schwarz methods

$\mathcal{H} = \sum_{k=0}^J I_k \mathcal{H}_k, \quad I_k \mathcal{H}_k \subseteq \mathcal{H}$	A Hilbert space $\mathcal{H}$ and subspaces.
$\mathcal{H}_0, \mathcal{H}_1, \cdots, \mathcal{H}_J, \quad \dim(\mathcal{H}_k) \le \dim(\mathcal{H})$	Subspaces of the Hilbert space $\mathcal{H}$.
$(\cdot,\cdot)$, $\|\cdot\|$	Inner-product and induced norm in $\mathcal{H}$.
$(\cdot,\cdot)_k$, $\|\cdot\|_k$	Inner-product and induced norm in $\mathcal{H}_k$.
I_k	Prolongation operator from $\mathcal{H}_k$ to $\mathcal{H}$.
I_k^T	(Variational) restriction operator from $\mathcal{H}$ to $\mathcal{H}_k$.
$Au = f$	Linear operator equation in $\mathcal{H}$.
$A_k = I_k^T A I_k$	Operator A restricted variationally to $\mathcal{H}_k$.
$R_k \approx A_k^{-1}$	Approximate subspace solver in $\mathcal{H}_k$.
$Q_k = I_k (I_k^T I_k)^{-1} I_k^T$	Orthogonal projector onto $I_k \mathcal{H}_k$.
$P_k = I_k A_k^{-1} I_k^T A = I_k (I_k^T A I_k)^{-1} I_k^T A$	A-orthogonal projector onto $I_k \mathcal{H}_k$.
$T_k = I_k R_k I_k^T A$	Approximate A-orthogonal projector onto $I_k \mathcal{H}_k$.
$E = (I - T_J) \cdots (I - T_1)(I - T_0)$	Product error propagator in $\mathcal{H}$.
$P = T_0 + T_1 + \cdots T_J$	Sum error propagator in $\mathcal{H}$.
C_0, ω	Product and sum operator theory constants.
Θ, Ξ	Interaction matrices.
ω_0, ω_1	Subspace solver spectral bounds.
S_0	Stability constant for subspace splittings.

Abbreviations

PDE	Partial differential equation
PBE	Poisson-Boltzmann equation
MG	Multigrid or multilevel
DD	Domain decomposition
PD	Positive definite.
SPD	Symmetric positive definite with respect to $(\cdot,\cdot)$.
X-SPD	Symmetric positive definite with respect to $(\cdot,\cdot)_X$.

1. The Physical Model

In this chapter, we motivate the material to follow by discussing the Poisson-Boltzmann equation in detail, along with some of its applications. We first review classical and molecular dynamics, Brownian dynamics, the Debye-Hückel Theory, and derive the Poisson-Boltzmann equation from a few basic assumptions. A linearized form of the equation is also obtained, and its validity is discussed. Some special situations are studied, in which analytical solutions can be obtained in the form of Green's functions for the linearized equation over the entire domain, or pieced together from analytical solutions in separate regions. We finish the chapter by specifying completely the two equations which provide the focus for the remainder of the work: the nonlinear and linearized Poisson-Boltzmann equations in bounded domains, where boundary conditions are approximated using some common analytical techniques.

While this chapter consists mainly of background material, our contributions here are as follows.

- It seems difficult to find a full derivation of the Poisson-Boltzmann equation in the literature; therefore, we have provided a detailed derivation.
- The two most useful analytical solutions are presented together, along with detailed explanations of the corresponding models and the complete derivations of the solutions.
- We collect together in one place most of the relevant references and sources for information about the Poisson-Boltzmann equation and its use, including references from the biophysics, physics, chemistry, and biology communities.

1.1 Introduction

Let us begin with a quote from an article appearing recently in *Chemical Review* [43]:

> Electromagnetism is *the* force of chemistry. Combined with the consequences of quantum and statistical mechanics, electromagnetic forces maintain the structure and drive the processes of the chemistry around us and inside us. Because of the long-range nature of Coulombic interactions, electrostatics plays a particularly vital role in intra- and intermolecular interactions of chemistry and biochemistry.

Currently, there is intensive study of the electrostatic properties of biomolecules in both the physics and chemistry communities. Excellent surveys outlining some of these efforts can be found in [32, 171]. Continuum models of molecules in ionic solutions, first proposed in 1923 by Debye and Hückel [45], are increasingly important tools for studying electrostatic interactions, and are now being incorporated into molecular and Brownian dynamics simulators [41, 154, 169]. Since the electrostatic behavior contributes to the structure, binding properties, as well as the kinetics of complex molecules such as proteins, modeling these interactions accurately is an important problem in biophysics.

The fundamental equation arising in the Debye-Hückel theory is a three-dimensional second order nonlinear partial differential equation describing the electrostatic potential $\Phi(\mathbf{r})$ at a field position $\mathbf{r}$. In the special case of a 1:1 electrolyte, this equation can be written for the dimensionless potential $u(\mathbf{r}) = e_c k_B^{-1} T^{-1} \Phi(\mathbf{r})$

as follows:

$$-\nabla \cdot (\epsilon(\mathbf{r})\nabla u(\mathbf{r})) + \bar{\kappa}^2(\mathbf{r}) \sinh(u(\mathbf{r})) = \left(\frac{4\pi e_c^2}{k_B T}\right) \sum_{i=1}^{N_m} z_i \delta(\mathbf{r} - \mathbf{r}_i), \tag{1.1}$$

where the permitivity $\epsilon(\mathbf{r})$ takes the values of the appropriate dielectric constants in the different regions of the model (the value ϵ_m in the molecular region, and a second value ϵ_w in both the solution region and an ion-exclusion layer surrounding the molecule). The *modified* Debye-Hückel parameter $\bar{\kappa}(\mathbf{r})$, which takes the values $\bar{\kappa}(\mathbf{r}) = \sqrt{\epsilon_w}\kappa$ in the solution region and $\bar{\kappa}(\mathbf{r}) = 0$ in the molecule region (where κ is the usual Debye-Hückel parameter), is proportional to the ionic strength of the solution (the modification makes $\bar{\kappa}(\mathbf{r})$ *dielectric independent*). The molecule is represented by N_m point charges $q_i = z_i e_c$ at positions $\mathbf{r}_i$, yielding the delta functions in (1.1), and the constants e_c, k_B, and T represent the charge of an electron, Boltzmann's constant, and the absolute temperature. Equation (1.1) is referred to as the *nonlinear Poisson-Boltzmann equation*, and its solution is usually approximated by solving the *linearized Poisson-Boltzmann equation*:

$$-\nabla \cdot (\epsilon(\mathbf{r})\nabla u(\mathbf{r})) + \bar{\kappa}^2(\mathbf{r}) u(\mathbf{r}) = \left(\frac{4\pi e_c^2}{k_B T}\right) \sum_{i=1}^{N_m} z_i \delta(\mathbf{r} - \mathbf{r}_i). \tag{1.2}$$

Analytical solutions to the linearized and nonlinear Poisson-Boltzmann equations are quite complex, even in the few simple situations for which they exist [176]; see our discussion later in this chapter. A very early analytical approximation approach to the nonlinear Poisson-Boltzmann equation appears in [31], whereas analytical approaches to closely related equations can be found in [69, 177].

Due to advances in computational algorithms and hardware in recent years, several investigations into the efficiency and accuracy of numerical methods for the linearized equation have appeared [42, 72, 116, 152, 154, 171, 186]. We mention in particular the popular computer programs DELPHI and UHBD, which are both based on fast numerical approximation of solutions to the linearized Poisson-Boltzmann equation. The UHBD (University of Houston Brownian Dynamics Simulator) program incorporates solutions of the linearized Poisson-Boltzmann equation into Brownian dynamics simulations [32, 41, 43].

The nonlinear Poisson-Boltzmann equation is only now beginning to find use as a tool for studying electrostatic properties, and various numerical methods are being proposed and investigated [3, 112, 135, 152, 160, 170]. In particular, recent extensions to the program DELPHI, developed at Columbia University, represent the first serious attempt at providing robust solutions to the full nonlinear Poisson-Boltzmann equation [72, 152, 170, 171].

We note that there have been no detailed studies or comparisons of the efficiency and robustness of the many of the proposed numerical methods. The focus of this dissertation is the development and analysis of more efficient and robust numerical methods for both the linear and nonlinear Poisson-Boltzmann equations. In particular, we will present new methods for both the linear and nonlinear equations, based on the multilevel iteration idea, which are substantially more efficient and robust than methods currently used for these equations. Extensive numerical experiments with implementations of these new methods on a number of vector and parallel computers will show that these methods can solve the full nonlinear problem in substantially less time than existing linear methods require for only the linear problem, and that the advantage of the new methods grows with the problem size. In addition, we will present detailed comparisons to many of the existing methods for both the linear and nonlinear equations.

In the remainder of this chapter we will discuss the Poisson-Boltzmann equation in more detail, giving a full derivation of the equation from a few assumptions, and will discuss some applications of the equation.

1.1.1 Classical and continuum mechanics

The fundamental problem of classical mechanics is the *n-Body Problem*:

> *Given n particles of mass m_i acted upon by forces $\mathbf{f}_i$, with initial particle positions $\mathbf{r}_i(0) = (x_{i0}, y_{i0}, z_{i0})$ and velocities $\mathbf{v}_i(0) = \dot{\mathbf{r}}_i(0)$, describe the positions of the particles, $\mathbf{r}_i(t)$, over time.*

For each particle $i = 1, \ldots, n$, the function $\mathbf{r}_i(t) : \mathbb{R} \mapsto \mathbb{R}^3$ represents the motion of the particle over time. The *configuration space* of the system of n particles is the direct product of the n copies of $\mathbb{R}^3$ required to represent the system over time, which is simply the space $\mathbb{R}^N = \mathbb{R}^3 \times \cdots \times \mathbb{R}^3$ n-times, so that $N = 3n$. We define the mapping $\mathbf{r}(t) = (\mathbf{r}_1(t)^T, \ldots, \mathbf{r}_n(t)^T)^T$, where $\mathbf{r} : \mathbb{R} \mapsto \mathbb{R}^N$ maps time into configuration space, and the force mapping $\mathbf{f}(\mathbf{r}, \dot{\mathbf{r}}, t) : \mathbb{R}^N \times \mathbb{R}^N \times \mathbb{R} \mapsto \mathbb{R}^N$. The mapping $\mathbf{r}(t)$ represents the motion of the system of n particles over time.

The solution to the problem above is given by Newton's second law of motion, $\mathbf{f} = M\mathbf{a}$, where M is an $N \times N$ diagonal matrix with the masses m_i repeated along the diagonal, $M = diag(m_1,\ m_1,\ m_1,\ m_2,\ m_2,\ m_2,\ \ldots,\ m_n,\ m_n,\ m_n)$. This yields a system of ordinary differential equations for the system configuration $\mathbf{r}$ at time t:

$$M\ddot{\mathbf{r}} = \mathbf{f}, \qquad \mathbf{r}(0) = \mathbf{r}_0, \qquad \dot{\mathbf{r}}(0) = \mathbf{v}_0.$$

Assuming that the total force on the particles is *conservative*, meaning that is is both *irrotational* ($\nabla \times \mathbf{f} = 0$) and *time-independent*, then the force can be derived from a potential function as $\mathbf{f} = -\nabla\Phi(\mathbf{r})$, where $\Phi : \mathbb{R}^N \mapsto \mathbb{R}$, and where the gradient operator is defined on the product of spaces as $\nabla = (\partial\Phi/\partial\mathbf{r}_1, \ldots, \partial\Phi/\partial\mathbf{r}_n)$. The system can then be written as:

$$M\ddot{\mathbf{r}} = -\nabla\Phi, \qquad \mathbf{r}(0) = \mathbf{r}_0, \qquad \dot{\mathbf{r}}(0) = \mathbf{v}_0.$$

This second order system of ordinary differential equations can be written as a first order *Hamiltonian dynamical system* by defining generalized position and momentum coordinates (see pages 60-65 of [4]), and the solution of this system of equations completely describes the complex behavior of the system of n particles over time. Integrating these equations analytically is in most cases not possible, so numerical methods must be employed; see for example [155] for specially designed numerical integrators for Hamiltonian systems. In any numerical integration procedure, the force function $\mathbf{f}$ must be evaluated, either directly or from the potential function Φ.

1.1.2 The potential function

The potential function is usually the sum of several distinct potential functions:

$$\Phi = \sum_{i=1}^{p} \Phi_i,$$

where for example the *near field* Φ_1 may include the Van der Waals potential of chemical physics, or the Lennard-Jones potential of noble gases, while the *external field* Φ_2 would include, for example, externally applied magnetic fields. The *far field* Φ_3 might include the gravitational potential, or the electrostatic potential of a system of charged particles.

The near field by definition exhibits rapid decay; for example, the Lennard-Jones 6-12 power potential of noble gases:

$$\Phi(\mathbf{r}) = \sum_{i=1}^{N} 4\epsilon \left[\left(\frac{\sigma_i(\mathbf{r})}{|\mathbf{r} - \mathbf{r}_i|} \right)^{12} - \left(\frac{\sigma_i(\mathbf{r})}{|\mathbf{r} - \mathbf{r}_i|} \right)^{6} \right],$$

decays as $|\mathbf{r}|^{-6}$, where $\sigma_i(\mathbf{r})$ is a measure of the equilibrium distance between particle i and field position $\mathbf{r}$, and ϵ is a material permitivity parameter. On the other hand, the far field decays much more slowly. For example, the electrostatic potential of a system of N charged particles is:

$$\Phi(\mathbf{r}) = \sum_{i=1}^{N} \frac{q_i}{\epsilon|\mathbf{r} - \mathbf{r}_i|},$$

which decays only as $|\mathbf{r}|^{-1}$.

To accurately approximate the near field potential Φ_1 for a system of N particles, only local interactions need be considered. In addition, calculating an approximation to the external field potential Φ_2 for each particle is independent of all the other particles. However, the approximation of the far field potential Φ_3 will require the calculation of all pair-wise interactions of the N particles, as the decay is sufficiently slow. Therefore, the complexity of computing each field to high accuracy is clearly:

$$\begin{array}{lll}\Phi_1 & \text{(near field – rapid decay)} & \longrightarrow \quad O(N) \\ \Phi_2 & \text{(external field – independent of } N\text{)} & \longrightarrow \quad O(N) \\ \Phi_3 & \text{(far field – slow decay)} & \longrightarrow \quad O(N^2).\end{array}$$

For this reason, computation of the far field potential presents the most difficulty in practical numerical computations. Recently, a fast direct method known as the *Fast Multipole Method* [75] has been developed, which reduces the complexity of approximating the far field potential. This method is already becoming extremely useful for N-body problems occurring in many application areas.

Continuum methods, in which some portion of the problem is treated as a continuum governed by partial rather than ordinary differential equations, offer an alternative in many cases where only the macro-properties of the system are important. The *Debye-Hückel Theory*, which we discuss more fully below, is a continuum approach for molecular systems consisting of proteins or other complex macromolecules lying in ionic solutions. This continuum approach may be particularly suitable for molecular dynamics or Brownian dynamics simulations, when the electrostatic force is the dominant force in determining the behavior of the system.

1.1.3 Molecular dynamics

The motions of large complex biomolecules (such as a proteins) obey the laws of classical mechanics, and in fact this problem is simply a particular instance of the N-body problem. The dynamics of such a molecule are again described by Newton's second law, where the electrostatic forces, chemical bonds, and other forces are represented by the potential function $\Phi(\mathbf{r})$. Since biomolecules always occur in solvents such as ionic water solutions, the equations of motion must incorporate the electrostatic effect of extremely large numbers of small solvent molecules. The $O(N^2)$ complexity for approximation of the far field, and the large numbers of ions required for accurate approximation, makes this approach infeasible for large biomolecules. Even in the case of a fast method with complexity $O(N)$, the number of ions required may still be too large for practical computations. Continuum representations of the solvent molecules offer a tractable alternative for electrostatic force calculations.

1.1.4 Brownian dynamics

If the electrostatic potential has been computed for a large complex molecule, for example an enzyme or an antibody protein, the binding properties of the enzyme or antibody can be investigated using Brownian dynamics simulations. *Brownian dynamics* refers to the simulation of an interacting system of particles combining the deterministic effects of Newton's second law of motion (dynamics) with stochastic effects (Brownian motion). A system of n interacting particles which also exhibit Brownian effects is described by the *stochastic system of ordinary differential equations*:

$$M\ddot{\mathbf{r}} = -\nabla\Phi + \mathbf{N}(t), \qquad \mathbf{r}(0) = \mathbf{r}_0, \qquad \dot{\mathbf{r}}(0) = \mathbf{v}_0,$$

which represents Newton's second law incorporating a *random* or *white noise* term $\mathbf{N}(t)$. The random terms are usually taken to be *Gaussian*, with each component function $N_i(t)$ independent, so that:

$$E[N_i(t)] = 0,$$

$$E[N_i(t)N_j(t+t_0)] = \delta_{ij}\delta(t_0),$$

where $E[\cdot]$ represents the expectation operation in the usual statistical sense.

The presence of the random term $\mathbf{N}(t)$ implies that the solution $\mathbf{r}(t)$ of the system can only be described statistically, i.e., by a *probability distribution* $P(t, \mathbf{r}, \dot{\mathbf{r}})$. It can be shown (page 254 of [190]) that the solution of a parabolic partial differential equation known as a *Fokker-Planck* equation yields the probability distribution P; to integrate this equation numerically requires evaluation of the potential Φ appearing above.

The distribution P can be used to calculate the probability that two agents which are undergoing both diffusive (Brownian) motion and force interactions will react; see for example [41] for a discussion of practical implementation techniques. Fast numerical approximation of the electrostatic potential field with the linearized Poisson-Boltzmann equation has proven extremely valuable for Brownian dynamics simulations [32, 41, 43].

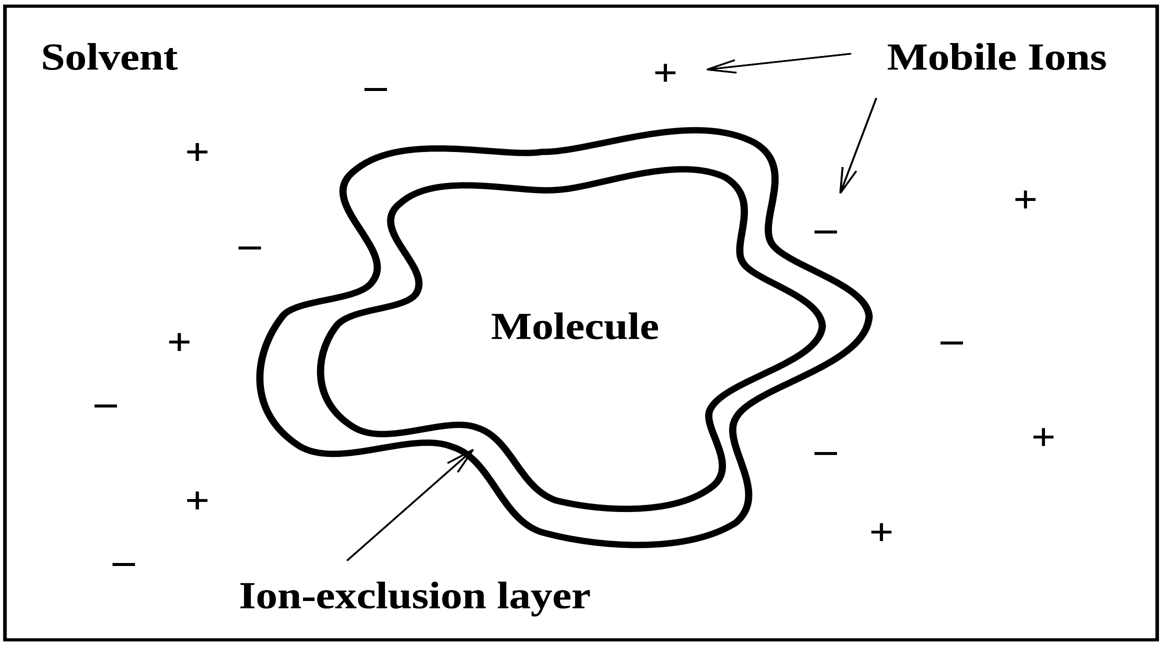


Figure 1.1: Two-dimensional view of the three-dimensional Debye-Hückel model.

1.2 Debye-Hückel theory and the Poisson-Boltzmann equation

In 1923, Debye and Hückel proposed a continuum method for the calculation of electrostatic free energy of small spherical ions in an ionic solution [45]. In their method, the ionic solution is treated as a continuum with a dielectric constant, and a partial differential equation governing the electrostatic potential is developed based on Gauss' law and the Boltzmann distribution law. Below, we describe an extension to the basic Debye-Hückel model, and derive the *Poisson-Boltzmann* equation. See any of [39, 91, 148, 168, 176] for more information.

1.2.1 The Debye-Hückel model

The model motivating the Debye-Hückel theory is given in Figure 1.1. In this model, the molecule for which we wish to determine the electrostatic potential is located in region Ω_1. In the original theory, Ω_1 simply contained a particular ion of the solution; however, the theory is easily extended to more complicated macromolecules such as proteins. We are interested in the more general model and will develop it briefly now.

Region Ω_3 consists of the solvent with dielectric constant ϵ_3, assumed to contain mobile ions. Region Ω_2 is an exclusion layer around the macromolecule in which no mobile charges of the solvent are present, but which has the same dielectric constant $\epsilon_2 = \epsilon_3$ as region Ω_3. If some solvent also penetrates region Ω_1, then this region will have a non-unit dielectric constant ϵ_1. Assuming that all mobile ions are univalent, we can treat them as positive and negative ions with charge $+e_c$ and $-e_c$, where e_c is the charge of an electron. (Higher valence ions can be treated in a fashion similar to the following.)

The electrostatic potential satisfies Gauss' law in each of the three regions. In differential form this yields a separate Poisson equation

$$\nabla^2 \Phi_k(\mathbf{r}) = \frac{-4\pi\rho_k(\mathbf{r})}{\epsilon_k},$$

for each region $\Omega_k, k = 1, 2, 3$. In order to use these equations to determine the potential $\Phi_k(\mathbf{r})$ in each region, the charge density functions $\rho_k(\mathbf{r})$ must be defined.

1.2.2 Gauss' law and the Boltzmann distribution law

Define a coordinate system in three-space, $\mathbf{r} = (x, y, z)$. If the molecule is represented by a series of N_m charges q_i at positions $\mathbf{r}_i$, where $q_i = z_i e_c, z_i \in \mathbb{R}, i = 1, \ldots, N_m$, then the potential in region Ω_1 can be computed directly as

$$\Phi_1(\mathbf{r}) = \sum_{i=1}^{N_m} \frac{q_i}{\epsilon_1 |\mathbf{r} - \mathbf{r}_i|}.$$

Recalling that the free space Green's function for the three-dimensional Laplacian is given by $(-4\pi|\mathbf{r}-\mathbf{r}_0|)^{-1}$, and by applying the Laplacian to both sides of the equation above, we have

$$\nabla^2 \Phi_1(\mathbf{r}) = \sum_{i=1}^{N_m} \frac{-4\pi q_i}{\epsilon_1} \delta(\mathbf{r} - \mathbf{r}_i),$$

where $\delta(\mathbf{r})$ is the *Dirac delta function.*

In Ω_2, the charge density function is given by $\rho_2(\mathbf{r}) = 0$ due to the absence of the mobile ions. Gauss' law for the potential in Ω_2 is then

$$\nabla^2 \Phi_2(\mathbf{r}) = \frac{-4\pi\rho_2(\mathbf{r})}{\epsilon_2} = 0.$$

In Ω_3, assume that the bulk concentration of ions is M per cubic centimeter for each of the two ions present, one of charge $+e_c$, the other of charge $-e_c$. The number M_+ of positive ions and M_- of negative ions per cubic centimeter will differ near the molecule in Ω_1. The fundamental assumption in the Debye-Hückel theory is that the ratio of the concentration of one type of ion near the molecule in Ω_1 to its concentration far from Ω_1 is given by the Boltzmann distribution law

$$e^{-W_i(\mathbf{r})/[k_B T]},$$

where T is the absolute temperature, k_B is Boltzmann's constant, and $W_i(\mathbf{r})$ is the work required to move the ion of type i from $|\mathbf{r}| = \infty$ $(\Phi(\mathbf{r}) = 0)$ to the point $\mathbf{r}$.

Since we have only two types of ions in our model, we have simply

$$W_1(\mathbf{r}) = +e_c \Phi_3(\mathbf{r}), \qquad W_2(\mathbf{r}) = -e_c \Phi_3(\mathbf{r}),$$

for the positive and negative ions, respectively. Therefore, the Boltzmann distribution law applied here gives

$$M_+ = M e^{-e_c \Phi_3(\mathbf{r})/[k_B T]}, \qquad M_- = M e^{+e_c \Phi_3(\mathbf{r})/[k_B T]},$$

where we assume that $M_+ = M_- = M$ far from Ω_1. The charge density at any point in Ω_3 will then be given by:

$$\rho_3(\mathbf{r}) = M_+ e_c - M_- e_c = M e_c e^{-e_c \Phi_3(\mathbf{r})/[k_B T]} - M e_c e^{e_c \Phi_3(\mathbf{r})/[k_B T]} = -2 M e_c \sinh\left(\frac{e_c \Phi_3(\mathbf{r})}{k_B T}\right).$$

Gauss' law for Ω_3 then becomes:

$$\nabla^2 \Phi_3(\mathbf{r}) = \frac{-4\pi\rho_3(\mathbf{r})}{\epsilon_3} = \left(\frac{8\pi M e_c}{\epsilon_3}\right) \sinh\left(\frac{e_c \Phi_3}{k_B T}\right).$$

1.2.3 The Debye-Hückel parameter

The *ionic strength* of the solvent is defined as

$$I_s = \frac{1}{2} \sum_{i=1}^{N_I} c_i z_i^2,$$

where N_I is the number of different types of ions, and c_i is the molar concentration of ion type i with charge $q_i = z_i e_c, z_i \in \mathbb{R}$. In our model, we have $N_I = 2$, $z_1 = z_2 = 1$, and $c_1 = c_2 = 1000M/N_A$, where N_A is Avogadro's number. The ionic strength is then

$$I_s = \frac{1}{2}\sum_{i=1}^{N_I} c_i z_i^2 = \frac{1000M}{N_A}.$$

This yields $M = I_s N_A/1000$, and with this we can rewrite the equation for Ω_3 as

$$\nabla^2\Phi_3(\mathbf{r}) = \left(\frac{8\pi N_A e_c I_s}{1000\epsilon_3}\right)\sinh\left(\frac{e_c\Phi_3(\mathbf{r})}{k_BT}\right).$$

With the *Debye-Hückel parameter* κ defined to be:

$$\kappa = \left(\frac{8\pi N_A e_c^2}{1000\epsilon_3 k_B T}\right)^{1/2} I_s^{1/2},$$

we can write the equation for Ω_3 in final form as

$$\nabla^2\Phi_3(\mathbf{r}) = \kappa^2\left(\frac{k_BT}{e_c}\right)\sinh\left(\frac{e_c\Phi_3(\mathbf{r})}{k_BT}\right).$$

1.2.4 Interface continuity conditions

Physically, we expect the function $\Phi(\mathbf{r})$ to be continuous at the interfaces of the regions, as well as the dielectric times the normal derivative of the function, $\epsilon\nabla\Phi(\mathbf{r})\cdot\mathbf{n}$, where $\mathbf{n}$ is the unit outward normal vector (see for example page 174 in [67] for a discussion of these types of continuity conditions). The discontinuous dielectric interface then implies a discontinuous normal derivative of $\Phi(\mathbf{r})$ at the interfaces. In particular, on $\Gamma_{12} = \Omega_1 \bigcap \Omega_2$, it must be true that:

$$\Phi_1(\mathbf{r}) = \Phi_2(\mathbf{r}), \qquad \epsilon_1\nabla\Phi_1(\mathbf{r})\cdot\mathbf{n} = \epsilon_2\nabla\Phi_2(\mathbf{r})\cdot\mathbf{n},$$

while on $\Gamma_{23} = \Omega_2 \bigcap \Omega_3$, we must have:

$$\Phi_2(\mathbf{r}) = \Phi_3(\mathbf{r}), \qquad \epsilon_2\nabla\Phi_2(\mathbf{r})\cdot\mathbf{n} = \epsilon_3\nabla\Phi_3(\mathbf{r})\cdot\mathbf{n}.$$

The appropriate boundary conditions for the infinite domain $\Omega = \Omega_1 \cup \Omega_2 \cup \Omega_3 = \mathbb{R}^3$ are $\Phi(\infty) = 0$.

1.2.5 The nonlinear and linearized Poisson-Boltzmann equations

We now define the piecewise constant function $\epsilon(\mathbf{r})$ on Ω, and define a *modified Debye-Hückel parameter* $\bar{\kappa}(\mathbf{r})$ (modified to be *dielectric independent*) as:

$$\epsilon(\mathbf{r}) = \left\{\begin{array}{l} \epsilon_1 \text{ if } \mathbf{r} \in \Omega_1, \\ \epsilon_2 (= \epsilon_3) \text{ if } \mathbf{r} \in \Omega_2 \text{ or } \Omega_3, \end{array}\right\}, \qquad \bar{\kappa}(\mathbf{r}) = \left\{\begin{array}{l} 0 \text{ if } \mathbf{r} \in \Omega_1 \text{ or } \Omega_2, \\ \sqrt{\epsilon_3}\kappa \text{ if } \mathbf{r} \in \Omega_3, \end{array}\right\}.$$

This extension of κ is consistent, since the ionic strength of the solvent in regions Ω_1 and Ω_2 is zero. With these definitions, we can write a single field equation, the *nonlinear Poisson-Boltzmann equation*, governing the electrostatic potential $\Phi(\mathbf{r})$ in all three regions:

$$-\nabla\cdot(\epsilon(\mathbf{r})\nabla\Phi(\mathbf{r})) + \bar{\kappa}^2(\mathbf{r})\left(\frac{k_BT}{e_c}\right)\sinh\left(\frac{e_c\Phi(\mathbf{r})}{k_BT}\right) = 4\pi\sum_{i=1}^{N_m} q_i\delta(\mathbf{r}-\mathbf{r}_i) \text{ in } \mathbb{R}^3, \tag{1.3}$$

$$\Phi(\infty) = 0.$$

Using the first term in the series expansion $\sinh x = x + \frac{x^3}{3!} + \frac{x^5}{5!} + \cdots$ as a linear approximation to $\sinh x$, we have the following linearized equation and boundary condition

$$-\nabla\cdot(\epsilon(\mathbf{r})\nabla\Phi(\mathbf{r})) + \bar{\kappa}^2(\mathbf{r})\Phi(\mathbf{r}) = 4\pi\sum_{i=1}^{N_m} q_i\delta(\mathbf{r}-\mathbf{r}_i) \text{ in } \mathbb{R}^3, \tag{1.4}$$

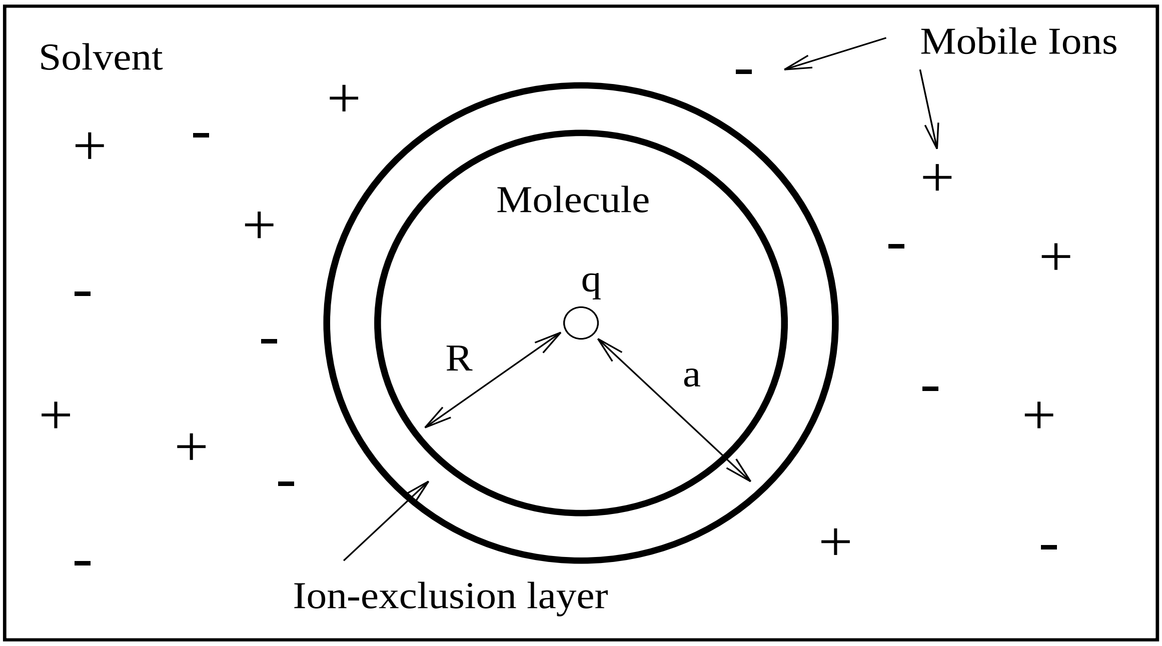


Figure 1.2: A spherical molecule with spherically symmetric charge.

$$\Phi(\infty) = 0,$$

which we refer to as the *linearized Poisson-Boltzmann equation.*

Remark 1.1. Note that the above equations cannot be interpreted "classically" for the following reason. The coefficients $\epsilon(\mathbf{r})$ and $\bar{\kappa}(\mathbf{r})$ are discontinuous at solvent interfaces, which implies that the first derivative of the potential $\Phi(\mathbf{r})$ must also be discontinuous at the interfaces. Therefore, the potential $\Phi(\mathbf{r}) \notin C^2(\mathbb{R}^3)$, and the derivatives in equation (1.3) and equation (1.4) cannot be interpreted in the classical sense. In Chapter 2, we will discuss the correct interpretation of equations (1.3) and (1.4) in terms of *weak solutions*.

1.3 Analytical solutions of the Poisson-Boltzmann equation

In special cases, analytical solutions to the linearized Poisson-Boltzmann equation can be explicitly constructed separately in different model regions and pieced together by enforcing continuity conditions at the region boundaries. In other situations, simplifying assumptions can be made which results in a single analytically solvable equation which governs the entire domain. We now present briefly two solutions as representative of each situation; these solutions appear in somewhat altered forms in [148] and [176]. The second of these solutions will be useful to us at the end of the chapter.

1.3.1 Spherical molecule with uniform charge

Consider a spherical molecule with spherically symmetric total charge q on the molecule surface, immersed in a solvent containing mobile univalent ions, as depicted in Figure 1.2. Defining the coordinate system to be centered at the molecule, the radius of the molecule defining region Ω_1 is denoted by R. The ions may approach only to a distance $a > R$, which defines the ion exclusion layer Ω_2 around the molecule. The ionic solvent then lies in the region Ω_3. The solvent dielectric constant in regions Ω_2 and Ω_3 is denoted ϵ_w, while the molecule dielectric in region Ω_1 is denoted ϵ_m. Since the charge q is assumed to be evenly distributed over molecule surface of area $4\pi R^2$, the molecule has the uniform charge density:

$$\sigma = \frac{q}{4\pi R^2}.$$

In spherical coordinates with spherical symmetry the linearized form of the Poisson-Boltzmann equation is easily seen to reduce to:

$$
\begin{aligned}
\text{region } \Omega_1: &\quad -\frac{1}{r^2}\frac{d}{dr}\left(r^2\frac{d}{dr}\Phi(r)\right) = 0, \qquad r < R,\\
\text{region } \Omega_2: &\quad -\frac{1}{r^2}\frac{d}{dr}\left(r^2\frac{d}{dr}\Phi(r)\right) = 0, \qquad R < r < a,\\
\text{region } \Omega_3: &\quad -\frac{1}{r^2}\frac{d}{dr}\left(r^2\frac{d}{dr}\Phi(r)\right) + \kappa^2\Phi(r) = 0, \qquad r > a,\\
\text{boundary condition}: &\quad \Phi(\infty) = 0,
\end{aligned}
$$

where the zero source functions are a result of placing the charge on the molecule surface. For boundary conditions on $\Gamma_{12} = \Omega_1 \bigcap \Omega_2$ (at $r = R$), we must have:

$$
\Phi_1(r) = \Phi_2(r), \qquad \epsilon_1\left(\frac{d\Phi_1}{dr}\right) - \epsilon_2\left(\frac{d\Phi_2}{dr}\right) = -4\pi\sigma = \frac{-q}{R^2},
$$

whereas on $\Gamma_{23} = \Omega_2 \bigcap \Omega_3$ (at $r = a$), we must have:

$$
\Phi_2(r) = \Phi_3(r), \qquad \epsilon_2\left(\frac{d\Phi_2}{dr}\right) = \epsilon_3\left(\frac{d\Phi_3}{dr}\right).
$$

It is quite easy to verify (by differentiating twice) that the general solution in each region is:

$$
\begin{aligned}
\text{region } \Omega_1: &\quad \Phi_1(r) = c_1 + \frac{c_2}{r}.\\
\text{region } \Omega_2: &\quad \Phi_2(r) = c_3 + \frac{c_4}{r}.\\
\text{region } \Omega_3: &\quad \Phi_3(r) = c_5\frac{e^{-\kappa r}}{r} + c_6\frac{e^{\kappa r}}{r}.
\end{aligned}
$$

The boundary and continuity conditions, and the requirement that $\Phi(r)$ be finite in region Ω_1, give six conditions for the six constants. The resulting expressions for the solution in each region are:

$$
\begin{aligned}
\text{region } \Omega_1: &\quad \Phi_1(r) = \frac{q}{\epsilon_w R}\left(1 - \frac{R\kappa}{1+\kappa a}\right).\\
\text{region } \Omega_2: &\quad \Phi_2(r) = \frac{q}{\epsilon_w r}\left(1 - \frac{r\kappa}{1+\kappa a}\right).\\
\text{region } \Omega_3: &\quad \Phi_3(r) = \frac{q e^{\kappa a}}{\epsilon_w(1+\kappa a)} \cdot \frac{e^{-\kappa r}}{r}.
\end{aligned}
$$

1.3.2 Complete solvent penetration

Consider a long rod-shaped molecule, which the ionic solution is assumed to completely penetrate, as depicted in Figure 1.3. In this case, the (nonlinear or linearized) equation developed earlier for the solvent region Ω_3 now governs the entire domain:

$$
-\nabla^2\Phi(\mathbf{r}) + \kappa^2\Phi(\mathbf{r}) = \left(\frac{4\pi}{\epsilon_w}\right)\sum_{i=1}^{N_m} q_i\delta(\mathbf{r}-\mathbf{r}_i) \text{ in } \mathbb{R}^3,
$$

$$
\Phi(\infty) = 0,
$$

where again the molecule is represented by set of N_m point charges $\{q_i\}$, and the solvent dielectric constant (valid now for all three regions) is denoted ϵ_w.

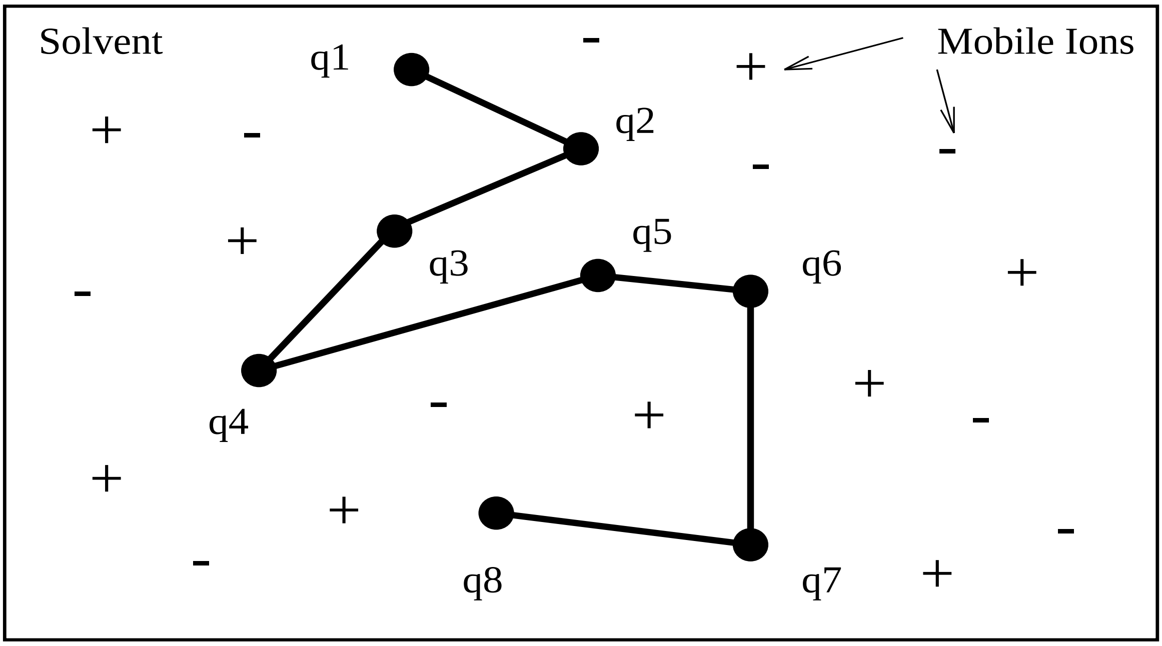


Figure 1.3: A long rod-shaped molecule in an ionic solution.

Recall the free-space Green's functions for the three-dimensional Helmholtz-like equation of the form:

$$-\nabla^2\Phi(\mathbf{r}) + \kappa^2\Phi(\mathbf{r}) = \delta(\mathbf{r} - \mathbf{r}_i),$$

which are given by:

$$G_1(\mathbf{r}, \mathbf{r}_i) \equiv \frac{e^{+\kappa|\mathbf{r}-\mathbf{r}_i|}}{4\pi|\mathbf{r} - \mathbf{r}_i|}, \qquad G_2(\mathbf{r}, \mathbf{r}_i) \equiv \frac{e^{-\kappa|\mathbf{r}-\mathbf{r}_i|}}{4\pi|\mathbf{r} - \mathbf{r}_i|}.$$

The boundary condition $\Phi(\infty) = 0$ clearly eliminates G_1, and we have then for a single charge q_i, a solution of the form:

$$\Phi^{(i)}(\mathbf{r}) = \frac{e^{-\kappa|\mathbf{r}-\mathbf{r}_i|}}{\epsilon_w|\mathbf{r} - \mathbf{r}_i|} q_i.$$

Since the equation is linear with homogeneous boundary conditions, the principle of superposition applies, yielding the full solution for the rod-like molecule in a penetrating solvent:

$$\Phi(\mathbf{r}) = \sum_{i=1}^{N_m} \Phi^{(i)}(\mathbf{r}) = \sum_{i=1}^{N_m} \frac{e^{-\kappa|\mathbf{r}-\mathbf{r}_i|}}{\epsilon_w|\mathbf{r} - \mathbf{r}_i|} q_i. \tag{1.5}$$

1.3.3 Other analytical solutions

Analytical solutions to the linearized form of the Poisson-Boltzmann equation have been given in slightly more complex situations [154, 176], in which a spherical molecular geometry is still required, but the spherical symmetry assumption on the charge distribution is relaxed. In this case, the Green's function which is obtained (see for example [154]) is extremely complex, involving a combination of spherical harmonics, Hankel functions, Neumann functions, and Bessel functions.

In the very special simplified case of an infinite planar molecule, there is an alternative to the nonlinear Poisson-Boltzmann model, referred to as the *Guy-Chapman theory* (cf. Guoy [77] and Chapman [35]). However, in the more general case without symmetries or other assumptions, even knowledge of existence and uniqueness of a solution, especially in the nonlinear case, is a difficult question. Without strong continuity assumptions on the coefficients, the standard potential theory [121] is not useful for proving that a unique solution exists. We consider a more general formulation of the problem in the next chapter, in which more powerful tools are available to prove existence and uniqueness results for a large class of problems. However, even with the knowledge that a unique solution exists, in most cases there remains no closed form expression for such a solution. Therefore, these solutions must be approximated numerically.

1.3.4 Some common analytical approximation techniques

The analytical solution (1.5) is often used in numerical computations to provide boundary conditions for either (1.3) or (1.4) in a bounded domain Ω with boundary Γ. The value of $\mathbf{r} = \mathbf{r}_{\max} \in \Gamma$ is chosen as large as necessary to make the above approximation as accurate as required. Note that the potential $\Phi(\mathbf{r}_{\max}) \rightarrow 0$ exponentially as the position $|\mathbf{r}_{\max}| \rightarrow \infty$. We will assume the validity of the Debye-Hückel model, as well as the validity of using a bounded domain as an approximation. By defining the *dimensionless potential* as

$$u(\mathbf{r}) = \frac{e_c \Phi(\mathbf{r})}{k_B T},$$

we will consider in the remainder of this work the following two equations with associated boundary conditions on bounded domains, which we will continue to denote as the *nonlinear Poisson-Boltzmann equation*

$$-\nabla \cdot (\epsilon(\mathbf{r}) \nabla u(\mathbf{r})) + \bar{\kappa}^2(\mathbf{r}) \sinh(u(\mathbf{r})) = \left(\frac{4\pi e_c^2}{k_B T} \right) \sum_{i=1}^{N_m} z_i \delta(\mathbf{r} - \mathbf{r}_i) \text{ in } \Omega \subset \mathbb{R}^3, \tag{1.6}$$

$$u(\mathbf{r}) = \left(\frac{e_c^2}{k_B T} \right) \sum_{i=1}^{N_m} \frac{e^{-\kappa |\mathbf{r} - \mathbf{r}_i|}}{\epsilon_w |\mathbf{r} - \mathbf{r}_i|} z_i \text{ on } \Gamma,$$

and the *linearized Poisson-Boltzmann equation*

$$-\nabla \cdot (\epsilon(\mathbf{r}) \nabla u(\mathbf{r})) + \bar{\kappa}^2(\mathbf{r}) u(\mathbf{r}) = \left(\frac{4\pi e_c^2}{k_B T} \right) \sum_{i=1}^{N_m} z_i \delta(\mathbf{r} - \mathbf{r}_i) \text{ in } \Omega \subset \mathbb{R}^3, \tag{1.7}$$

$$u(\mathbf{r}) = \left(\frac{e_c^2}{k_B T} \right) \sum_{i=1}^{N_m} \frac{e^{-\kappa |\mathbf{r} - \mathbf{r}_i|}}{\epsilon_w |\mathbf{r} - \mathbf{r}_i|} z_i \text{ on } \Gamma,$$

where in each case Ω is taken to be a bounded domain with a suitably smooth boundary Γ. It will be convenient later to choose Ω to be polygonal.

Finally, motivated both by physical considerations [127] and technical reasons which we will discuss in the next chapter, it is often the case that the delta functions appearing in the right hand sides of the equations are approximated with smooth bounded functions $f_i(\mathbf{r} - \mathbf{r}_i)$, representing a *smearing* of the point charges $z_i \delta(\mathbf{r} - \mathbf{r}_i)$. In some of the following chapters we will use this approximation.

1.4 Units and physical constants

Up to this point, we have ignored as much as possible the units, constants, and other physical considerations in our derivation of the equations. In this section, we compile some of this information, in order to provide a more complete understanding of the equations and the relative magnitudes of the various functions and parameters involved.

The fundamental CGS (centimeter-gram-second) units appearing in electrostatics are:

abbr.	unit	represents
cm	centimeter	distance
esu	electrostatic unit	charge
mol	mole	quantity
K	Kelvin	temperature

Some derived units, including some equivalent SI (international standard) units, are:

abbr.	unit	represents	equivalent expressions
dyn	dyne	force	esu^2/cm^2
erg	erg	energy	$dyn \cdot cm$
$\overset{o}{\mathrm{A}}$	angstrom	distance	$10^{-8}cm$
l	liter	volume	cm^3
molar	moles per liter	concentration	mol/l
m	meter	distance	$10^2 cm$
cal	calorie	energy	$4.184 \times 10^7 erg$
$kcal$	kilo-calorie	energy	$4.184 \times 10^{10} erg$, $10^3 cal$

The collection of required physical constants is as follows:

abbr.	name	value
N_A	Avagadro's number	6.0220450×10^{23}
k_B	Boltzmann's constant	$1.3806620 \times 10^{-16} erg/K$
e_c	Fundamental charge	$4.8032424 \times 10^{-10} esu$

We mention two commonly used energy units:

$$\frac{e_c^2}{\overset{o}{\mathrm{A}}} = \frac{(4.8032424 \times 10^{-10} esu)^2}{10^{-8} cm} \cdot \left(\frac{dyn \cdot cm^2}{esu^2}\right) \cdot \left(\frac{erg}{dyn \cdot cm}\right) \cdot \left(\frac{kcal}{4.184 \times 10^{10} erg}\right) \cdot \left(\frac{N_A}{mol}\right)$$

$$= 332.06364 \frac{kcal}{mol},$$

and, for a representative temperature of $T = 298K$, the "K-T" energy unit:

$$k_B T = (1.3806620 \times 10^{-16} erg/K) \cdot (298K) \cdot \left(\frac{kcal}{4.184 \times 10^{10} erg}\right) \cdot \left(\frac{N_A}{mol}\right) = 0.5921830 \frac{kcal}{mol}.$$

The electrostatic force exerted by charge q_1 at position $\mathbf{r}_1$ on charge q_2 at position $\mathbf{r}_2$ is given by the expression:

$$\mathbf{f} = k \frac{q_1 q_2}{r^2} \mathbf{u},$$

where $k = 1$ due to the choice of CGS units, yielding the force $\mathbf{f}$ in dyn units. The charges q_1 and q_2 are in esu units, r is the distance between the charges in cm, and $\mathbf{u}$ is the unit vector pointing from q_1 to q_2. The units of the electrostatic potential $\Phi(\mathbf{r})$ are $dyn \cdot cm/esu$, which yields energy units of $dyn \cdot cm$ for the potential energy $\int \Phi dq$, field units of dyn/esu for the electrostatic field $\mathbf{e} = -\nabla\Phi$, and force units of dyn for the electrostatic force $\mathbf{f} = q\mathbf{e}$. The dimensionless potential we defined earlier is as follows:

$$u(\mathbf{r}) = \frac{e_c \Phi(\mathbf{r})}{k_B T},$$

which is clearly dimensionless since Boltzmann's constant k_B has units $erg/K = dyn{\cdot}cm/K$, the temperature T has units K, and the charge e_c has units of esu.

Consider now the nonlinear Poisson-Boltzmann equation:

$$-\nabla \cdot (\epsilon(\mathbf{r}) \nabla u(\mathbf{r})) + \bar{\kappa}^2(\mathbf{r}) \sinh(u(\mathbf{r})) = \left(\frac{4\pi e_c^2}{k_B T}\right) \sum_{i=1}^{N_m} z_i \delta(\mathbf{r} - \mathbf{r}_i). \tag{1.8}$$

The dimensionless dielectric function $\epsilon(\mathbf{r})$ in (1.8) has been empirically determined to be:

$$\epsilon(\mathbf{r}) = \left\{ \begin{array}{l} \epsilon_1 \approx 2 \text{ if } \mathbf{r} \in \Omega_1, \\ \epsilon_2 = \epsilon_3 \approx 80 \text{ if } \mathbf{r} \in \Omega_2 \text{ or } \Omega_3. \end{array} \right\}.$$

Molecules of interest such as enzymes and proteins have the scale of ten to over one hundred angstroms, and it is common to choose a truncated domain of at least three times the size of the molecule for accurate approximation of the boundary conditions. Therefore, we will be working on the scale of angstroms, and the first term in (1.8) clearly has units of angstrom^{-2}. The second term $\sinh(u(\mathbf{r}))$ in (1.8) remains dimensionless, therefore the parameter $\bar{\kappa}^2(\mathbf{r})$ must have units of angstrom^{-2}.

We will first determine κ^2 for a representative temperature of $T = 298K$.

$$\kappa^2 = \left(\frac{8\pi N_A e_c^2}{1000\epsilon_3 k_B T}\right) I_s = \left(\frac{8\pi \cdot (6.0220450 \times 10^{23}/mol) \cdot (4.8032424 \times 10^{-10} esu)^2}{1000\epsilon_3 \cdot (1.3806620 \times 10^{-16} esu/K) \cdot (298K)}\right) I_s$$

$$= 8.486902807 \times 10^{16} \cdot \left(\frac{esu^2}{erg \cdot mol}\right) \cdot \frac{I_s}{\epsilon_3}.$$

Performing some unit conversion:

$$8.486902807 \times 10^{16} \cdot \left(\frac{esu^2}{erg \cdot mol}\right) \cdot \left(\frac{dyn \cdot cm^2}{esu^2}\right) \cdot \left(\frac{erg}{dyn \cdot cm}\right) \cdot \left(\frac{mol}{cm^3}\right) \cdot \left(\frac{10^4 cm^2}{10^{20} \overset{o}{\mathrm{A}}{}^2}\right) \cdot \frac{I_s}{\epsilon_3}$$

$$= 8.486902807 \overset{o}{\mathrm{A}}{}^{-2} \frac{I_s}{\epsilon_3},$$

we have the final expression for κ^2, where the now dimensionless number I_s is taken to be the ionic strength measured in moles per liter, referred to as the molar strength of the ionic solution.

Now, $\bar{\kappa}(\mathbf{r})$ is defined as:

$$\bar{\kappa}(\mathbf{r}) = \left\{ \begin{array}{c} 0 \text{ if } \mathbf{r} \in \Omega_1 \text{ or } \Omega_2, \\ \sqrt{\epsilon_3}\kappa \text{ if } \mathbf{r} \in \Omega_3, \end{array} \right\}.$$

Since $\bar{\kappa}^2 = \epsilon_3 \kappa^2$ in Ω_3, we have for $T = 298K$ that:

$$\bar{\kappa}^2(\mathbf{x} \in \Omega_3) = 8.486902807 \overset{o}{\mathrm{A}}{}^{-2} \; I_s.$$

For a typical ionic strength of 0.1 molar, or $I_s = 0.1$, this yields:

$$\bar{\kappa}^2(\mathbf{x} \in \Omega_3) = 0.8486902807 \overset{o}{\mathrm{A}}{}^{-2}.$$

For the source term in (1.8), the coefficients $z_i \in \mathbb{R}$ representing fractions of unit charge are dimensionless, and the delta functions $\delta(\cdot)$ contribute units of angstrom^{-3}. Taking a representative temperature of $T = 298K$, a single term in the sum has the form:

$$\left(\frac{4\pi e_c^2}{k_B T}\right) z_i \delta(\mathbf{r} - \mathbf{r}_i) = \left(\frac{4\pi \cdot (4.8032424 \times 10^{-10} esu)^2}{(1.3806620 \times 10^{-16} esu/K) \cdot (298K)}\right) z_i \delta(\mathbf{r} - \mathbf{r}_i)$$

$$= 7.046528885 \times 10^{-5} \cdot \left(\frac{esu^2}{erg}\right) \cdot z_i \delta(\mathbf{r} - \mathbf{r}_i).$$

Performing some unit conversion:

$$7.046528885 \times 10^{-5} \cdot \left(\frac{esu^2}{erg}\right) \cdot \left(\frac{dyn \cdot cm^2}{esu^2}\right) \cdot \left(\frac{erg}{dyn \cdot cm}\right) \cdot \left(\frac{10^8 \overset{o}{\mathrm{A}}}{cm}\right) \cdot \left(\frac{1}{\overset{o}{\mathrm{A}}{}^3}\right) \cdot z_i \delta(\mathbf{r} - \mathbf{r}_i)$$

$$= 7046.528885 \overset{o}{\mathrm{A}}{}^{-2} \; z_i \delta(\mathbf{r} - \mathbf{r}_i),$$

where $\delta(\cdot)$ is now taken as dimensionless.

We are now in a position to relate the magnitudes of the various problem parameters. Consider a very broad range of temperatures $T \in [200K, 400K]$, a broad range of ionic strengths $I_s \in [0, 10]$, and the following representative polygonal domain:

$$\Omega = [0, 100 \overset{o}{\mathrm{A}}] \times [0, 100 \overset{o}{\mathrm{A}}] \times [0, 100 \overset{o}{\mathrm{A}}].$$

We assume that the set of discrete charges $\{\mathbf{x}_1, \ldots, \mathbf{x}_{N_m}\}$ representing the molecule lie well within the domain, and hence far from the boundary Γ of Ω. The nonlinear Poisson-Boltzmann equation for the dimensionless potential $u(\mathbf{x})$ then has the general form:

$$-\nabla \cdot (\bar{\mathbf{a}}(\mathbf{x})\nabla u(\mathbf{x})) + b(\mathbf{x}, u(\mathbf{x})) = f(\mathbf{x}) \text{ in } \Omega \subset \mathbb{R}^3, \qquad u(\mathbf{x}) = g(\mathbf{x}) \text{ on } \Gamma.$$

From the above discussion, the problem coefficients are of the following forms, and satisfy the following bounds for the given temperature and ionic strength ranges:

(1) $\bar{\mathbf{a}} : \Omega \mapsto \mathbf{L}(\mathbb{R}^3, \mathbb{R}^3),\ a_{ij}(\mathbf{x}) = \delta_{ij}\epsilon(\mathbf{x}),\ 2 \le \epsilon(\mathbf{x}) \le 80,\ \forall \mathbf{x} \in \Omega.$

(2) $b : \Omega \times \mathbb{R} \mapsto \mathbb{R},\ b(\mathbf{x}, u(\mathbf{x})) = \bar{\kappa}^2(\mathbf{x}) \sinh(u(\mathbf{x})),\ 0 \le \bar{\kappa}^2(\mathbf{x}) \le 127.0,\ \forall \mathbf{x} \in \Omega.$

(3) $f : \Omega \mapsto \mathbb{R},\ f(\mathbf{x}) = C \cdot \sum_{i=1}^{N_m} z_i \delta(\mathbf{x} - \mathbf{x}_i),\ 5249.0 \le C \le 10500.0,\ -1 \le z_i \le 1,\ \forall \mathbf{x} \in \Omega.$

(4) $g : \Gamma \mapsto \mathbb{R},\ g(\mathbf{x}) = [C/(4\pi\epsilon_w)] \cdot \sum_{i=1}^{N_m} [z_i e^{-\bar{\kappa}(\mathbf{x})|\mathbf{x}-\mathbf{x}_i|/\sqrt{\epsilon_w}}]/|\mathbf{x} - \mathbf{x}_i|,\ \epsilon_w = 80,\ \forall \mathbf{x} \in \Gamma.$

By assuming that the charges $\mathbf{x}_i$ do not lie near the boundary Γ, which will always be the case for our choice of domain and boundary, we see that the boundary function $g(\mathbf{x})$ is a well-behaved continuously differentiable function of $\mathbf{x}$, $g \in C^1(\Gamma)$. Note that the linearized Poisson-Boltzmann equation is exactly as described above, with the following simple modification: $b(\mathbf{x}, u(\mathbf{x})) = \bar{\kappa}^2(\mathbf{x})u(\mathbf{x})$.

2. Analysis of the Equations

The purpose of this chapter is to present a theoretical study of the nonlinear Poisson-Boltzmann equation using modern tools. *A priori* estimates of solutions are derived in certain cases, and several basic existence and uniqueness theorems are established, for both the linearized and nonlinear Poisson-Boltzmann equations. We also develop the discrete analogues of these theorems for establishing that our discretizations lead to well-posed problems. While the continuous and discrete equations in the linear case are fairly well understood, we also review this material. Since we would like this work to be readable by a general scientific audience, we provide a summary of the relevant background material; however, this adds somewhat to the length of the chapter, since a fairly nontrivial framework must be constructed to analyze general partial differential equations. Since the Poisson-Boltzmann equation does not appear to have been previously studied in detail theoretically, we hope that this chapter will help to provide a better understanding of this problem. This type of analysis may help researchers in biophysics gain a better understanding of the mathematical properties of this equation, and what these properties may imply about the underlying physics of the problem.

As pertains to the Poisson-Boltzmann equation, our contributions here are as follows.

- We apply Lax-Milgram theory and the extension due to Babŭska to the linearized Poisson-Boltzmann equation, showing rigorously the existence of unique weak solutions.
- We establish some *a priori* estimates in the linear case, leading to well-posedness in both the continuous and discrete cases.
- We show existence and uniqueness of solutions to the nonlinear Poisson-Boltzmann equation by adapting a proof of Kerkhoven and Jerome, which uses techniques from convex functional analysis.
- We establish several properties of box-method discretizations of the linearized Poisson-Boltzmann equation.
- We prove that a box-method discretization of the full nonlinear Poisson-Boltzmann equation yields a nonlinear algebraic operator which is a homeomorphism, so that the discrete nonlinear problem is well-posed.

2.1 Review of elliptic theory

The linearized Poisson-Boltzmann equation is a member of a particular class of second order linear elliptic equations. To formulate a typical problem from this class in a general setting, consider a bounded region $\Omega \subset \mathbb{R}^d$ with boundary $\Gamma = \Gamma_D \cup \Gamma_N$, where $\Gamma_D \cap \Gamma_N = \emptyset$. We are concerned with general second order linear elliptic equations, which can be written in the strong, divergence form as:

$$-\nabla \cdot (\bar{\mathbf{a}} \nabla \hat{u}) + b\hat{u} = f \text{ in } \Omega, \tag{2.1}$$

$$\hat{u} = g_D \text{ on } \Gamma_D, \tag{2.2}$$

$$(\bar{\mathbf{a}} \nabla \hat{u}) \cdot \mathbf{n} + c\hat{u} = g_N \text{ on } \Gamma_N, \tag{2.3}$$

with $b(\mathbf{x}) : \Omega \mapsto \mathbb{R}$, $f(\mathbf{x}) : \Omega \mapsto \mathbb{R}$, $g_D(\mathbf{x}) : \Gamma_D \mapsto \mathbb{R}$, $g_N(\mathbf{x}) : \Gamma_N \mapsto \mathbb{R}$, $c(\mathbf{x}) : \Gamma_N \mapsto \mathbb{R}$, $\hat{u}(\mathbf{x}) : \Omega \mapsto \mathbb{R}$, and with the matrix function $\bar{\mathbf{a}}(\mathbf{x}) : \Omega \mapsto \mathbf{L}(\mathbb{R}^d, \mathbb{R}^d)$. The equation and boundary condition are often written as

$\mathcal{L}_\Omega \hat{u} = f_\Omega$ in Ω, $\mathcal{L}_\Gamma \hat{u} = f_\Gamma$ on Γ, or simply as the abstract equation $\mathcal{L}\hat{u} = f$. The equation is *elliptic* if the matrix $\bar{\mathbf{a}}(\mathbf{x}) = [a_{ij}(\mathbf{x})]$ is positive definite for all $\mathbf{x} \in \Omega$, and *strongly elliptic* if there exists $\lambda > 0$ such that $\sum_{ij} a_{ij}\eta_i\eta_j \geq \lambda|\eta|^2,\ \forall \mathbf{x} \in \Omega, \eta \in \mathbb{R}^d$.

The nonlinear Poisson-Boltzmann equation is a member of a class of second order semi-linear elliptic equations. On a bounded region $\Omega \subset \mathbb{R}^d$ with boundary $\Gamma = \Gamma_D \cup \Gamma_N$, where $\Gamma_D \cap \Gamma_N = \emptyset$, a typical problem from this class, written in strong, divergence form, is

$$-\nabla \cdot (\bar{\mathbf{a}}\nabla \hat{u}) + b(\mathbf{x}, \hat{u}) = f \text{ in } \Omega, \tag{2.4}$$

$$\hat{u} = g_D \text{ on } \Gamma_D, \tag{2.5}$$

$$(\bar{\mathbf{a}}\nabla \hat{u}) \cdot \mathbf{n} + c\hat{u} = g_N \text{ on } \Gamma_N, \tag{2.6}$$

where now $b(\mathbf{x}, \hat{u}(\mathbf{x})) : \Omega \times \mathbb{R} \mapsto \mathbb{R}$. Ellipticity is defined as in the linear case. We denote the equation and boundary condition abstractly as $\mathcal{N}(\hat{u}) = f$.

Both the nonlinear and linearized Poisson-Boltzmann equations are referred to as *interface problems*, due to the discontinuities in the coefficients representing material interfaces in the physical problem. These coefficient discontinuities preclude the use of classical potential theory for providing an existence and uniqueness theory for these problems (for example Chapter 4 in [70]). The question of well-posedness is a nontrivial matter, especially in the nonlinear case. The Bratu problem (page 432 in [40]), a nonlinear elliptic equation quite similar to the nonlinear PBE, represents an example for which the slightest variation of a single problem parameter alternatively produces a unique solution, multiple solutions, or no solution at all.

In special cases, solutions to the linearized and nonlinear Poisson-Boltzmann equation can be constructed analytically; more generally, this is not possible, and numerical techniques must be employed to construct an approximate solution. However, in order to justify any attempt to solve the problem numerically, some knowledge of the existence and uniqueness of a solution to the problem, even if this knowledge does not provide an expression for the solution itself, is desirable. The *generalized theory* of partial differential equations considers the existence and uniqueness of *weak* solutions in a Hilbert space, for which more powerful tools are available to provide existence and uniqueness theories for a large class of problems.

We first introduce some standard notation and develop some of the fundamental theorems and ideas used in the analysis of abstractly formulated partial differential equations. This background may be found for example in [44, 115, 161].

2.1.1 Notation and classical function spaces

We denote Euclidean d-space as $\mathbb{R}^d$, a point of which is denoted $\mathbf{x} = (x_1, \ldots, x_d)$, where $x_i \in \mathbb{R}$. The norm in $\mathbb{R}^d$ is defined as $|\mathbf{x}| = (\sum_{i=1}^d x_i^2)^{1/2}$. The set $\Omega \subset \mathbb{R}^d$ denotes a (usually open) bounded subset of $\mathbb{R}^d$, and the *boundary* of such a set is denoted Γ. The notation meas(Ω) is used to denote the (Lebesgue) measure or volume of the set Ω. By $\Omega_1 \subset\subset \Omega$ we mean that the closure of Ω_1 (closed and bounded in $\mathbb{R}^d$, hence compact) is contained in Ω, or $\bar{\Omega}_1 \subset \Omega$. The *support* of a function u defined over a set Ω is the closure $\bar{\Omega}_1$ of the set $\Omega_1 \subset \Omega$ on which $u \neq 0$. If $\bar{\Omega}_1 \subset \Omega$, we say that u has *compact support in* Ω. The *distance* between a point $\mathbf{x}$ and a set Ω is denoted:

$$\text{dist}(\mathbf{x}, \Omega) = \inf_{\mathbf{y} \in \Omega} |\mathbf{x} - \mathbf{y}|.$$

Scalar functions (zero order tensors) are denoted as $u(\mathbf{x}) : \Omega \mapsto \mathbb{R}$, vector functions (first order tensors) as $\mathbf{u}(\mathbf{x}) : \Omega \mapsto \mathbb{R}^d$, and matrix functions (second order tensors) as $\bar{\mathbf{u}}(\mathbf{x}) : \Omega \mapsto \mathbf{L}(\mathbb{R}^d, \mathbb{R}^d)$. Employing the Einstein summation convention, the usual notational conventions for tensor products are followed: the dot (scalar) product of two vectors is denoted $\mathbf{u} \cdot \mathbf{v} = u_i v_i$; the dyadic (tensor) product of two vectors is denoted $(\mathbf{uv})_{ij} = u_i v_j$; and the (vector) products of vectors and second order tensors are denoted $(\bar{\mathbf{u}} \cdot \mathbf{v})_i = u_{ij} v_j$, and $(\mathbf{v} \cdot \bar{\mathbf{u}})_i = v_j u_{ji}$.

By a *multi-index* α we mean the d-tuple $\alpha = (\alpha_1, \ldots, \alpha_d)$, α_i a nonnegative integer, where $|\alpha| = \sum_{i=1}^d \alpha_i$, which is used to denote mixed partial differentiation of order $|\alpha|$:

$$D^\alpha u = \frac{\partial^{|\alpha|} u}{\partial x_1^{\alpha_1} \cdots \partial x_d^{\alpha_d}}.$$

Single partial differentiation is denoted as $D_i u = \partial u/\partial x_i$, and by defining the vector $\nabla = (D_1, \ldots, D_d)$, the *gradient* and *divergence* operations can be written as tensor products: $(\text{grad } u)_i = (\nabla u)_i = D_i u$, $(\text{grad } \mathbf{u})_{ij} = (\nabla \mathbf{u})_{ij} = D_i u_j$, and $\text{div } \mathbf{u} = \nabla \cdot \mathbf{u} = D_i u_i$.

Volume integration of a function $u(\mathbf{x})$ over a set Ω is denoted $\int_\Omega u \, d\mathbf{x}$, whereas surface integration over the boundary Γ is denoted $\oint_\Gamma u \, ds$. Integration means in the *Lebesgue* sense, which is necessary due to the requirement that the function spaces which we discuss below, which employ the integral of a function as the norm, have the *completeness property*; see for example [119] for a thorough discussion of the deficiencies of the Riemann integral for this purpose.

The generalized theories of differential and integral equations are formulated in Banach (complete normed) spaces and Hilbert (complete inner product) spaces referred to as *Sobolev spaces*; the basic notions and properties of real Banach and Hilbert spaces are summarized in [44, 89]. By *real*, we mean that the scalar field associated with the Hilbert (Banach) space $\mathcal{H}$ as a topological vector space is $\mathbb{R}$. We now review a few classical function spaces which are needed in order to understand the Sobolev spaces.

The space of k-times continuously differentiable functions defined in Ω is denoted $C^k(\Omega)$. The subspace of $C^k(\Omega)$ with compact support is denoted $C_0^k(\Omega)$. It can be shown that if $u \in C_0^k(\Omega)$, then $D^\alpha u = 0$ on Γ, $0 \le |\alpha| \le k-1$.

The *Lebesgue space* $L^p(\Omega)$ of measurable functions on Ω represents the space of functions with finite norm defined by

$$\|u\|_{L^p(\Omega)} = \left(\int_\Omega |u|^p \, d\mathbf{x} \right)^{1/p}.$$

The spaces $L^p(\Omega)$, for $1 \le p < \infty$, are Banach spaces (the Riesz-Fischer Theorem, page 232 in [115]), and the special case of $p = 2$ is a Hilbert space when equipped with the inner product and norm:

$$(u, v)_{L^2(\Omega)} = \int_\Omega uv \, d\mathbf{x}, \qquad \|u\|_{L^2(\Omega)} = (u, u)^{1/2}_{L^2(\Omega)}.$$

By the nature of the Lebesgue integral, an element u of $L^p(\Omega)$ represents an equivalence class of functions which are equal almost everywhere (AE) in the sense that $\|u_1 - u_2\|_{L^p(\Omega)} = 0$, for every u_1 and u_2 in the class which u represents. The convention is to represent the class with the smoothest function in the class.

A function u for which there is a constant C such that $|u| \le C$ AE on Ω is called *essentially bounded* on Ω, and the greatest lower bound of all possible constants C is called the *essential supremum* of $|u|$, or $\text{ess sup}_{\mathbf{x} \in \Omega} |u|$. The space $L^\infty(\Omega)$ denotes the space of all essentially bounded functions on Ω. With norm given by $\|u\|_{L^\infty(\Omega)} = \text{ess sup}_{\mathbf{x} \in \Omega} |u|$, it can be shown that $L^\infty(\Omega)$ is a Banach space (page 237 in [115]).

An important relationship in the $L^p(\Omega)$ spaces is *Hölder's inequality*: if $1 < p < \infty$, $u \in L^p(\Omega)$, $u \in L^q(\Omega)$, where p and q are *conjugate exponents* in the sense that $1/p + 1/q = 1$, then $uv \in L^1(\Omega)$, and

$$\|uv\|_{L^1(\Omega)} \le \|u\|_{L^p(\Omega)} \|v\|_{L^q(\Omega)}.$$

It can be shown that Hölder's inequality also holds with $p = 1, q = \infty$, or $p = \infty, q = 1$. The case $p = q = 2$ is referred to as the *Cauchy-Schwarz inequality.* Also we have *Minkowski's inequality*: if $1 \le p < \infty$, and $u, v \in L^p(\Omega)$, then

$$\|u + v\|_{L^p(\Omega)} \le \|u\|_{L^p(\Omega)} + \|v\|_{L^p(\Omega)}.$$

Finally, a function u defined AE on Ω is said to be *locally integrable* on Ω if $u \in L^1(A)$ for each measurable set $A \subset\subset \Omega$. The space of locally integrable functions on Ω is denoted $L^1_{\text{loc}}(\Omega)$.

2.1.2 Some fundamental theorems in Hilbert space

Consider now the real Hilbert space $\mathcal{H}$, equipped with the inner product $(\cdot,\cdot)_\mathcal{H}$ inducing the norm $\|\cdot\|_\mathcal{H} = (\cdot,\cdot)^{1/2}_\mathcal{H}$. The theorems which will be important here involve linear functionals and bilinear forms operating on elements of $\mathcal{H}$.

Recall that a functional $F(u) : \mathcal{H} \mapsto \mathbb{R}$ is *linear* if $F(\alpha u + \beta v) = \alpha F(u) + \beta F(v)$, $\forall u, v \in \mathcal{H}$, $\alpha, \beta \in \mathbb{R}$, and *bounded* if for some $C \in \mathbb{R}$, $|F(u)| \le C\|u\|_\mathcal{H}$ $\forall u \in \mathcal{H}$. Similarly, the form $A(u, v) : \mathcal{H} \times \mathcal{H} \mapsto \mathbb{R}$ is called *bilinear* if for all $u, v, w \in \mathcal{H}$ and all $\alpha, \beta \in \mathbb{R}$ it is true that

$$A(\alpha u + \beta v, w) = \alpha A(u, w) + \beta A(v, w)$$

$$A(u, \alpha v + \beta w) = \alpha A(u, v) + \beta A(u, w).$$

The form $A(\cdot,\cdot)$ is called *bounded* if for some positive $M \in \mathbb{R}$, $|A(u, v)| \leq M\|u\|_{\mathcal{H}}\|v\|_{\mathcal{H}}$, $\forall u, v \in \mathcal{H}$, and $A(\cdot,\cdot)$ is called *coercive* if for some positive $m \in \mathbb{R}$, $A(u, u) \geq m\|u\|^2_{\mathcal{H}}$, $\forall u \in \mathcal{H}$.

If the bilinear form $A(\cdot,\cdot)$ is symmetric, meaning that $A(u, v) = A(v, u)$ $\forall u, v \in \mathcal{H}$, and positive in the sense that $A(u, u) > 0$ $\forall u \in \mathcal{H}$, $u \neq 0$, and if $A(u, u) = 0$ if and only if $u = 0$, then since the form is linear, it follows that $A(\cdot,\cdot)$ defines in inner-product on $\mathcal{H}$ and induces the norm $\|u\|_A = A(u, u)^{1/2}$.

The set of bounded linear functionals $F(u) : \mathcal{H} \mapsto \mathbb{R}$ forms the *dual space* of $\mathcal{H}$, denoted as $\mathcal{H}^*$. The dual space is a Banach space (Theorem 2.10-4 in [129]) when equipped with the norm:

$$\|F\|_{\mathcal{H}^*} = \sup_{\|u\|_{\mathcal{H}} \neq 0} \frac{|F(u)|}{\|u\|_{\mathcal{H}}} = \sup_{\|u\|_{\mathcal{H}}=1} |F(u)|.$$

We now state without proof three basic theorems relating bilinear forms and linear functionals to linear operators on and elements of a Hilbert space; these theorems are important tools for proving existence and uniqueness results for abstract formulations of partial differential equations.

Theorem 2.1 (Bounded Operator Theorem) *Let $A(u, v)$ be a bounded bilinear form on a Hilbert space $\mathcal{H}$. Then there exists a unique bounded linear operator $A : \mathcal{H} \mapsto \mathcal{H}$ such that*

$$A(u, v) = (Au, v) \qquad \forall u, v \in \mathcal{H}.$$

Proof. See for example Theorem 4.3.6 in [44]. □

Theorem 2.2 (Reisz Representation Theorem) *Let $F(u)$ be a bounded linear functional on a Hilbert space $\mathcal{H}$. Then there exists a unique $f \in \mathcal{H}$ such that*

$$F(u) = (u, f) \ \forall u \in \mathcal{H}, \qquad \text{and } \|F\|_* = \|f\|.$$

Proof. See for example Theorem 3.11.1 in [44]. □

Theorem 2.3 (Lax-Milgram Theorem) *Let $\mathcal{H}$ be a real Hilbert space, let the bilinear form $A(u, v)$ be bounded and coercive on $\mathcal{H}\times\mathcal{H}$, and let $F(u)$ be a bounded linear functional on $\mathcal{H}$. Then there exists a unique solution to the problem:*

$$\text{Find } u \in \mathcal{H} \text{ such that } A(u, v) = F(v) \qquad \forall v \in \mathcal{H}.$$

Proof. See for example Theorem 1.1.3 in [36]. □

2.1.3 Weak derivatives and Sobolev spaces

The appropriate Banach and Hilbert spaces in which to search for weak solutions to partial differential equations will be seen to be the Sobolev spaces. There are several equivalent ways to define the integer order Sobolev spaces. One way is to begin by defining the Sobolev norms and construct the Sobolev spaces by the completion of $C^k(\Omega)$ with respect to these norms. While this construction may seem artificial, recall that the real numbers are constructed from the rationals in precisely this same way, so that a Sobolev space is as “real” as $\mathbb{R}$ (the author found this argument in [132] quite persuasive).

A second equivalent approach (the equivalence is the Meyers-Serrin Theorem; see page 45 in [1]) begins by defining the *weak derivative* of a function $u \in L^1_{\text{loc}}(\Omega)$ corresponding to a multi-index α, which is the function $D^\alpha u$ satisfying:

$$\int_\Omega \phi D^\alpha u \ d\mathbf{x} = (-1)^{|\alpha|} \int_\Omega u D^\alpha \phi \ d\mathbf{x}, \qquad \forall \phi \in C_0^{|\alpha|}(\Omega).$$

The space of k-times weakly differentiable functions is denoted $W^k(\Omega)$, and the subspace of $W^k(\Omega)$ whose weak derivatives are Lebesgue p-integrable, $1 \leq p \leq \infty$, is a Banach space (Theorem 3.2 in [1]) called a *Sobolev space*, denoted along with its norm as:

$$W^{k,p}(\Omega) = \{u \in W^k(\Omega) : \ D^\alpha u \in L^p(\Omega), \ 0 \leq |\alpha| \leq k\},$$

$$\|u\|_{W^{k,p}(\Omega)} = (\sum_{0\le|\alpha|\le k} \|D^\alpha u\|^p_{L^p(\Omega)})^{1/p},\ 1 \le p < \infty, \qquad \|u\|_{W^{k,\infty}(\Omega)} = \max_{0\le|\alpha|\le k}\{\text{ess}\ \sup_{x\in\Omega} |D^\alpha u|\},$$

where D^α denotes weak differentiation. The norm in $W^{k,p}(\Omega)$ can be written in terms of the *semi-norm* $|\cdot|_{W^{k,p}(\Omega)}$, in the following way:

$$\|u\|^p_{W^{k,p}(\Omega)} = \sum_{j=0}^{k} |u|^p_{W^{k,p}(\Omega)}, \qquad \text{where } |u|_{W^{k,p}(\Omega)} = \left(\sum_{|\alpha|=k} \|D^\alpha u\|^p_{L^p(\Omega)}\right)^{1/p}.$$

Note that $W^{0,p}(\Omega) = L^p(\Omega)$, and that $|\cdot|_{W^{0,p}(\Omega)} = \|\cdot\|_{W^{0,p}(\Omega)} = \|\cdot\|_{L^p(\Omega)}$, $1 \le p \le \infty$.

For the case $p = 2$ the space is denoted $H^k(\Omega) = W^{k,2}(\Omega)$, and is a Hilbert space when equipped with the inner product and norm

$$(u,v)_{H^k(\Omega)} = \sum_{0\le|\alpha|\le k} (D^\alpha u, D^\alpha v)_{L^2(\Omega)}, \qquad \|u\|_{H^k(\Omega)} = \|u\|_{W^{k,2}(\Omega)} = (u,u)^{1/2}_{H^k(\Omega)}.$$

Again, note that $H^0(\Omega) = L^2(\Omega)$, and also $(\cdot,\cdot)_{H^0(\Omega)} = (\cdot,\cdot)_{L^2(\Omega)}$ and $|\cdot|_{H^0(\Omega)} = \|\cdot\|_{H^0(\Omega)} = \|\cdot\|_{L^2(\Omega)}$. The following subspace is important

$$H^k_0(\Omega) = \{u \in H^k(\Omega) :\ D^\alpha u = 0\ \forall\ \mathbf{x} \in \Gamma,\ 0 \le |\alpha| \le k-1\},$$

which is also a Hilbert space when equipped with $(\cdot,\cdot)_{H^k(\Omega)}$ and $\|\cdot\|_{H^k(\Omega)}$. Finally, we note that it is standard to denote the dual space of bounded linear functionals over the space $H^k(\Omega)$ as $H^{-k}(\Omega)$, with the corresponding dual norm $\|\cdot\|_{H^{-k}(\Omega)}$.

While the Sobolev spaces may be defined for arbitrary domains Ω, in order for the various well-known properties of these spaces to hold, the set Ω must satisfy certain conditions. These conditions usually include being bounded, and having a locally Lipshitz boundary, which is essentially a smoothness assumption on the boundary Γ excluding certain types of domains such as those with cusps. The Lipshitz condition is simply that for each point $x_0 \in \Gamma$ there exists $\delta > 0$ such that $\Gamma \bigcap \{\ x \mid \|x - x_0\| < \delta\}$ is the graph of a Lipshitz continuous function. If Ω is bounded and has a locally Lipshitz boundary, then the notation $\Omega \in \mathcal{C}^{0,1}$ is used (c.f. page 67 in [1] or page 47 in [68]). For example, bounded open convex sets $\Omega \subset \mathbb{R}^d$ satisfy $\Omega \in \mathcal{C}^{0,1}$ (Corollary 1.2.2.3 in [76]). Therefore, convex polygonal domains are in $\mathcal{C}^{0,1}$.

2.1.4 The Sobolev Imbedding Theorems and the Trace Theorem

The *Sobolev Imbedding Theorems* are a collection of theorems describing the relationships between the Sobolev spaces and some of the classical functions spaces. To say that a Banach space X is continuously imbedded in a Banach space Y, denoted as $X \hookrightarrow Y$, means that X is a subspace of Y, and that there exists a bounded and linear (hence continuous), one-to-one mapping A from X into Y. If the mapping A is compact (i.e., A maps bounded sets into pre-compact sets), then the imbedding is called *compact*; if the image $AX \subset Y$ is *dense* in Y (the closure of the image is Y, or $\overline{AX} = Y$), then the imbedding is called *dense*. One of the main Sobolev imbedding theorems is (case C of Theorem 5.4 in [1]):

Theorem 2.4 (Sobolev Imbedding Theorem) *If $\Omega \subset \mathbb{R}^d$ satisfies $\Omega \in \mathcal{C}^{0,1}$, then for nonnegative integers k and s and $1 \le p < \infty$ it is true that:*

$$W^{k,p}(\Omega) \hookrightarrow C^s(\bar\Omega), \qquad k > s + \frac{d}{p}.$$

Proof. See for example page 97 in [1]. □

In particular, this theorem implies that there exists C such that:

$$\max_{\mathbf{x}\in\bar\Omega} |D^\alpha u(\mathbf{x})| \le C\|u\|_{H^k(\Omega)}, \qquad 0 \le |\alpha| \le s.$$

These imbedding theorems may also be interpreted in terms of fractional exponents; fractional order Sobolev spaces can be defined in several equivalent ways, the most intuitive being through the use of the Fourier transform (page 109 in [128]).

To discuss the concept of a weak solution, we must have a notion of the restriction of a function in a Sobolev space to the boundary Γ of the domain Ω; the following theorem states that this is always possible for domains $\Omega \in \mathcal{C}^{0,1}$.

Theorem 2.5 (The Trace Theorem) *If $\Omega \in \mathbb{R}^d$ satisfies $\Omega \in \mathcal{C}^{0,1}$, then there exists exactly one bounded linear operator T, the "trace operator", which maps $u \in W^{1,p}(\Omega)$ to a function $Tu \in L^p(\Gamma)$, and such that if $u \in C^\infty(\bar{\Omega})$, then $Tu = u|_\Gamma$. This implies that there exists C such that $\|u\|_{L^2(\Gamma)} \leq C\|u\|_{H^1(\Omega)}$.*

Proof. See for example Theorem 8.15 in [128]. □

Remark 2.1. Note that if one defines the fractional order Sobolev spaces, then $L^p(\Gamma)$ can be replaced with the space $W^{1-1/p,p}(\Gamma)$. It can be shown that a function g is the trace of some function in $W^{1,p}(\Omega)$ if and only if $g \in W^{1-1/p,p}(\Gamma)$; this result can be found in Theorem 6.8.13 and Theorem 6.9.2 in [130]. If $g \in W^{1-1/p,p}(\Gamma)$ is the trace of $u \in W^{1,p}(\Omega)$, then we denote this as $g = u|_\Gamma$, or $g = \text{tr } u$.

An important relationship in Sobolev spaces involving a function and its gradient is the *Poincaré-Friedrichs inequality* (page 12 in [36]), also called *Friedrich's first inequality*, which can be derived from the Sobolev imbedding theorems:

$$\|u\|_{L^2(\Omega)} \leq \rho |u|_{H^1(\Omega)}, \qquad \forall u \in H_0^1(\Omega), \qquad \rho \in \mathbb{R}, \qquad \rho > 0. \tag{2.7}$$

2.1.5 Green's identities and weak solutions to elliptic equations

To relate the weak solution of a boundary value problem to the classical solution, we need Green's integral identities, generalized to Sobolev spaces.

Theorem 2.6 (Green's Integral Identities) *If $\Omega \subset \mathbb{R}^d$ satisfies $\Omega \in \mathcal{C}^{0,1}$, then for $p > 1$, $q > 1$, $1/p + 1/q = 1$, and $u \in W^{1,p}$, $v \in W^{1,q}$, it is true that:*

$$\int_\Omega D_i u(\mathbf{x}) v(\mathbf{x})\ d\mathbf{x} = -\int_\Omega u(\mathbf{x}) D_i v(\mathbf{x})\ d\mathbf{x} + \int_\Gamma u(\mathbf{x}) v(\mathbf{x}) \nu_i\ ds$$

where D_i is the weak derivative with respect to x_i, and ν_i is the i-th component of the unit vector normal to Γ. The functions in the surface integral are understood to be the traces of the functions u and v.

Proof. See Theorem 1.1 on page 121 of [150], or Theorem 13.12 in [68]. □

Now, consider the case of equations (2.1)–(2.3). If there are discontinuities present in the equation coefficients, specifically in the coefficient $\bar{\mathbf{a}}(\mathbf{x})$, then it makes no sense in terms of a classical solution. The definition of the weak solution centers on a weaker form of the problem; although the weak solution is defined very generally, one can show (using the Green's integral identities) that the weak solution is exactly the classical solution in the case that the coefficients and the domain are "nice" enough (c.f. Corollary 8.11 in [70]). In addition, finite element methods and variational multigrid methods are constructed from weak formulations of the given elliptic problem. Therefore, we will derive the general weak forms of problems (2.1)–(2.3) and (2.4)–(2.6).

Defining first the following subspace of $H^1(\Omega)$:

$$H^1_{0,D}(\Omega) = \{u \in H^1(\Omega) : u = 0 \text{ on } \Gamma_D\}.$$

Note that $H^1_{0,D}(\Omega)$ is also a Hilbert space. We now begin by multiplying the strong form equation (2.1) by a test function $v \in H^1_{0,D}(\Omega)$ and integrating (in the Lebesgue sense) over Ω to obtain:

$$\int_\Omega \left(-\nabla \cdot (\bar{\mathbf{a}} \nabla \hat{u}) + b\hat{u}\right) v\ d\mathbf{x} = \int_\Omega f v\ d\mathbf{x},$$

which becomes, after applying generalized Green's identities,

$$\int_\Omega (\bar{\mathbf{a}}\nabla \hat{u}) \cdot \nabla v \, d\mathbf{x} - \int_\Gamma v(\bar{\mathbf{a}}\nabla \hat{u}) \cdot \mathbf{n} \, ds + \int_\Omega b\hat{u}v \, d\mathbf{x} = \int_\Omega fv \, d\mathbf{x}. \tag{2.8}$$

The boundary integral above is reformulated using (2.3) as follows:

$$\int_\Gamma v(\bar{\mathbf{a}}\nabla \hat{u}) \cdot \mathbf{n} \, ds = \int_{\Gamma_D} v(\bar{\mathbf{a}}\nabla \hat{u}) \cdot \mathbf{n} \, ds + \int_{\Gamma_N} v(\bar{\mathbf{a}}\nabla \hat{u}) \cdot \mathbf{n} \, ds = 0 + \int_{\Gamma_N} v(g_N - c\hat{u}) \, ds. \tag{2.9}$$

If the boundary function g_D is regular enough so that $g_D \in H^{1/2}(\Gamma_D)$, then from the Trace Theorem (refer to Theorem 2.5 above and the discussion following the theorem), there exists $w \in H^1(\Omega)$ such that $g_D = \text{tr } w$. Employing such a function $w \in H^1(\Omega)$ satisfying $g_D = \text{tr } w$, we define the following affine or translated Sobolev space:

$$H^1_{g,D}(\Omega) = \{\hat{u} \in H^1(\Omega) : u + w, \ u \in H^1_{0,D}(\Omega), \ g_D = \text{tr } w\}.$$

It is easily verified that the solution $\hat{u}$ to the problem (2.1)–(2.3), if one exists, lies in $H^1_{g,D}(\Omega)$, although unfortunately $H^1_{g,D}(\Omega)$ is not a Hilbert space, since it is not linear. (Consider that if $u, v \in H^1_{g,D}(\Omega)$, it holds that $u + v \notin H^1_{g,D}(\Omega)$.) It is important that the problem be phrased in terms of Hilbert spaces such as $H^1_{0,D}(\Omega)$, in order that certain analysis tools and concepts be applicable. Therefore, we will do some additional transformation of the problem.

So far, we have shown that the solution to the original problem (2.1)–(2.3) also solves the following problem:

$$\text{Find } \hat{u} \in H^1_{g,D}(\Omega) \text{ such that } \hat{A}(\hat{u}, v) = \hat{F}(v) \quad \forall v \in H^1_{0,D}(\Omega), \tag{2.10}$$

where from equations (2.8) and (2.9), the bilinear form $\hat{A}(\cdot,\cdot)$ and the linear functional $\hat{F}(\cdot)$ are defined as:

$$\hat{A}(\hat{u}, v) = \int_\Omega (\bar{\mathbf{a}}\nabla \hat{u} \cdot \nabla v + b\hat{u}v) \, d\mathbf{x} + \int_{\Gamma_N} c\hat{u}v \, ds, \qquad \hat{F}(v) = \int_\Omega fv \, d\mathbf{x} + \int_{\Gamma_N} g_N v \, ds.$$

Since we can write the solution $\hat{u}$ to equation (2.10) as $\hat{u} = u + w$ for a fixed w satisfying $g_D = \text{tr } w$, we can rewrite the equations completely in terms of u and a new bilinear form $A(\cdot,\cdot)$ and linear functional $F(\cdot)$ as follows:

$$\text{Find } u \in H^1_{0,D}(\Omega) \text{ such that } A(u, v) = F(v) \quad \forall v \in H^1_{0,D}(\Omega), \tag{2.11}$$

$$A(u, v) = \int_\Omega \bar{\mathbf{a}}\nabla u \cdot \nabla v + buv \, d\mathbf{x} + \int_{\Gamma_N} cuv \, ds, \tag{2.12}$$

$$F(v) = \int_\Omega fv \, d\mathbf{x} + \int_{\Gamma_N} g_N v \, ds - A(w, v). \tag{2.13}$$

Clearly, the "weak" formulation of the problem given by equation (2.11) imposes only one order of differentiability on the solution u, and only in the weak sense. It is easily verified that $A(u, v) : H^1_{0,D}(\Omega) \times H^1_{0,D}(\Omega) \mapsto \mathbb{R}$ defines a bilinear form, and $F(u) : H^1_{0,D}(\Omega) \mapsto \mathbb{R}$ defines a linear functional. Since $H^1_{0,D}(\Omega)$ is a Hilbert space, if it can also be verified that $A(\cdot,\cdot)$ is bounded and coercive on $H^1_{0,D}(\Omega)$, and if $F(u)$ can be shown to be in the dual space $(H^1_{0,D}(\Omega))^* = H^{-1}(\Omega)$ of bounded linear functionals on $H^1_{0,D}(\Omega)$, then the existence and uniqueness of a weak solution to equation (2.11) follows from the Lax-Milgram Theorem (Theorem 2.3). The boundedness and coerciveness conditions must be verified on an individual basis for different coefficient functions $\{\bar{\mathbf{a}}, b, c, f, g_D, g_N\}$ and for different domains Ω.

If the boundary conditions supplied are only Dirichlet, so that $\Gamma_D \equiv \Gamma$ and $\Gamma_N \equiv \emptyset$, which gives that $H^1_{0,D}(\Omega) \equiv H^1_0(\Omega)$, then the weak formulation simplifies to

$$\text{Find } u \in H^1_0(\Omega) \text{ such that } A(u, v) = F(v) \quad \forall v \in H^1_0(\Omega), \tag{2.14}$$

$$A(u, v) = \int_\Omega \bar{\mathbf{a}}\nabla u \cdot \nabla v + buv \, d\mathbf{x}, \tag{2.15}$$

$$F(v) = \int_\Omega fv \ d\mathbf{x} - A(w,v). \tag{2.16}$$

The above discussion is also valid for nonlinear equations of the form (2.4)–(2.6). The required weak formulation is easily arrived at:

$$\text{Find } u \in H^1_{0,D}(\Omega) \text{ such that } A(u,v) + (N(u),v) = F(v) \quad \forall v \in H^1_{0,D}(\Omega), \tag{2.17}$$

where the corresponding forms are:

$$A(u,v) = \int_\Omega \bar{\mathbf{a}}\nabla u \cdot \nabla v \ d\mathbf{x} + \int_{\Gamma_N} cuv \ ds, \qquad (N(u),v) = \int_\Omega b(\mathbf{x}, w+u)v \ d\mathbf{x}, \tag{2.18}$$

$$F(v) = \int_\Omega fv \ d\mathbf{x} + \int_{\Gamma_N} g_N v \ ds - A(w,v), \tag{2.19}$$

and where the fixed known function $w \in H^1(\Omega)$ has trace $g = \text{tr } w$. Since the form $(N(\cdot),\cdot)$ is nonlinear, the Lax-Milgram Theorem cannot be applied, and more general methods must be used to show existence and uniqueness, such as topological, fixed-point, or variational methods (see for example the discussions in [68]). In fact, for the weak solution to even be defined, the integrals above must be finite; this is not immediately true if rapid nonlinearities are present.

If the boundary conditions supplied are only Dirichlet, then as in the linear case, the weak formulation simplifies to

$$\text{Find } u \in H^1_0(\Omega) \text{ such that } A(u,v) + (N(u),v) = F(v) \quad \forall v \in H^1_0(\Omega), \tag{2.20}$$

where the corresponding forms are:

$$A(u,v) = \int_\Omega \bar{\mathbf{a}}\nabla u \cdot \nabla v \ d\mathbf{x}, \qquad (N(u),v) = \int_\Omega b(\mathbf{x}, w+u)v \ d\mathbf{x}, \tag{2.21}$$

$$F(v) = \int_\Omega fv \ d\mathbf{x} - A(w,v), \tag{2.22}$$

Remark 2.2. For second order elliptic problems as discussed above, it is often possible to prove regularity theorems or "shift theorems" of the form:

$$\|u\|_{H^{1+\alpha}(\Omega)} \le C\|f\|_{H^{(1+\alpha)-2}(\Omega)} = C\|f\|_{H^{\alpha-1}(\Omega)},$$

where $0 < \alpha \le 1$. These theorems can be shown using the closed graph theorem when the problem coefficients and domain are smooth enough; see for example page 51 in [167], or page 75 in [172]. The book [76] is also devoted to establishing these types of theorems. For a discussion of regularity results and their impact on multilevel methods and convergence theory, see [46, 141]. In the analysis of finite element and multilevel numerical methods, these theorems are often essential, and are referred to as *elliptic regularity assumptions.* In fact, many multigrid convergence proofs rely on a particularly strong form of the above inequality called the *full elliptic regularity assumption*, which requires that the above inequality hold with $\alpha = 1$. These proofs rely on the use of duality arguments (which employ the elliptic regularity assumptions) originating in the finite element error analysis literature, often referred to as L^2-*lifting* or the *Aubin-Nitsche* trick (pages 136-139 in [36]). Unfortunately, for problems with discontinuous coefficients, these inequalities either do not hold at all, or hold only with extremely large constants C depending on the magnitudes of the coefficient jumps.

2.1.6 Nonlinear operators and the Gateaux and Frechet derivatives

In this section we present some background material regarding nonlinear operators on Hilbert spaces, and the basic ideas of Gateaux and Frechet derivatives of nonlinear operators.

Let $\mathcal{H}_1$, $\mathcal{H}_2$, and $\mathcal{H}$ be real Hilbert spaces, each with an associated inner-product $(\cdot,\cdot)$ inducing a norm $\|\cdot\| = (\cdot,\cdot)^{1/2}$. Let $F(\cdot)$ be a nonlinear operator such that $F : D \subset \mathcal{H}_1 \mapsto \mathcal{H}_2$. If $F(\cdot)$ is both one-to-one and onto, then it is called a *bijection*, in which case the inverse mapping $F^{-1}(\cdot)$ exists. If both $F(\cdot)$ and $F^{-1}(\cdot)$

are continuous, then $F(\cdot)$ is called a *homeomorphism.* Concerning the solution of the operator equation $F(u) = v$, it is important that $F(\cdot)$ be a homeomorphism for the problem to be *well-posed in the Hadamard sense.*[1]

The following notions of differentiation of operators on abstract spaces are important.

Definition 2.1 *The mapping $F : D \subset \mathcal{H}_1 \mapsto \mathcal{H}_2$ is called Gateaux- or G-differentiable at $u \in \operatorname{int}(D)$ if there exists $F'(u) \in \mathbf{L}(\mathcal{H}_1, \mathcal{H}_2)$ such that for any $h \in \mathcal{H}_1$:*

$$\lim_{t \to 0} \frac{1}{t} \|F(u + th) - F(u) - tF'(u)(h)\| = 0.$$

The linear operator $F'(u)$ is unique and is called the G-derivative of F at u. The *directional derivative* of F at u in the direction h is given by $F'(u)(h)$.

Definition 2.2 *The mapping $F : D \subset \mathcal{H}_1 \mapsto \mathcal{H}_2$ is called Frechet- or F-differentiable at $u \in \operatorname{int}(D)$ if there exists $F'(u) \in \mathbf{L}(\mathcal{H}_1, \mathcal{H}_2)$ such that for any $h \in \mathcal{H}_1$:*

$$\lim_{\|h\| \to 0} \frac{1}{\|h\|} \|F(u + h) - F(x) - F'(u)(h)\| = 0.$$

The unique linear operator $F'(u)$ is called the F-derivative of F at u. It is clear from the definitions above that the existence of the F-derivative implies the existence of the G-derivative, in which case it is easy to show they are identical; otherwise, the G-derivative can exist more generally than the F-derivative.

Consider the functional $J : \mathcal{H} \mapsto \mathbb{R}$, defined in terms of a bounded linear operator $A \in \mathbf{L}(\mathcal{H}, \mathcal{H})$ in the following way:

$$J(u) = \frac{1}{2}(Au, u) \quad \forall u \in \mathcal{H}.$$

From the definition of the F-derivative above, it is easy to see (cf. page 418 of [44]) that the F-derivative of J at $u \in \mathcal{H}$ is a bounded linear functional, or $J'(\cdot)(\cdot) : \mathcal{H} \mapsto \mathbf{L}(\mathcal{H}, \mathbb{R})$. (Refer to [117] for discussions of general multi-linear forms on a Hilbert space $\mathcal{H}$.) We can calculate the F-derivative of $J(\cdot)$ by identifying the component of $[J(u + h) - J(u)]$ which is linear in h as follows:

$$J(u + h) - J(u) = \frac{1}{2}(A(u + h), u + h) - \frac{1}{2}(Au, u)$$

$$= \frac{1}{2}(Au, u) + \frac{1}{2}(Au, h) + \frac{1}{2}(Ah, u) + \frac{1}{2}(Ah, h) - \frac{1}{2}(Au, u)$$

$$= \frac{1}{2}(Au, h) + \frac{1}{2}(h, A^T u) + \frac{1}{2}(Ah, h) = \frac{1}{2}((A + A^T)u, h) + \frac{1}{2}(Ah, h)$$

$$= \frac{1}{2}((A + A^T)u, h) + O(\|h\|^2).$$

The *F-derivative* of $J(\cdot)$ is then

$$J'(u)(v) = \frac{1}{2}((A + A^T)u, v) \quad \forall v \in \mathcal{H},$$

where A^T is the adjoint of A with respect to $(\cdot, \cdot)$. It follows from the Reisz Representation Theorem (Theorem 2.2) that $J'(u)(\cdot)$ can be identified with an element $J'(u) \in \mathcal{H}$,

$$J'(u)(v) = (J'(u), v) \quad \forall v \in \mathcal{H},$$

called the *gradient* or the *F-differential* of $J(\cdot)$ at u, which in this case is $J'(u) = \frac{1}{2}(A + A^T)u$.

It is not difficult to see (see for example [117] page 505 for discussion) that the second F-derivative of J at u can be interpreted as a (symmetric) bilinear form, or $B(\cdot, \cdot) = J''(\cdot)(\cdot, \cdot) : \mathcal{H} \mapsto \mathbf{L}(\mathcal{H} \times \mathcal{H}, \mathbb{R})$. To

[1] Well-posedness "in the sense of Hadamard" [124] refers to three criteria: existence of a solution, uniqueness of a solution, and continuous dependence of the solution on the data of the problem.

calculate $J''(\cdot)$, we simply identify the component of $[J(u+h)-J(u)]$ which is now *quadratic* in h. From above, we see that the quadratic term is:

$$B(h,h)=\frac{1}{2}(Ah,h).$$

To recover the full symmetric form $B(u,v)$ from $B(h,h)$, we can employ the following standard trick:

$$\begin{aligned} B(v+w,v+w) &= B(v,v)+2B(v,w)+B(w,w) \\ \Rightarrow \qquad B(v,w) &= \frac{1}{2}[B(v+w,v+w)-B(v,v)-B(w,w)]. \end{aligned}$$

This yields:

$$\begin{aligned} J''(u)(v,w) &= \frac{1}{2}[(A(v+w),v+w)-(Av,v)-(Aw,w)] \\ &= \frac{1}{2}[(Av,w)+(Aw,v)] = \frac{1}{2}[(Av,w)+(w,A^Tv)] \\ &= \frac{1}{2}((A+A^T)v,w) \qquad \forall v,w\in\mathcal{H}. \end{aligned}$$

It now follows (from Theorem 2.1) that $J''(u)(\cdot,\cdot)$ can be identified with the bounded linear operator $J''(u)\in\mathbf{L}(\mathcal{H},\mathcal{H})$ such that

$$J''(u)(v,w)=(J''(u)v,w)=\frac{1}{2}((A+A^T)v,w),$$

so that $J''(u)=\frac{1}{2}(A+A^T)$. Of course, $J''(\cdot)$ can be computed directly from the definition of the F-derivative, beginning with $J'(\cdot)$. Finally, note that if A is self-adjoint, then the expressions for $J'(\cdot)$ and $J''(\cdot)$ simplify to:

$$J'(u)=Au, \qquad \text{and} \qquad J''(u)=A.$$

Consider now the functional $J:\mathcal{H}\mapsto\mathbb{R}$, defined in terms of a nonlinear operator $F:\mathcal{H}\mapsto\mathcal{H}$ as follows:

$$J(u)=\frac{1}{2}\|F(u)\|^2=\frac{1}{2}(F(u),F(u)).$$

We will find the following result quite useful for the solution of nonlinear equations in a later chapter.

Lemma 2.7 *The Frechet derivative of* $J(u)=\frac{1}{2}\|F(u)\|^2$ *is given by* $J'(u)=F'(u)^TF(u)$.

Proof. We identify the component of $[J(u+h)-J(u)]$ linear in h by expanding $F(\cdot)$ in a generalized Taylor series in $\mathcal{H}$ (see for example page 255 in [124]) about the point u:

$$\begin{aligned} J(u+h)-J(u) &= \frac{1}{2}(F(u+h),F(u+h))-\frac{1}{2}(F(u),F(u)) \\ &= \frac{1}{2}(F(u)+F'(u)h+\cdots,F(u)+F'(u)h+\cdots)-\frac{1}{2}(F(u),F(u)) \\ &= \frac{1}{2}(F'(u)h,F(u))+\frac{1}{2}(F(u),F'(u)h)+O(\|h\|^2) \\ &= (F'(u)^TF(u),h)+O(\|h\|^2). \end{aligned}$$

Finally, from the Reisz Theorem we have $J'(u)=F'(u)^TF(u)$. □

Note that, in the case of linear operators, it is easy to show that boundedness is equivalent to continuity, and that all linear operators on finite-dimensional spaces are bounded. However, this is not true in the general case of nonlinear operators, and a separate notion of continuity is required. In $\epsilon-\delta$ verbage, the mapping $F:D\subset\mathcal{H}\mapsto\mathcal{H}$ is called *continuous* at $u\in D$ if, given $\epsilon>0$, there exists $\delta=\delta(u,\epsilon)>0$, such that if $v\in D$ and $\|u-v\|<\delta$, then $\|F(u)-F(v)\|<\epsilon$. If F is continuous at each $u\in D$ then F is called continuous on D, and further, if $\delta=\delta(\epsilon)$, then F is called *uniformly continuous* on D. An equivalent and perhaps more intuitive definition of continuity is that $\lim_{n\to\infty}u^n=u^*$ implies $\lim_{n\to\infty}F(u^n)=F(u^*)$, where $\{u^n\}$ is a sequence, $u^n\in\mathcal{H}$. It can be shown that F-differentiability implies continuity (see for example Theorem 3.1.6 in [158]).

2.1.7 Gradient mappings and convex functionals

In this section, we discuss the fundamental ideas about convex functionals and their associated gradient mappings. The partial differential equations we are considering in this work arise naturally from an associated minimization problem as the *Euler* or *Euler-Lagrange equations*, which represent the requirement that the first variation of the associated energy functional vanish at the minimum of the functional. The *calculus of variations* [131], developed in the 18th century, was the extension of the ideas of critical points and extreme points of a function $f : \mathbb{R} \mapsto \mathbb{R}$ to functionals on function spaces. It was made rigorous with the introduction of the G- and F-derivatives of operators on function spaces early in this century (cf. the excellent book [68] for some historical comments with regard to nonlinear partial differential equations). As a result of this historical development, it is common to refer to zero-point problems which are associated with minimization problems as *variational problems.*

Consider now the (energy) functional, $J : \mathcal{H} \mapsto \mathbb{R}$. A *global minimizer* of $J(\cdot)$ on $\mathcal{H}$ is a point $u^* \in \mathcal{H}$ such that $J(u^*) = \min_{v\in\mathcal{H}} J(v)$. A *local minimizer* of $J(\cdot)$ on $\mathcal{H}$ is a point $u^* \in D \subset \mathcal{H}$ such that $J(u^*) = \min_{v\in D} J(v)$. Assume now that

$$J'(u) = F(u), \qquad \forall u \in \mathcal{H}.$$

This leads us to the important concept of a *gradient mapping.*

Definition 2.3 *The mapping $F : D \subset \mathcal{H} \mapsto \mathcal{H}$ is called a gradient or potential mapping if for some G-differentiable functional $J : D \subset \mathcal{H} \mapsto \mathbb{R}$ it holds that $F(u) = J'(u)$ $\forall u \in D$.*

Regarding the functional $J(\cdot)$, the following are some minimal important concepts.

Definition 2.4 *The functional $J : D \subset \mathcal{H} \mapsto \mathbb{R}$ is called convex on D if $\forall u, v \in D$ and $\alpha \in (0, 1)$ it holds that:*

$$J(\alpha u + (1-\alpha)v) \le \alpha J(u) + (1-\alpha)J(v),$$

whenever the right-hand side of the inequality is defined.

The functional $J(\cdot)$ is called *strictly convex* if the inequality in Definition 2.4 is strict. If $J(u) \to +\infty$ when $\|u\| \to +\infty$, then $J(\cdot)$ is said to be *coercive.* The functional $J(\cdot)$ is called *lower semi-continuous* at the point $v_0 \in D$ if for any $t < J(v_0)$ there exists $\delta > 0$ such that for all v with $\|v - v_0\| < \delta$, it holds that $t < J(v)$. It can be shown (page 159 in [115]) that $J(\cdot)$ is lower semi-continuous at v_0 if and only if:

$$J(v_0) = \underline{\lim}_{v\to v_0} J(v) = \liminf_{v\to v_0} J(v) = \sup_{\delta>0} \inf\{J(v) \mid \|v - v_0\| < \delta\}.$$

If $J \not\equiv +\infty$, $J(v) > -\infty$ $\forall v \in D$, then $J(\cdot)$ is called *proper* on D.

We are interested in the connection between the following two problems:

Problem 1: Find $u \in D \subset \mathcal{H}$ such that $J(u) = \inf_{v\in D\subset\mathcal{H}} J(v)$.
Problem 2: Find $u \in D \subset \mathcal{H}$ such that $F(u) = J'(u) = 0$.

The *Euler necessary condition* for the existence of a local minimizer formalizes the idea of critical points of functionals $J(\cdot)$ on $\mathcal{H}$.

Theorem 2.8 *(Euler Condition) If the functional $J : D \subset \mathcal{H} \mapsto \mathbb{R}$ is G-differentiable with $F(u) = J'(u)$ $\forall u \in D$, and if u^* is a local minimizer of $J(\cdot)$, then the $F(u^*) = 0$.*

Proof. See Corollary 8.3.1 in [44]. □

The following theorem gives sufficient conditions for Problem 1 to be uniquely solvable.

Theorem 2.9 *(Ekland-Temam Theorem) If $J : \mathcal{D} \subset \mathcal{H} \mapsto \mathbb{R}$ is a convex, lower semi-continuous, proper, and coercive functional, with $\mathcal{D}$ a non-empty closed convex subset of $\mathcal{H}$, then $J(\cdot)$ has a local minimizer $u^* \in D$. Further, if $J(\cdot)$ is strictly convex on D, then u^* is unique.*

Proof. See Proposition 1.2 in [62]. □

2.2 A solution theory for the PBE

Consider the general second order linear elliptic equation (2.1), which as we have seen has the weak form:

$$\text{Find } u \in H_0^1(\Omega) \text{ such that } A(u,v) = F(v) \quad \forall v \in H_0^1(\Omega), \tag{2.23}$$

where

$$A(u,v) = \int_\Omega (\bar{\mathbf{a}}\nabla u \cdot \nabla v + buv)\ d\mathbf{x}, \qquad F(v) = \int_\Omega fv\ d\mathbf{x} - A(w,v), \qquad \Omega \subset \mathbb{R}^d, \tag{2.24}$$

with $\Omega \in \mathcal{C}^{0,1}$ and $g = \text{tr } w$, and where the problem coefficients satisfy:

1. $0 < a_{ij}(\mathbf{x}) \le c_1 < \infty,\ \forall \mathbf{x} \in \Omega,\ i,j = 1,\dots,d.$
2. $0 \le b(\mathbf{x}) \le c_2 < \infty,\ \forall \mathbf{x} \in \Omega.$
3. $f(\mathbf{x}) \in L^2(\Omega).$
4. $w(\mathbf{x}) \in H^1(\Omega),\ g(\mathbf{x}) \in H^{1/2}(\Gamma),\ g = \text{tr } w.$
5. The differential operator is *strongly elliptic*:
   $\exists \lambda > 0$ such that $\sum_{ij} a_{ij}(\mathbf{x})\eta_i\eta_j \ge \lambda|\eta|^2,\ \forall \mathbf{x} \in \Omega,\ \eta \in \mathbb{R}^d.$

We begin our analysis with this problem.

2.2.1 A preliminary lemma

Given the very weak assumptions on the coefficients in the above problem, we can prove the following preliminary result, which will be used later to prove the existence of a unique solution to the linearized Poisson-Boltzmann equation.

Lemma 2.10 *There exists a unique weak solution $u \in H_0^1(\Omega)$ to problem (2.23)-(2.24).*

Proof. We show that with the assumptions on the problem coefficients, the conditions of the Lax-Milgram Theorem are met, and the existence and uniqueness of a weak solution $u \in H_0^1(\Omega)$ to (2.23)–(2.24) follows by application of the Lax-Milgram Theorem.

First, it is immediately clear from the linearity of Lebesgue integration that $A(\cdot,\cdot)$ and $F(\cdot)$ define bilinear and linear forms on $H_0^1(\Omega)$, respectively. We must show that $F(\cdot)$ is bounded, and that $A(\cdot,\cdot)$ is bounded and coercive.

Consider the bilinear form $A(\cdot,\cdot)$. The strong ellipticity assumption and nonnegativity of $b(\mathbf{x})$ gives

$$A(u,u) = \int_\Omega (\bar{\mathbf{a}}\nabla u \cdot \nabla u + bu^2)\ d\mathbf{x} = \int_\Omega (\sum_{i,j=1}^d a_{ij}D_iuD_ju + bu^2)\ d\mathbf{x}$$

$$\ge \int_\Omega (\lambda \sum_{i=1}^d |D_iu|^2 + bu^2)\ d\mathbf{x} \ge \lambda \int_\Omega \sum_{i=1}^d |D_iu|^2\ d\mathbf{x} = \lambda|u|^2_{H^1(\Omega)}.$$

Since we have assumed only that $b(\mathbf{x})$ is nonnegative, we must employ Friedrich's first inequality to obtain the proper norm of u bounding $A(\cdot,\cdot)$ from below:

$$A(u,u) \ge \lambda|u|^2_{H^1(\Omega)} = \lambda\left(\frac{1}{2}|u|^2_{H^1(\Omega)} + \frac{1}{2}|u|^2_{H^1(\Omega)}\right)$$

$$\ge \lambda\left(\frac{1}{2\rho^2}\|u\|^2_{L^2(\Omega)} + \frac{1}{2}|u|^2_{H^1(\Omega)}\right) \ge m\left(\|u\|^2_{L^2(\Omega)} + |u|^2_{H^1(\Omega)}\right) = m\|u\|^2_{H^1(\Omega)},$$

where $m = \min\{\lambda/2\rho^2, \lambda/2\}$. Therefore $A(\cdot,\cdot)$ is coercive, with coercivity constant m.

It remains to show that $A(\cdot,\cdot)$ is bounded on $H_0^1(\Omega)$. By repeated application of Hölder's inequality, we have that

$$\begin{aligned}
|A(u,v)| &= |\int_\Omega (\bar{\mathbf{a}}\nabla u \cdot \nabla v + buv)\ d\mathbf{x}| = |\sum_{i,j=1}^d \int_\Omega a_{ij} D_i u D_j v\ d\mathbf{x} + \int_\Omega buv\ d\mathbf{x}| \\
&\le \sum_{i,j=1}^d \int_\Omega |a_{ij} D_i u D_j v|\ d\mathbf{x} + \int_\Omega |buv|\ d\mathbf{x} \le \sum_{i,j=1}^d \|a_{ij}\|_{L^\infty(\Omega)} \|D_i u D_j v\|_{L^1(\Omega)} + \|b\|_{L^\infty(\Omega)} \|uv\|_{L^1(\Omega)} \\
&\le \sum_{i,j=1}^d \|a_{ij}\|_{L^\infty(\Omega)} \|D_i u\|_{L^2(\Omega)} \|D_j v\|_{L^2(\Omega)} + \|b\|_{L^\infty(\Omega)} \|u\|_{L^2(\Omega)} \|v\|_{L^2(\Omega)} \\
&\le \sum_{i,j=1}^d \|a_{ij}\|_{L^\infty(\Omega)} \|u\|_{H^1(\Omega)} \|v\|_{H^1(\Omega)} + \|b\|_{L^\infty(\Omega)} \|u\|_{H^1(\Omega)} \|v\|_{H^1(\Omega)} \le M \|u\|_{H^1(\Omega)} \|v\|_{H^1(\Omega)},
\end{aligned}$$

where $M = 9c_1 + c_2 \ge \sum_{i,j=1}^d \|a_{ij}\|_{L^\infty(\Omega)} + \|b\|_{L^\infty(\Omega)}$.

Consider now the linear functional $F(\cdot)$. Since $A(\cdot,\cdot)$ is bounded, we have that:

$$\begin{aligned}
|F(v)| = |\int_\Omega fv\ d\mathbf{x} - A(w,v)| &\le \|f\|_{L^2(\Omega)} \|v\|_{L^2(\Omega)} + M\|w\|_{H^1(\Omega)} \|v\|_{H^1(\Omega)} \\
&\le \|f\|_{L^2(\Omega)} (\|v\|^2_{L^2(\Omega)} + |v|^2_{H^1(\Omega)})^{1/2} + M\|w\|_{H^1(\Omega)} \|v\|_{H^1(\Omega)} \\
&= (\|f\|_{L^2(\Omega)} + M\|w\|_{H^1(\Omega)}) \|v\|_{H^1(\Omega)} \le C\|v\|_{H^1(\Omega)},
\end{aligned}$$

where we have employed the Cauchy-Schwarz inequality, the definition of $\|\cdot\|_{H^1(\Omega)}$, the fact that $f \in L^2(\Omega)$, and that $w \in H^1(\Omega)$ is fixed. Therefore, $F(\cdot)$ is a bounded linear functional on $H_0^1(\Omega)$. □

2.2.2 A priori estimates in the linearized case

Assume that we are given some Hilbert space $\{\mathcal{H}, (\cdot,\cdot)_H, \|\cdot\|_H = (\cdot,\cdot)_H^{1/2}\}$ and the following problem:

$$\text{Find } u \in \mathcal{H} \text{ such that } A(u,v) = F(v) \quad \forall v \in \mathcal{H}, \tag{2.25}$$

where the bilinear form $A(\cdot,\cdot)$ is bounded and coercive on $\mathcal{H} \times \mathcal{H}$, the linear form $F(\cdot)$ is bounded on $\mathcal{H}$, or more explicitly:

$$m\|u\|_H^2 \le A(u,u), \qquad |A(u,v)| \le M\|u\|_H \|v\|_H, \qquad |F(v)| \le L\|u\|_H, \qquad \forall u,v \in \mathcal{H},$$

where m, M, and L are positive constants. It follows from the Lax-Milgram Theorem that problem (2.25) has a unique solution in the space $\mathcal{H}$.

However, in addition to simply saying that $u \in \mathcal{H}$ and that the $\mathcal{H}$-norm of u is finite,

$$\|u\|_H < \infty,$$

we can actually derive a *bound* on the magnitude of the $\mathcal{H}$-norm of u in terms of the parameters m, M, and L above in the following way. We begin with

$$m\|u\|_H^2 \le A(u,u) = F(u) \le L\|u\|_H,$$

and (assuming $u \ne 0$) we have by division

$$\|u\|_H \le \frac{L}{m}.$$

(Note that if $u = 0$ this bound is trivially satisfied.)

Recall now from the previous section that for the particular weak form PDE problem (2.23)–(2.24), the coercivity constant m and the continuity constants M and L took the forms

$$m = \min\{\lambda/2\rho^2, \lambda/2\}, \qquad M = 9c_1 + c_2, \qquad L = \|f\|_{L^2(\Omega)} + M\|w\|_{H^1(\Omega)},$$

where the Hilbert space in question of importance is $H^1(\Omega)$, and the various parameters arose from the Poincare inequality, ellipticity assumptions on the operator, and bounds on the PDE coefficients (refer to the previous section for details). An *a priori* bound on the magnitude of the solution to the problem (2.23)–(2.24) in the $H^1(\Omega)$ norm then takes the form:

$$\|u\|_{H^1(\Omega)} \leq \frac{\|f\|_{L^2(\Omega)} + (9c_1 + c_2)\|w\|_{H^1(\Omega)}}{\min\{\lambda/2\rho^2, \lambda/2\}}.$$

Whether it is possible to obtain such *a priori* bounds on the solution in stronger norms is an extremely difficult question. Such bounds are referred to as *elliptic regularity inequalities*; see Remark (2.2) for some additional comments.

2.2.3 Existence and uniqueness theorems for the linearized PBE

Consider the linearized Poisson-Boltzmann equation, where we allow a very broad range of temperatures $T \in [200K, 400K]$, a broad range of ionic strengths $I_s \in [0, 10]$, and the following representative polygonal domain:

$$\Omega = [0, 100 \overset{o}{\mathrm{A}}] \times [0, 100 \overset{o}{\mathrm{A}}] \times [0, 100 \overset{o}{\mathrm{A}}].$$

We assume that the set of discrete charges $\{\mathbf{x}_1, \ldots, \mathbf{x}_{N_m}\}$ representing the molecule lie well within the domain, and hence far from the boundary Γ of Ω. It is not difficult to show (Chapter 1 in [94]) that the linearized Poisson-Boltzmann equation for the dimensionless potential $u(\mathbf{x})$ then has the form of equation (2.1):

$$-\nabla \cdot (\bar{\mathbf{a}}(\mathbf{x})\nabla u(\mathbf{x})) + b(\mathbf{x})u(\mathbf{x}) = f(\mathbf{x}) \text{ in } \Omega \subset \mathbb{R}^3, \qquad u(\mathbf{x}) = g(\mathbf{x}) \text{ on } \Gamma,$$

with the equivalent weak formulation (2.23)–(2.24). It is shown in Chapter 1 of [94] that the problem coefficients have the following forms, and satisfy the following bounds for the given temperature and ionic strength ranges:

1. $\bar{\mathbf{a}} : \Omega \mapsto \mathbf{L}(\mathbb{R}^3, \mathbb{R}^3),\ a_{ij}(\mathbf{x}) = \delta_{ij}\epsilon(\mathbf{x}),\ 2 \leq \epsilon(\mathbf{x}) \leq 80,\ \forall \mathbf{x} \in \Omega.$
2. $b : \Omega \mapsto \mathbb{R},\ b(\mathbf{x}) = \bar{\kappa}^2(\mathbf{x}),\ 0 \leq \bar{\kappa}^2(\mathbf{x}) \leq 127.0,\ \forall \mathbf{x} \in \Omega.$
3. $f : \Omega \mapsto \mathbb{R},\ f(\mathbf{x}) = C \cdot \sum_{i=1}^{N_m} z_i \delta(\mathbf{x} - \mathbf{x}_i),\ 5249.0 \leq C \leq 10500.0,\ -1 \leq z_i \leq 1,\ \forall \mathbf{x} \in \Omega.$
4. $g : \Gamma \mapsto \mathbb{R},\ g(\mathbf{x}) = [C/(4\pi\epsilon_w)] \cdot\ \sum_{i=1}^{N_m}\ [z_i e^{-\bar{\kappa}(\mathbf{x})|\mathbf{x}-\mathbf{x}_i|/\sqrt{\epsilon_w}}]/|\mathbf{x} - \mathbf{x}_i|,\ \epsilon_w = 80,\ \forall \mathbf{x} \in \Gamma.$

By assuming that the charges $\mathbf{x}_i$ do not lie near the boundary Γ, which will always be the case for our choice of domain and boundary, we see that the boundary function $g(\mathbf{x})$ is a well-behaved continuously differentiable function of $\mathbf{x}$, $g \in C^1(\Gamma)$.

We can use Lemma 2.10 to prove the following result for the linearized Poisson-Boltzmann equation.

Theorem 2.11 *There exists a unique weak solution $u \in H^1(\Omega)$ to the linearized Poisson-Boltzmann equation if the source terms are approximated by $L^2(\Omega)$ functions.*

Proof. The proof consists of verifying the assumptions on the problem coefficients as required to apply Lemma 2.10. First, note that the assumptions on $\bar{\mathbf{a}}(\mathbf{x})$ and $b(\mathbf{x})$ are clearly satisfied. Since the source functions $\delta(\mathbf{x} - \mathbf{x}_i)$ are approximated with functions $f_i(\mathbf{x} - \mathbf{x}_i) \in L^2(\Omega)$, we have that the composite function $f(\mathbf{x}) = C \cdot \sum_{i=1}^{N_m} z_i f_i(\mathbf{x} - \mathbf{x}_i)$ is also clearly in $L^2(\Omega)$. By assuming that the fixed "charge" points $\mathbf{x}_i$ are located away from the boundary Γ, we have that $g \in C^1(\Gamma) \subset H^1(\Gamma)$, and from Remark 2.1 we know that there exists $w \in H^1(\Omega)$ such that $g = \mathrm{tr}\ w$. The strong ellipticity assumption follows from the lower bound on the tensor components $a_{ij}(\mathbf{x})$; in other words, the tensor $\bar{\mathbf{a}}(\mathbf{x})$ is uniformly positive definite in $\mathbf{x}$ due to the lower bound of 2 for $a_{ij}(\mathbf{x})$. The theorem now follows from Lemma 2.10. □

In the case that the function f in (2.24) consists of delta functions representing point charges (and so $f \notin L^2(\Omega)$), the Lax-Milgram Theorem cannot be used to show existence and uniqueness of solutions because the resulting linear functional F in (2.24) is no longer bounded; this is because the imbedding

$W^{1,2}(\Omega) \hookrightarrow C^0(\bar{\Omega})$ given in Theorem 2.4 fails if the spatial dimension d is greater than one. To understand what the problem is, consider the linear functional:

$$F(v) = \int_\Omega fv \; d\mathbf{x} = \int_\Omega \delta(\mathbf{x}_0)v \; d\mathbf{x}, \qquad \forall v \in H_0^1(\Omega). \tag{2.26}$$

For $F(\cdot)$ to be in the dual space of *bounded* linear functionals on $H_0^1(\Omega)$, required for the Lax-Milgram Theory, we must have that the norm:

$$\|F\|_{H^{-1}(\Omega)} = \sup_{v \in H_0^1(\Omega)} \frac{|\int_\Omega \delta(\mathbf{x}_0)v(\mathbf{x}) \; d\mathbf{x}|}{\|v(\mathbf{x})\|_{H^1(\Omega)}} = \sup_{v \in H_0^1(\Omega)} \frac{|v(\mathbf{x}_0)|}{\|v(\mathbf{x})\|_{H^1(\Omega)}}$$

is bounded. Now, if the imbedding $H^1(\Omega) \hookrightarrow C^0(\bar{\Omega})$ holds, then $v(\mathbf{x})$ is a continuous function on the bounded set Ω, and since $\bar{\Omega}$ is close and bounded and hence compact, we must have that $v(\mathbf{x})$ is uniformly continuous, and therefore bounded, $v(\mathbf{x}_0) \le C < \infty$, and so the functional above is bounded. However, if the imbedding fails, which is the case when $d = 2$ or $d = 3$, then $v(\mathbf{x})$ may not be bounded, and hence $v(\mathbf{x}_0)$ may not be finite, so that the linear functional is unbounded.

By this argument combined with Theorem 2.4, it holds that the function $v(\mathbf{x}) \in C^0(\bar{\Omega})$ (so that $v(\mathbf{x})$ is bounded for all $\mathbf{x} \in \Omega \subset \mathbb{R}^d$) only if $v(\mathbf{x}) \in H_0^k(\Omega)$, where k is such that:

- $d = 1$: $k > s + \frac{d}{p} = 0 + \frac{1}{2} = \frac{1}{2}$.
- $d = 2$: $k > s + \frac{d}{p} = 0 + \frac{2}{2} = 1$.
- $d = 3$: $k > s + \frac{d}{p} = 0 + \frac{3}{2} = \frac{3}{2}$.

Therefore, for $\Omega \subset \mathbb{R}^d$ where $d = 3$, if we select $v(\mathbf{x})$ from $H_0^2(\Omega)$ for example, then the linear functional $F(\cdot)$ in (2.26) will be bounded.

This leads to the following approach, which involves the selection of the test functions from a different space than the solution space. First, note that by applying Green's integral identities again, we can produce the following "weaker" form of the problem which imposes no differentiability on the solution u:

$$\text{Find } u \in H_0^0(\Omega) = L_0^2(\Omega) \text{ such that } A(u,v) = F(v) \quad \forall v \in H_0^2(\Omega), \tag{2.27}$$

where

$$A(u,v) = \int_\Omega [\nabla \cdot (\bar{\mathbf{a}}\nabla v)u + buv] \; d\mathbf{x}, \qquad F(v) = \int_\Omega fv \; d\mathbf{x} - A(w,v), \qquad \Omega \subset \mathbb{R}^3. \tag{2.28}$$

The following theorem allows one to work with these two separate Hilbert spaces, allowing the test functions to be chosen from the higher regularity space.

Theorem 2.12 (Generalized Lax-Milgram Theorem) *Let $\mathcal{H}_1$ and $\mathcal{H}_2$ be two real Hilbert spaces, let $A(u,v)$ be a bilinear form on $\mathcal{H}_1 \times \mathcal{H}_2$ which is bounded in the sense that:*

$$|A(u,v)| \le M\|u\|_{\mathcal{H}_1}\|v\|_{\mathcal{H}_2}, \qquad \forall u \in \mathcal{H}_1, \qquad \forall v \in \mathcal{H}_2,$$

and coercive in the sense that:

$$\inf_{\|u\|_{\mathcal{H}_1}=1} \; \sup_{\|v\|_{\mathcal{H}_2}\le 1} |A(u,v)| \ge m > 0, \;\; \sup_{u \in \mathcal{H}_1} |A(u,v)| > 0, \;\; v \ne 0,$$

and let $F(u)$ be a bounded linear functional on $\mathcal{H}_2$. Then there exists a unique solution to the problem:

$$\text{Find } u \in \mathcal{H}_1 \text{ such that } A(u,v) = F(v) \quad \forall v \in \mathcal{H}_2.$$

Proof. See for example Theorem 5.2.1, page 112 in [10]. □

Theorem 2.13 *There exists a unique weak solution $u \in L^2(\Omega)$ to the linearized Poisson-Boltzmann equation.*

Proof. We only outline the proof here, which consists of taking $\mathcal{H}_1 = L^2(\Omega)$, $\mathcal{H}_2 = H_0^2(\Omega)$, and verifying the assumptions of the Generalized Lax-Milgram Theorem. □

2.2.4 An existence and uniqueness theorem for the nonlinear PBE

The standard existence and uniqueness theory for nonlinear elliptic equations, as presented for example in [68], involves restricting the theory to a class of nonlinearities satisfying an extended *Carathéodory property*, denoted CAR(p). Among several properties of the nonlinear functions $b(\mathbf{x}, u(\mathbf{x})) \in$ CAR(p) is a polynomial growth condition (see page 82 of [68]), which requires that the nonlinear function $b(\mathbf{x}, u(\mathbf{x}))$ be bounded by a function which grows as a polynomial of degree p in u. (The degree p is tied to the solution space $W^{k,p}(\Omega)$.) This excludes the nonlinear Poisson-Boltzmann equation, where we are faced with the nonlinear function

$$b(\mathbf{x}, u(\mathbf{x})) = \bar{\kappa}^2(\mathbf{x}) \sinh(u(\mathbf{x})) = \frac{\bar{\kappa}^2(\mathbf{x})}{2}(e^{u(\mathbf{x})} - e^{-u(\mathbf{x})}).$$

The need for the polynomial growth condition is easy to see, if one considers the nonlinear term as it appears in the weak form of the Poisson-Boltzmann equation:

$$\int_\Omega b(\mathbf{x}, u+w)v \; d\mathbf{x} = \int_\Omega \bar{\kappa}^2 \sin(u+w)v \; d\mathbf{x} = \int_\Omega \frac{\bar{\kappa}^2}{2}(e^{u+w} - e^{-u-w})v \; d\mathbf{x}, \qquad \forall v \in H_0^1(\Omega).$$

The problem is that $u \in H^1(\Omega)$ does not guarantee that $b(\mathbf{x}, u) \in L^2(\Omega)$; in other words, in order for the weak solution to even be defined, the integral above involving $b(\cdot,\cdot)$ must be finite. It can be shown that the polynomial growth condition guarantees this for a certain Sobolev space (not necessarily $H^1(\Omega)$). Therefore, it is not clear that the weak solution $u \in H^1(\Omega)$ is even well-defined in the case of the nonlinear Poisson-Boltzmann equation and similar equations.

There appear to be two approaches available for problems with these types of rapid nonlinearities for which the standard theory does not apply. The first is an extended form of *monotone operator theory*, which we will not discuss further; essentially, the idea is to restrict the solution space to a subspace of $H^1(\Omega)$ for which the weak solution is well-defined, and then use a monotone operator theory for the subspace problem. This theory is presented in the papers [33, 60, 92].

A second approach, which we will follow below, is to side-step the question altogether of exactly when $b(\mathbf{x}, u) \in L^2(\Omega)$, by appealing to some results from convex analysis. This approach is taken in [113, 114, 123] for a very similar equation arising in semiconductor physics. Our proof below follows closely the proof of Lemma 3.1 in [114].

Theorem 2.14 *There exists a unique weak solution $u \in H^1(\Omega)$ to the nonlinear Poisson-Boltzmann equation if the source terms are approximated by $L^2(\Omega)$ functions.*

Proof. The idea of the proof is to identify a convex functional $J(\cdot)$ for which the weak form of the nonlinear Poisson-Boltzmann equation is the associated gradient mapping as discussed in §2.1.7 above, and then appeal to the Ekland-Temam Theorem (Theorem 2.9). First, recall the weak formulation of the nonlinear Poisson-Boltzmann equation:

$$\text{Find } u \in H_0^1(\Omega) \text{ such that } A(u,v) + (N(u),v) = F(v) \quad \forall v \in H_0^1(\Omega), \tag{2.29}$$

where the corresponding forms are:

$$A(u,v) = \int_\Omega \bar{\mathbf{a}} \nabla u \cdot \nabla v \; d\mathbf{x}, \qquad (N(u),v) = \int_\Omega b(\mathbf{x}, w+u)v \; d\mathbf{x},$$

$$F(v) = \int_\Omega fv \; d\mathbf{x} - A(w,v),$$

and where the fixed known function $w \in H^1(\Omega)$ has trace $g = \text{tr } w$. The coefficients defining the nonlinear Poisson-Boltzmann equation satisfy:

1. $\bar{\mathbf{a}} : \Omega \mapsto \mathbb{R}, \; a_{ij}(\mathbf{x}) = \delta_{ij}\epsilon(\mathbf{x}), \; 2 \le \epsilon(\mathbf{x}) \le 80, \; \forall \mathbf{x} \in \Omega.$
2. $b : \Omega \times \mathbb{R} \mapsto \mathbb{R}, \; b(\mathbf{x}, u(\mathbf{x})) = \bar{\kappa}^2(\mathbf{x}) \sinh(u(\mathbf{x})), \; 0 \le \bar{\kappa}^2(\mathbf{x}) \le 127.0, \; \forall \mathbf{x} \in \Omega.$
3. $f : \Omega \mapsto \mathbb{R}, \; f(\mathbf{x}) = C \cdot \sum_{i=1}^{N_m} z_i \delta(\mathbf{x} - \mathbf{x}_i), \; 5249.0 \le C \le 10500.0, \; -1 \le z_i \le 1, \; \forall \mathbf{x} \in \Omega.$

4. $g : \Gamma \mapsto \mathbb{R},\ g(\mathbf{x}) = [C/(4\pi\epsilon_w)] \cdot \ \sum_{i=1}^{N_m} \ [z_i e^{-\bar{\kappa}(\mathbf{x})|\mathbf{x}-\mathbf{x}_i|/\sqrt{\epsilon_w}}]/|\mathbf{x}-\mathbf{x}_i|,\ \epsilon_w = 80,\ \forall \mathbf{x} \in \Gamma.$

We assume that the fixed "charges" $\mathbf{x}_i$ do not lie near the boundary Γ, so that the boundary function $g(\mathbf{x})$ is a well-behaved continuously differentiable function of $\mathbf{x}$, $g \in C^1(\Gamma)$.

Following Lemma 3.1 in [113], we begin by defining a nonlinear function, specifically constructed for the case of the Poisson-Boltzmann problem:

$$p(\mathbf{x}, y) = \frac{\bar{\kappa}^2(\mathbf{x})}{2} \left[(e^y - 1)e^{w(\mathbf{x})} + (e^{-y} - 1)e^{-w(\mathbf{x})} \right]. \tag{2.30}$$

Note that $p(\mathbf{x}, 0) = 0$, and that the derivative of $p(\mathbf{x}, y)$ with respect to y yields the Poisson-Boltzmann nonlinearity:

$$p'(\mathbf{x}, y) = \frac{\partial p(\mathbf{x}, y)}{\partial y} = \frac{\bar{\kappa}^2(\mathbf{x})}{2} \left(e^y e^{w(\mathbf{x})} - e^{-y} e^{-w(\mathbf{x})} \right) = \bar{\kappa}^2(\mathbf{x}) \sinh(w(\mathbf{x}) + y) = b(\mathbf{x}, w + y).$$

We now define the functional $J(\cdot) : H_0^1(\Omega) \mapsto \mathbb{R}$ conditionally as:

$$J(v) = \left\{ \begin{array}{ll} \frac{1}{2}A(v, v) + \int_\Omega p(\mathbf{x}, v)\ d\mathbf{x} - F(v), & p(\cdot, v) \in L^2(\Omega) \text{ for } v \in H_0^1(\Omega), \\ +\infty, & p(\cdot, v) \notin L^2(\Omega) \text{ for } v \in H_0^1(\Omega). \end{array} \right\}. \tag{2.31}$$

We will compute directly the F-derivative of $J(\cdot)$, to show that the Euler condition for the particular functional $J(\cdot)$ we have constructed yields the weak form of the nonlinear PBE in equation (2.29) from $J'(u) = 0$, or actually from:

$$\text{Find } u \in \mathcal{H} \text{ such that } \ (J'(u), v) = 0 \quad \forall v \in \mathcal{H}.$$

As discussed earlier, we simply identify the component of $[J(u + h) - J(u)]$ which is linear in h. First, note that $J(\cdot)$ can be written in terms of the weak form components and the nonlinear function $p(\cdot)$ as:

$$J(u) = \frac{1}{2}A(u, u) + (p(\mathbf{x}, u), 1) - F(u).$$

Now consider $[J(u + h) - J(u)]$, where we employ a Taylor expansion of $p(\mathbf{x}, \cdot)$ about $u(\mathbf{x})$:

$$\begin{aligned} J(u+h) - J(u) &= [\frac{1}{2}A(u+h, u+h) + (p(\mathbf{x}, u+h), 1) - F(u+h)] - [\frac{1}{2}A(u, u) + (p(\mathbf{x}, u), 1) - F(u)] \\ &= \frac{1}{2}[A(u, h) + A(h, u) + A(h, h)] + [(p(\mathbf{x}, u) + p'(\mathbf{x}, u)h + \cdots, 1) - (p(\mathbf{x}, u), 1)] \\ &\quad +[F(u) + F(h) - F(u)] \\ &= A(u, h) + (p'(\mathbf{x}, u), h) - F(h) + O(\|h\|^2). \end{aligned}$$

Therefore, the F-derivative of the functional $J(\cdot)$ and the corresponding Euler condition are:

$$\text{Find } u \in \mathcal{H} \text{ such that } \ (J'(u), v) = A(u, v) + (N(u), v) - F(v) = 0 \quad \forall v \in \mathcal{H},$$

where $N(u) = p'(\mathbf{x}, u) = b(\mathbf{x}, w + u)$. This is exactly the weak form of the nonlinear PBE in equation (2.29), so that the weak form of the nonlinear PBE is the gradient mapping formulation of the functional $J(\cdot)$ we have constructed in equation (2.31).

To show that $J(\cdot)$ has a minimum $u \in H_0^1(\Omega)$, we can use Theorem 2.9 (the Ekland-Temam Theorem) if we can show that the functional $J(\cdot)$ is proper, convex, lower semi-continuous, and coercive on $H_0^1(\Omega)$. First, due to the conditional definition of $J(\cdot)$, it follows immediately that $J(v) > -\infty\ \forall v \in H_0^1(\Omega)$, and that $J \not\equiv +\infty$ (take $v \equiv 0$), so by the discussion in §2.1.7, we have that $J(\cdot)$ is a proper functional on the space $H_0^1(\Omega)$. It remains to verify the other three properties for the functional $J(\cdot)$.

In the proof of Lemma 3.1 in [113], it is stated without proof that $J(\cdot)$ as defined for the semiconductor equation is convex. It is then shown that $J(\cdot)$ is lower semi-continuous and coercive by re-norming $H_0^1(\Omega)$ with an equivalent but more convenient norm, and using Fatou's lemma and the Cauchy-Schwarz inequality. We will use similar approach, but will employ some simple results from [162, 172] to establish these three properties for $J(\cdot)$ below.

To begin, recall that all linear functions and linear spaces are trivially convex by the definition of linearity:

$$L(\alpha u + (1-\alpha))v \equiv \alpha Lu + (1-\alpha)Lv.$$

For example, the set $\mathbb{R}$ is a convex set. In addition, a linear combination $F(u) = \sum_{i=1}^{N} c_i f_i(u)$ of general convex functions $f_i(u)$ is again a convex function, since

$$\begin{aligned} F(\alpha u + (1-\alpha)v) &= \sum_{k=1}^{N} c_i f_i(\alpha u + (1-\alpha)v) \leq \sum_{k=1}^{N} c_i[\alpha f_i(u) + (1-\alpha)f_i(v)] \\ &= \alpha\left(\sum_{k=1}^{N} c_i f_i(u)\right) + (1-\alpha)\left(\sum_{k=1}^{N} c_i f_i(v)\right) = \alpha F(u) + (1-\alpha)F(v). \end{aligned}$$

Now, it can be shown (Theorem 2.B, page 155 in [172]) that if $p : K \mapsto \mathbb{R}$ is G-differentiable on a convex set K (e.g., $K = \mathbb{R}$), then $p(\cdot)$ is convex if and only if

$$(p'(u) - p'(v))(u - v) \geq 0, \qquad \forall u, v \in K.$$

Note that, for a fixed $\mathbf{x}$ and a fixed $w(\mathbf{x})$, the Poisson-Boltzmann nonlinearity given by $p'(\mathbf{x}, u) = \bar{\kappa}^2(\mathbf{x})$ $\sinh(w(\mathbf{x}) + u)$ is a monotonically increasing function of u. This follows from the fact that

$$p''(\mathbf{x}, u) = \bar{\kappa}^2(\mathbf{x})\cosh(w(\mathbf{x}) + u) \geq 0, \qquad \forall u \in K.$$

Therefore, $[u - v]$ and $[p(\mathbf{x}, u) - p(\mathbf{x}, v)]$ always have the same sign for any $u, v \in K$, so that $p(\cdot, u)$ is convex for $u \in K$. Now, let $P(u) = \int_\Omega p(\cdot, u)\ d\mathbf{x}$. Since $p(\cdot, u)$ is convex, we have that

$$\begin{aligned} P(\alpha u + (1-\alpha)v) &= \int_\Omega p(\mathbf{x}, \alpha u + (1-\alpha)v)\ d\mathbf{x} \leq \int_\Omega [\alpha p(\mathbf{x}, u) + (1-\alpha)p(\mathbf{x}, v)]\ d\mathbf{x} \\ &= \alpha \int_\Omega p(\mathbf{x}, u)\ d\mathbf{x} + (1-\alpha)\int_\Omega p(\mathbf{x}, v)\ d\mathbf{x} = \alpha P(u) + (1-\alpha)P(v), \end{aligned}$$

so that also $P(u)$ is a convex function of u. This is of course due to the fact that the integral operator is linear. Now, since the other terms comprising the functional $J(\cdot)$ are linear and hence convex, the composite functional $J(\cdot)$ is a linear combination of convex functions, and it follows from our discussion above that $J(\cdot)$ itself is convex on K.

To show lower semi-continuity of $J(\cdot)$, we note that it can be shown (Corollary 2.D in [172]) that if $p(\cdot)$ is G-differentiable and convex on a convex set K, then $p(\cdot)$ is lower semi-continuous on K. Therefore, $p(\cdot)$ as defined in equation (2.30) is lower semi-continuous. In addition, by the Tonelli Theorem (Theorem 9.16, page 347 in [162]), the function $P(u)$ is lower semicontinuous on $L^p(\Omega)$ if and only if $p(\cdot, u)$ is continuous and convex in u; both of these conditions on $p(\cdot, u)$ hold, so that $P(u)$ is lower semicontinuous. Since all linear functions are trivially lower semi-continuous, it follows immediately that the linear combination of lower semicontinuous functionals forming $J(\cdot)$ is itself a lower semi-continuous functional.

To employ the Ekland-Temam Theorem, it remains to show that $J(\cdot)$ is coercive. First, note that if $\alpha = \inf_{\bar{\Omega}} w$, $\beta = \sup_{\bar{\Omega}} w$, we have that:

$$p(\mathbf{x}, v) = \frac{\bar{\kappa}^2}{2}\left[(e^v - 1)\,e^w + \left(e^{-v} - 1\right)e^{-w}\right] \geq -\frac{\bar{\kappa}^2}{2}\left[e^\beta + e^{-\alpha}\right] > -\infty, \tag{2.32}$$

from which it follows that

$$\int_\Omega p(\mathbf{x}, v)\ d\mathbf{x} \geq -\text{meas}(\Omega)\frac{\bar{\kappa}^2}{2}\left[e^\beta + e^{-\alpha}\right] > -\infty.$$

Using a simple argument as in Lemma 2.10 above, due to the boundedness, positivity, and symmetry of the tensor $\bar{\mathbf{a}}(\mathbf{x})$, we have that the energy norm

$$\|v\|_A = A(v, v)^{1/2} = \left(\int_\Omega \bar{\mathbf{a}}\nabla v \cdot \nabla v\ d\mathbf{x}\right)^{1/2}$$

is equivalent to the norm in $H_0^1(\Omega)$ in the sense that:

$$\sqrt{m}\|v\|_{H^1(\Omega)} \le \|v\|_A \le \sqrt{M}\|v\|_{H^1(\Omega)}, \tag{2.33}$$

where m and M are the boundedness and coerciveness constants (for the linear form $A(u,v)$) as in the previous Lemma 2.10. Now, consider that

$$J(v) = \frac{1}{2}\int_\Omega \bar{\mathbf{a}}\nabla v \cdot \nabla v \ d\mathbf{x} + \int_\Omega p(\mathbf{x}, v)\ d\mathbf{x} - F(v)$$

$$\ge \frac{m}{2}\|v\|_{H^1(\Omega)}^2 - \text{meas}(\Omega)\frac{\bar{\kappa}^2}{2}\left[e^\beta + e^{-\alpha}\right] - \|f\|_{L^2(\Omega)}\|v\|_{L^2(\Omega)} - M\|w\|_{H^1(\Omega)}\|v\|_{H^1(\Omega)},$$

where we have employed equations (2.32) and (2.33), and the bound for $F(v)$ which was derived in the proof of Lemma 2.10. Since f and w are fixed, and since

$$\lim_{\|v\|_{L^2(\Omega)}\to+\infty} \frac{\|v\|_{H^1(\Omega)}^2}{\|v\|_{L^2(\Omega)}} = \lim_{\|v\|_{L^2(\Omega)}\to+\infty} \frac{\|v\|_{L^2(\Omega)}^2 + |v|_{H^1(\Omega)}^2}{\|v\|_{L^2(\Omega)}} \ge \lim_{\|v\|_{L^2(\Omega)}\to+\infty} \|v\|_{L^2(\Omega)} = +\infty,$$

we have that $J(v) \to +\infty$ as $\|v\|_{H^1(\Omega)} \to +\infty$, so that $J(\cdot)$ is coercive on $H_0^1(\Omega)$.

Therefore, since we have shown that the functional $J(\cdot)$ is proper, convex, lower semi-continuous, and coercive on $H_0^1(\Omega)$, by the Ekland-Temam Theorem there exists a minimizer $u \in H_0^1(\Omega)$ of $J(\cdot)$.

Finally, it remains to show that the minimizer of $J(\cdot)$, and therefore a solution to the nonlinear PBE, is unique. The uniqueness can be seen by the following argument. Assume that two solutions u_1 and u_2 exist, each of which satisfy the weak form (2.29). If we subtract the two equations, we have by the linearity of $A(\cdot,\cdot)$:

$$A(u_1 - u_2, v) + (N(u_1) - N(u_2), v) = 0 \qquad \forall v \in H_0^1(\Omega).$$

Taking $v = u_1 - u_2$, since $N(u)$ is monotonically increasing in u, it holds that $(N(u_1) - N(u_2), u_1 - u_2) \ge 0$, so that we must have $A(u_1 - u_2, u_1 - u_2) \le 0$. But since $A(\cdot,\cdot)$ defines a norm as discussed above, we must also have

$$A(u_1 - u_2, u_1 - u_2) = \|u_1 - u_2\|_A^2 \ge 0.$$

Thus, $\|u_1 - u_2\|_A = 0$ must hold, which is true only if $u_1 = u_2$. □

Remark 2.3. Note that, by the conditional definition of $J(\cdot)$, the question of whether $e^u \notin L^2(\Omega)$ is avoided altogether; these situations are lumped into the conditional $J(\cdot) = +\infty$, which is a valid definition of a proper convex functional as required for the use of the Ekland-Temam Theorem. We note that a similar proof can be constructed along the lines of the proof of Theorem 2.1 on page 543 of [73].

It is also possible to show uniqueness through the Ekland-Temam Theorem rather than directly as above; the functional $J(\cdot)$ can be shown to be strictly convex rather than simply convex.

2.3 The box-method and properties of the discrete PBE

Consider the linear equation:

$$-\nabla \cdot (\bar{\mathbf{a}}\nabla u) + bu = f \text{ in } \Omega \subset \mathbb{R}^3, \qquad u = g \text{ on } \Gamma. \tag{2.34}$$

We are concerned with the case (as with the linearized Poisson-Boltzmann equation) that the functions $\{\bar{\mathbf{a}}, b, f\}$ are piecewise C^k functions on Ω, with k large. We also assume that the coefficient discontinuities are regular, and can be identified during the discretization process.

We will also consider the strong form of the nonlinear equation:

$$-\nabla \cdot (\bar{\mathbf{a}}\nabla u) + b(\mathbf{x}, u) = f \text{ in } \Omega \subset \mathbb{R}^3, \qquad u = g \text{ on } \Gamma. \tag{2.35}$$

We begin by partitioning the domain Ω into the finite elements or volumes τ^j, such that:

- $\Omega \equiv \bigcup_{j=1}^M \tau^j$, where the *elements* τ^j are arbitrary hexahedra or tetrahedra.

- $\{\bar{\mathbf{a}}, b, f\}$ have discontinuities along boundaries of the τ^j.
- The union of the l (eight or four) corners of all the τ^j form the *nodes* $\mathbf{x}^i$.
- $\{\tau^{j;i}\} \equiv \{\tau^j : \mathbf{x}^i \in \tau^j\}$.
- $\tau^{(i)} \equiv \bigcup_j \tau^{j;i} \equiv \{\bigcup_j \tau^j : \mathbf{x}^i \in \tau^j\}$.
- Continuity required of $u(\mathbf{x})$ and of $\bar{\mathbf{a}}\nabla u \cdot \mathbf{n}$ across interfaces.

We now briefly discuss the box-method, a method for discretizing interface problems which yields reliably accurate approximations. This method, in one form or another, has been one of the standard approaches for discretizing two- and three-dimensional interface problems occurring in reactor physics and reservoir simulation [179, 180]. Similar methods are used in computational fluid dynamics. The motivation for these methods has been the attempt to enforce conservation of certain physical quantities in the discretization process.

Remark 2.4. Note that a standard Taylor series approach to discretization is not reliable since the truncation error depends on various derivatives of the solution, which may be large (or nonexistent) at or near interfaces. A simple application of the finite element method ignoring the interfaces is not reliable either, as the standard error estimates will be invalid due to solution singularities at interface corners or intersections (see for example [174], pg. 266, or [9], pg. 245, or §8.4 in [76]).

2.3.1 General formulation

We begin by integrating (2.34) over an arbitrary $\tau^{(i)}$. Note that in many cases the underlying conservation law was originally phrased in integral form. The resulting equation is:

$$-\sum_j \int_{\tau^{j;i}} \nabla \cdot (\bar{\mathbf{a}}\nabla u)\; d\mathbf{x} + \sum_j \int_{\tau^{j;i}} bu\; d\mathbf{x} = \sum_j \int_{\tau^{j;i}} f\; d\mathbf{x}.$$

Using the divergence theorem, we can rewrite the first term on the left, yielding:

$$-\sum_j \int_{\partial\tau^{j;i}} (\bar{\mathbf{a}}\nabla u) \cdot \mathbf{n}\; d\mathbf{s} + \sum_j \int_{\tau^{j;i}} bu\; d\mathbf{x} = \sum_j \int_{\tau^{j;i}} f\; d\mathbf{x},$$

where $\partial\tau^{j;i}$ is the boundary of $\tau^{j;i}$, and $\mathbf{n}$ is the unit normal to the surface of $\tau^{j;i}$.

Note that all interior surface integrals in the first term vanish, since $\bar{\mathbf{a}}\nabla u \cdot \mathbf{n}$ must be continuous across the interfaces. We are left with:

$$-\int_{\partial\tau^{(i)}} (\bar{\mathbf{a}}\nabla u) \cdot \mathbf{n}\; d\mathbf{s} + \sum_j \int_{\tau^{j;i}} bu\; d\mathbf{x} = \sum_j \int_{\tau^{j;i}} f\; d\mathbf{x}, \tag{2.36}$$

where $\partial\tau^{(i)}$ denotes the boundary of $\tau^{(i)}$.

Since this last relationship holds exactly in each $\tau^{(i)}$, we can use (2.36) to develop an approximation at the nodes $\mathbf{x}_i$ at the "centers" of the $\tau^{(i)}$ by employing quadrature rules and difference formulas. In particular, the volume integrals in the second two terms in (2.36) can be approximated with quadrature rules. Similarly, the surface integrals required to evaluate the first term in (2.36) can be approximated with quadrature rules, where ∇u is replaced with an approximation. Error estimates can be obtained from difference and quadrature formulas, as in Chapter 6 of [179], or more generally by analyzing the box-method as a special Petrov-Galerkin finite element method [18, 122].

Remark 2.5. This procedure is sometimes referred to as the *integral method* in one dimension, the *box-method* in two dimensions, and the *finite volume* method in three dimensions, although it is standard to refer to the method in any dimension as the box-method.

2.3.2 Non-uniform Cartesian meshes

We now restrict ourselves to the case that the τ^j are hexahedral elements, whose six sides are parallel to the coordinate axes. With regard to the notation above, since we are working with $\mathbb{R}^3$, we will define $x = x_1, y = x_2, z = x_3$. By restricting our discussion to elements which are non-uniform Cartesian (or *axi-parallel*), the spatial mesh may be characterized by the nodal points

$$\mathbf{x} = (x, y, z) \text{ such that } \left\{ \begin{array}{c} x \in \{x_0, x_1, \ldots, x_{I+1}\} \\ y \in \{y_0, y_1, \ldots, y_{J+1}\} \\ z \in \{z_0, z_1, \ldots, z_{K+1}\} \end{array} \right\}.$$

Any such mesh point we denote as $\mathbf{x}_{ijk} = (x_i, y_j, z_k)$, and we define the *mesh spacings* as

$$h_i = x_{i+1} - x_i, \qquad h_j = y_{j+1} - y_j, \qquad h_k = z_{k+1} - z_k,$$

which are not required to be equal or uniform.

To each mesh point $\mathbf{x}_{ijk} = (x_i, y_j, z_k)$, we associate the closed three-dimensional hexahedral region $\tau^{(ijk)}$ "centered" at $\mathbf{x}_{ijk}$, defined by

$$x \in \left[x_i - \frac{h_{i-1}}{2}, x_i + \frac{h_i}{2}\right], \quad y \in \left[y_j - \frac{h_{j-1}}{2}, y_j + \frac{h_j}{2}\right], \quad z \in \left[y_k - \frac{h_{k-1}}{2}, z_k + \frac{h_k}{2}\right].$$

Integrating (2.34) over $\tau^{(ijk)}$ for each mesh-point $\mathbf{x}_{ijk}$ and employing the divergence theorem as above yields:

$$\int_{\partial\tau^{(ijk)}} (\bar{\mathbf{a}}\nabla u) \cdot \mathbf{n} \, ds + \int_{\tau^{(ijk)}} bu \, d\mathbf{x} = \int_{\tau^{(ijk)}} f \, d\mathbf{x}.$$

The volume integrals are now approximated with the quadrature rule:

$$\int_{\tau^{(ijk)}} p \, d\mathbf{x} \approx p_{ijk} \left[\frac{(h_{i-1} + h_i)(h_{j-1} + h_j)(h_{k-1} + h_k)}{8}\right].$$

Assuming that the tensor $\bar{\mathbf{a}}$ is diagonal, $\bar{\mathbf{a}} = diag(a^{(11)}, a^{(22)}, a^{(33)})$, the surface integral then reduces to:

$$\int_{\partial\tau^{(ijk)}} \left[a^{(11)} u_x + a^{(22)} u_y + a^{(33)} u_z\right] \cdot \mathbf{n} \, ds.$$

This integral reduces further to six two-dimensional plane integrals on the six faces of the $\tau^{(ijk)}$, and are approximated with the analogous two-dimensional rule, after approximating the partial derivatives with centered differences. Introducing the notation $p_{i-1/2,j,k} = p(x_i - h_{i-1}/2, y_j, z_k)$, and $p_{i+1/2,j,k} = p(x_i + h_i/2, y_j, z_k)$, the resulting discrete equations can be written as:

$$\begin{aligned}
& a^{(11)}_{i-1/2,j,k} \left(\frac{u_{ijk} - u_{i-1,j,k}}{h_{i-1}}\right) \frac{(h_{j-1} + h_j)(h_{k-1} + h_k)}{4} \\
& + a^{(11)}_{i+1/2,j,k} \left(\frac{u_{ijk} - u_{i+1,j,k}}{h_i}\right) \frac{(h_{j-1} + h_j)(h_{k-1} + h_k)}{4} \\
& + a^{(22)}_{i,j-1/2,k} \left(\frac{u_{ijk} - u_{i,j-1,k}}{h_{j-1}}\right) \frac{(h_{i-1} + h_i)(h_{k-1} + h_k)}{4} \\
& + a^{(22)}_{i,j+1/2,k} \left(\frac{u_{ijk} - u_{i,j+1,k}}{h_j}\right) \frac{(h_{i-1} + h_i)(h_{k-1} + h_k)}{4} \\
& + a^{(33)}_{i,j,k-1/2} \left(\frac{u_{ijk} - u_{i,j,k-1}}{h_{k-1}}\right) \frac{(h_{i-1} + h_i)(h_{j-1} + h_j)}{4} \\
& + a^{(33)}_{i,j,k+1/2} \left(\frac{u_{ijk} - u_{i,j,k+1}}{h_k}\right) \frac{(h_{i-1} + h_i)(h_{j-1} + h_j)}{4}
\end{aligned}$$

$$+\left(b_{ijk}u_{ijk}\right)\frac{(h_{i-1}+h_i)(h_{j-1}+h_j)(h_{k-1}+h_k)}{8}=\left(f_{ijk}\right)\frac{(h_{i-1}+h_i)(h_{j-1}+h_j)(h_{k-1}+h_k)}{8}.$$

Collecting the coefficients of the unknown nodes u_{ijk} yields:

$$\begin{aligned}
&\Bigg[a^{(11)}_{i-1/2,j,k}\frac{(h_{j-1}+h_j)(h_{k-1}+h_k)}{4h_{i-1}}+a^{(11)}_{i+1/2,j,k}\frac{(h_{j-1}+h_j)(h_{k-1}+h_k)}{4h_i}\\
&+a^{(22)}_{i,j-1/2,k}\frac{(h_{i-1}+h_i)(h_{k-1}+h_k)}{4h_{j-1}}+a^{(22)}_{i,j+1/2,k}\frac{(h_{i-1}+h_i)(h_{k-1}+h_k)}{4h_j}\\
&+a^{(33)}_{i,j,k-1/2}\frac{(h_{i-1}+h_i)(h_{j-1}+h_j)}{4h_{k-1}}+a^{(33)}_{i,j,k+1/2}\frac{(h_{i-1}+h_i)(h_{j-1}+h_j)}{4h_k}\\
&+b_{ijk}\frac{(h_{i-1}+h_i)(h_{j-1}+h_j)(h_{k-1}+h_k)}{8}\Bigg]u_{ijk}\\
&+\left[-a^{(11)}_{i-1/2,j,k}\frac{(h_{j-1}+h_j)(h_{k-1}+h_k)}{4h_{i-1}}\right]u_{i-1,j,k}+\left[-a^{(11)}_{i+1/2,j,k}\frac{(h_{j-1}+h_j)(h_{k-1}+h_k)}{4h_i}\right]u_{i+1,j,k}\\
&+\left[-a^{(22)}_{i,j-1/2,k}\frac{(h_{i-1}+h_i)(h_{k-1}+h_k)}{4h_{j-1}}\right]u_{i,j-1,k}+\left[-a^{(22)}_{i,j+1/2,k}\frac{(h_{i-1}+h_i)(h_{k-1}+h_k)}{4h_j}\right]u_{i,j+1,k}\\
&+\left[-a^{(33)}_{i,j,k-1/2}\frac{(h_{i-1}+h_i)(h_{j-1}+h_j)}{4h_{k-1}}\right]u_{i,j,k-1}+\left[-a^{(33)}_{i,j,k+1/2}\frac{(h_{i-1}+h_i)(h_{j-1}+h_j)}{4h_k}\right]u_{i,j,k+1}\\
&=\left[\frac{(h_{i-1}+h_i)(h_{j-1}+h_j)(h_{k-1}+h_k)}{8}\right]f_{ijk}.
\end{aligned}$$

After employing the Dirichlet boundary conditions from (2.34), the above set of equations for the approximations u_{ijk} to the solution values $u(\mathbf{x}_{ijk})$ at the nodes $\mathbf{x}_{ijk}$ can be written together as the single matrix equation:

$$Au=f. \tag{2.37}$$

As a result of considering the non-uniform Cartesian mesh, if we order the unknowns u_{ijk} in the vector u in the natural ordering, the matrix A will have seven-banded block-tridiagonal form.

In the case of the nonlinear equation (2.35), if we assume that the nonlinear term is *autonomous* in the sense that $b(\mathbf{x},u)=b(u)$, then the derivation is as above, and the resulting system of *nonlinear* algebraic equations is:

$$\begin{aligned}
&\Bigg[a^{(11)}_{i-1/2,j,k}\frac{(h_{j-1}+h_j)(h_{k-1}+h_k)}{4h_{i-1}}+a^{(11)}_{i+1/2,j,k}\frac{(h_{j-1}+h_j)(h_{k-1}+h_k)}{4h_i}\\
&+a^{(22)}_{i,j-1/2,k}\frac{(h_{i-1}+h_i)(h_{k-1}+h_k)}{4h_{j-1}}+a^{(22)}_{i,j+1/2,k}\frac{(h_{i-1}+h_i)(h_{k-1}+h_k)}{4h_j}\\
&+a^{(33)}_{i,j,k-1/2}\frac{(h_{i-1}+h_i)(h_{j-1}+h_j)}{4h_{k-1}}+a^{(33)}_{i,j,k+1/2}\frac{(h_{i-1}+h_i)(h_{j-1}+h_j)}{4h_k}\Bigg]u_{ijk}\\
&+\left[\frac{(h_{i-1}+h_i)(h_{j-1}+h_j)(h_{k-1}+h_k)}{8}\right]b_{ijk}(u_{ijk})\\
&+\left[-a^{(11)}_{i-1/2,j,k}\frac{(h_{j-1}+h_j)(h_{k-1}+h_k)}{4h_{i-1}}\right]u_{i-1,j,k}+\left[-a^{(11)}_{i+1/2,j,k}\frac{(h_{j-1}+h_j)(h_{k-1}+h_k)}{4h_i}\right]u_{i+1,j,k}\\
&+\left[-a^{(22)}_{i,j-1/2,k}\frac{(h_{i-1}+h_i)(h_{k-1}+h_k)}{4h_{j-1}}\right]u_{i,j-1,k}+\left[-a^{(22)}_{i,j+1/2,k}\frac{(h_{i-1}+h_i)(h_{k-1}+h_k)}{4h_j}\right]u_{i,j+1,k}\\
&+\left[-a^{(33)}_{i,j,k-1/2}\frac{(h_{i-1}+h_i)(h_{j-1}+h_j)}{4h_{k-1}}\right]u_{i,j,k-1}+\left[-a^{(33)}_{i,j,k+1/2}\frac{(h_{i-1}+h_i)(h_{j-1}+h_j)}{4h_k}\right]u_{i,j,k+1}\\
&=\left[\frac{(h_{i-1}+h_i)(h_{j-1}+h_j)(h_{k-1}+h_k)}{8}\right]f_{ijk}.
\end{aligned}$$

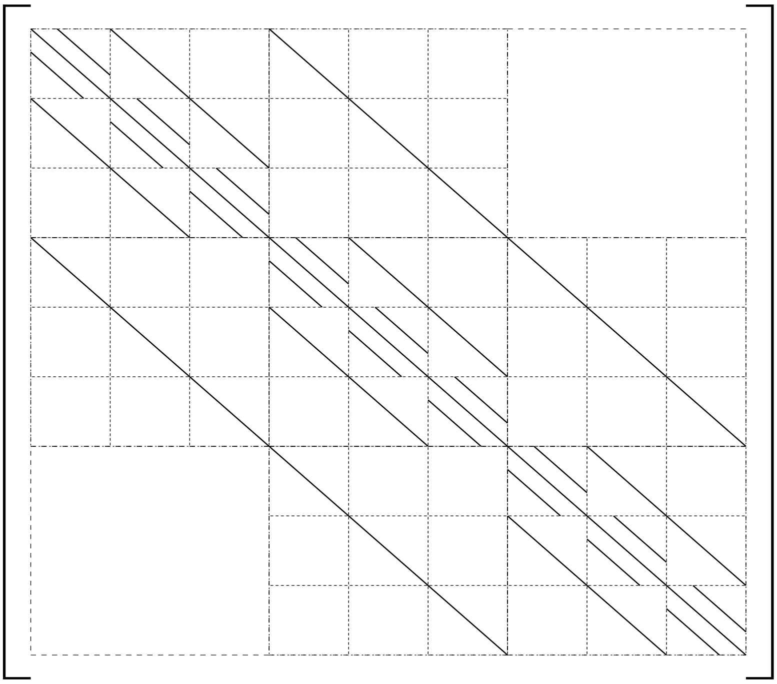


Figure 2.1: Banded matrix structure produced by the box-method.

If the nonlinear term is not autonomous but is at least *separable* in the sense that:

$$b(\mathbf{x}, u) = \eta(\mathbf{x})\beta(u),$$

then the derivation is the same, except that $\beta(u)$ will replace $b(u)$ above, and an averaging of $\eta(\mathbf{x})$ will multiply the nonlinear term $\beta(u)$.

In either case, the above set of equations for the approximations at the nodes $\mathbf{x}_{ijk}$ can be written together as the single *nonlinear* algebraic equation:

$$Au + N(u) = f, \tag{2.38}$$

where again the matrix A representing the linear part of (2.38) is seven-banded and block-tridiagonal.

Remark 2.6. The banded structure in the case of non-uniform Cartesian meshes allows for very efficient implementations of iterative methods for numerical solution of the discrete linear and nonlinear equations; the seven-banded form is depicted in Figure 2.6 for a 3 × 3 × 3 non-uniform Cartesian mesh.

2.3.3 Linear and nonlinear algebraic equations

We wish to establish some properties of the linear and nonlinear algebraic equations

$$Au = f \tag{2.39}$$

$$Au + N(u) = f \tag{2.40}$$

arising from box-method discretizations of the linearized and nonlinear Poisson-Boltzmann equations, respectively. First, we review some background material following [179] and [158].

Recall that an $n \times n$ matrix A is called *reducible* if it can be written as

$$PAP^T = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix},$$

where P is a permutation matrix. If there does not exist a matrix P such that the above holds, then the matrix A is called *irreducible.* The reducibility of a matrix can be determined by examining its *finite directed graph*, which is defined as follows. With the $n \times n$ matrix A is associated the n nodes $N_i, i = 1, \ldots n$. If the entry a_{ij} of A is nonzero, then the directed graph of A contains a directed path *from* N_i *to* N_j. If there exists a directed path *from* any node i *to* any node j, then the graph is called *strongly connected.* The following condition is useful.

Theorem 2.15 *An $n \times n$ matrix A is irreducible if and only if its directed graph is strongly connected.*

Proof. See Theorem 1.6 in [179]. □

Recall that the matrix A is *diagonally dominant* if

$$|a_{ii}| \geq \sum_{j=1, j\neq i}^{n} |a_{ij}|, \qquad i = 1, \ldots, n. \tag{2.41}$$

Further, the matrix A is *irreducibly diagonally dominant* if A is diagonally dominant and irreducible, and if inequality (2.41) holds in the strict sense for at least one i. The matrix A is *strictly diagonally dominant* if (2.41) holds for all i in the strict sense. The following theorem follows easily from the Gerschgorin Circle Theorem.

Theorem 2.16 *If the $n \times n$ matrix A is either strictly or irreducibly diagonally dominant with positive diagonal entries, then A is positive definite.*

Proof. See the Corollary to Theorem 1.8 in [179]. □

Recall that a *partial ordering* of the space $\mathbf{L}(\mathbb{R}^n, \mathbb{R}^n)$ of linear operators mapping $\mathbb{R}^n$ into itself may defined in the following way. Let $A \in \mathbf{L}(\mathbb{R}^n, \mathbb{R}^n)$ and $B \in \mathbf{L}(\mathbb{R}^n, \mathbb{R}^n)$. The ordering is defined as:

$$A \leq B \text{ if and only if } a_{ij} \leq b_{ij}, \qquad \forall i, j.$$

In particular, we can now write expressions such as $A > 0$, meaning that all the entries of the matrix A are positive.

There are two important classes of matrices which we now mention, which often arise from the discretization of partial differential equations. First, we note that if A is irreducibly diagonally dominant, with positive diagonal entries and non-positive off-diagonal entries, it can be shown that $A^{-1} > 0$. Similarly, if A is irreducible and symmetric, with nonpositive off-diagonal entries, then $A^{-1} > 0$ if and only if A is also positive definite. Now, A is called a *Stieltjes matrix* if A is symmetric positive definite with non-positive off-diagonal entries. Finally, if A is nonsingular, $A^{-1} > 0$, and if A has non-positive off-diagonal entries, then A is called an *M-matrix.* Clearly, we have that if the matrix A is a Stieltjes matrix then A is also an M-matrix; in fact, a Stieltjes matrix is simply a symmetric M-matrix. As a final remark, the following result will be useful.

Theorem 2.17 *If A is an M-matrix, and if D a non-negative diagonal matrix, then $A+D$ is an M-matrix, and $(A+D)^{-1} \leq A^{-1}$.*

Proof. See Theorem 2.4.11 in [158]. □

Consider now the nonlinear algebraic equation

$$F(u) = Au + N(u) = f, \tag{2.42}$$

where $A \in \mathbf{L}(\mathbb{R}^n, \mathbb{R}^n)$, and where $N(u) : \mathbb{R}^n \mapsto \mathbb{R}^n$ is a nonlinear operator. We will be interested in conditions which guarantee that F is a *homeomorphism* of $\mathbb{R}^n$ onto $\mathbb{R}^n$, meaning that the mapping F is one-to-one and onto, and that both F and F^{-1} are continuous. If this is established for the mapping F, then clearly for any function f, the equation (2.42) has a unique solution u which depends continuously on f.

We first introduce some notation and then state a useful theorem. A nonlinear operator $N = (b_1, \ldots, b_n)$ is called *diagonal* if the i-th component function b_i is a function only of u_i, where $u = (u_1, \ldots, u_n)$. The composite function F is then called *almost linear*, if A is linear and N is diagonal.

The following theorem (Theorem 5.3.10 in [158]) gives an important sufficient condition for a nonlinear operator to be a homeomorphism.

Theorem 2.18 *(The Hadamard Theorem) If $F : \mathbb{R}^n \mapsto \mathbb{R}^n$ is continuously differentiable on $\mathbb{R}^n$ and $\|F'(\mathbf{x})^{-1}\| \leq \gamma < +\infty$ $\forall \mathbf{x} \in \mathbb{R}^n$, then F is a homeomorphism of $\mathbb{R}^n$ onto $\mathbb{R}^n$.*

Proof. See Theorem 5.3.10 in [158]. □

2.3.4 Properties of the discrete linearized PBE operator

Various properties of systems generated with the box-method are discussed in Chapter 6 of [179] for one- and two-dimensional problems. In the three-dimensional case, it is immediately clear from the form of the discrete equations (2.37) that the resulting matrix is symmetric, with positive diagonal entries and non-positive off-diagonal entries. It is also immediate from its directed graph in the case of a natural ordering (which is simply the non-uniform Cartesian mesh itself) that the resulting matrix is irreducible. Finally, if at least one Dirichlet point is specified, then strict inequality will hold in the definition of diagonal dominance for at least one equation; hence, the matrix is irreducibly diagonally dominant. This gives the following three-dimensional version of Theorem 6.4 in [179].

Theorem 2.19 *If the matrix A represents the discrete equations (2.37) generated by a box-method discretization, and at least one Dirichlet point is specified on the boundary, then A is symmetric, irreducibly diagonally dominant, and has positive diagonal entries as well as non-positive off-diagonal entries. In addition, A is positive definite, and is therefore a Stieltjes matrix.*

Proof. See the discussion above. □

Corollary 2.20 *A box-method discretization of the linearized PBE, using non-uniform Cartesian hexahedral elements, yields a matrix A which is a symmetric M-matrix.*

Proof. The linearized Poisson-Boltzmann equation is a representative of the class (2.34) for which theorem (2.19) applies. □

Remark 2.7. In the case of uniform meshes, it is well-known that finite element, box, and finite difference (Taylor series) methods are similar or even equivalent. However, it should be noted that a Taylor series approach does not yield a symmetric matrix in the general case of non-uniform Cartesian meshes, as do box and finite element methods. Additionally, if natural boundary conditions (Neumann conditions) are present, a box or finite element discretization will always give rise to a symmetric matrix; this is not generally the case with a Taylor series approach.

2.3.5 Existence and uniqueness of discrete nonlinear PBE solutions

We now consider the general nonlinear equation

$$Au + N(u) = f, \tag{2.43}$$

arising from the box-method discretization of (2.35) as described earlier using hexahedral non-uniform Cartesian elements. The matrix A clearly has the properties described in Theorem 2.19 for the linear case. If we make some simple assumptions about the nonlinear operator $N(u)$, then we have the following result (Theorem 5.4.1 in [158]), the short proof of which we include for completeness.

Theorem 2.21 *If A is an M-matrix, if $N(u)$ is continuously differentiable, and if $N'(u)$ is diagonal and non-negative for all $u \in \mathbb{R}^n$, then the composite operator $F(u) = Au + N(u)$ is a homeomorphism of $\mathbb{R}^n$ onto $\mathbb{R}^n$.*

Proof. Since N is continuously differentiable, so is the composite operator F, and $F'(u) = A + N'(u)$. For each u, since the linear operator $N'(u)$ is a non-negative diagonal matrix, we have by Theorem 2.17 that the linear operator $F'(u)$ is an M-matrix, and that

$$0 < F'(u)^{-1} \leq A^{-1}, \quad \forall u \in \mathbb{R}^n.$$

Thus, $\|F'(u)^{-1}\|$ is bounded uniformly. The theorem then follows by the Hadamard Theorem. □

Corollary 2.22 *A box-method discretization of the nonlinear PBE, using non-uniform Cartesian hexahedral elements, yields a nonlinear algebraic operator F which is a homeomorphism from $\mathbb{R}^n$ onto $\mathbb{R}^n$, so that the discrete nonlinear Poisson-Boltzmann problem is well-posed.*

Proof. First, it is clear from the previous section that the matrix A which arises from the discretization is a symmetric M-matrix. Now consider the nonlinear term $N(u)$ in the case of the nonlinear Poisson-Boltzmann equation, which has the form:

$$\bar{\kappa}^2(\mathbf{x})\sinh(u(\mathbf{x})).$$

While the function $\bar{\kappa}(\mathbf{x})$ is only piecewise continuous over the domain, we have that for each nodal point $\mathbf{x}_{ijk}$, the coefficient $\bar{\kappa}(\mathbf{x}_{ijk})$ is constant. Therefore, at each mesh point the component function

$$b_{ijk}(u_{ijk}) = \bar{\kappa}^2(\mathbf{x}_{ijk})\sinh(u_{ijk})$$

is a continuously differentiable function of $u_{ijk} \in \mathbb{R}$, and the full function $N(u)$ is then continuously differentiable and diagonal. Further, since $\bar{\kappa}(\mathbf{x})$ is always non-negative, the derivative of the component function is continuous and non-negative since

$$b'_{ijk}(u) = \bar{\kappa}^2(\mathbf{x}_{ijk})\cosh(u) \geq 0, \qquad \forall u \in \mathbb{R}.$$

Therefore, $N'(u)$ is non-negative and diagonal for all $u \in \mathbb{R}^n$. The corollary then follows from Theorem 2.21. □

Remark 2.8. In the course of this research, it became important to establish Corollary 2.22, because initial numerical results appeared to yield non-physical situations. We note that Corollary 2.22 can also be shown using either a discrete convex analysis approach or discrete monotone operator theory, or using *M-function theory* (page 468 in [158]); an M-function is the nonlinear extension of an M-matrix.

2.4 Motivation for use of the box-method

In a recent paper [18], after recasting the box-method as a Petrov-Galerkin method with piecewise linear trial functions and piecewise constant test functions, it was shown that in the two-dimensional case, the box-method generates approximations comparable in accuracy to finite element methods with piecewise linear basis functions. Their study was motivated by the observation that both methods often generate exactly the same matrices in two-dimensions (this is always the case for the Laplacean operator on general two-dimensional meshes).

The three-dimensional case has been considered in detail by Kerkhoven [122]; it is demonstrated that box and finite element methods do not have the same equivalence properties in three dimensions that they have in two. Error estimates are derived by considering the box-method to be a Petrov-Galerkin discretization, using piecewise linear trial and test functions, of a closely associated "relaxed" problem. The relaxed method is then analyzed, using a modification of Strang's First Lemma (Theorem 4.1.1 in [36]), as a perturbation to a standard Galerkin discretization. The error induced by moving to the relaxed problem is then analyzed. As a result, error estimates comparable to Galerkin error estimates with piecewise linear trial and test functions are obtained for the box-method.

Note that some elliptic regularity is required for finite element error analysis, and this requirement is also imposed on the above box-method analysis, as it is based on the finite element case. Therefore, if one has no elliptic regularity estimates (such as in the case of discontinuous coefficients), the standard error analysis is not available. One can show, however, that the Galerkin method will still converge, although at an undetermined rate (Theorem 3.2.3 in [36]).

In case of nonlinear equations of the type we have considered in this chapter, the discretized nonlinear term produced by a finite element discretization is non-diagonal; this is also true in the linear case with a Helmholtz-like term in the operator, which produces a non-diagonal mass matrix. The box-method produces only diagonal terms from the Helmholtz-like or nonlinear terms; by *mass lumping* the elements of the mass matrix onto the diagonal this property is regained for finite element methods. However, some of the nice features of the finite element method are sacrificed, such as the *variational conditions* which we describe in Chapter 3 and Chapter 4.

We remark that our choice of the box-method for the linearized and nonlinear Poisson-Boltzmann equations was for both practical and theoretical considerations. From the practical side, there exists sophisticated biophysical modeling software, designed specifically for three-dimensional non-uniform Cartesian meshes. In particular, the UHBD and DELPHI programs discussed in Chapter 1 are specifically designed around an underlying non-uniform Cartesian mesh, and the non-uniform Cartesian formulation is required for efficient calculation of a posteriori quantities from the potential map produced by solving the Poisson-Boltzmann equation. These quantities include forces for molecular and Brownian dynamics simulations, as well as energies, reaction rates, and many other quantities which lead to a better understanding of these biological systems.

From the theoretical side is the following consideration: one of the most (perhaps the most) important and useful features of the finite element method with regard to multilevel methods is that discretizations on successively refined meshes automatically satisfy the so-called *variational conditions*:

$$A_{k-1} = I_k^{k-1} A_k I_{k-1}^k, \qquad I_k^{k-1} = (I_{k-1}^k)^T,$$

where A_k is the matrix or operator on the fine mesh, A_{k-1} is the matrix or operator on the coarse mesh, and the operator I_{k-1}^k is a prolongation operator which maps functions from the coarse space to the fine space. Unfortunately, if discontinuities in the coefficients of the problem are only resolvable on the finest mesh, meaning that they lie along element boundaries on the finest mesh but lie withing elements on coarser meshes, then the variational conditions will be violated. This is due to the fact that these conditions can be shown to hold theoretically (see Chapter 3) with exact evaluation of the integrals forming the components of the stiffness matrices, but quadrature error will be large due to the presence of discontinuous functions within elements; the only hope is to attempt to enforce these conditions algebraically on the coarse mesh problems, given the fine mesh operator and the prolongation operators. If we must enforce the variational conditions algebraically anyway, then we lose one of the most desirable features of the finite element method, from the perspective of multilevel theory. The importance of imposing the variational conditions in multilevel methods, either exactly or approximately, is easily demonstrated; we will give some simple examples in Chapter 6. We will discuss the variational conditions in more detail in the following chapters.

Of course one solution would be to begin with a coarse mesh for which all discontinuities lie along element boundaries, and then successively refine the mesh. Unfortunately, this is not always possible, especially in the case of the Poisson-Boltzmann equation. Large and complex molecules have very complex surfaces, which are only resolvable with a mesh size which already taxes the available computer resources. We mention, for example, the use of a discretization with more than seven million unknowns for a large molecule in [100], and in that particular case we would have preferred even more unknowns to more accurately resolve the surface and approximate the boundary conditions. An alternate approach is to use an integral equation formulation of the Poisson-Boltzmann equation and use surface tessellations of the molecule along with boundary finite element methods; we do not discuss these techniques here.

Note that perhaps the greatest obstacle one faces when employing multilevel methods for a “real world” application is the following: how does one define the coarse problems, when the real problem is really only well-defined with a very fine mesh? The two multilevel approaches we investigate for dealing with this difficulty here are: enforcement of the variational conditions exactly using algebraic means, an approach which is somewhat costly; and approximate enforcement of the variational conditions by various coefficient averaging strategies. We will discuss these techniques in more detail in the chapters to follow.

3. Linear Multilevel Methods

In this chapter, we summarize the main ideas behind linear multilevel methods, and present in detail the methods we study in the remainder of the work. We first discuss classical linear methods, their convergence properties, and conjugate gradient accelerations, motivating the discussion of linear multilevel methods. Multilevel methods are defined using a recursive operator formulation, setting the stage for a discussion in Chapter 5 of theoretical results which apply to these methods in certain situations. We also discuss modifications of the methods for equations with problem coefficients such as the linearized Poisson-Boltzmann equation. We finish the chapter with a short look at the complexity properties of various methods.

Our contributions here are as follows.

- We establish several simple but useful properties of the error propagator of an abstract linear method; these properties have been exploited in the literature, but their short proofs seem hard to find.
- We derive recursive and product forms of the multilevel error propagator explicitly in terms of interpolation and restriction operators; these are generalizations of some abstract recursions which have appeared in the finite element multilevel literature.
- We study two linear multilevel methods for interface problems, one based on coefficient averaging, and the other based on algebraic enforcement of variational or Galerkin conditions.
- We show how variational conditions can be enforced algebraically in an efficient way using a *stencil calculus* originally developed by R. Falgout, and we develop MAPLE and MATHEMATICA symbolic stencil calculators for producing the Galerkin matrix stencil entries in one, two, and three dimensions.
- We establish some relationships between coefficient averaging methods and algebraic Galerkin methods in certain cases in one and two dimensions which were not previously known.

3.1 Linear operator equations

In this section, we first review the theory of self-adjoint linear operators on a Hilbert space. The results required for the analysis of linear methods, as well as conjugate gradient methods, are summarized. We then develop carefully the theory of classical linear methods for operators equations. The conjugate gradient method is then considered, and the relationship between the convergence rate of linear methods as preconditioners and the convergence rate of the resulting preconditioned conjugate gradient method is explored in some detail.

As a motivation, consider that if either the box-method or the finite element method is used to discretize the second order linear elliptic partial differential equation $\mathcal{L}u = f$, a set of linear algebraic equations results, which we denote as:

$$A_k u_k = f_k. \tag{3.1}$$

The subscript k denotes the discretization level, with larger k corresponding to a more refined mesh, and with an associated mesh parameter h_k representing the diameter of the largest element or volume in the mesh Ω_k. For a self-adjoint strongly elliptic partial differential operator, the matrix A_k produced by the box or finite element method is SPD. In this work, we are interested in linear iterations for solving the matrix equation (3.1) which have the general form:

$$u_k^{n+1} = (I - B_k A_k) u_k^n + B_k f_k, \tag{3.2}$$

where B_k is an SPD matrix approximating A_k^{-1} in some sense. The classical stationary linear methods fit into this framework, as well as domain decomposition methods and multigrid methods.

3.1.1 Linear operators and spectral theory

In this section we compile some material on self-adjoint linear operators in finite-dimensional spaces which will be used throughout the work.

Let $\mathcal{H}$, $\mathcal{H}_1$, and $\mathcal{H}_2$ be a real finite-dimensional Hilbert spaces equipped with the inner-product $(\cdot,\cdot)$ inducing the norm $\|\cdot\| = (\cdot,\cdot)^{1/2}$. Since we are concerned only with finite-dimensional spaces, a Hilbert space $\mathcal{H}$ can be thought of as the Euclidean space $\mathbb{R}^n$; however, the preliminary material below and the algorithms we develop are phrased in terms of the unspecified space $\mathcal{H}$, so that the algorithms may be interpreted directly in terms of finite element spaces as well. This is necessary to set the stage for our discussion of multigrid and domain decomposition theory later in the work.

If the operator $A : \mathcal{H}_1 \mapsto \mathcal{H}_2$ is linear, we denote this as $A \in \mathbf{L}(\mathcal{H}_1, \mathcal{H}_2)$. The *adjoint* of a linear operator $A \in \mathbf{L}(\mathcal{H}, \mathcal{H})$ with respect to $(\cdot,\cdot)$ is the unique operator A^T satisfying $(Au, v) = (u, A^T v)\ ,\ \forall u, v \in \mathcal{H}$. An operator A is called *self-adjoint* or *symmetric* if $A = A^T$; a self-adjoint operator A is called *positive definite* or simply *positive*, if $(Au, u) > 0\ ,\ \forall u \in \mathcal{H},\ \ u \neq 0$.

If A is self-adjoint positive definite (SPD) with respect to $(\cdot,\cdot)$, then the bilinear form $A(u, v) = (Au, v)$ defines another inner-product on $\mathcal{H}$, which we sometimes denote as $(\cdot,\cdot)_A = A(\cdot,\cdot)$ to emphasize the fact that it is an inner-product rather than simply a bilinear form. The A-inner-product then induces the A-norm $\|\cdot\|_A = (\cdot,\cdot)_A^{1/2}$. For each inner-product the Cauchy-Schwarz inequality holds:

$$|(u, v)| \leq (u, u)^{1/2} (v, v)^{1/2}, \qquad |(u, v)_A| \leq (u, u)_A^{1/2} (v, v)_A^{1/2}, \qquad \forall u, v \in \mathcal{H}.$$

The adjoint of an operator M with respect to $(\cdot,\cdot)_A$, the A-*adjoint*, is the unique operator M^* satisfying $(Mu, v)_A = (u, M^* v)_A\ ,\ \forall u, v \in \mathcal{H}$. From this definition it follows that

$$M^* = A^{-1} M^T A\ . \tag{3.3}$$

An operator M is called A-*self-adjoint* if $M = M^*$, and A-*positive* if $(Mu, u)_A > 0\ ,\ \forall u \in \mathcal{H},\ \ u \neq 0$.

If $N \in \mathbf{L}(\mathcal{H}_1, \mathcal{H}_2)$, then the adjoint satisfies $N^T \in \mathbf{L}(\mathcal{H}_2, \mathcal{H}_1)$, and relates the inner-products in $\mathcal{H}_1$ and $\mathcal{H}_2$ as follows:

$$(Nu, v)_{\mathcal{H}_2} = (u, N^T v)_{\mathcal{H}_1}\ , \quad \forall u \in \mathcal{H}_1\ , \quad \forall v \in \mathcal{H}_2\ .$$

Since it is usually clear from the arguments which inner-product is involved, we shall drop the subscripts on inner-products (and norms) throughout the paper, except when necessary to avoid confusion.

For the operator M we denote the eigenvalues satisfying $Mu_i = \lambda_i u_i$ for eigenfunctions $u_i \neq 0$ as $\lambda_i(M)$. The spectral theory for self-adjoint linear operators states that the eigenvalues of the self-adjoint operator M are real and lie in the closed interval $[\lambda_{\min}(M), \lambda_{\max}(M)]$ defined by the Raleigh quotients:

$$\lambda_{\min}(M) = \min_{u \neq 0} \frac{(Mu, u)}{(u, u)}, \qquad \lambda_{\max}(M) = \max_{u \neq 0} \frac{(Mu, u)}{(u, u)}.$$

Similarly, if an operator M is A-self-adjoint, then the eigenvalues are real and lie in the interval defined by the Raleigh quotients generated by the A-inner-product:

$$\lambda_{\min}(M) = \min_{u \neq 0} \frac{(Mu, u)_A}{(u, u)_A}, \qquad \lambda_{\max}(M) = \max_{u \neq 0} \frac{(Mu, u)_A}{(u, u)_A}.$$

We denote the set of eigenvalues as the spectrum $\sigma(M)$ and the largest of these in absolute value as the spectral radius as $\rho(M) = \max(|\lambda_{\min}(M)|, |\lambda_{\max}(M)|)$. For SPD (or A-SPD) operators M, the eigenvalues of M are real and positive, and the powers M^s for real s are well-defined through the spectral decomposition;

see for example §79 and §82 in [90]. Finally, recall that a matrix representing the operator M with respect to any basis for $\mathcal{H}$ has the same eigenvalues as the operator M.

Linear operators on finite-dimensional spaces are always bounded, and these bounds define the operator norms induced by the norms $\|\cdot\|$ and $\|\cdot\|_A$:

$$\|M\| = \max_{u\neq 0} \frac{\|Mu\|}{\|u\|}, \qquad \|M\|_A = \max_{u\neq 0} \frac{\|Mu\|_A}{\|u\|_A}.$$

A well-known property is that if M is self-adjoint, then $\rho(M) = \|M\|$. This property can also be shown to hold for A-self-adjoint operators. The following lemma can be found in [5] (as Lemma 4.1), although the proof there is for A-normal matrices rather than A-self-adjoint operators.

Lemma 3.1 *If A is SPD and M is A-self-adjoint, then $\|M\|_A = \rho(M)$.*

Proof. We simply note that

$$\|M\|_A = \max_{u\neq 0} \frac{\|Mu\|_A}{\|u\|_A} = \max_{u\neq 0} \frac{(AMu, Mu)^{1/2}}{(Au, u)^{1/2}} = \max_{u\neq 0} \frac{(AM^*Mu, u)^{1/2}}{(Au, u)^{1/2}} = \lambda_{\max}^{1/2}(M^*M),$$

since M^*M is always A-self-adjoint. Since by assumption M itself is A-self-adjoint, we have that $M^* = M$, which yields: $\|M\|_A = \lambda_{\max}^{1/2}(M^*M) = \lambda_{\max}^{1/2}(M^2) = (\max_i[\lambda_i^2(M)])^{1/2} = \max[|\lambda_{\min}(M)|, |\lambda_{\max}(M)|] = \rho(M)$. □

3.1.2 The basic linear method

In this section, we introduce the basic linear method which we study and use in the remainder of the work.

Assume we are faced with the operator equation $Au = f$, where $A \in \mathbf{L}(\mathcal{H}, \mathcal{H})$ is SPD, and we desire the unique solution u. Given a *preconditioner* (approximate inverse) $B \approx A^{-1}$, consider the equivalent *preconditioned system* $BAu = Bf$. The operator B is chosen so that the simple linear iteration:

$$u^1 = u^0 - BAu^0 + Bf = (I - BA)u^0 + Bf,$$

which produces an improved approximation u^1 to the true solution u given an initial approximation u^0, has some desired convergence properties. This yields the following basic linear iterative method which we study in the remainder of this work:

Algorithm 3.1 *(Basic Linear Method for solving $Au = f$)*

$$u^{n+1} = u^n + B(f - Au^n) = (I - BA)u^n + Bf.$$

Subtracting the iteration equation from the identity $u = u - BAu + Bf$ yields the equation for the error $e^n = u - u^n$ at each iteration:

$$e^{n+1} = (I - BA)e^n = (I - BA)^2 e^{n-1} = \cdots = (I - BA)^{n+1} e^0. \tag{3.4}$$

The convergence of Algorithm 3.1 is determined completely by the spectral radius of the error propagation operator $E = I - BA$.

Theorem 3.2 *The condition $\rho(I - BA) < 1$ is necessary and sufficient for convergence of Algorithm 3.1.*

Proof. See for example Theorem 10.11 in [128] or Theorem 7.1.1 in [157]. □

Since $|\lambda|\|u\| = \|\lambda u\| = \|Mu\| \leq \|M\|\,\|u\|$ for any norm $\|\cdot\|$, it follows that $\rho(M) \leq \|M\|$ for all norms $\|\cdot\|$. Therefore, $\|I - BA\| < 1$ and $\|I - BA\|_A < 1$ are both sufficient conditions for convergence of Algorithm 3.1. In fact, it is the norm of the error propagation operator which will bound the reduction of the error at each iteration, which follows from (3.4):

$$\|e^{n+1}\|_A \leq \|I - BA\|_A \|e^n\|_A \leq \|I - BA\|_A^{n+1} \|e^0\|_A. \tag{3.5}$$

The spectral radius $\rho(E)$ of the error propagator E is called the *convergence factor* for Algorithm 3.1, whereas the norm of the error propagator $\|E\|$ is referred to as the *contraction number* (with respect to the particular choice of norm $\|\cdot\|$).

3.1.3 Properties of the error propagation operator

In this section, we establish some simple properties of the error propagation operator of an abstract linear method. We note that several of these properties are commonly used, especially in the multigrid literature, although the short proofs of the results seem difficult to locate. The particular framework we construct here for analyzing linear methods is based on the recent work of Xu [185], on the recent papers on multigrid and domain decomposition methods referenced therein, and on the text by Varga [179].

An alternate sufficient condition for convergence of the basic linear method is given in the following lemma, which is similar to *Stein's Theorem* (Theorem 7.1.8 in [157], or Theorem 6.1, page 80 in [187]).

Lemma 3.3 *If E^* is the A-adjoint of E, and $I - E^*E$ is A-positive, then it holds that $\rho(E) \leq \|E\|_A < 1$.*

Proof. By hypothesis, $(A(I - E^*E)u, u) > 0 \ \forall u \in \mathcal{H}$. This implies that $(AE^*Eu, u) < (Au, u) \ \forall u \in \mathcal{H}$, or $(AEu, Eu) < (Au, u) \ \forall u \in \mathcal{H}$. But this last inequality implies that

$$\rho(E) \leq \|E\|_A = \max_{u \neq 0} \frac{(AEu, Eu)}{(Au, u)} < 1.$$

□

We now state three very simple lemmas that we use repeatedly in the following sections.

Lemma 3.4 *If A is SPD, then BA is A-self-adjoint if and only if B is self-adjoint.*

Proof. Simply note that: $(ABAx, y) = (BAx, Ay) = (Ax, B^T Ay) \ \forall x, y \in \mathcal{H}$. The lemma follows since $BA = B^T A$ if and only if $B = B^T$. □

Lemma 3.5 *If A is SPD, then $I - BA$ is A-self-adjoint if and only if B is self-adjoint.*

Proof. Begin by noting that: $(A(I - BA)x, y) = (Ax, y) - (ABAx, y) = (Ax, y) - (Ax, (BA)^* y) = (Ax, (I - (BA)^*)y), \ \forall x, y \in \mathcal{H}$. Therefore, $E^* = I - (BA)^* = I - BA = E$ if and only if $BA = (BA)^*$. But by Lemma 3.4, this holds if and only if B is self-adjoint, so the result follows. □

Lemma 3.6 *If A and B are SPD, then BA is A-SPD.*

Proof. By Lemma 3.4, BA is A-self-adjoint. Also, we have that:

$$(ABAu, u) = (BAu, Au) = (B^{1/2}Au, B^{1/2}Au) > 0 \quad \forall u \neq 0, \ u \in \mathcal{H}.$$

Therefore, BA is also A-positive, and the result follows. □

We noted above that the property $\rho(M) = \|M\|$ holds in the case that M is self-adjoint with respect to the inner-product inducing the norm $\|\cdot\|$. If B is self-adjoint, the following theorem states that the resulting error propagator $E = I - BA$ has this property with respect to the A-norm.

Theorem 3.7 *If A is SPD and B is self-adjoint, then $\|I - BA\|_A = \rho(I - BA)$.*

Proof. By Lemma 3.5, $I - BA$ is A-self-adjoint, and by Lemma 3.1 the result follows. □

The following simple lemma, similar to Lemma 3.3, will be very useful later in the work.

Lemma 3.8 *If A and B are SPD, and $E = I - BA$ is A-non-negative, then it holds that $\rho(E) = \|E\|_A < 1$.*

Proof. By Lemma 3.5, E is A-self-adjoint, and by assumption E is A-non-negative, and so from §3.1.1 we see that E must have real non-negative eigenvalues. By hypothesis, $(A(I - BA)u, u) \geq 0 \ \forall u \in \mathcal{H}$, which implies that $(ABAu, u) \leq (Au, u) \ \forall u \in \mathcal{H}$. By Lemma 3.6, BA is A-SPD, and we have that

$$0 < (ABAu, u) \leq (Au, u) \quad \forall u \in \mathcal{H}, \quad u \neq 0,$$

which implies that $0 < \lambda_i(BA) \leq 1 \ \forall \lambda_i \in \sigma(BA)$. Thus, since $\lambda_i(E) = \lambda_i(I - BA) = 1 - \lambda_i(BA) \ \forall i$, we have that

$$\rho(E) = \max_i \lambda_i(E) = 1 - \min_i \lambda_i(BA) < 1.$$

Finally, by Theorem 3.7, we have $\|E\|_A = \rho(E) < 1$. □

The following simple lemma relates the contraction number bound to two simple inequalities; it is a standard result which follows directly from the spectral theory of self-adjoint linear operators.

Lemma 3.9 *If A is SPD and B is self-adjoint, and $E = I - BA$ is such that:*

$$-C_1(Au, u) \le (AEu, u) \le C_2(Au, u), \qquad \forall u \in \mathcal{H},$$

for $C_1 \ge 0$ and $C_2 \ge 0$, then $\rho(E) = \|E\|_A \le \max\{C_1, C_2\}$.

Proof. By Lemma 3.5, $E = I - BA$ is A-self-adjoint, and by the spectral theory outlined in §3.1.1, the inequality above simply bounds the most negative and most positive eigenvalues of E with $-C_1$ and C_2, respectively. The result then follows by Theorem 3.7. □

Corollary 3.10 *If A and B are SPD, then Lemma 3.9 holds for some $C_2 < 1$.*

Proof. By Lemma 3.6, BA is A-SPD, which implies that the eigenvalues of BA are real and positive by the discussion in §3.1.1. By Lemma 3.5, $E = I - BA$ is A-self-adjoint, and therefore has real eigenvalues. The eigenvalues of E and BA are related by $\lambda_i(E) = \lambda_i(I - BA) = 1 - \lambda_i(BA)\ \forall i$, and since $\lambda_i(BA) > 0\ \forall i$, we must have that $\lambda_i(E) < 1\ \forall i$. Since C_2 in Lemma 3.9 bounds the largest positive eigenvalue of E, we have that $C_2 < 1$. □

We now define the A-*condition number* of an invertible operator M by extending the standard notion to the A-inner-product:

$$\kappa_A(M) = \|M\|_A \|M^{-1}\|_A.$$

In the next section we show (Lemma 3.12) that if M is an A-self-adjoint operator, then in fact the following simpler expression holds:

$$\kappa_A(M) = \frac{\lambda_{\max}(M)}{\lambda_{\min}(M)}.$$

The generalized condition number κ_A is employed in the following lemma, which states that there is an optimal relaxation parameter for a basic linear method, and gives the best possible convergence estimate for the method employing the optimal parameter. This lemma has appeared many times in the literature in one form or another; cf. [159].

Lemma 3.11 *If A and B are SPD, then*

$$\rho(I - \alpha BA) = \|I - \alpha BA\|_A < 1.$$

if and only if $\alpha \in (0, 2/\rho(BA))$. Convergence is optimal when $\alpha = 2/[\lambda_{\min}(BA) + \lambda_{\max}(BA)]$, giving

$$\rho(I - \alpha BA) = \|I - \alpha BA\|_A = 1 - \frac{2}{1 + \kappa_A(BA)} < 1.$$

Proof. Note that $\rho(I - \alpha BA) = \max_\lambda |1 - \alpha\lambda(BA)|$, so that $\rho(I - \alpha BA) < 1$ if and only if $\alpha \in (0, 2/\rho(BA))$, proving the first part. Taking $\alpha = 2/[\lambda_{\min}(BA) + \lambda_{\max}(BA)]$, we have

$$\rho(I - \alpha BA) = \max_\lambda |1 - \alpha\lambda(BA)| = \max_\lambda (1 - \alpha\lambda(BA))$$

$$= \max_\lambda \left(1 - \frac{2\lambda(BA)}{\lambda_{\min}(BA) + \lambda_{\max}(BA)}\right) = 1 - \frac{2\lambda_{\min}(BA)}{\lambda_{\min}(BA) + \lambda_{\max}(BA)} = 1 - \frac{2}{1 + \frac{\lambda_{\max}(BA)}{\lambda_{\min}(BA)}}.$$

Since BA is A-self-adjoint, by Lemma 3.12 we have that $\kappa_A(BA) = \lambda_{\max}(BA)/\lambda_{\min}(BA)$, so that if $\alpha = 2/[\lambda_{\min}(BA) + \lambda_{\max}(BA)]$, then

$$\rho(I - \alpha BA) = \|I - \alpha BA\|_A = 1 - \frac{2}{1 + \kappa_A(BA)}.$$

To show this is optimal, we must solve $\min_\alpha[\max_\lambda |1-\alpha\lambda|]$, where $\alpha \in (0, 2/\lambda_{\max})$. Note that each α defines a polynomial of degree zero in λ, namely $P_o(\lambda) = \alpha$. Therefore, we can rephrase the problem as

$$P_1^{\text{opt}}(\lambda) = \min_{P_o} \left[\max_\lambda |1 - \lambda P_o(\lambda)| \right].$$

It is well-known that the scaled and shifted Chebyshev polynomials give the solution to this “mini-max” problem:

$$P_1^{\text{opt}}(\lambda) = 1 - \lambda P_o^{\text{opt}} = \frac{T_1\left(\frac{\lambda_{\max}+\lambda_{\min}-2\lambda}{\lambda_{\max}-\lambda_{\min}}\right)}{T_1\left(\frac{\lambda_{\max}+\lambda_{\min}}{\lambda_{\max}-\lambda_{\min}}\right)}.$$

Since $T_1(x) = x$, we have simply that

$$P_1^{\text{opt}}(\lambda) = \frac{\frac{\lambda_{\max}+\lambda_{\min}-2\lambda}{\lambda_{\max}-\lambda_{\min}}}{\frac{\lambda_{\max}+\lambda_{\min}}{\lambda_{\max}-\lambda_{\min}}} = 1 - \lambda\left(\frac{2}{\lambda_{\min}+\lambda_{\max}}\right),$$

showing that in fact $\alpha_{\text{opt}} = 2/[\lambda_{\min} + \lambda_{\max}]$. □

Remark 3.1. Theorem 3.7 will be exploited later since $\rho(E)$ is usually much easier to compute numerically than $\|E\|_A$, and since it is the energy norm $\|E\|_A$ of the error propagator E which is typically bounded in various convergence theories for iterative processes.

Note that if we wish to reduce the initial error $\|e^0\|_A$ by the factor ϵ, then equation (3.5) implies that this will be guaranteed if

$$\|E\|_A^{n+1} \leq \epsilon.$$

Taking natural logarithms of both sides and solving for n, we see that the maximum number of iterations required to reach the desired tolerance, as a function of the contraction number, is given by:

$$n \leq \frac{|\ln \epsilon|}{|\ln \|E\|_A|}. \tag{3.6}$$

If the bound on the norm is of the form in Lemma 3.11, then to achieve a tolerance of ϵ after n iterations will require:

$$n \leq \frac{|\ln \epsilon|}{\left|\ln\left(1 - \frac{2}{1+\kappa_A(BA)}\right)\right|} = \frac{|\ln \epsilon|}{\left|\ln\left(\frac{\kappa_A(BA)-1}{\kappa_A(BA)+1}\right)\right|}. \tag{3.7}$$

Using the approximation:

$$\ln\left(\frac{a-1}{a+1}\right) = \ln\left(\frac{1+(-1/a)}{1-(-1/a)}\right) = 2\left[\left(\frac{-1}{a}\right) + \frac{1}{3}\left(\frac{-1}{a}\right)^3 + \frac{1}{5}\left(\frac{-1}{a}\right)^5 + \cdots\right] < \frac{-2}{a},$$

we have that $|\ln[(\kappa_A(BA)-1)/(\kappa_A(BA)+1)]| > 2/\kappa_A(BA)$, so that:

$$n \leq \frac{1}{2}\kappa_A(BA)|\ln \epsilon| + 1.$$

We then have that the maximum number of iterations required to reach an error on the order of the tolerance ϵ is:

$$n = O\left(\kappa_A(BA)|\ln \epsilon|\right).$$

If a single iteration of the method costs $O(N)$ arithmetic operators, then the overall complexity to solve the problem is $O(|\ln \|E\|_A|^{-1} N |\ln \epsilon|)$, or $O(\kappa_A(BA) N |\ln \epsilon|)$. If the quantity $\|E\|_A$ can be bounded less than one independent of N, or if $\kappa_A(BA)$ can be bounded independent of N, then the complexity is near optimal $O(N|\ln \epsilon|)$.

Note that if E is A-self-adjoint, then we can replace $\|E\|_A$ by $\rho(E)$ in the above discussion. Even when this is not the case, $\rho(E)$ is often used above in place of $\|E\|_A$ to obtain an estimate, and the quantity

$R_\infty(E) = -\ln \rho(E)$ is referred to as the *asymptotic convergence rate* (see page 67 of [179], or page 88 of [187]).

In [179], the *average convergence rate of m iterations* is defined as $R(E^m) = -\ln(\|E^m\|^{1/m})$, the meaning of which is intuitively clear from equation (3.5). As noted on page 95 in [179], since $\rho(E) = \lim_{m\to\infty} \|E^m\|^{1/m}$ for all bounded linear operators E and norms $\|\cdot\|$ (Theorem 7.5-5 in [129]), it follows that $\lim_{m\to\infty} R(E^m) = R_\infty(E)$.

While $R_\infty(E)$ is considered the standard measure of convergence of linear iterations (it is called the "convergence rate" in [187], page 88), this is really an asymptotic measure, and the convergence behavior for the early iterations may be better monitored by using the norm of the propagator E directly in (3.6); an example is given on page 67 of [179] for which $R_\infty(E)$ gives a poor estimate of the number of iterations required.

3.1.4 Conjugate gradient acceleration of linear methods

Consider now the linear equation $Au = f$ in the space $\mathcal{H}$. The conjugate gradient method was developed by Hestenes and Stiefel [93] for linear systems with symmetric positive definite operators A. It is common to *precondition* the linear system by the SPD *preconditioning operator* $B \approx A^{-1}$, in which case the generalized or preconditioned conjugate gradient method [38] results. Our purpose in this section is to briefly examine the algorithm, its contraction properties, and establish some simple relationships between the contraction number of a basic linear preconditioner and that of the resulting preconditioned conjugate gradient algorithm. These relationships are commonly used, but some of the short proofs seem unavailable.

In [7], a general class of conjugate gradient methods obeying three-term recursions is studied, and it is shown that each instance of the class can be characterized by three operators: an inner product operator X, a preconditioning operator Y, and the system operator Z. As such, these methods are denoted as CG(X,Y,Z). We are interested in the special case that $X = A$, $Y = B$, and $Z = A$, when both B and A are SPD. Choosing the *Omin* [7] algorithm to implement the method CG(A,B,A), the *preconditioned conjugate gradient method* results:

Algorithm 3.2 *(Preconditioned Conjugate Gradient Algorithm)*

```
Let $u^0 \in \mathcal{H}$ be given.
$r^0 = f - Au^0, \quad s^0 = Br^0, \quad p^0 = s^0.$
Do $i = 0, 1, \ldots$ until convergence:
    $\alpha_i = (r^i, s^i)/(Ap^i, p^i)$
    $u^{i+1} = u^i + \alpha_i p^i$
    $r^{i+1} = r^i - \alpha_i Ap^i$
    $s^{i+1} = Br^{i+1}$
    $\beta_{i+1} = (r^{i+1}, s^{i+1})/(r^i, s^i)$
    $p^{i+1} = s^{i+1} + \beta_{i+1} p^i$
End do.
```

If the dimension of $\mathcal{H}$ is n, then the algorithm can be shown to converge in n steps since the preconditioned operator BA is A-SPD [7]. Note that if $B = I$, then this algorithm is exactly the Hestenes and Stiefel algorithm.

Since we wish to understand a little about the convergence properties of the conjugate gradient method, and how these will be effected by a linear method representing the preconditioner B, we will briefly review a well-known conjugate gradient contraction bound. To begin, it is not difficult to see that the error at each iteration of Algorithm 3.2 can be written as a polynomial in BA times the initial error:

$$e^{i+1} = [I - BAp_i(BA)]e^0,$$

where $p_i \in \mathcal{P}_i$, the space of polynomials of degree i. At each step the energy norm of the error $\|e^{i+1}\|_A = \|u - u^{i+1}\|_A$ is minimized over the Krylov subspace:

$$V_{i+1}(BA, Br^0) = \text{span}\ \{Br^0, (BA)Br^0, (BA)^2Br^0, \ldots, (BA)^iBr^0\}.$$

Therefore, it must hold that:

$$\|e^{i+1}\|_A = \min_{p_i \in \mathcal{P}_i} \|[I - BAp_i(BA)]e^0\|_A.$$

Since BA is A-SPD, the eigenvalues $\lambda_j \in \sigma(BA)$ of BA are real and positive, and the eigenvectors v_j of BA are A-orthonormal. By expanding $e^0 = \sum_{j=1}^n \alpha_j v_j$, we have:

$$\|[I - BAp_i(BA)]e^0\|_A^2 = (A[I - BAp_i(BA)]e^0, [I - BAp_i(BA)]e^0)$$

$$= (A[I - BAp_i(BA)](\sum_{j=1}^n \alpha_j v_j), [I - BAp_i(BA)](\sum_{j=1}^n \alpha_j v_j))$$

$$= (\sum_{j=1}^n [1 - \lambda_j p_i(\lambda_j)]\alpha_j \lambda_j v_j, \sum_{j=1}^n [1 - \lambda_j p_i(\lambda_j)]\alpha_j v_j) = \sum_{j=1}^n [1 - \lambda_j p_i(\lambda_j)]^2 \alpha_j^2 \lambda_j$$

$$\leq \max_{\lambda_j \in \sigma(BA)} [1 - \lambda_j p_i(\lambda_j)]^2 \sum_{j=1}^n \alpha_j^2 \lambda_j = \max_{\lambda_j \in \sigma(BA)} [1 - \lambda_j p_i(\lambda_j)]^2 \sum_{j=1}^n (A\alpha_j v_j, \alpha_j v_j)$$

$$= \max_{\lambda_j \in \sigma(BA)} [1 - \lambda_j p_i(\lambda_j)]^2 (A \sum_{j=1}^n \alpha_j v_j, \sum_{j=1}^n \alpha_j v_j) = \max_{\lambda_j \in \sigma(BA)} [1 - \lambda_j p_i(\lambda_j)]^2 \|e^0\|_A^2.$$

Thus, we have that

$$\|e^{i+1}\|_A \leq \left(\min_{p_i \in \mathcal{P}_i} \left[\max_{\lambda_j \in \sigma(BA)} |1 - \lambda_j p_i(\lambda_j)| \right] \right) \|e^0\|_A.$$

The scaled and shifted Chebyshev polynomials $T_{i+1}(\lambda)$, extended outside the interval $[-1, 1]$ as in the Appendix A of [9], yield a solution to this *mini-max* problem. Using some simple well-known relationships valid for $T_{i+1}(\cdot)$, the following contraction bound is easily derived:

$$\|e^{i+1}\|_A \leq 2 \left(\frac{\sqrt{\frac{\lambda_{\max}(BA)}{\lambda_{\min}(BA)}} - 1}{\sqrt{\frac{\lambda_{\max}(BA)}{\lambda_{\min}(BA)}} + 1} \right)^{i+1} \|e^0\|_A = 2\ \delta_{\mathrm{cg}}^{i+1}\ \|e^0\|_A. \tag{3.8}$$

The ratio of the extreme eigenvalues of BA appearing in the bound is often mistakenly called the (spectral) condition number $\kappa(BA)$; in fact, since BA is not self-adjoint (it is A-self-adjoint), this ratio is not in general equal to the condition number (this point is discussed in great detail in [5]). However, the ratio does yield a condition number in a different norm. The following lemma is a special case of Corollary 4.2 in [5].

Lemma 3.12 *If A and B are SPD, then*

$$\kappa_A(BA) = \|BA\|_A \|(BA)^{-1}\|_A = \frac{\lambda_{\max}(BA)}{\lambda_{\min}(BA)}. \tag{3.9}$$

Proof. For any A-SPD M, it is easy to show that M^{-1} is also A-SPD, so that from §3.1.1 both M and M^{-1} have real, positive eigenvalues. From Lemma 3.1 it then holds that:

$$\|M^{-1}\|_A = \rho(M^{-1}) = \max_{u \neq 0} \frac{(AM^{-1}u, u)}{(Au, u)} = \max_{u \neq 0} \frac{(AM^{-1/2}u, M^{-1/2}u)}{(AMM^{-1/2}u, M^{-1/2}u)}$$

$$= \max_{v \neq 0} \frac{(Av, v)}{(AMv, v)} = \left[\min_{v \neq 0} \frac{(AMv, v)}{(Av, v)} \right]^{-1} = \lambda_{\min}(M)^{-1}.$$

By Lemma 3.6, BA is A-SPD, which together with Lemma 3.1 implies that $\|BA\|_A = \rho(BA) = \lambda_{\max}(BA)$. From above we have that $\|(BA)^{-1}\|_A = \lambda_{\min}(BA)^{-1}$, implying that the A-condition number is given as the ratio of the extreme eigenvalues of BA as in equation (3.9). □

More generally, it can be shown that if the operator D is C-normal for some SPD inner-product operator C, then the generalized condition number given by $\kappa_C(D) = \|D\|_C \|D^{-1}\|_C$ is equal to the ratio of the extreme eigenvalues of the operator D. A proof of this fact is given in Corollary 4.2 of [5], along with a detailed discussion of this and other relationships for more general conjugate gradient methods. The conjugate gradient contraction number $\delta_{\rm cg}$ can now be written as:

$$\delta_{\rm cg} = \frac{\sqrt{\kappa_A(BA)} - 1}{\sqrt{\kappa_A(BA)} + 1} = 1 - \frac{2}{1 + \sqrt{\kappa_A(BA)}}.$$

The following lemma is used in the analysis of multigrid and other linear preconditioners (it appears for example as Proposition 5.1 in [184]) to bound the condition number of the operator BA in terms of the extreme eigenvalues of the linear preconditioner error propagator $E = I - BA$. We have given our own short proof of this result for completeness.

Lemma 3.13 *If A and B are SPD, and $E = I - BA$ is such that:*

$$-C_1(Au, u) \le (AEu, u) \le C_2(Au, u), \qquad \forall u \in \mathcal{H},$$

for $C_1 \ge 0$ and $C_2 \ge 0$, then the above must hold with $C_2 < 1$, and it follows that:

$$\kappa_A(BA) \le \frac{1 + C_1}{1 - C_2}.$$

Proof. First, since A and B are SPD, by Corollary 3.10 we have that $C_2 < 1$. Since $(AEu, u) = (A(I - BA)u, u) = (Au, u) - (ABAu, u)$, $\forall u \in \mathcal{H}$, it is immediately clear that

$$-C_1(Au, u) - (Au, u) \le -(ABAu, u) \le C_2(Au, u) - (Au, u), \qquad \forall u \in \mathcal{H}.$$

After multiplying by -1, we have

$$(1 - C_2)(Au, u) \le (ABAu, u) \le (1 + C_1)(Au, u), \qquad \forall u \in \mathcal{H}.$$

By Lemma 3.6, BA is A-SPD, and it follows from §3.1.1 that the eigenvalues of BA are real and positive, and lie in the interval defined by the Raleigh quotients of §3.1.1, generated by the A-inner-product. From above, we see that the interval is given by $[(1 - C_2), (1 + C_1)]$, and by Lemma 3.12 the result follows. □

The next corollary appears for example as Theorem 5.1 in [184].

Corollary 3.14 *If A and B are SPD, and BA is such that:*

$$C_1(Au, u) \le (ABAu, u) \le C_2(Au, u), \qquad \forall u \in \mathcal{H},$$

for $C_1 \ge 0$ and $C_2 \ge 0$, then the above must hold with $C_1 > 0$, and it follows that:

$$\kappa_A(BA) \le \frac{C_2}{C_1}.$$

Proof. This follows easily from the argument used in the proof of Lemma 3.13. □

The following corollary, which relates the contraction property of a linear method to the condition number of the operator BA, appears without proof as Proposition 2.2 in [185].

Corollary 3.15 *If A and B are SPD, and $\|I - BA\|_A \le \delta < 1$, then*

$$\kappa_A(BA) \le \frac{1 + \delta}{1 - \delta}. \tag{3.10}$$

Proof. This follows immediately from Lemma 3.13 with $\delta = \max\{C_1, C_2\}$. □

We comment briefly on an interesting implication of Lemma 3.13, which was apparently first noticed in [184]. It seems that even if a linear method is not convergent, for example if $C_1 > 1$ so that $\rho(E) > 1$, it may still be a good preconditioner. For example, if A and B are SPD, then by Corollary 3.10 we always have $C_2 < 1$. If it is the case that $C_2 << 1$, and if $C_1 > 1$ does not become too large, then $\kappa_A(BA)$ will be small and the conjugate gradient method will converge rapidly. A multigrid method will often diverge when applied to a problem with discontinuous coefficients unless special care is taken. Simply using conjugate gradient acceleration in conjunction with the multigrid method often yields a convergent (even rapidly convergent) method without employing any of the special techniques that have been developed for these problems; Lemma 3.13 may be the explanation for this behavior.

The following result from [185] connects the contraction number of the linear method used as the preconditioner to the contraction number of the resulting conjugate gradient method, and it shows that the conjugate gradient method always accelerates a linear method.

Theorem 3.16 *If A and B are SPD, and $\|I - BA\|_A \leq \delta < 1$, then $\delta_{\rm cg} < \delta$.*

Proof. An abbreviated proof appears in [185]; we fill in the details here for completeness. Assume that the given linear method has contraction number bounded as $\|I - BA\|_A < \delta$. Now, since the function:

$$\frac{\sqrt{\kappa_A(BA)} - 1}{\sqrt{\kappa_A(BA)} + 1}$$

is an increasing function of $\kappa_A(BA)$, we can use the result of Lemma 3.13, namely $\kappa_A(BA) \leq (1+\delta)/(1-\delta)$, to bound the contraction rate of preconditioned conjugate gradient method as follows:

$$\delta_{\rm cg} \leq \left(\frac{\sqrt{\kappa_A(BA)} - 1}{\sqrt{\kappa_A(BA)} + 1} \right) \leq \frac{\sqrt{\frac{1+\delta}{1-\delta}} - 1}{\sqrt{\frac{1+\delta}{1-\delta}} + 1} \cdot \frac{\sqrt{\frac{1+\delta}{1-\delta}} - 1}{\sqrt{\frac{1+\delta}{1-\delta}} - 1} = \frac{\frac{1+\delta}{1-\delta} - 2\sqrt{\frac{1+\delta}{1-\delta}} + 1}{\frac{1+\delta}{1-\delta} - 1} = \frac{1 - \sqrt{1-\delta^2}}{\delta}.$$

Note that this last term can be rewritten as:

$$\delta_{\rm cg} \leq \frac{1 - \sqrt{1-\delta^2}}{\delta} = \delta \left(\frac{1}{\delta^2} [1 - \sqrt{1-\delta^2}] \right).$$

Now, since $0 < \delta < 1$, clearly $1 - \delta^2 < 1$, so that $1 - \delta^2 > (1 - \delta^2)^2$. Thus, $\sqrt{1-\delta^2} > 1 - \delta^2$, or $-\sqrt{1-\delta^2} < \delta^2 - 1$, or finally $1 - \sqrt{1-\delta^2} < \delta^2$. Therefore, $(1/\delta^2) \left[1 - \sqrt{1-\delta^2}\right] < 1$, or

$$\delta_{\rm cg} \leq \delta \left(\frac{1}{\delta^2} \left[1 - \sqrt{1-\delta^2}\right] \right) < \delta.$$

A more direct proof follows by recalling from Lemma 3.11 that the *best* possible contraction of the linear method, when provided with an optimal parameter, is given by:

$$\delta_{\rm opt} = 1 - \frac{2}{1 + \kappa_A(BA)},$$

whereas the conjugate gradient contraction is

$$\delta_{\rm cg} = 1 - \frac{2}{1 + \sqrt{\kappa_A(BA)}}.$$

Assuming $B \neq A^{-1}$, we always have $\kappa_A(BA) > 1$, so we must have that $\delta_{\rm cg} < \delta_{\rm opt} \leq \delta$. □

Remark 3.2. This result implies that it always pays in terms of an improved contraction number to use the conjugate gradient method to accelerate a linear method; the question remains of course whether the additional computational labor involved will be amortized by the improvement. This is not clear from the above analysis, and seems to be problem-dependent in practice.

Remark 3.3. Note that if a given linear method requires a parameter α as in Lemma 3.11 in order to be competitive, one can simply use the conjugate gradient method as an accelerator for the method without a parameter, avoiding the possibly costly estimation of a good parameter α. Theorem 3.16 guarantees that the resulting method will have superior contraction properties, without requiring the parameter estimation. This is exactly why additive multigrid and domain decomposition methods (which we discuss in more detail later) are used almost exclusively as preconditioners for conjugate gradient methods; in contrast to the multiplicative variants, which can be used effectively without a parameter, the additive variants always require a good parameter α to be effective, unless used as preconditioners.

To finish this section, we remark briefly on the complexity of Algorithm 3.2. If a tolerance of ϵ is required, then the computational cost to reduce the energy norm of the error below the tolerance can be determined from the expression above for δ_{cg} and from equation (3.8). To achieve a tolerance of ϵ after n iterations will require:

$$2\ \delta_{\text{cg}}^{n+1} = 2\ \left(\frac{\sqrt{\kappa_A(BA)}-1}{\sqrt{\kappa_A(BA)}+1} \right)^{n+1} < \epsilon.$$

Dividing by 2 and taking natural logarithms yields:

$$n \le \frac{\left|\ln \frac{\epsilon}{2}\right|}{\left|\ln\left(\frac{\sqrt{\kappa_A(BA)}-1}{\sqrt{\kappa_A(BA)}+1}\right)\right|}.$$

Using the approximation:

$$\ln\left(\frac{a-1}{a+1}\right) = \ln\left(\frac{1+(-1/a)}{1-(-1/a)}\right) = 2\left[\left(\frac{-1}{a}\right) + \frac{1}{3}\left(\frac{-1}{a}\right)^3 + \frac{1}{5}\left(\frac{-1}{a}\right)^5 + \cdots\right] < \frac{-2}{a},$$

we have that $|\ln[(\kappa_A^{1/2}(BA)-1)/(\kappa_A^{1/2}(BA)+1)]| > 2/\kappa_A^{1/2}(BA)$, so that:

$$n \le \frac{1}{2}\kappa_A^{1/2}(BA)\left|\ln\frac{\epsilon}{2}\right| + 1.$$

We then have that the maximum number of iterations required to reach an error on the order of the tolerance ϵ is:

$$n = O\left(\kappa_A^{1/2}(BA)\left|\ln\frac{\epsilon}{2}\right|\right).$$

If the cost of each iteration is $O(N)$, which will hold in the case of the sparse matrices generated by standard discretizations of elliptic partial differential equations, then the overall complexity to solve the problem is $O(\kappa_A^{1/2}(BA)N|\ln[\epsilon/2]|)$. If the preconditioner B is such that $\kappa_A^{1/2}(BA)$ can be bounded independently of the problem size N, then the complexity becomes (near) optimal order $O(N|\ln[\epsilon/2]|)$.

We make some final remarks regarding the idea of *spectral equivalence.*

Definition 3.1 *The SPD operators $B \in \mathbf{L}(\mathcal{H},\mathcal{H})$ and $A \in \mathbf{L}(\mathcal{H},\mathcal{H})$ are called spectrally equivalent if there exists constants $C_1 > 0$ and $C_2 > 0$ such that:*

$$C_1(Au,u) \le (Bu,u) \le C_2(Au,u), \qquad \forall u \in \mathcal{H}.$$

In other words, B defines an inner-product which induces a norm equivalent to the norm induced by the A-inner-product. If a given preconditioner B is spectrally equivalent to A^{-1}, then the condition number of the preconditioned operator BA is uniformly bounded.

Lemma 3.17 *If the SPD operators B and A^{-1} are spectrally equivalent, then:*

$$\kappa_A(BA) \le \frac{C_2}{C_1}.$$

Proof. By hypothesis, we have that $C_1(A^{-1}u,u) \le (Bu,u) \le C_2(A^{-1}u,u)$, $\forall u \in \mathcal{H}$. But this can be written as: $C_1(A^{-1/2}u, A^{-1/2}u) \le (A^{1/2}BA^{1/2}A^{-1/2}u, A^{-1/2}u) \le C_2(A^{-1/2}u, A^{-1/2}u)$, or:

$$C_1(\tilde{u},\tilde{u}) \le (A^{1/2}BA^{1/2}\tilde{u},\tilde{u}) \le C_2(\tilde{u},\tilde{u}), \qquad \forall \tilde{u} \in \mathcal{H}.$$

Now, since $BA = A^{-1/2}(A^{1/2}BA^{1/2})A^{1/2}$, we have that BA is similar to the SPD operator $A^{1/2}BA^{1/2}$. Therefore, the above inequality bounds the extreme eigenvalues of BA, and as a result the lemma follows by Lemma 3.12. □

Remark 3.4. Of course, since all norms on finite-dimensional spaces are equivalent (which follows from the fact that all linear operators on finite-dimensional spaces are bounded), the idea of spectral equivalence is only important in the case of infinite-dimensional spaces, or when one considers how the equivalence constants behave as one increases the sizes of the spaces. This is exactly the issue in multigrid and domain decomposition theory: as one decreases the mesh size (increases the size of the spaces involved), one would like the quantity $\kappa_A(BA)$ to remain nicely bounded (in other words, one would like the equivalence constants to remain constant or grow only slowly). A discussion of these ideas appears in [159].

3.1.5 Discrete linear elliptic equations

Consider the second order linear elliptic partial differential equation:

$$-\nabla \cdot (\bar{\mathbf{a}}\nabla u) + bu = f \text{ in } \Omega \subset \mathbb{R}^d, \qquad u = g \text{ on } \Gamma, \tag{3.11}$$

where the coefficients are as described in Chapter 2, §2.2, so that the problem is uniquely solvable. The equivalent weak form of the problem is:

$$\text{Find } u \in H_0^1(\Omega) \text{ such that } A(u,v) = F(v) \quad \forall v \in H_0^1(\Omega), \tag{3.12}$$

where:

$$A(u,v) = \int_\Omega (\bar{\mathbf{a}}\nabla u \cdot \nabla v + buv)\, d\mathbf{x}, \qquad F(v) = \int_\Omega fv\, d\mathbf{x} - A(w,v) = (f,v)_{L^2(\Omega)} - A(w,v),$$

and where $g = \text{tr } w$. In this section, we briefly discuss box and finite element discretizations of this problem, and some of the properties of the discrete equations and spaces which arise. We remark that such discretizations were examined in detail in Chapter 2; our purpose here is to summarize some information which will be useful later, and to explain how the discrete equations which arise fit into the framework we are constructing in this chapter.

The discretized domain is made up of the n_k nodes $\Omega_k = \{\mathbf{x}_k^1, \ldots, \mathbf{x}_k^{n_k}\}$ and the l_k elements (or *volumes* or *boxes*) $\mathcal{T}_k = \{\tau_k^1, \ldots, \tau_k^{l_k}\}$. The parameters $\{h_k^1, \ldots, h_k^{n_k}\}$ represent the diameters of the circumscibing spheres of the set of elements $\mathcal{T}_k$, and $\{\rho_k^1, \ldots, \rho_k^{l_k}\}$ represent the diameters of the inscribed spheres. The mesh parameter h_k represents the diameter of the largest element or volume in $\mathcal{T}_k$, or $h_k = \max_i\{h_k^i\}$, and ρ_k represents the smallest of the inscribing diameters, $\rho_k = \min_i\{\rho_k^i\}$. In the finite element case, there is also an associated set of C^0-piecewise linear basis functions $\{\phi_k^1, \ldots, \phi_k^{n_k}\}$ defined over the tessellation $\mathcal{T}_k$. The basis functions are taken to be *Lagrangian* (or *nodal*) in the sense that $\phi_k^i(\mathbf{x}_k^j) = \delta_{ij}$.

We will be concerned with a sequence of refinements of the discretization Ω_k and the tessellation $\mathcal{T}_k$, where $k = 1$ corresponds to the initial or coarsest mesh, and $k = J$ corresponds to the final or most highly refined mesh. Typically, the following *shape-regular* (Assumption H1, page 132 of [36]) and *quasi-uniform* (the "inverse" assumption, page 140 of [36]) assumptions are made on the elements comprising $\mathcal{T}_k$ at each level k:

$$\frac{h_k^i}{\rho_k^i} \le \sigma \quad \forall i, \quad k = 1,2,\ldots,J \tag{3.13}$$

$$\frac{h_k}{h_k^i} \le \nu \quad \forall i, \quad k = 1,2,\ldots,J \tag{3.14}$$

where the constants σ and ν are independent of i and k. The first condition (3.13), which is a natural condition to impose, states that the elements in the mesh do not become too *skewed* or *degenerate* at any discretization level. The ratio h_k/ρ_k appears directly in error estimates for the finite element method (page 111 in [36]), although when (3.13) is satisfied, the error estimates can be written completely in terms of the maximal diameter h_k (Theorem 3.2.2, page 134 in [36]). The second condition (3.14) is also natural, and simply states that the elements at any particular level remain comparable in size asymptotically. When these conditions are satisfied, on each level k the number of nodes is related to the size of the elements as $h_k = O(n_k^{-1/d})$, or $n_k = O(h_k^{-d})$.

When problem (3.11) is discretized at level k with either the box or finite element method, the following matrix problem is generated:

$$\text{Find } u_k \in \mathcal{U}_k \text{ such that } A_k u_k = f_k, \tag{3.15}$$

where A_k is an SPD matrix, and where the solution space $\mathcal{U}_k$ is defined as:

$$\mathcal{U}_k = \{u_k \in \mathbb{R}^{n_k} : u_k(\mathbf{x}_k^i) \in \mathbb{R},\ \forall \mathbf{x}_k^i \in \Omega_k\}.$$

The space $\mathcal{U}_k$ represents the space of grid functions $u_k = (u_k(\mathbf{x}_k^1), \ldots, u_k(\mathbf{x}_k^{n_k}))^T$ with values of each of the nodes $\{\mathbf{x}_k^i\}$. The space $\mathcal{U}_k$ (which is simply $\mathbb{R}^{n_k}$) is a finite-dimensional Hilbert space when equipped with the inner-product and norm:

$$(u_k, v_k)_k = h_k^d \sum_{i=1}^{n_k} u_k(\mathbf{x}_k^i) v_k(\mathbf{x}_k^i), \qquad \|u_k\|_k = (u_k, u_k)_k^{1/2}, \qquad \forall u_k, v_k \in \mathcal{U}_k. \tag{3.16}$$

The A_k-inner-product and A_k-norm in $\mathcal{U}_k$ are defined by:

$$(u_k, v_k)_{A_k} = (A_k u_k, v_k)_k, \qquad \|u_k\|_{A_k} = (u_k, u_k)_{A_k}^{1/2}, \qquad \forall u_k, v_k \in \mathcal{U}_k. \tag{3.17}$$

It is well-known that for either a box or finite element discretization satisfying assumptions (3.13) and (3.14), the eigenvalues and condition number of the matrix which arises can be bounded by:

$$\lambda_{\min}(A_k) \geq C_1 h_k^d, \qquad \lambda_{\max}(A_k) \leq C_2 h_k^{d-2}, \qquad \kappa(A_k) = \frac{\lambda_{\max}(A_k)}{\lambda_{\min}(A_k)} \leq \left(\frac{C_2}{C_1}\right) h_k^{-2}, \tag{3.18}$$

where a division or multiplication of the matrix (and hence the eigenvalue bounds above) by various powers of h_k is common. For the derivation of these bounds, see for example Theorem 5.1 in [174] or page 236 of [9] for the finite element case, or the discussion of the box-method for a model problem below in §3.1.6.

In the case of the finite element method, the discrete problem can also be interpreted abstractly, which is required for the recent multilevel theories. To begin, the finite element space of C^0-piecewise linear functions defined over the tessellation $\mathcal{T}_k$ at level k is denoted:

$$\mathcal{M}_k = \{u_k \in H_0^1(\Omega) : u_k|_{\tau_k^i} \in \mathcal{P}_1(\tau_k^i),\ \forall \tau_k^i \in \mathcal{T}_k\},$$

where $\mathcal{P}_1(\tau)$ is the space of polynomials of degree one over τ. The space $\mathcal{M}_k$ is a finite-dimensional Hilbert space when equipped with the inner-product and norm in $H_0^1(\Omega)$. If $u_k \in \mathcal{M}_k$, and $\tilde{u}_k \in \mathcal{U}_k$ such that $u_k = \sum_{i=1}^{n_k} \tilde{u}_k(\mathbf{x}_k^i)\phi_k^i$ for Lagrangian $\{\phi_k^i\}$, then it can be shown (see Assumption 3.2 of [23]) that the discrete norm defined in equation (3.16) is equivalent to the L^2-norm in $\mathcal{M}_k$ in the sense:

$$C_1\|u_k\|_{L^2(\Omega)} \leq \|\tilde{u}_k\|_k \leq C_2\|u_k\|_{L^2(\Omega)}, \qquad \forall u_k = \sum_{i=1}^{n_k} \tilde{u}_k(\mathbf{x}_k^i)\phi_k^i \in \mathcal{M}_k, \qquad \tilde{u}_k \in \mathcal{U}_k.$$

The finite element approximation to the solution of the partial differential equation (3.12) has the form:

$$\text{Find } u_k \in \mathcal{M}_k \text{ such that } A(u_k, v_k) = (f, v_k)_{L^2(\Omega)} - A(w_k, v_k) \quad \forall v_k \in \mathcal{M}_k. \tag{3.19}$$

Since $A(\cdot,\cdot)$ is a bilinear form on the finite-dimensional space $\mathcal{M}_k$, there exists a bounded linear operator $A_k : \mathcal{M}_k \mapsto \mathcal{M}_k$ such that $(A_k u_k, v_k)_{L^2(\Omega)} = A(u_k, v_k)\ \forall u_k, v_k \in \mathcal{M}_k$ (Theorem 1 page 38 of [89]). If we denote the L^2-projector onto $\mathcal{M}_k$ as $Q_k : L_2(\Omega) \mapsto \mathcal{M}_k$ such that $(Q_k f, v_k)_{L^2(\Omega)} = (f, v_k)_{L^2(\Omega)},\ \forall f \in L^2(\Omega), v_k \in \mathcal{M}_k$, then problem (3.19) is equivalent to the abstract problem:

$$\text{Find } u_k \in \mathcal{M}_k \text{ such that } A_k u_k = f_k, \qquad \text{where } f_k = Q_k f - A_k w_k. \tag{3.20}$$

The operator A_k is an abstract (SPD) operator on $\mathcal{M}_k$, and its representation with respect to the set of finite element basis functions $\{\phi_k^i\}$ is the stiffness matrix $[K_k]_{ij} = A(\phi_k^i, \phi_k^j)$.

Bounds for the maximum and minimum eigenvalues of the abstract operator A_k can be derived, analogous to those in equation (3.18) for the matrices arising in box or finite element discretizations, using a well-known inverse inequality available in the finite element literature. To begin, we first note that if the elements $\mathcal{T}_k$ satisfy the shape-regular and quasi-uniform assumptions (3.13) and (3.14), then the following *inverse inequality* can be derived for the resulting space $\mathcal{M}_k$ (see inequality 3.2.37, page 142 in [36]):

$$\|u_k\|_{H^1(\Omega)} \leq \gamma h_k^{-1} \|u_k\|_{L^2(\Omega)}, \qquad \forall u_k \in \mathcal{M}_k. \tag{3.21}$$

This inequality can be combined with the usual boundedness and coerciveness conditions on the underlying bilinear form to prove the next result; the bound for $\lambda_{\max}$ is given in [23], and a more careful derivation of both upper and lower bounds on each eigenvalue is given in [184].

Theorem 3.18 *The following bounds hold for the abstract discrete operator A_k:*

$$\lambda_{\min}(A_k) \geq C_1, \qquad \lambda_{\max}(A_k) \leq C_2 h_k^{-2}, \qquad \kappa(A_k) \leq \left(\frac{C_2}{C_1}\right) h_k^{-2}.$$

Proof. Given the usual boundedness condition on the bilinear form defined by A_k:

$$A(u_k, v_k) = (A_k u_k, v_k)_{L^2(\Omega)} \leq M \|u_k\|_{H^1(\Omega)} \|v_k\|_{H^1(\Omega)}, \qquad \forall u_k, v_k \in \mathcal{M}_k,$$

we can combine this with the inverse inequality (3.21) to bound the largest eigenvalue of the operator A_k:

$$\lambda_{\max}(A_k) = \max_{u_k \neq 0} \frac{(A_k u_k, u_k)_{L^2(\Omega)}}{(u_k, u_k)_{L^2(\Omega)}} \leq \max_{u_k \neq 0} \frac{M \|u_k\|^2_{H^1(\Omega)}}{\|u_k\|^2_{L^2(\Omega)}}$$

$$\leq \max_{u_k \neq 0} \frac{M \gamma^2 h_k^{-2} \|u_k\|^2_{L^2(\Omega)}}{\|u_k\|^2_{L^2(\Omega)}} \leq M \gamma^2 h_k^{-2} = C_2 h_k^{-2}.$$

Similarly, the usual coerciveness condition on the bilinear form defined by A_k is given as:

$$A(u_k, u_k) = (A_k u_k, u_k)_{L^2(\Omega)} \geq m \|u_k\|^2_{H^1(\Omega)}, \qquad \forall u_k \in \mathcal{M}_k,$$

which can be used directly to yield a bound on the smallest eigenvalue of A_k:

$$\lambda_{\min}(A_k) = \min_{u_k \neq 0} \frac{(A_k u_k, u_k)_{L^2(\Omega)}}{(u_k, u_k)_{L^2(\Omega)}} \geq \min_{u_k \neq 0} \frac{m \|u_k\|^2_{H^1(\Omega)}}{\|u_k\|^2_{L^2(\Omega)}}$$

$$= \min_{u_k \neq 0} \frac{m(\|u_k\|^2_{L^2(\Omega)} + |u_k|^2_{H^1(\Omega)})}{\|u_k\|^2_{L^2(\Omega)}} \geq \min_{u_k \neq 0} \frac{m \|u_k\|^2_{L^2(\Omega)}}{\|u_k\|^2_{L^2(\Omega)}} \geq m = C_1.$$

The final result follows immediately since

$$\kappa(A_k) = \frac{\lambda_{\max}(A_k)}{\lambda_{\min}(A_k)} \leq \left(\frac{C_2}{C_1}\right) h_k^{-2}.$$

□

To conclude, we see that the discrete approximation to the problem (3.11) can be characterized as the solution to the operator equation:

$$\text{Find } u_k \in \mathcal{H}_k \text{ such that } A_k u_k = f_k, \tag{3.22}$$

for some SPD $A_k \in \mathbf{L}(\mathcal{H}_k, \mathcal{H}_k)$, where $\mathcal{H}_k$ can be interpreted as the grid function space $\mathcal{U}_k$, in which case this is a matrix equation generated by a box or finite element discretization of (3.11), or $\mathcal{H}_k$ may be interpreted as the space $\mathcal{M}_k$, in which case this is an abstract operator equation. In either case, we have seen above that with appropriate assumptions (3.13) and (3.14) on the underlying mesh $\mathcal{T}_k$, we can relate the maximal and minimal eigenvalues of the resulting matrix or abstract operator A_k to the mesh parameter h_k.

3.1.6 Convergence and complexity of classical linear methods

We mention now some classical linear iterations for discrete elliptic equations $Au = f$ on the space $\mathcal{U}$ (leaving off the subscript k here and below since only one space is involved), where A is an SPD matrix. Our purpose is to explain briefly the motivation for considering multilevel methods as alternatives to the classical methods.

Since A is SPD, we may write $A = D - L - L^T$, and where D is a diagonal matrix and L a strictly lower-triangular matrix. The Richardson variation of Algorithm 3.1 takes λ^{-1} as the approximate inverse $B \approx A^{-1}$ of A, where λ is a bound on the largest eigenvalue of A:

$$u^{n+1} = (I - \lambda^{-1}A)u^n + \lambda^{-1}f. \tag{3.23}$$

The Jacobi variation of Algorithm 3.1 takes D^{-1} as the approximate inverse B:

$$u^{n+1} = (I - D^{-1}A)u^n + D^{-1}f. \tag{3.24}$$

In the Gauss-Seidel variant, the approximate inverse is taken to be $(D - L)^{-1}$, giving:

$$u^{n+1} = (I - (D - L)^{-1}A)u^n + (D - L)^{-1}f. \tag{3.25}$$

The SOR variant takes the approximate inverse as $\omega(D - \omega L)^{-1}$, giving:

$$u^{n+1} = (I - \omega(D - \omega L)^{-1}A)u^n + \omega(D - \omega L)^{-1}f. \tag{3.26}$$

In the case that the model problem of the Poisson equation on a uniform mesh is considered, then the eigenvalues of both A and the error propagation matrix $I - BA$ can be determined analytically. This allows for an analysis of the convergence rates of the Richardson, Jacobi, and Gauss-Seidel iterations.

To give an example of the convergence results which are available for these classical methods, first recall that for the real square matrix A, the splitting $A = M - N$ is called a *regular splitting* (page 88 of [179]) of A if $N > 0$, M is nonsingular, and $M^{-1} \geq 0$. Note that an alternative construction of the Jacobi and Gauss-Seidel methods is through matrix splittings. For example, given the splitting $A = M - N = D - (L + U)$ which corresponds to the Jacobi iteration, the resulting iteration can be writing in terms of M and N as follows:

$$u^{n+1} = (I - D^{-1}A)u^n + D^{-1}f = (I - M^{-1}(M - N))u^n + M^{-1}f = M^{-1}Nu^n + M^{-1}f.$$

Therefore, for a splitting $A = M - N$, the convergence of the resulting linear method is governed completely by the spectral radius of the error propagation matrix, $\rho(M^{-1}N)$. The following standard theorem gives a sufficient condition for converge of the Jacobi and Gauss-Seidel iterations, which can be considered to be regular splittings of A.

Theorem 3.19 *If A is an M-matrix, and M is obtained from A by setting off-diagonal elements of A to zero, then the splitting $A = M - N$ is regular and the corresponding linear iteration defined by the splitting is convergent; i.e., $\rho(M^{-1}N) < 1$.*

Proof. See page 90, Theorem 3.14 in [179]. □

Given that λ is the largest eigenvalue (or an upper bound on the largest eigenvalue) of A, we remark that Richardson's method is always trivially convergent since each eigenvalue $\lambda_i(E)$ of E is bounded by one:

$$\lambda_i(E) = \lambda_i(I - BA) = \lambda_i(I - \lambda^{-1}A) = 1 - \lambda^{-1}\lambda_i(A) < 1.$$

However, the following difficulty makes these classical linear methods impractical for large problems. Consider the case of the three-dimensional Poisson's equation on the unit square with zero Dirichlet boundary conditions, discretized with the box-method on a uniform mesh with m mesh-points in each mesh direction ($n = m^3$) and mesh spacing $h = 1/(m + 1)$. It is well-known that the eigenvalues of the resulting matrix A can be expressed in closed form:

$$\lambda_i = \lambda_{\{p,q,r\}} = 6 - 2\cos(p\pi h) - 2\cos(q\pi h) - 2\cos(r\pi h), \qquad p, q, r = 1, \ldots, m.$$

Clearly, the largest eigenvalue of A is $\lambda = 6(1 - cos(m\pi h))$, and the smallest is $\lambda_1 = 6(1 - cos(\pi h))$. It is not difficult to show (see pages 201-205 in [179], or pages 127-132 in [187] for the two-dimensional case) that the largest eigenvalue of the Jacobi error propagation matrix $I - D^{-1}A$ is in this case equal to $cos(\pi h)$. It is also well-known that for consistently ordered matrices with *Property $\mathcal{A}$* (page 42 in [187]), the spectral radius of the Gauss-Seidel error propagation matrix is the square of the Jacobi matrix spectral radius; more generally, the relationship between the Jacobi and Gauss-Seidel spectral radii is given by the *Stein-Rosenberg Theorem* (see Theorem 3.3, page 70 of [179], or the extended form appearing as Theorem 5.1 and Corollary 5.2, pages 120-122 of [187]). An expression for the spectral radius of the SOR error propagation matrix can also be derived; the spectral radii for the classical methods are then:

- Richardson: $\rho(E) = 1 - 6\lambda^{-1}(1 - cos(\pi h)) \approx 1 - 3\lambda^{-1}\pi^2 h^2 = 1 - O(h^2)$
- Jacobi: $\rho(E) = cos(\pi h) \approx 1 - \frac{1}{2}\pi^2 h^2 = 1 - O(h^2)$
- Gauss-Seidel: $\rho(E) = cos^2(\pi h) \approx 1 - \pi^2 h^2 = 1 - O(h^2)$
- SOR: $\rho(E) \approx 1 - O(h)$

The same dependence on h is exhibited for one- and two-dimensional problems. Therein lies the problem: as $h \to 0$, then for the classical methods $\rho(E) \to 1$, so that the methods converge more and more slowly as the problem size is increased.

Remark 3.5. An alternative convergence proof for the Jacobi and Gauss-Seidel iterations follows simply by noting that the matrix $I - E^*E$ is A-positive for both the Jacobi and Gauss-Seidel error propagators E, and by employing Lemma 3.3, or the related Stein's Theorem. Stein's Theorem is the basis for the proof of the Ostrowski-Reich SOR convergence theorem (Theorem 7.1.10 in [157]).

3.2 Linear multilevel methods

Multilevel (or *multigrid*) methods are highly efficient numerical techniques for solving the algebraic equations arising from the discretization of partial differential equations. These methods were developed in direct response to the deficiencies of the classical iterations discussed in the previous section. Some of the early fundamental papers are [29, 84, 175], and a comprehensive analysis of the many different aspects of these methods is given in [85]. The following derivation of two-level and multilevel methods in a recursive operator framework is motivated by some very recent work on finite element-based multilevel and domain decomposition methods, represented for example by [26, 58, 185]. Our notation follows the currently established convention for these types of methods, represented for example by [185].

3.2.1 Linear equations in a nested sequence of spaces

In what follows, we will often be concerned with a nested sequence of spaces $\mathcal{H}_1 \subset \mathcal{H}_2 \subset \cdots \subset \mathcal{H}_J \equiv \mathcal{H}$, where $\mathcal{H}_J$ corresponds to the finest or largest space and $\mathcal{H}_1$ the coarsest or smallest. Each space $\mathcal{H}_k$ is taken to be a Hilbert space, equipped with an inner-product $(\cdot,\cdot)_k$ which induces the norm $\|\cdot\|_k$. Regarding notation, if $A \in \mathbf{L}(\mathcal{H}_k, \mathcal{H}_k)$ then we denote the operator as A_k. Similarly, if $A \in \mathbf{L}(\mathcal{H}_k, \mathcal{H}_i)$ then we denote the operator as A_k^i. Finally, if $A \in \mathbf{L}(\mathcal{H}_k, \mathcal{H}_k)$ but its operation concerns somehow a specific subspace $\mathcal{H}_i \subset \mathcal{H}_k$, then we denote the operator as $A_{k;i}$. For quantities involving the finest space $\mathcal{H}_J$, we will often leave off the subscripts, without danger of confusion.

Now, given such a nested sequence of Hilbert spaces, we assume that associated with each space $\mathcal{H}_k$ is an SPD operator A_k, which defines a second inner-product $(\cdot,\cdot)_{A_k} = (A_k\cdot,\cdot)_k$, inducing a second norm $\|\cdot\|_{A_k} = (\cdot,\cdot)_{A_k}^{1/2}$. The spaces $\mathcal{H}_k$ are connected by *prolongation* operators $I_{k-1}^k \in \mathbf{L}(\mathcal{H}_{k-1}, \mathcal{H}_k)$ and *restriction* operators $I_k^{k-1} \in \mathbf{L}(\mathcal{H}_k, \mathcal{H}_{k-1})$, where we assume that $\mathrm{NULL}(I_{k-1}^k) = \{0\}$, and usually that $I_k^{k-1} = (I_{k-1}^k)^T$, where the adjoint is with respect to the inner products on the sequence of spaces $\mathcal{H}_k$:

$$(u_k, I_{k-1}^k v_{k-1})_k = ((I_{k-1}^k)^T u_k, v_{k-1})_{k-1}, \qquad \forall u_k \in \mathcal{H}_k, \qquad \forall v_{k-1} \in \mathcal{H}_{k-1}. \tag{3.27}$$

We are given the operator equation $Au = f$ in the finest space $\mathcal{H} \equiv \mathcal{H}_J$, where $A \in \mathbf{L}(\mathcal{H}, \mathcal{H})$ is SPD, and we are interested in iterative algorithms for determining the unique solution u which involve solving problems in the coarser spaces $\mathcal{H}_k$ for $1 \le k < J$. If the equation in $\mathcal{H}$ has arisen from a box or finite element discretization of an elliptic partial differential equation, then operators A_k (and the associated coarse problems $A_k u_k = f_k$) in coarser spaces $\mathcal{H}_k$ for $k < J$ may be defined naturally with the same discretization on a coarser mesh. Alternatively, it is convenient (for theoretical reasons which we will discuss later in the chapter) to take the so-called *variational approach* of constructing the coarse operators, where the operators $A_k \in \mathbf{L}(\mathcal{H}_k, \mathcal{H}_k)$ satisfy:

$$A_{k-1} = I_k^{k-1} A_k I_{k-1}^k, \qquad I_k^{k-1} = (I_{k-1}^k)^T. \tag{3.28}$$

The first condition in (3.28) is sometimes referred to as the *Galerkin condition*, whereas the two conditions (3.28) together are known as the *variational conditions*, due to the fact that both conditions are satisfied naturally by variational or Galerkin (finite element) discretizations on successively refined meshes. Note that if A_k is SPD, then A_{k-1} produced by (3.28) will also be SPD.

In the case that $\mathcal{H}_k = \mathcal{U}_k$, the prolongation operator I_{k-1}^k typically corresponds to d-dimensional interpolation of u_{k-1} to $u_k = I_{k-1}^k u_{k-1}$, where u_{k-1} and u_k are interpreted as grid functions defined over two successively refined (box or finite element) discretizations Ω_{k-1} and Ω_k of the domain $\Omega \subset \mathbb{R}^d$. Since the coarse grid function space has by definition lower dimension than the fine space, I_{k-1}^k takes the form of a rectangular matrix with more rows than columns. A constant $c \in \mathbb{R}$ will appear in the second condition in (3.28), which will become $I_k^{k-1} = c(I_{k-1}^k)^T$, due to taking I_k^{k-1} to be the adjoint of I_{k-1}^k with respect to the inner-product (3.16), and since $h_k < h_{k-1}$.

In the case that $\mathcal{H}_k = \mathcal{M}_k$, the prolongation corresponds to the natural inclusion of a coarse space function into the fine space, and the restriction corresponds to the L^2-projection of a fine space function onto the coarse space. The variational conditions (3.28) then hold for the abstract operators A_k on the spaces $\mathcal{M}_k$, with inclusion and L^2-projection for the prolongation and restriction (see the proof of Note 3.6.6 in §3.6 of [85]). In addition, the stiffness matrices representing the abstract operators A_k also satisfy the conditions (3.28), where now the prolongation and restriction operators are as in the case of the space $\mathcal{U}_k$. However, we remark that this is true only with *exact evaluation* of the integrals forming the matrix components; the conditions (3.28) are violated if quadrature is used, and are violated radically if the quadrature error is large (see [74] for a discussion and analysis).

Recent results have been obtained for multilevel methods in the spaces $\mathcal{H}_k = \mathcal{M}_k$, which rely on certain operator recursions (we point out in particular the papers [24, 26, 184, 185]). Some of these results [26, 185] are "regularity-free" in the sense that they do not require the usual regularity or smoothness assumptions on the solution to the problem, which is important since these are not valid for problems such as those with discontinuous coefficients. Since our interest is exactly those problems with difficulties such as discontinuous coefficients, our hope is to use these recent results as guidelines to develop and apply multilevel methods which (1) provide optimal or near optimal complexity numerical solution of the problem, and (2) are supplied with a rigorous convergence theory.

Unfortunately, we are in a situation in which it is impossible to satisfy all of the necessary assumptions to apply even these recent results; however, we will see that some of the results can be adapted to our situation. We will develop the multilevel algorithms here and in the next chapter in a recursive form in the abstract spaces $\mathcal{H}_k$. In Chapter 5, we will review the existing results for the case $\mathcal{H}_k = \mathcal{M}_k$. We will then attempt to formulate an abstract theory in $\mathcal{H}_k$, which will also apply in the case $\mathcal{H}_k = \mathcal{U}_k$, when the operators satisfy some (but not all) of the relationships arising naturally in the case of $\mathcal{H}_k = \mathcal{M}_k$. Later in this chapter, we will expand in some detail on how some of these relationships can be algebraically imposed on discrete elliptic equations in a reasonably efficient way.

Some purely "algebraic" multilevel theories exist (see for example [30, 166]); however, we are interested only in the specific case of discrete elliptic equations rather than general matrix equations, and we wish to try to incorporate some of the recent results available from the finite element multilevel literature.

3.2.2 Two-level methods

As we noted earlier, the convergence rate of the classical methods deteriorate as the mesh size $h_k \to 0$. However, using the same spectral analysis one can easily see that the components of the error corresponding to the small eigenvalues of the error propagation operator are actually being decreased quite effectively even as $h_k \to 0$; these are the rapidly varying or *high frequency* components in the error. This effect is illustrated graphically in Figure 3.1 for Gauss-Seidel iteration applied to the two-dimensional Poisson's equation on the unit square. In the figure, the error in both physical and Fourier (or frequency) space is shown initially and after one, two, and five iterations. In the Fourier space plots, the low-frequency components of the error

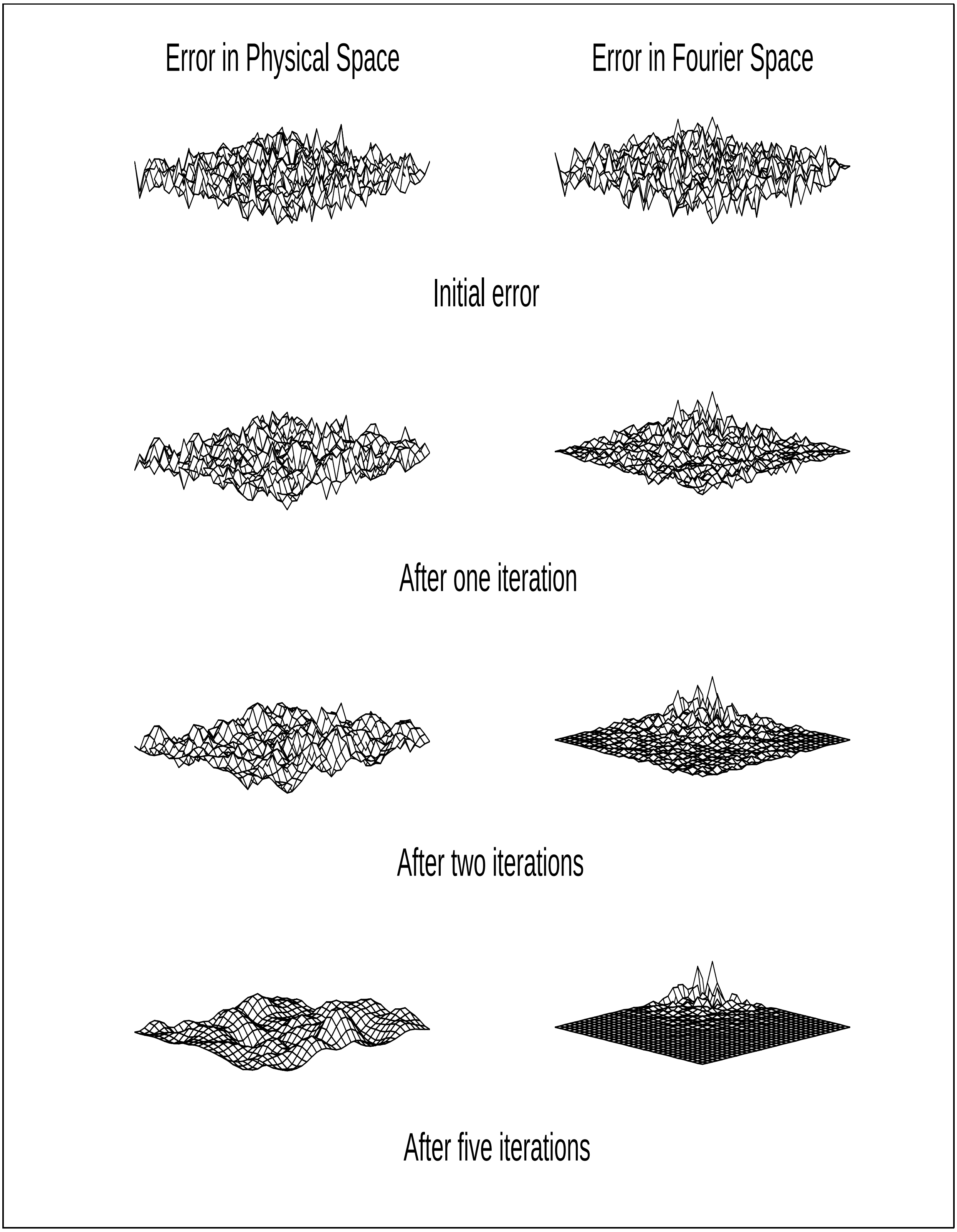


Figure 3.1: The error-smoothing effect of Gauss-Seidel iteration.

are found in the rear, whereas the high-frequency components are found to the far left, the far right, and in the foreground. The source function for this example was constructed from a random field (to produce all frequencies in the solution) and the initial guess was taken to be zero.

The observation that classical linear methods are very efficient at reducing the high frequency modes is the motivation for the multilevel method: a classical linear method can be used to handle the high frequency components of the error (or to *smooth* the error), and the low frequency components can be eliminated efficiently on a coarser mesh with fewer unknowns, where the low frequency modes are well represented.

For the equation $A_k u_k = f_k$ on level k, the smoothing method takes the form of Algorithm 3.1 for some operator R_k, the *smoothing operator*, as the approximate inverse of the operator A_k:

$$u_k^{n+1} = u_k^n + R_k(f_k - A_k u_k^n). \tag{3.29}$$

In the case of two spaces $\mathcal{H}_k$ and $\mathcal{H}_{k-1}$, the error equation $e_k = A_k^{-1} r_k$ is solved approximately using the coarse space, with the *coarse level correction operator* $C_k = I_{k-1}^k A_{k-1}^{-1} I_k^{k-1}$ representing exact solution with A_{k-1}^{-1} in the coarse level subspace $\mathcal{H}_{k-1}$. The solution is then adjusted by the correction:

$$u_k^{n+1} = u_k^n + C_k(f_k - A_k u_k^n). \tag{3.30}$$

There are several ways in which these two procedures can be combined.

By viewing multilevel methods as compositions of the simple linear methods (3.29) and (3.30), a simple yet complete framework for understanding these methods can be constructed. The most important concepts can be discussed with regard to two-level methods, and then generalized to more than two levels using an implicit recursive definition of an approximate coarse level inverse operator.

Consider the case of two nested spaces $\mathcal{H}_{k-1} \subset \mathcal{H}_k$, and the following two-level method:

Algorithm 3.3 *(Nonsymmetric Two-Level Method)*

```
(1) Coarse level correction:   $v_k = u_k^n + C_k(f_k - A_k u_k^n)$
(2) Post-smoothing:            $u_k^{n+1} = v_k + R_k(f_k - A_k v_k)$.
```

The coarse level correction operator has the form $C_k = I_{k-1}^k A_{k-1}^{-1} I_k^{k-1}$, and the smoothing operator is one of the classical iterations. This two-level iteration, a composition of two linear iterations of the form of Algorithm 3.1, can itself be written in the form of Algorithm 3.1:

$$\begin{aligned} u_k^{n+1} = v_k + R_k(f_k - A_k v_k) &= u_k^n + C_k(f_k - A_k u_k^n) + R_k f_k - R_k A_k(u_k^n + C_k(f_k - A_k u_k^n)) \\ &= u_k^n + C_k f_k - C_k A_k u_k^n + R_k f_k - R_k A_k u_k^n - R_k A_k C_k f_k + R_k A_k C_k A_k u_k^n \\ &= (I - C_k A_k - R_k A_k + R_k A_k C_k A_k) u_k^n + (C_k + R_k - R_k A_k C_k) f_k \\ &= (I - B_k A_k) u_k^n + B_k f_k. \end{aligned}$$

The *two-level operator* B_k, the approximate inverse of A_k which is implicitly defined by the nonsymmetric two-level method, has the form:

$$B_k = C_k + R_k - R_k A_k C_k. \tag{3.31}$$

The error propagation operator for the two-level method has the usual form $E_k = I - B_k A_k$, which now can be factored due to the above form for B_k:

$$E_k = I - B_k A_k = (I - R_k A_k)(I - C_k A_k). \tag{3.32}$$

In the case that ν post-smoothing iterations are performed in step (2) instead of one, it is not difficult to show that the error propagation operator takes the altered form:

$$I - B_k A_k = (I - R_k A_k)^\nu (I - C_k A_k).$$

Now consider a symmetric form of the above two-level method:

Algorithm 3.4 *(Symmetric Two-Level Method)*

(1) Pre-smoothing:	$w_k = u_k^n + R_k^T(f_k - A_k u_k^n)$
(2) Coarse level correction:	$v_k = w_k + C_k(f_k - A_k w_k)$
(3) Post-smoothing:	$u_k^{n+1} = v_k + R_k(f_k - A_k v_k)$.

As in the nonsymmetric case, it is a simple task to show that this two-level iteration can be written in the form of Algorithm 3.1:

$$u_k^{n+1} = (I - B_k A_k)u_k^n + B_k f_k,$$

where, after a simple expansion as for the nonsymmetric method above, the *two-level operator* B_k implicitly defined by the symmetric method can be seen to be:

$$B_k = R_k + C_k + R_k^T - R_k A_k C_k - R_k A_k R_k^T - C_k A_k R_k^T + R_k A_k C_k A_k R_k^T.$$

It is easily verified that the factored form of the resulting error propagator E_k^s for the symmetric algorithm is:

$$E_k^s = I - B_k A_k = (I - R_k A_k)(I - C_k A_k)(I - R_k^T A_k).$$

Note that the operator $I - B_k A_k$ is A_k-self-adjoint, which by Lemma 3.5 is true if and only if B_k is symmetric, implying the symmetry of B_k. The operator B_k constructed by the symmetric two-level iteration is always symmetric if the smoothing operator R_k is symmetric; however, it is also true in the symmetric algorithm above when general nonsymmetric smoothing operators R_k are used, because we use the adjoint R_k^T of the post-smoothing operator R_k as the pre-smoothing operator. The symmetry of B_k is important for use as a preconditioner for the conjugate gradient method, which requires that B_k be symmetric for convergence.

Remark 3.6. Note that this alternating technique for producing symmetric operators B_k can be extended to multiple nonsymmetric smoothing iterations, as suggested in [25]. Denote the variable nonsymmetric smoothing operator $R_k^{(i)}$ as:

$$R_k^{(i)} = \left\{ \begin{array}{ll} R_k, & i \text{ odd} \\ R_k^T, & i \text{ even} \end{array} \right\}.$$

If ν pre-smoothings are performed, alternating between R_k and R_k^T, and ν post-smoothings are performed alternating in the opposite way, then a tedious computation shows that the error propagator has the factored form:

$$I - B_k A_k = \left(\prod_{i=1}^{\nu} (I - R_k^{(i)} A_k) \right) (I - C_k A_k) \left(\prod_{i=1}^{\nu} (I - (R_k^{(i)})^T A_k) \right),$$

where we take the convention that the highest indices in the products appear on the left. It is easy to verify that $I - B_k A_k$ is A_k-self-adjoint, so that B_k is symmetric.

3.2.3 The variational conditions and A-orthogonal projection

Up to this point, we have specified the approximate inverse corresponding to the coarse level subspace correction only as $C_k = I_{k-1}^k A_{k-1}^{-1} I_k^{k-1}$, for some coarse level operator A_{k-1}. Consider the case that the variational conditions (3.28) are satisfied. The error propagation operator for the coarse level correction then takes the form:

$$I - C_k A_k = I - I_{k-1}^k A_{k-1}^{-1} I_k^{k-1} A_k = I - I_{k-1}^k [(I_{k-1}^k)^T A_k I_{k-1}^k]^{-1} (I_{k-1}^k)^T A_k.$$

This last expression is simply the A_k-orthogonal projector $I - P_{k;k-1}$ onto the complement of the coarse level subspace, where the unique orthogonal and A_k-orthogonal projectors $Q_{k;k-1}$ and $P_{k;k-1}$ projecting $\mathcal{H}_k$ onto $I_{k-1}^k \mathcal{H}_{k-1}$ can be written as:

$$Q_{k;k-1} = I_{k-1}^k [(I_{k-1}^k)^T I_{k-1}^k]^{-1} (I_{k-1}^k)^T, \qquad P_{k;k-1} = C_k A_k = I_{k-1}^k [(I_{k-1}^k)^T A_k I_{k-1}^k]^{-1} (I_{k-1}^k)^T A_k.$$

In other words, if the variational conditions are satisfied, and the coarse level equations are solved exactly, then the coarse level correction projects the error onto the A_k-orthogonal complement of the coarse level subspace. It is now not surprising that successively refined finite element discretizations satisfy the variational conditions naturally, since they are defined in terms of A_k-orthogonal projections.

Note the following interesting relationship between the symmetric and nonsymmetric two-level methods, which is a consequence of the A_k-orthogonal projection property.

Lemma 3.20 *If the variational conditions (3.28) hold, then the nonsymmetric and symmetric propagators E_k and E_k^s are related by:*

$$\|E_k^s\|_{A_k} = \|E_k\|_{A_k}^2.$$

Proof. Since $I - C_k A_k$ is a projector, we have $(I - C_k A_k)^2 = I - C_k A_k$. It follows that:

$$E_k^s = (I - R_k A_k)(I - C_k A_k)(I - R_k^T A_k) = (I - R_k A_k)(I - C_k A_k)(I - C_k A_k)(I - R_k^T A_k) = E_k E_k^*,$$

where E_k^* as the A_k-adjoint of E_k. Therefore, the convergence of the symmetric algorithm is related to that of the nonsymmetric algorithm as:

$$\|E_k^s\|_{A_k} = \|E_k E_k^*\|_{A_k} = \|E_k\|_{A_k}^2.$$

□

Remark 3.7. The relationship between the symmetric and nonsymmetric error propagation operators in Lemma 3.20 was first pointed out by McCormick in [144], and has been exploited in several recent papers [24, 185]. It allows one to use the symmetric form of the algorithm as may be necessary for use with conjugate gradient methods, while exploiting the above relationship to work only with the nonsymmetric error propagator E_k in analysis, which may be easier to analyze.

3.2.4 Multilevel methods

Consider now the full nested sequence of spaces $\mathcal{H}_1 \subset \mathcal{H}_2 \subset \cdots \subset \mathcal{H}_J \equiv \mathcal{H}$. The idea of the multilevel method is to begin with the two-level method, but rather than solve the course level equations exactly, yet another two-level method is used to solve the coarse level equations approximately, beginning with an initial approximation of zero on the coarse level. The idea is applied recursively, until the cost of solving the coarse system is negligible, or until the coarsest possible level is reached.

The following is a recursively defined multilevel algorithm which corresponds to the form of the algorithm commonly implemented on a computer. For the system $Au = f$, the algorithm returns the approximate solution u^{n+1} after one iteration of the method applied to the initial approximate u^n.

Algorithm 3.5 *(Nonsymmetric Multilevel Method – Implementation Form)*

$$u^{n+1} = ML(J, u^n, f)$$

where the operation $u_k^{\text{NEW}} = ML(k, u_k^{\text{OLD}}, f_k)$ is defined recursively:

```
IF $(k = 1)$ THEN:
    (1) Solve directly:            $u_1^{\text{NEW}} = A_1^{-1} f_1$.
ELSE:
    (1) Coarse level correction:   $v_k = u_k^{\text{OLD}} + I_{k-1}^k (ML(k-1, 0, I_k^{k-1}(f_k - A_k u_k^{\text{OLD}})))$
    (2) Post-smoothing:            $u_k^{\text{NEW}} = v_k + R_k(f_k - A_k v_k)$.
END.
```

As with the two-level Algorithm 3.3, it is a straightforward calculation to write the multilevel Algorithm 3.5 in the standard form of Algorithm 3.1, where now the *multilevel operator* $B \equiv B_J$ is defined recursively. To begin, assume that the approximate inverse of A_{k-1} at level $k-1$ implicitly defined by Algorithm 3.5 has been explicitly identified and denoted as B_{k-1}. The coarse level correction step of Algorithm 3.5 at level k can then be written as:

$$v_k = u_k^{\text{OLD}} + I_{k-1}^k B_{k-1} I_k^{k-1}(f_k - A_k u_k^{\text{OLD}}).$$

At level k, Algorithm 3.5 can now be thought of as the two-level Algorithm 3.3, where the two-level operator $C_k = I_{k-1}^k A_{k-1}^{-1} I_k^{k-1}$ has been replaced by the approximation $C_k = I_{k-1}^k B_{k-1} I_k^{k-1}$. From (3.31), we see that the expression for the multilevel operator B_k at level k in terms of the operator B_{k-1} at level $k-1$ is given by:

$$B_k = I_{k-1}^k B_{k-1} I_k^{k-1} + R_k - R_k A_k I_{k-1}^k B_{k-1} I_k^{k-1}. \tag{3.33}$$

We can now state a second multilevel algorithm, which is mathematically equivalent to Algorithm 3.5, but which is formulated explicitly in terms of the recursively defined multilevel operators B_k.

Algorithm 3.6 *(Nonsymmetric Multilevel Method – Operator Form)*

$$u^{n+1} = u^n + B(f - Au^n)$$

where the multilevel operator $B \equiv B_J$ is defined by the recursion:

(1) Let $B_1 = A_1^{-1}$, and assume B_{k-1} has been defined.
(2) $B_k = I_{k-1}^k B_{k-1} I_k^{k-1} + R_k - R_k A_k I_{k-1}^k B_{k-1} I_k^{k-1}, \qquad k = 2, \ldots, J.$

As was noted for the two-level case, the error propagator at level k can be factored as:

$$E_k = I - B_k A_k = (I - R_k A_k)(I - I_{k-1}^k B_{k-1} I_k^{k-1} A_k). \tag{3.34}$$

Remark 3.8. The recursive definition of the multilevel operators B_k first appeared in [24], although operator recursions for the error propagators $E_k = I - B_k A_k$ had appeared earlier in [136]. Many of the recent results on finite element-based multilevel methods depend on the recursive definition of the multilevel operators B_k; see the next section and Chapter 5.

3.2.5 Recursive and product forms of the multilevel error propagator

Two useful results can be derived from (3.34). The two expressions we derive below are slight generalizations of results appearing recently in the finite element multilevel literature [26, 184]; these results have been the impetus for the most recent theoretical advances on finite element-based multilevel methods.

First, let us define the operator $\tilde{P}_k^{k-1}$:

$$\tilde{P}_k^{k-1} = A_{k-1}^{-1} I_k^{k-1} A_k. \tag{3.35}$$

Note that if the variational conditions (3.28) are satisfied, then this operator is related to the A_k-orthogonal projector $P_{k;k-1}$ as:

$$P_{k;k-1} = I_{k-1}^k [(I_{k-1}^k)^T A_k I_{k-1}^k]^{-1} (I_{k-1}^k)^T A_k = I_{k-1}^k A_{k-1}^{-1} I_k^{k-1} A_k = I_{k-1}^k \tilde{P}_k^{k-1}.$$

Using $\tilde{P}_k^{k-1}$ and $P_{k;k-1}$, we can write the error propagator E_k in terms of E_{k-1}; the following lemma first appeared (in a different notation) as Proposition 5 in [136], and it is used in many recent papers (see equation (2.6) in [24], and Proposition 7.1 in [184]).

Lemma 3.21 *The error propagators E_k of Algorithm 3.6 are generated by the recursion:*

$$E_k = (I - R_k A_k)(I - I_{k-1}^k \tilde{P}_k^{k-1} + I_{k-1}^k E_{k-1} \tilde{P}_k^{k-1}). \tag{3.36}$$

If variational conditions (3.28) hold, this recursion becomes:

$$E_k = (I - R_k A_k)(I - P_{k;k-1} + I_{k-1}^k E_{k-1} \tilde{P}_k^{k-1}). \tag{3.37}$$

Proof. The second term in equation (3.34) can be written as:

$$I - I_{k-1}^k B_{k-1} I_k^{k-1} A_k = I - I_{k-1}^k B_{k-1} A_{k-1} A_{k-1}^{-1} I_k^{k-1} A_k$$
$$= I - I_{k-1}^k B_{k-1} A_{k-1} \tilde{P}_k^{k-1} = I - I_{k-1}^k \tilde{P}_k^{k-1} + I_{k-1}^k \tilde{P}_k^{k-1} - I_{k-1}^k B_{k-1} A_{k-1} \tilde{P}_k^{k-1}$$
$$= I - I_{k-1}^k \tilde{P}_k^{k-1} + I_{k-1}^k (I - B_{k-1} A_{k-1}) \tilde{P}_k^{k-1} = I - I_{k-1}^k \tilde{P}_k^{k-1} + I_{k-1}^k E_{k-1} \tilde{P}_k^{k-1}.$$

Therefore, the error propagators E_k can be defined recursively:

$$E_k = (I - R_k A_k)(I - I_{k-1}^k B_{k-1} I_k^{k-1} A_k) = (I - R_k A_k)(I - I_{k-1}^k \tilde{P}_k^{k-1} + I_{k-1}^k E_{k-1} \tilde{P}_k^{k-1}).$$

If the variational conditions (3.28) hold, then as noted above the A_k-orthogonal projector $P_{k;k-1}$ is related to the operator $\tilde{P}_k^{k-1}$ as $P_{k;k-1} = I_{k-1}^k \tilde{P}_k^{k-1}$, so that the recursion becomes:

$$E_k = (I - R_k A_k)(I - P_{k;k-1} + I_{k-1}^k E_{k-1} \tilde{P}_k^{k-1}).$$

□

The second result which can be derived from (3.34) involves a certain product formulation. We first introduce the following notation for the composition of prolongations and restrictions:

$$I_{k-i}^{k} = I_{k-1}^{k} I_{k-2}^{k-1} \cdots I_{k-i+1}^{k-i+2} I_{k-i}^{k-i+1}, \qquad I_{k}^{k-i} = I_{k-i+1}^{k-i} I_{k-i+2}^{k-i+1} \cdots I_{k-1}^{k-2} I_{k}^{k-1}, \qquad I_k^k = I.$$

We will also use a simplified notation for composite prolongation operators to the finest space $\mathcal{H}$:

$$I_J = I, \quad I_k = I_{J-1}^{J} I_{J-2}^{J-1} \cdots I_{k+1}^{k+2} I_{k}^{k+1}, \quad k = 1, \ldots, J-1.$$

A generalized product form is established in the following lemma, which in the special case of the finite element spaces $H_k = \mathcal{M}_k$ reduces to the product form first derived in [26].

Lemma 3.22 *If variational conditions (3.28) hold, the error propagator E of Algorithm 3.6 can be factored:*

$$E = I - BA = (I - T_J)(I - T_{J-1}) \cdots (I - T_1), \tag{3.38}$$

where

$$T_1 = I_1 A_1^{-1} I_1^T A, \qquad T_k = I_k R_k I_k^T A, \qquad k = 2, \ldots, J.$$

Proof. Let us begin by expanding the second term in (3.34) more fully and then factoring again:

$$\begin{aligned} I - I_{k-1}^{k} B_{k-1} I_{k}^{k-1} A_k &= I - I_{k-1}^{k} (I_{k-2}^{k-1} B_{k-2} I_{k-1}^{k-2} + R_{k-1} - R_{k-1} A_{k-1} I_{k-2}^{k-1} B_{k-2} I_{k-1}^{k-2}) I_k^{k-1} A_k \\ &= I - I_{k-2}^{k} B_{k-2} I_k^{k-2} A_k - I_{k-1}^{k} R_{k-1} I_k^{k-1} A_k + I_{k-1}^{k} R_{k-1} (I_k^{k-1} A_k I_{k-1}^{k}) I_{k-2}^{k-1} B_{k-2} I_k^{k-2} A_k \\ &= I - I_{k-2}^{k} B_{k-2} I_k^{k-2} A_k - I_{k-1}^{k} R_{k-1} I_k^{k-1} A_k + (I_{k-1}^{k} R_{k-1} I_k^{k-1} A_k)(I_{k-2}^{k} B_{k-2} I_k^{k-2} A_k) \\ &= (I - I_{k-1}^{k} R_{k-1} I_k^{k-1} A_k)(I - I_{k-2}^{k} B_{k-2} I_k^{k-2} A_k), \end{aligned}$$

where we have assumed that the first part of the variational conditions (3.28) holds. In general, we have:

$$I - I_{k-i}^{k} B_{k-i} I_k^{k-i} A_k = (I - I_{k-i}^{k} R_{k-i} I_k^{k-i} A_k)(I - I_{k-i-1}^{k} B_{k-i-1} I_k^{k-i-1} A_k).$$

Using this result inductively, beginning with $k = J$, the error propagator $E \equiv E_J$ takes the *product form*:

$$E = I - BA = (I - T_J)(I - T_{J-1}) \cdots (I - T_1).$$

The second part of the variational conditions (3.28) implies that the T_k are A-self-adjoint and have the form:

$$T_1 = I_1 A_1^{-1} I_1^T A, \qquad T_k = I_k R_k I_k^T A, \qquad k = 2, \ldots, J.$$

□

Remark 3.9. For $J = 2$, we recover the two-level error propagator (3.32) with both the recursive and product forms above. We note that in the derivation of both of the above results, it was required that the variational conditions (3.28) hold. In Chapter 5, we will show how (3.37) and (3.38) can be used to bound the norm of the error propagator E.

The recursive form (3.37) was first exploited in [136], and has been used extensively in recent papers [24, 184]. A product form of the error propagator similar to (3.38) has been used in recent papers [26, 185], but a similar form was perhaps first noted in [147] for practical rather than theoretical considerations. Error propagators which have product forms occur naturally in multiplicative Schwarz domain decomposition methods [58], and in fact it has been shown recently that both multiplicative domain decomposition and multilevel methods can be viewed as particular instances of a general class of successive subspace decomposition and correction methods [185].

Our presentation here is a slight generalization of existing results, in that we have derived both of the forms (3.37) and (3.38) explicitly in terms of the prolongation and restriction operators that occur in multigrid implementations; in other words, these expressions can be interpreted completely in terms of matrices and the spaces $\mathcal{U}_k$ rather than operator recursions on the abstract spaces $\mathcal{H}_k$ or the finite element spaces $\mathcal{M}_k$. In the case of finite element spaces $\mathcal{M}_k$, the prolongation and restriction operators correspond

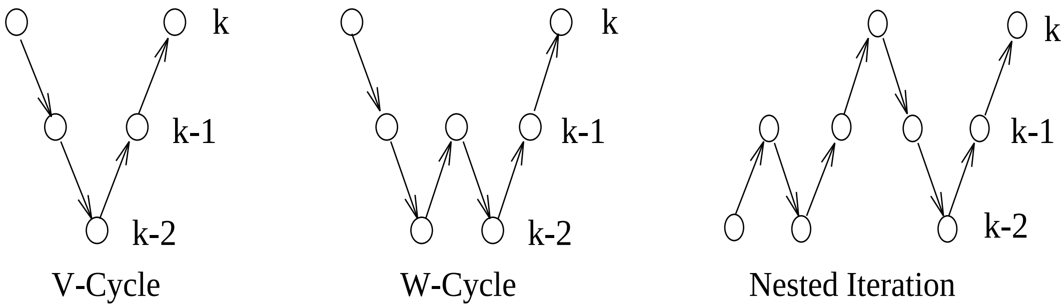


Figure 3.2: The V-cycle, the W-cycle, and nested iteration.

to the natural inclusion operation and the L^2-projection, respectively; in this case, the expressions above take on particularly simple forms which can be exploited to great advantage (see for example [185] for a discussion, or Chapter 5).

The symmetric forms of Algorithm 3.5 and Algorithm 3.6 are easily defined using this framework by inserting the pre-smoothing step as in the two-level case. It follows from the product form of the error propagator (3.38) that if the adjoint of R_k is used as the pre-smoothing operator, and if the variational conditions (3.28) are satisfied, then as in the two-level case we have that $E^s = EE^*$, where E^s is the error propagator of the symmetric multilevel algorithm. Therefore, it suffices to analyze E, the error propagator of the nonsymmetric multilevel algorithm. Again, this was first noted in [144].

3.2.6 The V-cycle, the W-cycle, and nested iteration

The methods we have just described are standard examples of *multigrid* or *multilevel methods* [85], where we have introduced a few restrictions for convenience, such as equal numbers of pre- and post-smoothings, one coarse space correction per iteration, and pre-smoothing with the adjoint of the post-smoothing operator. These restrictions are unnecessary in practice, but are introduced to make the analysis of the methods somewhat simpler, and to result in a symmetric preconditioner as required for combination with the conjugate gradient method.

The procedure just outlined involving correcting with the coarse space once each iteration is referred to as the *V-cycle* [29]. Another variation is termed the *W-cycle*, in which two coarse space corrections are performed per level at each iteration. More generally, the *p-cycle* would involve p coarse space corrections per level at each iteration for some integer $p \geq 1$. The *full multigrid method* [29] or *nested iteration technique* [85] begins with the coarse space, prolongates the solution to a finer space, performs a p-cycle, and repeats the process, until a p-cycle is performed on the finest level. The methods can be depicted as in Figure 3.2.

3.2.7 Convergence and complexity of multilevel methods

Multilevel methods first appeared in the Russian literature in [64]. In his 1961 paper Fedorenko described a two-level method for solving elliptic equations, and in a second paper from 1964 [65] proved convergence of a multilevel method for Poisson's equation on the square. Many theoretical results have been obtained since these first two papers. In short, what can be proven for multilevel methods under reasonable conditions is that the convergence rate or contraction number (usually the energy norm of the error propagator E^s) is bounded by a constant below one, independent of the meshsize and the number of levels, and hence the number of unknowns:

$$\|E^s\|_A \leq \delta_J < 1. \tag{3.39}$$

In more general situations (such as problems with discontinuous coefficients), analysis yields contraction numbers which decay as the number of levels employed in the method is increased; we will discuss these situations in Chapter 5.

If a tolerance of ϵ is required, then the computational cost to reduce the energy norm of the error below the tolerance can be determined from (3.6) and (3.39):

$$i \le \frac{|\ln \epsilon|}{|\ln \|E^s\|_A|} \le \frac{|\ln \epsilon|}{|\ln \delta_J|}.$$

The discretization error of $O(h_J^s)$ for some $s > 0$ yields a practical tolerance of $\epsilon = O(h_J^s)$. As remarked in §3.1.5, for a shape-regular and quasi-uniform mesh, the meshsize h_J is related to the number of discrete unknowns n_J through the dimension d of the spatial domain as $n_J = O(h_J^{-d})$. Assuming that $\delta_J < 1$ independently of J and h_J, we have that the maximum number of iterations i required to reach an error on the order of discretization error is:

$$i \le \frac{|\ln \epsilon|}{|\ln \delta_J|} = O(|\ln h_J|) = O(|\ln n_J^{-1/d}|) = O(\ln n_J). \tag{3.40}$$

Consider now that the operation count o_J of a single (p-cycle) iteration of Algorithm 3.5 with J levels is given by:

$$o_J = po_{J-1} + Cn_J = p(po_{J-2} + Cn_{J-1}) + Cn_J = \cdots = p^{J-1}o_1 + C\sum_{k=2}^{J} p^{J-k}n_k,$$

where we assume that the post-smoothing iteration has cost Cn_k for some constant C independent of the level k, and that the cost of a single coarse level correction is given by o_{k-1}. Now, assuming that the cost to solve the coarse problem o_1 can be ignored, then it is not difficult to show from the above expression for o_J that the computational cost of each multilevel iteration is $O(n_J)$ if (and only if) the dimensions of the spaces $\mathcal{H}_k$ satisfy:

$$n_{k_1} < \frac{C_1}{p^{k_2-k_1}} n_{k_2}, \qquad \forall k_1, k_2, \qquad k_1 < k_2 \le J,$$

where C_1 is independent of k. This implies both of the following:

$$n_k < \frac{C_1}{p} n_{k+1}, \qquad n_k < \frac{C_1}{p^{J-k}} n_J, \qquad k = 1, \ldots, J-1.$$

Consider the case of non-uniform Cartesian meshes which are successively refined, so that $h_{k_1} = 2^{k_2-k_1}h_{k_2}$ for $k_1 < k_2$, and in particular $h_{k-1} = 2h_k$. This gives

$$n_{k_1} = C_2 h_{k_1}^{-d} = C_2(2^{k_2-k_1}h_{k_2})^{-d} = C_2 2^{-d(k_2-k_1)}(C_3 n_{k_2}^{-1/d})^{-d} = \frac{C_2 C_3^{-d}}{(2^d)^{k_2-k_1}} n_{k_2}.$$

Therefore, if $2^{d(k_2-k_1)} > p^{k_2-k_1}$, or if $2^d > p$, which is true in two dimensions ($d = 2$) for $p \le 3$, and in three dimensions ($d = 3$) for $p \le 7$, then each multilevel iteration has complexity $O(n_J)$. In particular, one V-cycle ($p = 1$) or W-cycle ($p = 2$) iteration has complexity $O(n_J)$ for non-uniform Cartesian meshes in two and three dimensions.

If these conditions on the dimensions of the spaces are satisfied, so that each multilevel iteration has cost $O(n_J)$, then combining this with equation (3.40) implies that the overall complexity to solve the problem with a multilevel method is $O(n_J \ln n_J)$. By using the nested iteration, it is not difficult to show using an inductive argument (see for example §5.3 of [85]) that the multilevel method improves to optimal order $O(n_J)$ if $\delta_J < 1$ independent of J and h_J, meaning that the computational cost to solve to solve a problem with n_J pieces of data is Cn_J, for some constant C which does not depend on n_J. As we will discuss in more detail in Chapter 5, the theoretical multilevel studies first appearing in the West in the late 1970's and continuing up through the present have focussed on extending the proofs of optimality (or near optimality) to larger classes of problems.

3.3 Multilevel methods for discontinuous coefficients

In the case of elliptic problems with smooth coefficients, a red/black Gauss-Seidel smoothing method, transfer operators corresponding to linear interpolation, box-method discretization on all levels, and direct or iterative solution on the coarse level, combine to yield a very efficient method [104, 105].

However, in the case of interface problems occurring in reservoir simulation and reactor physics as well as in biophysics, the convergence rates of multilevel methods degrade drastically, and the methods may not converge at all. Numerous studies have appeared addressing this problem, most notably the studies by Alcouffe et al. [2], Dendy and Hyman [53], Dendy [48, 49, 50, 51, 52], and Behie and Forsythe [19, 20]. Numerical experiments indicate that forming the coarse equations by either the Galerkin approach (3.28) or a coefficient averaging technique, and coupling either of these with transfer operators which enforce continuity conditions across material interfaces (referred to as *operator-based prolongation*), leads to multilevel methods which regain their usual good convergence rates. Our interest here is to study and employ effectively some of these techniques for three-dimensional linear and nonlinear interface problems.

3.3.1 Interface problems: a one-dimensional example

Before discussing averaging, Galerkin, and operator-based prolongation methods, we first review the box-method discretization for a simple one-dimensional interface problem. Consider the following example, which will be used to explain each of these procedures:

$$-\frac{d}{dx}\left(a(x)\frac{d}{dx}u(x)\right)+b(x)u(x)=f(x) \text{ in } (c,d), \qquad u(c)=u(d)=0. \tag{3.41}$$

The functions $a(x)$ and $b(x)$ are positive for all x in $[c,d]$, and $a(x), b(x)$, and $f(x)$ are continuously differentiable everywhere, except that one or more of the three may be discontinuous at the *interface* point $x=\xi\in(c,d)$.

Define a discrete mesh $c=x_0<x_1<\ldots<x_{n+1}=d$, with $x_{i+1}=x_i+h_i$ for $h_i>0$, such that the point of discontinuity coincides with some mesh point $x_i=\xi$. Then the *integral method* (§6.2 in [179], also called the *box* or *finite volume method* in two or three dimensions) provides a reasonably rigorous technique for obtaining a discrete form of (3.41) at each mesh point x_i, despite the presence of the discontinuities. One considers the interval $[x_i-h_{i-1}/2, x_i+h_i/2]$ containing the point x_i, and integrates (3.41) over the interval. Let us denote the half-mesh points as $x_{i-1/2}=x_i-h_{i-1}/2$ and $x_{i+1/2}=x_i+h_i/2$. After performing the integration of the first term of (3.41) separately over the half-intervals $[x_{i-1/2},x_i]$ and $[x_i,x_{i+1/2}]$, and enforcing the continuity condition at the interface point $x_i=\xi$

$$\lim_{x\to x_i-} a(x)\frac{d}{dx}u(x)=\lim_{x\to x_i+} a(x)\frac{d}{dx}u(x), \tag{3.42}$$

the following expression is obtained, which is exact for the solution $u(x)$ in the interval:

$$\left(a(x_{i-1/2})\frac{d}{dx}u(x_{i-1/2})\right)-\left(a(x_{i+1/2})\frac{d}{dx}u(x_{i+1/2})\right)+\int_{x_{i-1/2}}^{x_{i+1/2}} b(x)u(x)dx=\int_{x_{i-1/2}}^{x_{i+1/2}} f(x)dx.$$

An algebraic expression is then obtained for an approximation $u_h(x_i)$ to $u(x_i)$ by replacing the derivatives with differences, and replacing the integrals with quadrature formulas separately over the half intervals.

Denoting the discretized functions as $u_h(x_i)$, we can for example write down an $O(h^2)$ (if $h=h_{i-1}=h_i$) approximation using centered differences and the rectangle rule:

$$a_h(x_{i-1/2})\left(\frac{u_h(x_i)-u_h(x_{i-1})}{h_{i-1}}\right)-a_h(x_{i+1/2})\left(\frac{u_h(x_{i+1})-u_h(x_i)}{h_i}\right)$$

$$+u_h(x_i)\left(\frac{h_{i-1}b_h(x_i^-)+h_ib_h(x_i^+)}{2}\right)=\left(\frac{h_{i-1}f_h(x_i^-)+h_if_h(x_i^+)}{2}\right). \tag{3.43}$$

All approximations are performed over intervals where the functions are smooth; therefore, error estimates from the difference and quadrature formulas are valid. The three-dimensional version of the box-method was discussed in detail in Chapter 2.

3.3.2 Coefficient averaging methods

From the previous discussion, it should be clear that if discontinuities in the equation coefficients lie along mesh lines and planes on all coarse meshes, then the standard box or finite element method discretization on all levels will produce accurate approximations. However, if the discontinuities are complex in shape, then the discontinuities may necessarily lie within individual elements on coarse meshes, resulting in poor coarse approximations and poor multilevel convergence rates.

One approach to handling this problem is to explicitly average the coefficients in the equation to produce a new problem with smoother coefficients, essentially *smearing* the interfaces so that their effect may be captured by discrete methods. The new problem is discretized on a coarser mesh, and the process is continued to produce discrete equations on a sequence of coarser meshes. These techniques are discussed in the studies of Alcouffe et al. [2] and Liu et al. [134] for two-dimensional problems.

For example, in our one-dimensional problem (3.41), the discrete equations (3.43) require that the function $a(x)$ be sampled at the *half*-mesh points $x_{i-1/2}$ and $x_{i+1/2}$. In multigrid implementations, coarse meshes are often constructed to be subsets of the next finer mesh, referred to as successively refined meshes or grids. In this situation, assume that the fine points x_{i-1} and x_{i+1} correspond to *adjacent* coarse points. For discretization on the coarse level, the function $a(x)$ must be sampled at the coarse level half-mesh point, which will correspond to the fine point x_i. Therefore, given the function values $a_h(x_{i-1/2})$ and $a_h(x_{i+1/2})$, we wish to produce a value $a_H(x_i)$ for use in the coarse discrete equations, such that $a_H(x_i)$ in some sense represents the discontinuity in $a(x)$ at x_i.

Using electrical network arguments and noting a connection to homogenization theory, Alcouffe et al. [2] suggest various combinations of the *harmonic* and *arithmetic* averages

$$a_H(x_i) = HARM(a_h(x_{i-1/2}), a_h(x_{i+1/2})), \qquad HARM(x,y) = \frac{2xy}{x+y}, \qquad ARITH(x,y) = \frac{x+y}{2},$$

to represent the coefficients across interfaces. The sequence of graphs in Figure 3.3 shows the different effects of four successive arithmetic and harmonic averagings of the coefficient $a(x)$ on four successively coarser meshes, where $a(x)$ is piecewise constant and defined as:

$$a(x) = \left\{ \begin{array}{c} 1, \text{ if } 0 < x < 1 \\ 10, \text{ if } 1 \le x \le 2 \\ 1, \text{ if } 2 < x < 3 \end{array} \right\}.$$

The discontinuities at $x = 1$ and $x = 2$ lie on mesh lines only on the finest of the four meshes, taken to have 80 mesh points. The coarser meshes have 40, 20, 10, and 5 points each. The analogous averagings are shown for two-dimensions in Figure 3.4 and Figure 3.5.

While the harmonic average appears to preserve more correctly the effect of the discontinuity in both one and two dimensions, it is difficult to tell from these graphs which of the two approaches would be more effective at producing equations with good coarse level approximation properties. We will justify our preference for the harmonic average by showing its equivalence to the Galerkin approach in certain situations below, and by showing its effectiveness for problems with discontinuities of the type occurring in the linearized Poisson-Boltzmann equation with a series of numerical experiments in Chapter 6.

Consider now the two-dimensional problem:

$$-\nabla \cdot (\bar{\mathbf{a}}(\mathbf{x}) \nabla u(\mathbf{x})) + b(\mathbf{x}) u(\mathbf{x}) = f(\mathbf{x}) \text{ in } \Omega \subset \mathbb{R}^2, \qquad u(\mathbf{x}) = 0 \text{ on } \Gamma,$$

where now $\mathbf{x} = (x, y)$, and where the tensor $\bar{\mathbf{a}} : \Omega \mapsto \mathbf{L}(\mathbb{R}^2, \mathbb{R}^2)$ has the diagonal form:

$$\bar{\mathbf{a}}(\mathbf{x}) = \left(\begin{array}{cc} a^{(11)}(\mathbf{x}) & 0 \\ 0 & a^{(22)}(\mathbf{x}) \end{array} \right).$$

A box-method discretization of this problem on a non-uniform Cartesian mesh (see Chapter 2) requires evaluation of the coefficient $a^{(11)}$ on the fine mesh along y-mesh-lines, but midway between x-mesh-points; we denote this discretized coefficient as $a_h^{(11)}$. Similarly, the coefficient $a^{(22)}$ requires evaluation on the fine mesh along x-mesh-lines, but midway between y-mesh-points; we denote this discretized coefficient as $a_h^{(22)}$.

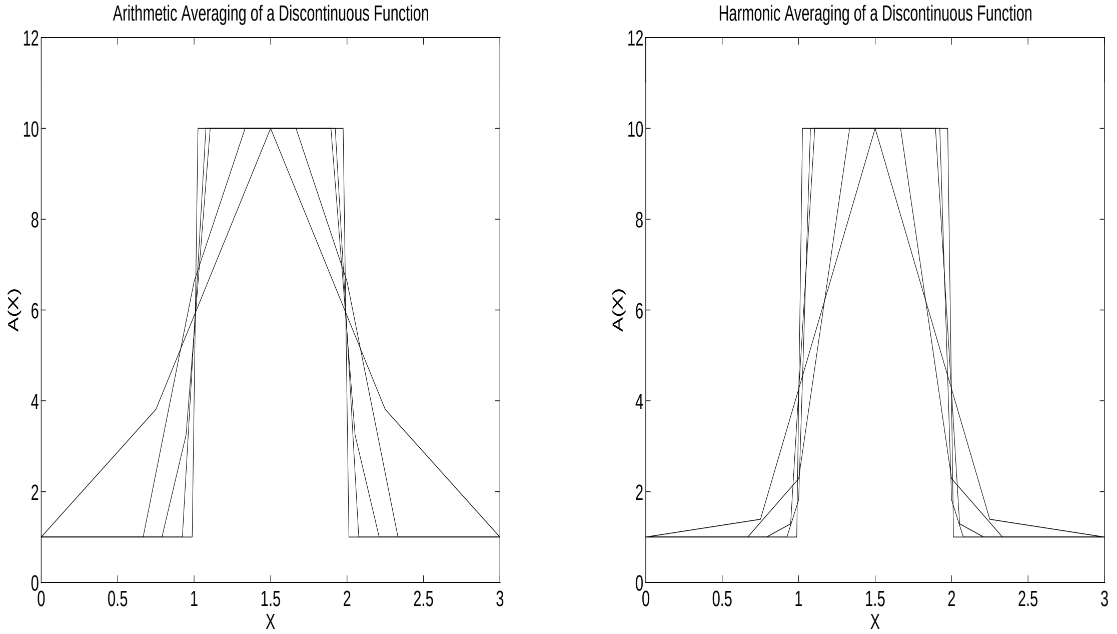


Figure 3.3: Arithmetic and harmonic averaging in one dimension.

In [2], the averaging approach is discussed in detail for two-dimensional problems, where the harmonic average across interfaces is combined with arithmetic averages in the second grid direction to produce coefficients $a_H^{(11)}$ and $a_H^{(22)}$ for coarser levels; one of several possible schemes is depicted in Figure 3.6.

In Figure 3.6, the black circles represent the fine mesh points, the larger open circles represent the coarse mesh points, the small open squares represent the half-mesh-points at which the fine mesh coefficients $a_h^{(22)}$ are located, the small open triangles represent the half-mesh-points at which the fine mesh coefficients $a_h^{(11)}$ are located, the large open squares are the new coarse mesh coefficients $a_H^{(22)}$, and the large open triangles are the new coarse mesh coefficients $a_H^{(11)}$. The straight arrows indicate a harmonic average of two points, and the curved arrows represent the arithmetic averaging of three values which were produced by harmonic averaging. Discontinuities are assumed to lie along the mesh lines, so that the harmonic average is taken across the possible discontinuity.

The numerical experiments in [2] for two-dimensional problems indicate that this technique is effective for representing the effect of discontinuities on coarse meshes for many types of interface problems, and when combined with the prolongation operators discussed below results in effective multilevel methods. This approach is easily extended to three dimensions in the obvious way, and our experiments [107] indicate that it is a very effective technique for the discontinuities that occur in the linearized Poisson-Boltzmann equation and similar equations. We will present experiments with this approach in Chapter 6.

Note that this approach requires little extra computation over a standard discretization on coarse levels, and in the three-dimensional case results in seven-point stencils on all coarse levels if a box-method on a non-uniform Cartesian mesh is used. However, a serious flaw in this approach is that there are no proof techniques available for analyzing the convergence properties of multilevel methods with coefficient averaging of discontinuous coefficients (in fact, it is easy to construct a sample problem with large discontinuity jumps for which this approach will lead to a divergent method). Analysis techniques (see for example [85] for a discussion) which allow violation of the variational conditions (3.28) require elliptic regularity assumptions which are not available for problems with discontinuous coefficients. We will discuss the regularity-free techniques employing the variational conditions more fully in Chapter 5.

While we cannot analyze the averaging methods using the available theoretical tools, we will numerically

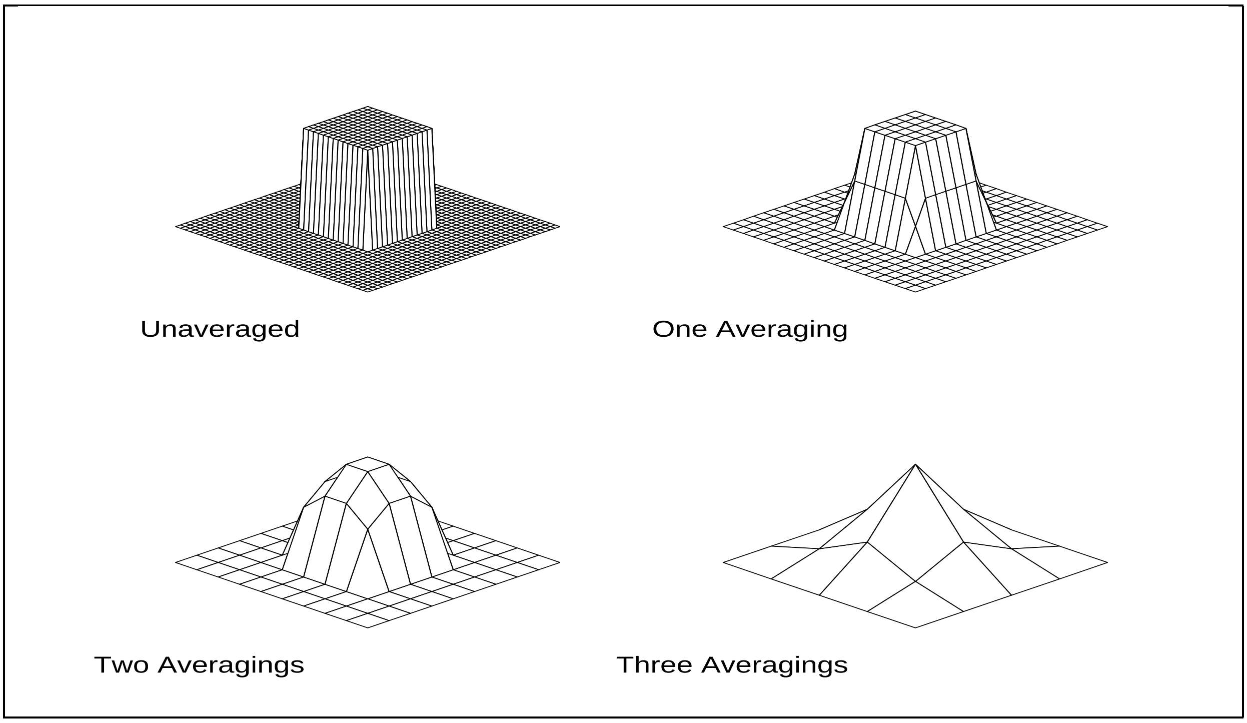


Figure 3.4: Arithmetic averaging in two dimensions.

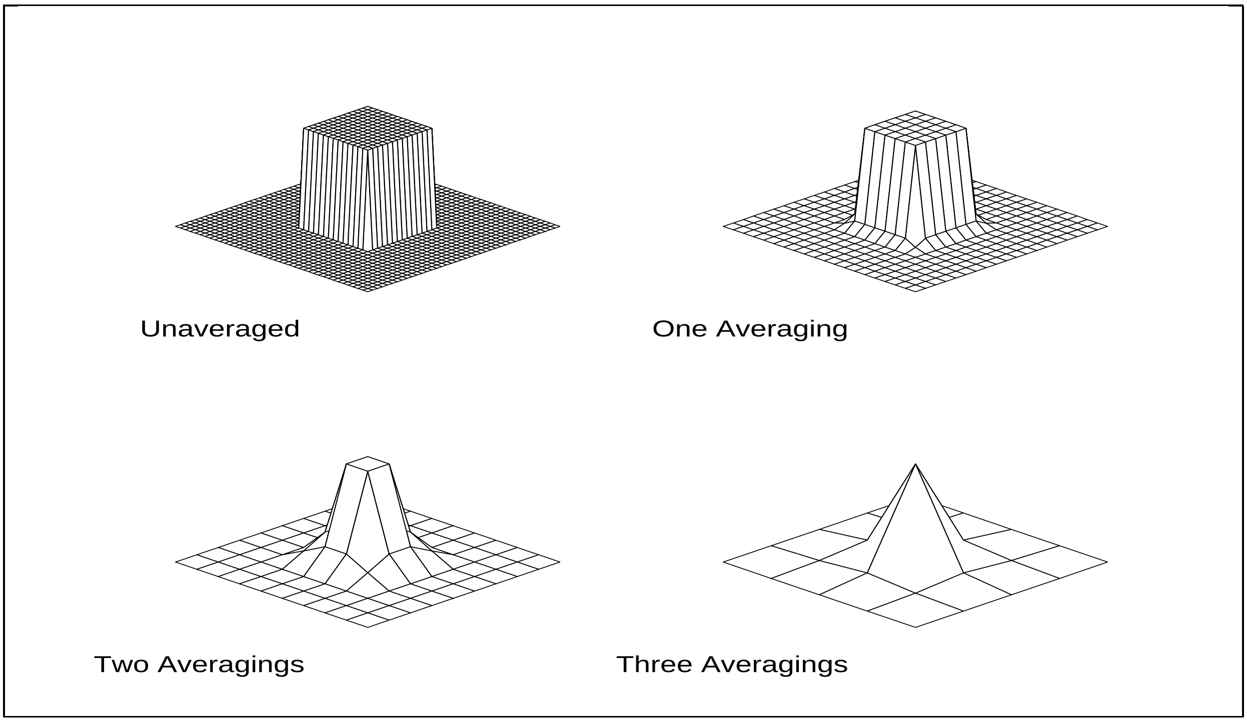


Figure 3.5: Harmonic averaging in two dimensions.

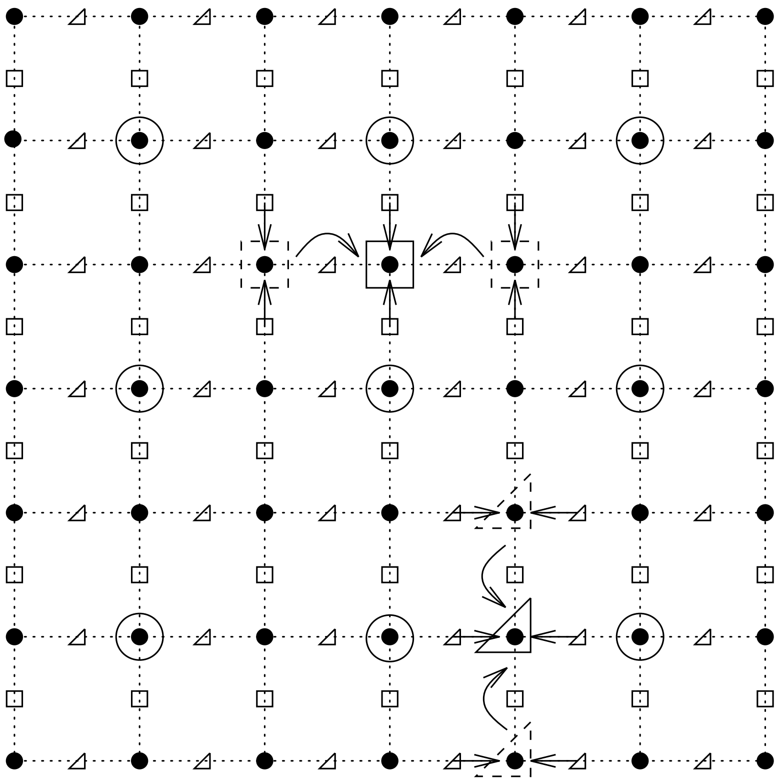


Figure 3.6: Combination of arithmetic and harmonic averaging in two dimensions.

investigate the effectiveness of these methods for three-dimensional linear and nonlinear interface problems in Chapter 6 and Chapter 7 with a series of experiments.

3.3.3 Algebraic Galerkin methods

The *Galerkin approach* to forming the coarse level equations is the algebraic enforcement of the variational conditions (3.28). This approach provides an algebraic mechanism in which the fine level equation coefficients are averaged to produce the coarse level equation coefficients, where the averaging is performed by the prolongation and restriction operators.

It should be noted that this approach is difficult to implement and computationally expensive in three dimensions, since seven-point stencils produced by the box-method on a non-uniform Cartesian fine mesh expand to twenty-seven point stencils on all coarser meshes when standard transfer operators are used (this is easily shown using the stencil calculus which we outline below and in Appendix A). This results in more extensive set-up computations, as well as more expensive coarse level computations during the iteration itself due to the less sparse matrices on the coarse levels.

While it is more costly, this technique for improving the coarse level approximation properties may be the preferred one in many situations for several reasons. First, it is non-heuristic, in that once the prolongation operator is selected, the coarse mesh equations are constructed automatically without the need to specify an averaging scheme. Second, it is known that this approach is always convergent (we present our own proof in Chapter 5). The study by Dendy [51] presents a detailed empirical investigation of the convergence properties of these methods for a set of very difficult test problems. This excellent study clearly established the effectiveness of these methods for three-dimensional problems, as the earlier studies [2, 53] had for two-dimensional problems. Unfortunately, to the author's knowledge, there are no publicly available three-dimensional multilevel codes using the Galerkin approach; in fact, it appears that no details have been published on how to obtain the Galerkin coarse-level matrix expressions (at least in the three-dimensional case). Explicit expressions for the Galerkin coefficients have appeared in the literature only for the two-dimensional case, and these expressions were only for symmetric matrices and a particular prolongation operator (the expressions appeared in the Appendix of [19]).

Note that the variational conditions are satisfied naturally by successively refined finite element discretizations, but only with *exact* evaluation of the integrals forming the components of the stiffness matrices. If quadrature is used to approximate the integrals, then the variational conditions will be violated. Goldstein [74] has investigated the effect of numerical integration on multilevel convergence, and has given conditions on the accuracy of the quadrature formula which will insure convergence. If discontinuities occur within elements, several problems occur and the approach in [74] cannot be used. Therefore, even with a finite element discretization on the fine mesh, if discontinuities occur within elements on coarser levels, then the variational conditions must be imposed algebraically in order to guarantee convergence.

In order to explain clearly how the variational conditions (3.28) can be imposed algebraically in a reasonably efficient manner for a certain class of problems, we will first introduce a stencil calculus for the computation of general matrix-matrix products in an efficient way. This calculus was first developed by R. Falgout, and he has outlined the calculus in detail for one- and two-dimensional problems in his thesis [63]. Although our notation below is somewhat different, our approach is based on his very clear presentation of the one- and two-dimensional cases. One limitation of the calculus is that matrices must be representable as stencils, implying that there must be an underlying, logically non-uniform Cartesian mesh. Note that this is not really a restriction for many problems, as *virtual points* may be added to the set of unknowns with a corresponding "identity" stencil for that mesh point.

3.3.4 Stencil calculus for computing the Galerkin equations

We will be concerned here only with the two-level case, and to avoid confusion with matrix entries which will appear, we will denote fine and coarse level spaces, grids, matrices and functions with the subscripts h and H, respectively, rather than k and $k-1$. With this minor change in notation, we will use the grid function space notation outlined in §3.1.5 for matrix equations arising from partial differential equations.

The main ideas in the stencil calculus for producing the Galerkin equations can be explained most clearly by considering first the one-dimensional case with only two levels, and then extending the key ideas to higher dimensions and more levels. Therefore, we once again consider the one-dimensional example (3.41), and assume we have discretized the equation with the box-method producing the discrete equations (3.43) for each mesh point $u_h(x_i),\ i=1,\ldots,n$. For simplicity, we will assume in the following discussion that the coefficient $b(x)$ in (3.41) is zero.

Recalling the standard approach of eliminating the Dirichlet boundary points from the set of unknowns, we see that all of the equations are identical in form, except that the two equations which border the left and right boundaries will be decoupled from the boundary points, with corresponding modifications to the right hand side entries (in this case, with zero Dirichlet conditions, the right hand side entries remain unchanged). Taking all of the equations (3.43) together for the special case that $n=7$, and ordering the unknowns $u_h(x_i)$ from $i=1$ to $i=7$ consecutively, we produce the linear algebraic system $A_h u_h = f_h$, which has the tridiagonal form:

$$\left[\begin{array}{ccccccc} C_1 & -E_1 & & & & & \\ -W_2 & C_2 & -E_2 & & & & \\ & -W_3 & C_3 & -E_3 & & & \\ & & -W_4 & C_4 & -E_4 & & \\ & & & -W_5 & C_5 & -E_5 & \\ & & & & -W_6 & C_6 & -E_6 \\ & & & & & -W_7 & C_7 \end{array}\right] \left[\begin{array}{c} u_h(x_1) \\ u_h(x_2) \\ u_h(x_3) \\ u_h(x_4) \\ u_h(x_5) \\ u_h(x_6) \\ u_h(x_7) \end{array}\right] = \left[\begin{array}{c} f_h(x_1) \\ f_h(x_2) \\ f_h(x_3) \\ f_h(x_4) \\ f_h(x_5) \\ f_h(x_6) \\ f_h(x_7) \end{array}\right].$$

The equations in the above system represent equation (3.43) for each unknown grid function value; for the single unknown value $u_h(x_i)$, the equation is explicitly:

$$\left[-\frac{a_h(x_{i-1/2})}{h_{i-1}} \quad \left(\frac{a_h(x_{i-1/2})}{h_{i-1}} + \frac{a_h(x_{i+1/2})}{h_i}\right) \quad -\frac{a_h(x_{i+1/2})}{h_i} \right] \left[\begin{array}{c} u_h(x_{i-1}) \\ u_h(x_i) \\ u_h(x_{i+1}) \end{array}\right]$$

$$= \left[\frac{h_{i-1}\tilde{f}_h(x_i^-) + h_i\tilde{f}_h(x_i^+)}{2} \right]. \tag{3.44}$$

If we establish the convention that a term in the above *stencil* expression is identically zero if its corresponding index satisfies $i \leq 0$ or $i \geq n+1$, meaning that the stencil touches or goes outside the boundary of the discretized domain, then we can represent all of the equations with the above *stencil representation* of equation (3.43). For simplicity, we will write this *stencil equation* symbolically as:

$$\left[\begin{array}{ccc} -W_i & C_i & -E_i \end{array} \right]_h^h \left[\begin{array}{c} u_h(x_{i-1}) \\ u_h(x_i) \\ u_h(x_{i+1}) \end{array} \right] = [f_h(x_i)], \tag{3.45}$$

where the subscript h on the stencil represents the "domain" of the stencil, and the superscript h represents the "range" of the stencil. In this case, both simply the fine mesh function space $\mathcal{U}_h$. We can represent any tri-diagonal system of any dimension using the above stencil notation, although our discretization yields two special properties, namely $W_i = E_{i-1}$ (symmetry), and $C_i = W_i + E_i$ for all points not lying next to a boundary point.

The functions $u_h \in \mathcal{U}_h$ and $f_h \in \mathcal{U}_h$ are interpreted as *grid functions*, and the matrix A_h is a *grid function operator* which we represent using the *stencil form*:

$$A_h = \left[\begin{array}{ccc} -W_i & C_i & -E_i \end{array} \right]_h^h.$$

The operation of A_h on a grid function u_h is calculated by centering the stencil of A_h over each component of the grid function u_h and simply applying the stencil (multiplying each component of the stencil with the corresponding component of the grid function, and adding the results). Therefore, we can compute the action of the matrix A_h on the vector u_h by considering the action of the stencil for A_h on u_h interpreted as a grid function.

Given the matrix A_h and its stencil representation, along with the prolongation matrix I_H^h and its corresponding stencil, we are interested in computing the Galerkin coarse matrix $A_H = (I_H^h)^T A_h I_H^h$ (or equivalently its stencil representation). We will see shortly that the stencil for the coarse mesh system matrix A_H will have the form:

$$A_H = \left[\begin{array}{ccc} -W_i^H & C_i^H & -E_i^H \end{array} \right]_H^H,$$

where the domain and range are the coarse mesh function space $\mathcal{U}_H$.

First we consider the form of the prolongation matrix. With the fine mesh $\Omega_h = \{x_1, \ldots, x_n\}$ where n is odd, it is standard in both box-method-based and finite element-based multilevel methods to employ successively refined meshes, in which the points of the coarse mesh form a subset of the fine mesh points; in this case, the coarse mesh points will consist of the even-numbered fine mesh points, or $\Omega_H = \{x_2, \ldots, x_{2i}, \ldots, x_{n-1}\}$. To *prolongate* a coarse mesh function $u_H = [u_H(x_2), \ldots, u_H(x_{2i}), \ldots, u_H(x_{n-1})]^T \in \mathcal{U}_H$ to the fine mesh function $u_h = [u_H(x_1), \ldots, u_H(x_n)]^T \in \mathcal{U}_h$, we will employ two separate rules:

$$u_h(x_i) = \left\{ \begin{array}{ll} i \text{ even}: & PC_i u_H(x_i) \\ i \text{ odd}: & PE_{i-1}u_H(x_{i-1}) + PW_{i+1}u_H(x_{i+1}) \end{array} \right\}.$$

For the case of $n = 7$, this prolongation operation can be represented in matrix form as $u_h = I_H^h u_H$, or as:

$$\left[\begin{array}{c} u_h(x_1) \\ u_h(x_2) \\ u_h(x_3) \\ u_h(x_4) \\ u_h(x_5) \\ u_h(x_6) \\ u_h(x_7) \end{array} \right] = \left[\begin{array}{ccc} PW_2 & & \\ PC_2 & & \\ PE_2 & PW_4 & \\ & PC_4 & \\ & PE_4 & PW_6 \\ & & PC_6 \\ & & PE_6 \end{array} \right] \left[\begin{array}{c} u_H(x_2) \\ u_H(x_4) \\ u_H(x_6) \end{array} \right].$$

Since there are two special cases, we must represent the stencil for this operation in a slightly different way than the simple case represented by A_h. The following is a modification of a notational convenience introduced by R. Falgout [63] to represent these types of matrices as *composite stencils*:

$$I_H^h = \left[\begin{array}{ccc} PE_{i-1} & 0 & PW_{i+1} \end{array} \right]_{H(h)}^h \vee \left[\begin{array}{c} PC_i \end{array} \right]_{H(H)}^h.$$

The operation of I_H^h on a grid function u_h is calculated by centering the stencil of I_H^h over the coarse grid function u_H as seen on the fine mesh, i.e., to the fine mesh function constructed by *injecting* the coarse mesh function into the zero fine mesh function, producing:

$$v_h = [\ 0\ ,\ u_H(x_2)\ ,\ 0\ ,\ u_H(x_4)\ ,\ 0\ ,\ \dots\ ,\ 0\ ,\ u_H(x_{2i})\ ,\ 0\ ,\ \dots\ ,\ 0\ ,\ u_H(x_{n-1})\ ,\ 0\ ]^T.$$

The stencil of I_H^h is centered over each component of v_h, and the stencil is simply applied, taking into account whether to apply rule one or rule two. If the stencil is centered over a coarse mesh point, then rule one is applied (indicated by the subscript $H(H)$). If the stencil is centered over a fine mesh point not corresponding to a coarse mesh point, then rule two is applied (indicated by the subscript $H(h)$).

The application of the restriction operator $u_H = I_h^H u_h$, which we take to be the transpose of the prolongation operator $I_h^H = (I_H^h)^T$, is for the special case of $n = 7$ as follows:

$$\left[\begin{array}{c} u_H(x_2) \\ u_H(x_4) \\ u_H(x_6) \end{array}\right] = \left[\begin{array}{ccccccc} PW_2 & PC_2 & PE_2 & & & & \\ & & PW_4 & PC_4 & PE_4 & & \\ & & & & PW_6 & PC_6 & PE_6 \end{array}\right] \left[\begin{array}{c} u_h(x_1) \\ u_h(x_2) \\ u_h(x_3) \\ u_h(x_4) \\ u_h(x_5) \\ u_h(x_6) \\ u_h(x_7) \end{array}\right].$$

We see that the restriction operator can be represented as the single stencil

$$I_h^H = \left[\ PW_i \quad PC_i \quad PE_i\ \right]_{h(H)}^H,$$

if we take the convention that this stencil is only applied when centered over fine mesh points which coincide with coarse mesh points (hence the use of the subscript $h(H)$ to represent this stencil).

To summarize, the rules for the stencil calculus on two successively refined meshes are as follows:

(1) All stencils operate only on fine mesh functions.

(2) If coarse mesh functions are involved, they are first injected to the fine mesh.

(3) A fine mesh system matrix stencil operates on each entry of a fine mesh function.

(4) A prolongation matrix stencil operates on each entry of the fine mesh function (the injection of a coarse mesh function), taking into account which rule to apply.

(5) A restriction matrix stencil operates only on the fine mesh function components which correspond to a coarse mesh component.

Given the above rules, we are now in a position to compute the general product $A_H = (I_H^h)^T A_h I_H^h$ using only the stencil entries. To see how this is possible, consider the product $A_H e_H = (I_H^h)^T A_h I_H^h e_H$, were e_H is the unit grid function $e_H = [0, \dots, 0, 1, 0, \dots, 0]^T$ having the value of unity at mesh point x_i. The product $A_H e_H$ is simply the inner-product of all of the *rows* of A_H with the vector e_H, which will extract the $i-th$ *column* of the matrix A_H. We can calculate the i-th column as a grid function, by applying the stencils for each of the operators forming A_H successively to the grid function e_H. Since the result can be viewed as a *column stencil* for A_H, or a *row* stencil for A_H^T, we can then compute the final row stencil for A_H by applying the row stencil of A_H^T to the grid function e_H.

For our one-dimensional example, the calculation proceeds as follows. We first collect the expressions for the stencils together:

$$A_h = \left[\ -W_i \quad C_i \quad -E_i\ \right]_h^h, \qquad I_h^H = \left[\ PW_i \quad PC_i \quad PE_i\ \right]_{h(H)}^H, \tag{3.46}$$

$$I_H^h = [\ PE_{i-1} \quad 0 \quad PW_{i+1}\ ]_{H(h)}^h \vee [\ PC_i\ ]_{H(H)}^h, \qquad i = 1, \dots, n.$$

Applying the prolongation stencil to the unit grid function yields:

$$I_H^h e_H = [\ 0\ ,\ \dots\ ,\ 0\ ,\ PW_i\ ,\ PC_i\ ,\ PE_i\ ,\ 0\ ,\ \dots\ ,\ 0\ ]^T,$$

which is simply a column of the interpolation operator, or a row of its transpose, which is the restriction operator. Applying now the system matrix stencil yields:

$$A_h I_H^h e_H = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ -E_{i-2} PW_i \\ C_{i-1} PW_i - E_{i-1} PC_i \\ C_i PC_i - W_i PW_i - E_i PE_i \\ C_{i+1} PE_i - W_{i+1} PC_i \\ -W_{i+2} PE_i \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

Finally, applying the restriction operator gives a column of A_H:

$$I_h^H A_h I_H^h e_H = [\ 0\ ,\ \ldots\ ,\ 0\ ,\ -E_{i-2}^H\ ,\ C_i^H\ ,\ -W_{i+2}^H\ ,\ 0\ ,\ \ldots\ ,\ 0\ ]^T,$$

where

$$\begin{aligned} C_i^H &= (PC_i)^2 C_i + (PW_i)^2 C_{i-1} + (PE_i)^2 C_{i+1} \\ &\quad - PW_i PC_i E_i - PE_i PC_i W_{i+1} - PW_i W_i - PE_i E_i, \\ -E_{i-2}^H &= PE_{i-2} PW_i C_{i-1} - PC_{i-2} PW_i E_{i-2} - PC_i PE_{i-2} E_{i-1}, \\ -W_{i+2}^H &= PW_{i+2} PE_i C_{i+1} - PC_{i+2} PE_i W_{i+2} - PC_i PW_{i+2} W_{i+1}. \end{aligned}$$

Since this is a *column* of A_H, the *row stencil* for A_H^T is:

$$A_H^T = \left[\ -E_{i-2}^H \quad C_i^H \quad -W_{i+2}^H\ \right]_H^H.$$

Applying this stencil to the unit grid function e_H yields the final expression for the row stencil of A_H:

$$A_H = \left[\ -W_i^H \quad C_i^H \quad -E_i^H\ \right]_H^H, \qquad i = 2, 4, 6, \ldots, n-1, \tag{3.47}$$

where

$$\begin{aligned} C_i^H &= (PC_i)^2 C_i + (PW_i)^2 C_{i-1} + (PE_i)^2 C_{i+1} \\ &\quad - PW_i PC_i E_{i-1} - PE_i PC_i W_{i+1} - PW_i W_i - PE_i E_i, \\ -E_i^H &= PE_i PW_{i+2} C_{i+1} - PC_i PW_{i+2} E_i - PC_{i+2} PE_i E_{i+1}, \\ -W_i^H &= PW_i PE_{i-2} C_{i-1} - PC_i PE_{i-2} W_i - PC_{i-2} PW_i W_{i-1}. \end{aligned}$$

To verify this result, for the simple case $n = 7$ we can compute the product directly:

$$A_H = I_h^H A_h I_H^h = (I_H^h)^T A_h I_H^h$$

$$= \begin{bmatrix} C_2^H & -E_2^H & \\ -W_4^H & C_4^H & -E_4^H \\ & -W_6^H & C_6^H \end{bmatrix} = \begin{bmatrix} PW_2 & PC_2 & PE_2 & & & & \\ & & PW_4 & PC_4 & PE_4 & & \\ & & & & PW_6 & PC_6 & PE_6 \end{bmatrix} \cdot$$

$$\begin{bmatrix} C_1 & -E_1 & & & & & \\ -W_2 & C_2 & -E_2 & & & & \\ & -W_3 & C_3 & -E_3 & & & \\ & & -W_4 & C_4 & -E_4 & & \\ & & & -W_5 & C_5 & -E_5 & \\ & & & & -W_6 & C_6 & -E_6 \\ & & & & & -W_7 & C_7 \end{bmatrix} \cdot \begin{bmatrix} PW_2 & & \\ PC_2 & & \\ PE_2 & PW_4 & \\ & PC_4 & \\ & PE_4 & PW_6 \\ & & PC_6 \\ & & PE_6 \end{bmatrix}.$$

It is easily verified that the entries of the resulting matrix are identical to those produced using the stencil calculus in equation (3.47).

Note that this calculus extends to two and three dimensions easily, where we must now introduce the restriction that the grid functions are defined over a logically non-uniform Cartesian mesh. However, the expressions for the Galerkin coarse level operator stencil entries become extremely complex, even in the two-dimensional case. To assist in the calculation of these expressions, we have written a set of MATHEMATICA and MAPLE routines implementing the stencil calculus in two and three dimensions. These routines are useful for computing the Galerkin coarse level matrix entries symbolically from the fine level matrix entries and the prolongation matrix entries.

The symbolic manipulation routines were used to produce the expressions for the Galerkin coarse matrix entries in our three-dimensional linear and nonlinear multilevel methods as an alternative to the coefficient averaging approach described in the previous section. Extensive numerical experiments with both the Galerkin and coefficient averaging methods are presented in Chapter 6. In Appendix A, we have given the explicit expressions for Galerkin coarse level matrix stencils for the one, two, and three-dimensional cases, since these expressions do not appear to have been widely published (the two-dimensional expressions for a special case have appeared in the Appendix of [19]).

3.3.5 Operator-based prolongation and stencil compression

These techniques were originally developed in [2] and used extensively in [19, 20, 48, 49, 50, 53], and more recently in [51, 52]. They can be explained by considering again our example (3.41). With two successively refined meshes, assume we are given a coarse level function at points which correspond to the fine level points x_{i-1} and x_{i+1}, and we wish to *prolongate* (or *interpolate*) the coarse level function to the fine mesh points x_{i-1}, x_i, and x_{i+1}.

For the fine points x_{i-1} and x_{i+1} which correspond to coarse points, we can take the values of the new fine level function to be equal to the coarse level function, referred to as *injection.* To obtain the fine level function value at the point x_i not coincident with a coarse point, a standard linear interpolation can be used:

$$u_h(x_i) = \left(\frac{h_{i-1}}{h_{i-1}+h_i}\right) u_h(x_{i-1}) + \left(\frac{h_i}{h_{i-1}+h_i}\right) u_h(x_{i+1}).$$

In other words, we define the the prolongation operator as:

$$I_H^h = [\ PE_{i-1} \quad 0 \quad PW_{i+1}\ ]_{H(h)}^h \vee [\ PC_i\ ]_{H(H)}^h,$$

$$PC_i = 1, \qquad PE_{i-1} = \frac{h_{i-1}}{h_{i-1}+h_i}, \qquad PW_{i+1} = \frac{h_i}{h_{i-1}+h_i}. \tag{3.48}$$

On the other hand, in the case that the new point x_i is an interface point, we would like to impose the continuity condition (3.42). We can approximate this by imposing:

$$a_h(x_{i-1/2}) \left(\frac{u_h(x_i) - u_h(x_{i-1})}{h_{i-1}}\right) = a_h(x_{i+1/2}) \left(\frac{u_h(x_{i+1}) - u_h(x_i)}{h_i}\right).$$

Solving for $u_h(x_i)$ gives the more general prolongation formula:

$$u_h(x_i) = \left(\frac{a_h(x_{i-1/2})/h_{i-1}}{a_h(x_{i-1/2})/h_{i-1} + a_h(x_{i+1/2})/h_i}\right) u_h(x_{i-1})$$
$$+ \left(\frac{a_h(x_{i+1/2})/h_i}{a_h(x_{i-1/2})/h_{i-1} + a_h(x_{i+1/2})/h_i}\right) u_h(x_{i+1}),$$

which reduces to (3.48) in the case that $a_h(x_{i-1/2}) = a_h(x_{i+1/2})$. In this case, with again $PC_i = 1$, the prolongation operator is defined as follows:

$$PE_{i-1} = \frac{a_h(x_{i-1/2})/h_{i-1}}{a_h(x_{i-1/2})/h_{i-1} + a_h(x_{i+1/2})/h_i}, \qquad PW_{i+1} = \frac{a_h(x_{i+1/2})/h_i}{a_h(x_{i-1/2})/h_{i-1} + a_h(x_{i+1/2})/h_i}.$$

This type of prolongation can be extended to two and three dimensions in a number of ways [2, 85, 134]. First, note that from the definition of the fine grid matrix stencil (3.44) and (3.45), an equivalent representation of this prolongation is as follows:

$$PC_i = 1, \qquad PE_{i-1} = \frac{W_i}{C_i}, \qquad PW_{i+1} = \frac{E_i}{C_i}. \tag{3.49}$$

In other words, an alternative procedure for producing the more general prolongation formula (3.49) is by solving the i^{th} equation of the system $A_k u_k = 0$. The coefficients in the prolongation rule then come from the discrete stencil for the i^{th} equation of A_k, which in the one-dimensional example became (3.49).

The difficulty with this approach in dimensions higher than one is that the resulting prolongation formula for the center stencil point involves not only coarse points, but as yet undefined fine points as well, unless the meshes are defined in a non-standard fashion (see [63, 85] for examples). This difficulty can be avoided with standard successively refined meshes through *stencil compression*, an idea perhaps originating in [2].

We now describe this in three dimensions, essentially as employed in [51]. One must consider four fine mesh point types in the prolongation procedure:

(1) Fine points coincident with coarse mesh points.
(2) Fine points lying on a coarse mesh line but not of Type (1).
(3) Fine points lying on a coarse mesh plane not of Type (1) or Type (2).
(4) Fine points points not on a coarse mesh line or plane.

Assume now that our stencil is a full twenty-seven point stencil. Injection is used for Type (1) points. For Type (2) points, dependencies in the discrete stencil corresponding to directions not on the coarse mesh line are removed by *compressing* the three-dimensional stencil to a one-dimensional stencil (by simply summing the entries), producing a two-point prolongation formula, as in the one-dimensional case (3.49). An eight-point prolongation formula for Type (3) points results by summing away dependencies (*compressing*) in the direction not coincident with a coarse mesh plane. Type (4) points will require all twenty-six surrounding points in the prolongation formula.

Note that if the prolongation is performed in the order Type (1) → Type (4), then all computations involve only fine mesh quantities that have been previously computed by the preceding prolongation formulas. Alternatively, the prolongation weights can be pre-computed following the same procedure above, and used exactly as a standard interpolation; this approach will be necessary for Galerkin methods where the prolongation stencil entries must be available to compute the Galerkin coarse grid matrix. In Appendix A, we give the complete expressions for pre-computed operator-based prolongation stencil weights in the one-, two-, and three-dimensional cases, based on the idea of *stencil compression* as discussed above.

Even if the Galerkin approach is not used, it is common to take the restriction operator to be $I_k^{k-1} = (I_{k-1}^k)^T$, the *adjoint* of the prolongation operator with respect to the inner-product (3.16), where d is the dimension of the problem, and I_{k-1}^k is the d-dimensional version of either the standard interpolation (3.48) or the operator-based interpolation (3.49).

The effectiveness of operator-based prolongations through stencil compression, as compared to standard linear prolongations, will be evaluated in detail numerically in Chapter 6.

3.3.6 Equivalence of averaging and algebraic Galerkin methods

Consider the stencils (3.46) and linear prolongation (3.48) on a uniform mesh:

$$A_h = \left[\begin{array}{ccc} -W_i & C_i & -E_i \end{array}\right]_h^h, \qquad I_h^H = \left[\begin{array}{ccc} \frac{1}{2} & 1 & \frac{1}{2} \end{array}\right]_{h(H)}^H,$$

where we recall our box-method discretization stencil components:

$$-W_i = -\frac{a_h(x_{i-1/2})}{h}, \qquad -E_i = -\frac{a_h(x_{i+1/2})}{h}, \qquad C_i = W_i + E_i.$$

This combination yields as the Galerkin coarse matrix stencil (3.47):

$$A_H = \left[\begin{array}{ccc} -W_i^H & C_i^H & -E_i^H \end{array}\right]_H^H,$$

where

$$\begin{aligned}
C_i^H &= C_i + \frac{C_{i-1}}{4} + \frac{C_{i+1}}{4} - \frac{E_{i-1}}{2} - \frac{W_{i+1}}{2} - \frac{W_i}{2} - \frac{E_i}{2}, \\
-E_i^H &= \frac{C_{i+1}}{4} - \frac{E_i}{2} - \frac{E_{i+1}}{2}, \\
-W_i^H &= \frac{C_{i-1}}{4} - \frac{W_i}{2} - \frac{W_{i-1}}{2}.
\end{aligned}$$

First, note that:

$$\begin{aligned}
C_i^H &= (W_i^H + E_i^H) - (W_i^H + E_i^H) + C_i + \frac{C_{i-1}}{4} + \frac{C_{i+1}}{4} - \frac{E_{i-1}}{2} - \frac{W_{i+1}}{2} - \frac{W_i}{2} - \frac{E_i}{2} \\
&= (W_i^H + E_i^H) + (\frac{C_{i-1}}{4} - \frac{W_i}{2} - \frac{W_{i-1}}{2} + \frac{C_{i+1}}{4} - \frac{E_i}{2} - \frac{E_{i+1}}{2}) \\
&\quad + C_i + \frac{C_{i-1}}{4} + \frac{C_{i+1}}{4} - \frac{E_{i-1}}{2} - \frac{W_{i+1}}{2} - \frac{W_i}{2} - \frac{E_i}{2} \\
&= (W_i^H + E_i^H) + (C_i - W_i - E_i) \\
&\quad + (\frac{C_{i-1}}{4} + \frac{C_{i-1}}{4} - \frac{W_{i-1}}{2} - \frac{E_{i-1}}{2}) + (\frac{C_{i+1}}{4} + \frac{C_{i+1}}{4} - \frac{W_{i+1}}{2} - \frac{E_{i+1}}{2}) \\
&= W_i^H + E_i^H,
\end{aligned}$$

where we have employed the relationship $C_i = W_i + E_i$, which holds for the discretization producing the fine mesh stencil as we remarked earlier. Therefore, if the fine mesh stencil has this property, then it is inherited by the galerkin coarse stencil when linear prolongation is employed.

Let us now attempt to relate the Galerkin stencil components to the coefficients of the original differential equation on the fine mesh. First, we have that:

$$\begin{aligned}
-W_i^H &= \frac{C_{i-1}}{4} - \frac{W_i}{2} - \frac{W_{i-1}}{2} = \frac{W_{i-1}}{4} + \frac{E_{i-1}}{4} - \frac{W_i}{2} - \frac{W_{i-1}}{2} \\
&= \frac{E_{i-1}}{4} - \frac{W_i}{2} - \frac{W_{i-1}}{4} = \frac{a_h(x_{i-1/2})}{4h} - \frac{a_h(x_{i-1/2})}{2h} - \frac{a_h(x_{i-3/2})}{4h} \\
&= -\frac{a_h(x_{i-1/2})}{4h} - \frac{a_h(x_{i-3/2})}{4h} = -\frac{1}{2h}\left(\frac{a_h(x_{i-3/2}) + a_h(x_{i-1/2})}{2}\right).
\end{aligned}$$

Similarly,

$$-E_i^H = -\frac{1}{2h}\left(\frac{a_h(x_{i+1/2}) + a_h(x_{i+3/2})}{2}\right).$$

As we have just shown, $C_i^H = W_i^H + E_i^H$, so that we have have the following proposition.

Proposition 3.23 *In one dimension on a uniform mesh, arithmetic averaging of the discretized problem coefficients in the problem (3.41) followed by a standard box-method discretization is equivalent to enforcing the variational conditions with linear prolongation.*

Proof. First, we remark that the coefficient $b(x)$ in (3.41) must be zero for this result. Define the arithmetically averaged coefficients:

$$a_H(x_i) = \frac{a_h(x_{i-1/2}) + a_h(x_{i+1/2})}{2}, \qquad i = 1, 3, 5, \ldots, n.$$

On the coarse mesh, the mesh size will be twice the fine mesh size, $H = 2h$. A standard box-method discretization, using the averaged coefficient on the coarse mesh will yield:

$$A_H = \left[\begin{array}{ccc} -W_i^H & C_i^H & -E_i^H \end{array}\right]_H^H, \qquad i = 2, 4, 6, \ldots, n-1,$$

where

$$-W_i^H = -\frac{a_H(x_{i-1})}{H}, \qquad -E_i^H = -\frac{a_H(x_{i+1})}{H}, \qquad C_i^H = W_i^H + E_i^H.$$

These expressions are equivalent to the components of the Galerkin coarse matrix stencil above produced by linear prolongation. □

Consider now the stencils (3.46) and operator-based prolongation (3.49) on a uniform mesh:

$$A_h = \left[\begin{array}{ccc} -W_i & C_i & -E_i \end{array}\right]_h^h, \qquad I_h^H = \left[\begin{array}{ccc} \frac{E_{i-1}}{C_{i-1}} & 1 & \frac{W_{i+1}}{C_{i+1}} \end{array}\right]_{h(H)}^H.$$

The corresponding Galerkin coarse matrix stencil components are directly from (3.47):

$$\begin{aligned} C_i^H &= C_i + \left(\frac{E_{i-1}}{C_{i-1}}\right)^2 C_{i-1} + \left(\frac{E_{i+1}}{C_{i+1}}\right)^2 C_{i+1} \\ &\quad -\frac{E_{i-1}}{C_{i-1}}E_{i-1} - \frac{W_{i+1}}{C_{i+1}}W_{i+1} - \frac{E_{i-1}}{C_{i-1}}W_i - \frac{W_{i+1}}{C_{i+1}}E_i, \\ -E_i^H &= \frac{W_{i+1}}{C_{i+1}}\frac{E_{i+1}}{C_{i+1}}C_{i+1} - \frac{E_{i+1}}{C_{i+1}}E_i - \frac{W_{i+1}}{C_{i+1}}E_{i+1}, \\ -W_i^H &= \frac{E_{i-1}}{C_{i-1}}\frac{W_{i-1}}{C_{i-1}}C_{i-1} - \frac{W_{i-1}}{C_{i-1}}W_i - \frac{E_{i-1}}{C_{i-1}}W_{i-1}. \end{aligned}$$

It is not difficult to show that these expressions simplify to:

$$-W_i^H = -\frac{W_{i-1}W_i}{C_{i-1}}, \qquad -E_i^H = -\frac{E_iE_{i+1}}{C_{i+1}}, \qquad C_i^H = W_i^H + E_i^H.$$

Let us now attempt to relate the Galerkin stencil components to the coefficients of the original differential equation on the fine mesh. First, we have that:

$$-W_i^H = -\frac{[a_h(x_{i-3/2})/h]\cdot[a_h(x_{i-1/2})/h]}{[a_h(x_{i-3/2})/h] + [a_h(x_{i-1/2})/h]} = -\frac{1}{2h}\left[\frac{2\cdot a_h(x_{i-3/2})\cdot a_h(x_{i-1/2})}{a_h(x_{i-3/2}) + a_h(x_{i-1/2})}\right].$$

Similarly,

$$-E_i^H = -\frac{1}{2h}\left[\frac{2\cdot a_h(x_{i+1/2})\cdot a_h(x_{i+3/2})}{a_h(x_{i+1/2}) + a_h(x_{i+3/2})}\right].$$

This leads us to the following proposition.

Proposition 3.24 *In one dimension on a uniform mesh, harmonic averaging of the discretized problem coefficients in the problem (3.41) followed by a standard box-method discretization is equivalent to enforcing the variational conditions with operator-based prolongation.*

Proof. The coefficient $b(x)$ in (3.41) must be zero for this result. Define the harmonically averaged coefficients:

$$a_H(x_i) = HARM(a_h(x_{i-1/2}), a_h(x_{i+1/2})), \qquad i = 1, 3, 5, \ldots, n, \qquad HARM(x, y) = \frac{2xy}{x+y}.$$

In other words, the coefficients on the coarse mesh are defined as:

$$a_H(x_i) = \frac{2\cdot a_h(x_{i-1/2})\cdot a_h(x_{i+1/2})}{a_h(x_{i-1/2}) + a_h(x_{i+1/2})}, \qquad i = 1, 3, 5, \ldots, n.$$

On the coarse mesh, the mesh size will be twice that of the fine mesh, $H = 2h$. A standard box-method discretization, using the averaged coefficient on the coarse mesh will yield:

$$A_H = \left[\begin{array}{ccc} -W_i^H & C_i^H & -E_i^H \end{array} \right]_H^H, \qquad i = 2, 4, 6, \ldots, n-1,$$

where

$$-W_i^H = -\frac{a_H(x_{i-1})}{H}, \qquad -E_i^H = -\frac{a_H(x_{i+1})}{H}, \qquad C_i^H = W_i^H + E_i^H.$$

These expressions are equivalent to the components of the Galerkin coarse matrix stencil above produced by operator-based prolongation. □

We now consider the two-dimensional case; the stencil calculus for the two-dimensional case is outlined in more detail in Appendix A. We have implemented the stencil calculus in MATHEMATICA and MAPLE, since the Galerkin stencil calculations become extremely complex, even in two-dimensions. Therefore, we will present essentially the same investigation of the relationship between coefficient averaging and the variational conditions in the two-dimensional case, but will employ our stencil calculator to avoid writing out the details of the calculations.

Consider now the two-dimensional problem:

$$-\nabla \cdot (\bar{\mathbf{a}}(\mathbf{x})\nabla u(\mathbf{x})) = f(\mathbf{x}) \text{ in } \Omega \subset \mathbb{R}^2, \qquad u(\mathbf{x}) = 0 \text{ on } \Gamma, \tag{3.50}$$

where now $\mathbf{x} = (x, y)$, and where the tensor $\bar{\mathbf{a}} : \Omega \mapsto \mathbf{L}(\mathbb{R}^2, \mathbb{R}^2)$ has the diagonal form:

$$\bar{\mathbf{a}}(\mathbf{x}) = \left(\begin{array}{cc} a^{(11)}(\mathbf{x}) & 0 \\ 0 & a^{(22)}(\mathbf{x}) \end{array} \right).$$

The box-method discretization of the problem on a non-uniform Cartesian mesh (see Chapter 2) for mesh points $\mathbf{x}_{ij} = (x_i, y_j)$ has the stencil form:

$$A_h = \left[\begin{array}{ccc} 0 & -N_{ij} & 0 \\ -W_{ij} & C_{ij} & -E_{ij} \\ 0 & -S_{ij} & 0 \end{array} \right]_h^h,$$

where

$$E_{ij} = a_h^{(11)}(\mathbf{x}_{i+1/2,j}) \left(\frac{h_{j-1} + h_j}{2h_i} \right), \qquad N_{ij} = a_h^{(22)}(\mathbf{x}_{i,j+1/2}) \left(\frac{h_{i-1} + h_i}{2h_j} \right),$$

$$W_{ij} = E_{i-1,j}, \qquad S_{ij} = N_{i,j-1}, \qquad C_{ij} = N_{ij} + S_{ij} + E_{ij} + W_{ij}.$$

In the case of a uniform mesh $h_i = h_j = h$, then the above expressions simplify to:

$$E_{ij} = a_h^{(11)}(\mathbf{x}_{i+1/2,j}), \qquad N_{ij} = a_h^{(22)}(\mathbf{x}_{i,j+1/2}),$$

where the fine mesh points are $\mathbf{x}_{ij}, i, j = 1, \ldots, n$, and the coarse mesh points are $\mathbf{x}_{ij}, i, j = 2, 4, 6, \ldots, n-1$. Therefore, $a^{(11)}$ is evaluated midway between two fine mesh i- or x-points, and the $a^{(22)}$ is evaluated midway between two fine mesh j- or y-points; we denote these discretized coefficients as $a_h^{(11)}$ and $a_h^{(22)}$.

On a uniform mesh, one representation of linear interpolation in two dimensions is the following (see Appendix A), written in short form as its transpose, the corresponding restriction operator:

$$(I_H^h)^T = I_h^H = \left[\begin{array}{ccc} 0 & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & 1 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & 0 \end{array} \right]_{h(H)}^H.$$

We now employ the symbolic MAPLE stencil calculator described in Appendix A, using the system matrix stencil above together with the interpolation operator above (this computation is quite tedious by hand).

After exploiting the known symmetries in the fine matrix stencil, the following Galerkin coarse matrix stencil is produced:

$$A_H = \left[\begin{array}{ccc} 0 & -N_{ij}^H & 0 \\ -W_{ij}^H & C_{ij}^H & -E_{ij}^H \\ 0 & -S_{ij}^H & 0 \end{array} \right]_h^h ,$$

where, after extensive simplification of the results, the following expressions for the Galerkin coarse matrix stencil components result:

$$E_{ij}^H = \frac{1}{4}\left(E_{i,j-1} + E_{ij} + E_{i+1,j} + E_{i+1,j+1}\right), \qquad N_{ij}^H = \frac{1}{4}\left(N_{i-1,j} + N_{ij} + N_{i,j+1} + N_{i+1,j+1}\right),$$

$$W_{ij}^H = E_{i-1,j}^H, \qquad S_{ij}^H = N_{i,j-1}^H, \qquad C_{ij}^H = N_{ij}^H + S_{ij}^H + E_{ij}^H + W_{ij}^H .$$

Since $E_{ij} = a_h^{(11)}(\mathbf{x}_{i+1/2,j})$ and $N_{ij} = a_h^{(22)}(\mathbf{x}_{i,j+1/2})$, we can express the Galerkin matrix entries in terms of the original problem coefficients on the fine mesh:

$$E_{ij}^H = \frac{1}{4}\left(a_h^{(11)}(\mathbf{x}_{i+1/2,j-1}) + a_h^{(11)}(\mathbf{x}_{i+1/2,j}) + a_h^{(11)}(\mathbf{x}_{i+3/2,j}) + a_h^{(11)}(\mathbf{x}_{i+3/2,j+1}) \right),$$

$$N_{ij}^H = \frac{1}{4}\left(a_h^{(22)}(\mathbf{x}_{i-1,j+1/2}) + a_h^{(22)}(\mathbf{x}_{i,j+1/2}) + a_h^{(22)}(\mathbf{x}_{i,j+3/2}) + a_h^{(22)}(\mathbf{x}_{i+1,j+3/2}) \right),$$

$$W_{ij}^H = E_{i-1,j}^H, \qquad S_{ij}^H = N_{i,j-1}^H, \qquad C_{ij}^H = N_{ij}^H + S_{ij}^H + E_{ij}^H + W_{ij}^H .$$

This gives the following result.

Proposition 3.25 *In two dimensions on a uniform mesh, a certain arithmetic averaging of the discretized problem coefficients in the problem (3.50) followed by a standard box-method discretization is equivalent to enforcing the variational conditions with linear prolongation. The arithmetic averaging is defined by:*

$$a_H^{(11)}(\mathbf{x}_{ij}) = \frac{1}{4}\left(a_h^{(11)}(\mathbf{x}_{i+1/2,j-1}) + a_h^{(11)}(\mathbf{x}_{i+1/2,j}) + a_h^{(11)}(\mathbf{x}_{i+3/2,j}) + a_h^{(11)}(\mathbf{x}_{i+3/2,j+1}) \right),$$

$$i = 1, 3, 5, \ldots, n, \quad j = 2, 4, 6, \ldots, n-1;$$

$$a_H^{(22)}(\mathbf{x}_{ij}) = \frac{1}{4}\left(a_h^{(22)}(\mathbf{x}_{i-1,j+1/2}) + a_h^{(22)}(\mathbf{x}_{i,j+1/2}) + a_h^{(22)}(\mathbf{x}_{i,j+3/2}) + a_h^{(22)}(\mathbf{x}_{i+1,j+3/2}) \right),$$

$$i = 2, 4, 6, \ldots, n-1, \quad j = 1, 3, 5, \ldots, n.$$

Proof. This is immediately clear from the above discussion. Recall that the fine mesh points are $\mathbf{x}_{ij}, i, j = 1, \ldots, n$, and the coarse mesh points are $\mathbf{x}_{ij}, i, j = 2, 4, 6, \ldots, n-1$. The new coarse mesh coefficient $a_H^{(11)}$ lies midway between two coarse mesh points in the i-direction, and the new coefficient $a_H^{(22)}$ lies midway between two coarse mesh points in the j-direction. The averaging scheme is depicted in Figure 3.7, where: the black circles represent the fine mesh points, the larger open circles represent the coarse mesh points, the small open squares represent the half-mesh-points at which the coefficients $a_h^{(22)}$ are located, the small open triangles represent the half-mesh-points at which the coefficients $a_h^{(11)}$ are located, the large open squares are the new coarse mesh coefficients $a_H^{(22)}$, and the large open triangles are the new coarse mesh coefficients $a_H^{(11)}$. The arrows indicate which points are involved in the averaging to produce the coarse mesh coefficients. □

Remark 3.10. We have established some simple relationships between coefficient averaging techniques and the variational conditions. There is an analogous proposition relating two-dimensional Galerkin expressions using operator-based prolongation and harmonically averaged coefficients; unfortunately, the averaging expressions are more complex than the direct Galerkin expressions (given in Appendix A). We have not written them out here, since one objective was to derive a simple and more intuitive procedure for constructing the coarse problem, without sacrificing the variational conditions. For the case we have presented above, comparing the above expressions to those appearing in Appendix A, it is easy to see that the averaging approach

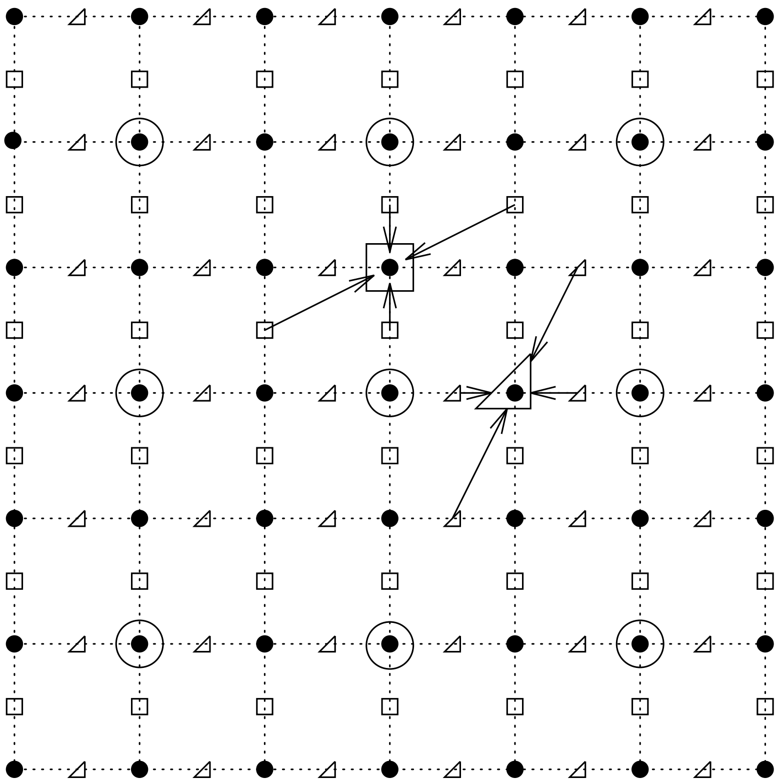


Figure 3.7: Two-dimensional averaging scheme enforcing variational conditions.

to enforcing the variational conditions is much less costly than the direct algebraic enforcement, when the averaging approach can be used.

Unfortunately, our approach here for one- and two-dimensional problems does not extend easily to three dimensions. The difficulty is that the stencil produced by the standard box-method in three dimensions is seven-point; the Galerkin coarse matrix stencil will necessarily have more than seven nonzero components, employing either linear or tri-linear interpolation as the prolongation operator (this is discussed in Appendix A). It may be possible to use a non-standard box-method stencil, and relate the Galerkin coarse matrix stencil entries to some averaging of the problem coefficients and a discretization with this non-standard stencil.

We remark that the material in this section is not simply a restatement of the fact that finite element and box-methods produce the same matrices in certain situations. The operator considered here in problem 3.50 is general, and the coefficients involved may be highly varying or even discontinuous within elements on all but the finest mesh; as such, a finite element discretization will not automatically satisfy the variational conditions, since quadrature error will clearly be large. This averaging approach is intuitive, it guarantees that the variational conditions hold in cases where the finite element method would not guarantee this, and it is computationally much less costly than the equivalent approach of enforcing the conditions by using the full general algebraic Galerkin expressions given in Appendix A.

We wish to make a final comment on the generality of the two-dimensional result above. Assume that we are given a symmetric positive definite matrix with the stencil representation:

$$A_h = \begin{bmatrix} 0 & -N_{ij} & 0 \\ -W_{ij} & C_{ij} & -E_{ij} \\ 0 & -S_{ij} & 0 \end{bmatrix}_h^h ,$$

where

$$E_{ij} > 0, \qquad N_{ij} > 0, \qquad W_{ij} = E_{i-1,j}, \qquad S_{ij} = N_{i,j-1}, \qquad C_{ij} = N_{ij} + S_{ij} + E_{ij} + W_{ij}.$$

Box and finite element discretizations of (3.50) on logically (not necessarily physically) non-uniform Cartesian meshes produce these types of matrices, although our interest here is only the general matrix problem above. By our discussion above, if we form coarse grid equations as follows:

$$A_H = \left[\begin{array}{ccc} 0 & -N_{ij}^H & 0 \\ -W_{ij}^H & C_{ij}^H & -E_{ij}^H \\ 0 & -S_{ij}^H & 0 \end{array} \right]_h^h,$$

using the very simple and inexpensive averaging procedure:

$$E_{ij}^H = \frac{1}{4}\left(E_{i,j-1} + E_{ij} + E_{i+1,j} + E_{i+1,j+1}\right), \qquad N_{ij}^H = \frac{1}{4}\left(N_{i-1,j} + N_{ij} + N_{i,j+1} + N_{i+1,j+1}\right),$$

$$W_{ij}^H = E_{i-1,j}^H, \qquad S_{ij}^H = N_{i,j-1}^H, \qquad C_{ij}^H = N_{ij}^H + S_{ij}^H + E_{ij}^H + W_{ij}^H,$$

then we are guaranteed that the variational conditions hold, where the prolongation operator corresponds to:

$$(I_H^h)^T = I_h^H = \left[\begin{array}{ccc} 0 & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & 1 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & 0 \end{array} \right]_{h(H)}^H.$$

This yields a very efficient technique for enforcing the variational conditions for general matrix equations which have stencil representations. Efficient algebraic multilevel methods, using the coarse mesh equations and interpolation operators above, can be based on this approach. The only restrictions are that the original matrix be symmetric, five-point, and satisfy the row sum criterion: $C_{ij} = N_{ij} + S_{ij} + E_{ij} + W_{ij}$. Multilevel methods based on this approach should be more robust than standard methods which don't enforce variational conditions, and more efficient than general algebraic methods which do enforce the variational conditions. Again, note that this approach guarantees that the variational conditions hold regardless of how the original matrix was generated, and will therefore be effective for problems such as those with discontinuous coefficients or other difficulties.

3.4 Summary of complexity properties of standard methods

In the case of a uniform $m \times m \times m$ mesh and the standard box-method discretization of Poisson's equation on the unit square, the resulting algebraic system is of dimension $N = m^3$. It is well-known that the computational complexities of dense, banded, and sparse Gaussian elimination are $O(N^3)$, $O(N^{7/3})$, and $O(N^2)$, respectively. In order to understand how the iterative methods we have discussed in this chapter will compare to direct methods as well as to each other in terms of *complexity*, we must translate their respective known convergence properties for the model problem into a complexity estimate.

Assume now that the discretization error is $O(h^s)$ for some $s > 0$, which yields a practical linear iteration tolerance of $\epsilon = O(h^s)$. As remarked earlier in §3.1.5, if the mesh is shape-regular and quasi-uniform, then the meshsize h is related to the number of discrete unknowns N through the dimension d of the spatial domain as $h = O(N^{-1/d})$. Now, for the model problem, we showed earlier in §3.1.6 that the spectral radii of the Richardson, Jacobi, and Gauss-Seidel behave as $1 - O(h^2)$. Since $-\ln(1 - ch^2) \approx ch^2 + O(h^4)$, we can estimate the number of iterations required to solve the problem to the level of discretization error from (3.6) as follows:

$$n \leq \frac{|\ln \epsilon|}{|\ln \rho(E)|} = \frac{|\ln h^s|}{|\ln(1 - ch^2)|} \approx \frac{|s \ln h|}{h^2} = O\left(\frac{|\ln N^{-1/d}|}{N^{-2/d}}\right) = O(N^{2/d} \ln N).$$

Assuming the cost of each iteration is $O(N)$ due to the sparsity of the matrices produced by standard discretization methods, we have that the total computational cost to solve the problem using any of the three methods above for $d = 3$ is $O(N^{5/3} \ln N)$. A similar model problem analysis can be done for other methods.

Table 3.1: Model problem complexities of various solvers.

Method	2D	3D
Dense Gaussian elimination	$O(N^3)$	$O(N^3)$
Banded Gaussian elimination	$O(N^2)$	$O(N^{2.33})$
Sparse Gaussian elimination	$O(N^{1.5})$	$O(N^2)$
Richardson's Method	$O(N^2 \ln N)$	$O(N^{1.67} \ln N)$
Jacobi iteration	$O(N^2 \ln N)$	$O(N^{1.67} \ln N)$
Gauss-Seidel iteration	$O(N^2 \ln N)$	$O(N^{1.67} \ln N)$
SOR	$O(N^{1.5} \ln N)$	$O(N^{1.33} \ln N)$
Conjugate gradients (CG)	$O(N^{1.5} \ln N)$	$O(N^{1.33} \ln N)$
Preconditioned CG	$O(N^{1.25} \ln N)$	$O(N^{1.17} \ln N)$
Multilevel methods	$O(N \ln N)$	$O(N \ln N)$
Nested Multilevel methods	$O(N)$	$O(N)$

In contrast, one multilevel iteration costs $O(N)$ operations, and we have remarked that with nested iteration the number of iterations required to reach discretization error remains constant as N increases. To summarize, the complexities of the methods we have discussed in this chapter plus a few others are given in Table 3.1. The complexities for the conjugate gradient methods applied to the model problem may be found in [9]. This table states clearly the motivation for considering the use of multilevel and multigrid methods for the numerical solution of elliptic partial differential equations.

4. Methods for Nonlinear Equations

We begin by reviewing some important concepts about nonlinear equations, nonlinear iterations, and conditions for and types of convergence. Some of the classical nonlinear iterations and nonlinear conjugate gradient methods are then discussed, along with their convergence properties. Newton-like methods are then reviewed, including inexact variations and global convergence modifications. We then discuss damped-inexact-Newton-multilevel methods, which involve the coupling of damped-Newton methods with linear multilevel methods for approximate solution of the Jacobian systems. We attempt to combine the damping parameter selection and linear iteration tolerance specification to insure global superlinear convergence. We also present a nonlinear multilevel method similar to one proposed by Hackbusch, which does not involve an outer Newton iteration. We conclude the chapter by introducing a nonlinear operator-based prolongation procedure; this is a nonlinear extension of the stencil compression ideas of Chapter 3.

Our contributions here are as follows.

- We prove a simple result which yields a necessary and sufficient condition on the residual of the Jacobian system for the inexact Newton direction to be a descent direction; a corollary to this result is a simple sufficient condition for descent, which is easy to combine with superlinear convergence tolerance strategies.
- We combine the (necessary and) sufficient descent condition(s) with inexact tolerance selection strategies for superlinear convergence; this guarantees global, asymptotically superlinear convergence for a damped-inexact-Newton-multilevel iteration which we present.
- We study a true nonlinear multilevel method not involving an outer Newton iteration. This method requires the calculation of a damping parameter for global convergence by solving a one-dimensional minimization problem, similar to that required in the nonlinear conjugate gradient method.
- We develop a nonlinear operator-based prolongation as a generalization of the idea of stencil compression for linear problems.

4.1 Nonlinear operator equations

A discretization of the nonlinear elliptic equation $\mathcal{N}(u) = f$ will in general produce a set of nonlinear algebraic equations $M_k(u_k) = f_k$, although for the particular problems we are considering here, the algebraic equations have both a linear and nonlinear part:

$$A_k u_k + N_k(u_k) = f_k. \tag{4.1}$$

As in Chapter 3, the subscript k denotes the discretization level, with larger k corresponding to a more refined mesh, and hence a larger number of unknowns.

For certain classes of differential operators $\mathcal{N}(\cdot)$, such as those with nonlinearities of the type occurring in the nonlinear Poisson-Boltzmann equation, the differential equation $\mathcal{N}(u) = f$ is uniquely solvable (this is discussed in Chapter 2). The algebraic equations produced by standard discretizations can also be shown to inherit this property (this is also discussed in Chapter 2); throughout this chapter, we will assume that both the original elliptic equation $\mathcal{N}(u) = f$ and the resulting algebraic equations (4.1) are always uniquely solvable for all source functions f and f_k.

In this chapter, we are interested in nonlinear iterations for solving the algebraic equation (4.1) which have the general form:

$$u_k^{n+1} = G_k(u_k^n), \tag{4.2}$$

where $G_k(\cdot)$ is a mapping which is constructed to have as its fixed-point the unique solution u_k of (4.1). The nonlinear extensions of the classical linear methods fit into this framework, as well as the Newton-like methods. Our interest in improved convergence, efficiency, and robustness properties will lead us to damped-inexact-Newton-multilevel methods and nonlinear multilevel methods. These methods will be studied in detail numerically in Chapter 7.

4.1.1 Nonlinear operators, differentiation, and continuity

In this section we will compile some background material on nonlinear operators in finite-dimensional spaces which is used throughout the chapter. We will use the notation from §3.1.1, and assume familiarity with the material on linear operators presented there.

Let $\mathcal{H}_1$, $\mathcal{H}_2$, and $\mathcal{H}$ be real finite-dimensional Hilbert spaces, each with an associated inner-product $(\cdot,\cdot)$ inducing a norm $\|\cdot\| = (\cdot,\cdot)^{1/2}$. Since we are concerned only with finite-dimensional spaces, the spaces $\mathcal{H}$, $\mathcal{H}_1$, and $\mathcal{H}_2$ can be thought of as the Euclidean spaces $\mathbb{R}^n$, $\mathbb{R}^{n_1}$, and $\mathbb{R}^{n_2}$; however, following our approach in the previous chapter, the preliminary material below and the algorithms we develop are phrased in terms of the unspecified spaces $\mathcal{H}$, so that the algorithms may be interpreted directly in terms of finite element spaces as well. This is necessary to set the stage for our discussion of multilevel theory in Chapter 5.

Let $F(\cdot)$ be a nonlinear operator such that $F : D \subset \mathcal{H}_1 \mapsto \mathcal{H}_2$. If $F(\cdot)$ is both one-to-one and onto, then it is called a *bijection*, in which case the inverse mapping $F^{-1}(\cdot)$ exists. If both $F(\cdot)$ and $F^{-1}(\cdot)$ are continuous, then $F(\cdot)$ is called a *homeomorphism.* Concerning the solution of the operator equation $F(u) = v$, it is important that $F(\cdot)$ be a homeomorphism for the problem to be *well-posed in the Hadamard sense.*[1]

The notions of F-(Frechet) and G-(Gateaux) derivatives of an arbitrary function $F : \mathcal{H}_1 \mapsto \mathcal{H}_2$, for arbitrary Hilbert spaces $\mathcal{H}_1$ and $\mathcal{H}_2$, are important here; these ideas were presented in detail in Chapter 2, and familiarity with this material will be assumed here. In Chapter 2 we discussed the special functional $J : \mathcal{H} \mapsto \mathbb{R}$, defined in terms of a bounded linear operator $A \in \mathbf{L}(\mathcal{H},\mathcal{H})$ as follows

$$J(u) = \frac{1}{2}(Au,u), \qquad \forall u \in \mathcal{H}.$$

This particular functional arises often in this chapter, and we remark that in Chapter 2 we discussed the details of calculating its F-derivatives. In Chapter 2 we also considered the functional

$$J(u) = \frac{1}{2}\|F(u)\| = \frac{1}{2}(F(u),F(u)), \qquad \forall u \in \mathcal{H},$$

and we presented as Lemma 2.7 the following result:

$$(J'(u),v) = (F'(u)^T F(u),v), \qquad \forall u \in \mathcal{H},$$

which will be quite useful in this chapter.

The basic notions of continuity of functions were also discussed in Chapter 2. The following special notion of continuity will be used later in this chapter.

Definition 4.1 *The mapping $F : D \subset \mathcal{H} \mapsto \mathcal{H}$ is called Hölder-continuous on D with constant γ and exponent p if there exists $\gamma \geq 0$ and $p \in (0,1]$ such that*

$$\|F(u) - F(v)\| \leq \gamma \|u - v\|^p \qquad \forall u,v \in D \subset \mathcal{H}.$$

If $p = 1$, then F is called uniformly Lipschitz-continuous on D, with Lipschitz constant γ.

[1] Well-posedness "in the sense of Hadamard" [124] refers to three criteria: existence of a solution, uniqueness of a solution, and continuous dependence of the solution on the data of the problem.

Remark 4.1. In the case that $\mathcal{H}_1 = \mathbb{R}^n$ and $\mathcal{H}_2 = \mathbb{R}^m$, then the matrix of all directional derivatives of $F : D \subset \mathbb{R}^n \mapsto \mathbb{R}^m$ taken in the coordinate directions is called the *Jacobian matrix*:

$$F'(\mathbf{x}) = \nabla F(\mathbf{x})^T = \left[\frac{\partial F_i(\mathbf{x})}{\partial x_j} \right],$$

where $F(\mathbf{x}) = (F_1(\mathbf{x}), \ldots, F_m(\mathbf{x}))^T$ and $\mathbf{x} = (x_1, \ldots, x_n)^T$. If $\mathcal{H}_2 = \mathbb{R}^1$, so that $F : D \subset \mathbb{R}^n \mapsto \mathbb{R}$ is a linear functional, then $F'(\mathbf{x})$ is the usual *gradient* vector. It is clear from the definitions that the existence of the G-derivative implies the existence of the Jacobian matrix (the existence of all partial derivatives of F). In the case that the F-derivative exists, the Jacobian matrix is the representation of both the F- and G-derivatives.

4.1.2 Notions of convergence and fixed-point theorems

Let $u^* \in \mathcal{H}$. There are several notions of convergence we will be concerned with regarding the sequence $\{u^n\}$, $u^n \in \mathcal{H}$. Recall that the sequence $\{u^n\}$ is said to *converge strongly* to u^* if $\lim_{n\to\infty} \|u^* - u^n\| = 0$. The sequence $\{u^n\}$ is said to *converge weakly* to u^* if $\lim_{n\to\infty}(u^* - u^n, v) = 0 \ \forall v \in \mathcal{H}$. It can be shown (page 76 in [37]) that strong convergence implies weak convergence.

Regarding strong convergence, there are several notions of the rate of convergence.

Definition 4.2 *The sequence $\{u^n\}$ is said to converge Q-linearly to u^* if there exists $c \in [0, 1)$ and $\bar{n} \geq 0$ such that for $n \geq \bar{n}$,*

$$\|u^* - u^{n+1}\| \leq c\|u^* - u^n\|.$$

Definition 4.3 *The sequence $\{u^n\}$ is said to converge Q-superlinearly to u^* if there exists $\{c_n\}$ such that $c_n \to 0$ and:*

$$\|u^* - u^{n+1}\| \leq c_n\|u^* - u^n\|.$$

Definition 4.4 *The sequence $\{u^n\}$ is said to converge to u^* with rate Q-order(p) if there exists $p > 1$, $c \geq 0$, and $\bar{n} \geq 0$ such that for $n \geq \bar{n}$,*

$$\|u^* - u^{n+1}\| \leq c\|u^* - u^n\|^p.$$

Definition 4.5 *The sequence $\{u^n\}$ is said to converge to u^* with rate R-order(p) if $\|u^* - u^n\|$ is bounded above by another sequence converging with rate Q-order(p).*

As we mentioned at the beginning of the chapter, we are interested in *fixed-point iterations* of the form:

$$u^{n+1} = G(u^n),$$

where the nonlinear mapping $G(\cdot)$ is the *fixed-point mapping.* If $G(\cdot)$ represents some iterative technique for obtaining the solution to a problem, it is important to understand what are necessary or at least sufficient conditions for this iteration to converge to a solution.

First, recall that a *contraction operator* is a mapping $G : D \subset \mathcal{H} \mapsto D$ such that for some *contraction constant* $\alpha \in [0, 1)$ it is true that

$$\|G(u) - G(v)\| \leq \alpha\|u - v\| \quad \forall u, v \in D.$$

The following powerful theorem not only states that a fixed-point iteration will converge to a fixed-point, but also that the fixed point is unique, the convergence rate is Q-linear, and that the error in an iterate may be estimated by the contraction constant.

Theorem 4.1 (Contraction Mapping Theorem) *Let $G : D \subset \mathcal{H} \mapsto D$, where D is closed. If $G(\cdot)$ is a contraction with contraction constant α, then:*

(1) There exists a unique $u^* \in D$ such that $G(u^*) = u^*$.

(2) For any $u^0 \in D$, the sequence $\{u^n\}$ generated by $u^{n+1} = G(u^n)$ converges to u^*.

(3) The convergence rate is Q-linear with constant α, and an error estimate is given by:

$$\|u^* - u^{n+1}\| \leq \frac{\alpha}{1 - \alpha}\|u^{n+1} - u^n\|.$$

Proof. See for example [117] page 474. □

4.1.3 Gradient mappings, energy functionals, and convex analysis

In Chapter 2, we presented the main ideas behind formal calculus of variations involving functionals, gradient mappings, F- and G-derivatives, the Euler or Euler-Lagrange equations, and some fundamental results from convex analysis. The material in this chapter relies on these ideas, and therefore we will assume some familiarity with the material in Chapter 2. However, since we are restricting ourselves to finite-dimensional spaces in this chapter, we will present below a slightly simplified form of the convex analysis material, applicable in the finite-dimensional case.

The three algorithms which we will consider in detail in this chapter, which are the nonlinear conjugate gradient method, the damped-inexact-Newton-multilevel method, and the nonlinear multilevel method, all rely on and exploit the connection between the zero-point problem $F(u) = 0$ and an associated functional $J(u)$ which is minimized at the solution to $F(u) = 0$. Therefore, we will give some background material here regarding the connection between these two problems. Although the following discussion is again in terms of the unspecified space $\mathcal{H}$, it can be interpreted completely in terms of functionals $J : \mathbb{R}^n \mapsto \mathbb{R}$.

Assume we are given the following nonlinear equation:

$$F(u) = Au + N(u) - f = 0,$$

where $F(\cdot)$ and $N(\cdot)$ are nonlinear operators mapping the finite-dimensional space $\mathcal{H}$ onto itself, and where the operator A is SPD. Consider now the (energy) functional, $J : \mathcal{H} \mapsto \mathbb{R}$, where:

$$J(u) = \frac{1}{2}(Au, u) + B(u) - (f, u).$$

A *global minimizer* of $J(\cdot)$ on $\mathcal{H}$ is a point $u^* \in \mathcal{H}$ such that $J(u^*) = \min_{v \in \mathcal{H}} J(v)$. A *local minimizer* of $J(\cdot)$ on $\mathcal{H}$ is a point $u^* \in \mathcal{H}$ such that $J(u^*) = \min_{v \in \mathcal{V}} J(v)$, where $\mathcal{V} \subset \mathcal{H}$ is an open neighborhood of u^*.

Given that A is SPD, and that $N(u) = B'(u)$, it follows immediately from our discussions in §4.1.1 and §2.1.7 that $J' : \mathcal{H} \mapsto \mathcal{H}$ is defined by:

$$J'(u) = Au + B'(u) - f = Au + N(u) - f = F(u).$$

The second F-derivative of $J(\cdot)$ is then the first F-derivative of $F(\cdot)$:

$$J''(u) = F'(u) = A + N'(u).$$

This leads us to the important concept of a *gradient mapping.*

Definition 4.6 *The mapping $F : D \subset \mathcal{H} \mapsto \mathcal{H}$ is called a gradient or potential mapping if for some G-differentiable functional $J : D \subset \mathcal{H} \mapsto \mathbb{R}$ it holds that $F(u) = J'(u)\ \forall u \in D$.*

Regarding the functional $J(\cdot)$, the following are some minimal important concepts.

Definition 4.7 *The functional $J : D \subset \mathcal{H} \mapsto \mathbb{R}$ is called convex on D if $\forall u, v \in D$ and $\alpha \in (0, 1)$ it holds that:*

$$J(\alpha u + (1 - \alpha)v) \leq \alpha J(u) + (1 - \alpha)J(v).$$

The functional $J(\cdot)$ is called *strictly convex* if the inequality in Definition 4.7 is strict. If $J(u) \rightarrow +\infty$ when $\|u\| \rightarrow +\infty$, then $J(\cdot)$ is said to be *coercive.* The following stronger notion of convexity implies both coerciveness and strict convexity.

Definition 4.8 *The functional $J : D \subset \mathcal{H} \mapsto \mathbb{R}$ is called uniformly convex on D if $\forall u, v \in D$ and $\alpha \in (0, 1)$ there exists $c > 0$ such that:*

$$c\alpha(1 - \alpha)\|x - y\|^2 \leq \alpha J(u) + (1 - \alpha)J(v) - J(\alpha u + (1 - \alpha)v).$$

We are interested in the connection between the following two problems:

Problem 1: Find $u \in \mathcal{H}$ such that $J(u) = \min_{v \in \mathcal{H}} J(v)$.
Problem 2: Find $u \in \mathcal{H}$ such that $F(u) = J'(u) = 0$.

The *Euler necessary condition* for the existence of a local minimizer formalizes the idea of critical points of functionals $J(\cdot)$ on $\mathcal{H}$.

Theorem 4.2 (Euler Condition) *If the functional $J : \mathcal{H} \mapsto \mathbb{R}$ is G-differentiable with $F(u) = J'(u)$ $\forall u \in \mathcal{H}$, and if u^* is a local minimizer of $J(\cdot)$, then the $F(u^*) = 0$.*

Proof. See Corollary 8.3.1 in [44]. □

The following theorem gives sufficient conditions for Problem 1 to be uniquely solvable.

Theorem 4.3 (Ekland-Temam Theorem) *If $J : \mathcal{H} \mapsto \mathbb{R}$ is a strictly convex, continuous, and coercive functional, then $J(\cdot)$ has a unique global minimizer u^*.*

Proof. See the proof of Proposition 1.2 in [62], Theorems 4.3.4 and 4.3.7 of [158], or Theorem 26.8 of [68]. □

We finish the section with a theorem that connects Problems 1 and 2 more fully than the Euler condition, and which gives a simple condition on the F-derivative of a gradient mapping $F(\cdot)$ which will guarantee that it is a homeomorphism.

Theorem 4.4 *If $F : \mathcal{H} \mapsto \mathcal{H}$ is a gradient mapping of an associated functional $J(\cdot)$, if $F(\cdot)$ is continuously differentiable on $\mathcal{H}$, and if there exists $c > 0$ such that*

$$(F'(u)v, v) \geq c(v, v), \qquad \forall u, v \in \mathcal{H}, \tag{4.3}$$

then $F(\cdot)$ is a homeomorphism from $\mathcal{H}$ onto $\mathcal{H}$, and $J(\cdot)$ has a unique global minimizer. Moreover, the unique solution to Problem 1 above is also the unique solution to Problem 2.

Proof. A proof (see for example Theorem 26.11 in [68] or Theorem 4.3.10 in [158]) can be constructed by showing condition (4.3) implies that the associated functional $J(\cdot)$ is uniformly convex, and hence has a unique global minimizer u^* by Theorem 4.3. Theorem 4.2 then states that $F(u^*) = 0$. To show that u^* is the unique solution to $F(u) = 0$ involves showing that (4.3) implies that $F(\cdot)$ is a uniformly monotone operator (Theorem 5.4.3 in [158]), which implies that $F(\cdot)$ is a homeomorphism (Theorem 5.4.5 in [158]). □

Remark 4.2. The coerciveness condition (4.3) along with the natural symmetry of $J''(\cdot)$ (Theorem 4.1.6 in [158]) imply that the linear operator $F'(\cdot)$ is SPD with smallest eigenvalue $c > 0$.

The gradient mapping approach can be used to give an existence and uniqueness proof for solutions of the discretized nonlinear Poisson-Boltzmann equation, as an alternative to the proof presented in Chapter 2 which was based on M-function-like arguments. If one can show that a nonlinear operator $F(\cdot)$ is a gradient mapping, then simply bounding the smallest eigenvalue of the Jacobian matrix $F'(\cdot)$ away from zero guarantees that $F(\cdot)$ is a homeomorphism.

We remark that if $F(\cdot)$ is continuously differentiable, then the condition (4.3) alone, without existence of $J(\cdot)$ or even symmetry of $F'(\cdot)$, implies that $F(\cdot)$ is a *uniformly monotone operator* (Theorem 5.4.3 in [158]), which can be used to show that $F(\cdot)$ is a homeomorphism (Theorem 5.4.5 in [158]).

4.1.4 Discrete nonlinear elliptic equations

We are interested in the nonlinear equations which arise from discretizations of nonlinear elliptic partial differential equations of the type discussed in Chapter 2 (such as the Poisson-Boltzmann equation), and so we will use the notation of §3.1.5 and §3.2.1. In particular, we are interested in equations of the form:

$$A_k u_k + N_k(u_k) = f_k, \tag{4.4}$$

where these equations correspond to a box or finite element discretization of a nonlinear (semi-linear) elliptic partial differential equation as discussed in detail in §3.1.5. The space of grid functions u_k with values at the nodes of the mesh is denoted as $\mathcal{U}_k$, and equation (4.4) may be interpreted as a nonlinear algebraic equation in the space $\mathcal{U}_k$. Equation (4.4) may also be interpreted as an abstract operator equation in the finite element space $\mathcal{M}_k$, as discussed in detail in §3.1.5. In either case, the operator A_k is necessarily SPD for the problems and discretization methods we consider, while the form and properties of the nonlinear term $N_k(\cdot)$ depend on the particular problem.

To discuss iterative algorithms for (4.4), and in particular multilevel algorithms, we will require a nested sequence of finite-dimensional spaces $\mathcal{H}_1 \subset \mathcal{H}_2 \subset \cdots \mathcal{H}_J \equiv \mathcal{H}$, which are connected by prolongation and restriction operators, as discussed in detail in §3.2.1. We are given the abstract nonlinear problem in the finest space $\mathcal{H}$:

$$\text{Find } u \in \mathcal{H} \text{ such that } Au + N(u) = f, \tag{4.5}$$

where $A \in \mathbf{L}(\mathcal{H}, \mathcal{H})$ is SPD and $N(\cdot) : \mathcal{H} \mapsto \mathcal{H}$ is a nonlinearity which yields a uniquely solvable problem (see Chapter 2 and the next section), and we are interested in iterative algorithms for determining the unique solution u which involve solving problems of the form:

$$\text{Find } u_k \in \mathcal{H}_k \text{ such that } A_k u_k + N_k(u_k) = f_k, \tag{4.6}$$

in the coarser spaces $\mathcal{H}_k$ for $1 \le k < J$. When only the finest space $\mathcal{H}$ is employed, we will leave off the subscripts from the functions, operators, and spaces to simplify the notation.

4.2 Standard nonlinear methods

In this section, we briefly review the nonlinear extensions of the classical linear methods, nonlinear conjugate gradient methods, and Newton-like methods. We expand at some length on a couple of topics, including: the one-dimensional line search required in the Fletcher-Reeves nonlinear conjugate gradient method, which we will use later for computing a global convergence damping parameter in our nonlinear multilevel methods; and the idea of inexactness and damping in a Newton iteration, which we will use later in our damped-inexact-Newton-multilevel methods.

4.2.1 Nonlinear extensions of the classical iterations

The classical linear methods discussed in Chapter 3, such as Jacobi and Gauss-Seidel, can be extended in the obvious way to nonlinear algebraic equations of the form (4.4). In each case, the method can be viewed as a fixed-point iteration:

$$u^{n+1} = G(u^n).$$

Of course, implementation of these methods, which we refer to as nonlinear Jacobi and nonlinear Gauss-Seidel methods, now requires the solution of a sequence of one-dimensional nonlinear problems for each unknown in one step of the method. A variation that works well, even compared to newer methods, is the nonlinear SOR method.

The convergence properties of these types of methods, as well as a myriad of variations and related methods, are discussed in detail in [158]. Note, however, that the same difficulty arising in the linear case also arises here: as the problem size is increased (the mesh size is reduced), these methods converge more and more slowly. As a result, we will consider alternative methods, such as nonlinear conjugate gradient methods, Newton-like methods, and nonlinear multilevel methods.

Remark 4.3. Since the one-dimensional problems arising in the nonlinear Jacobi and nonlinear Gauss-Seidel methods are often solved with Newton's method, the methods are also referred to as Jacobi-Newton and Gauss-Seidel-Newton methods, meaning that the Jacobi or Gauss-Seidel iteration is the main or outer iteration, whereas the inner iteration is performed by Newton's method. We will consider momentarily the other situation; namely, the use of Newton's method as the outer iteration, and a linear iterative method such as multigrid for solution of the linear Jacobian system at each outer Newton iteration. We will refer to this method as a Newton-multilevel method.

4.2.2 Nonlinear conjugate gradient methods

As we have seen, the following minimization problem:

$$\text{Find } u \in \mathcal{H} \text{ such that } J(u) = \min_{v \in \mathcal{H}} J(v), \qquad \text{where} \qquad J(u) = \frac{1}{2}(Au, u) + B(u) - (f, u)$$

is equivalent to the associated zero-point problem:

$$\text{Find } u \in \mathcal{H} \text{ such that } F(u) = Au + N(u) - f = 0,$$

where $N(u) = B'(u)$. We assume here that both problems are uniquely solvable. An effective approach for solving the zero-point problem, by exploiting the connection with the minimization problem, is the *Fletcher-Reeves* version [66] of the nonlinear conjugate gradient method, which takes the form:

Algorithm 4.1 *(Fletcher-Reeves Nonlinear CG Method)*

```
Let $u^0 \in \mathcal{H}$ be given.
$r^0 = f - N(u^0) - Au^0, \quad p^0 = r^0$.
Do $i = 0, 1, \ldots$ until convergence:
    $\alpha_i =$ (see below)
    $u^{i+1} = u^i + \alpha_i p^i$
    $r^{i+1} = r^i + N(u^i) - N(u^{i+1}) - \alpha_i A p^i$
    $\beta_{i+1} = (r^{i+1}, r^{i+1})/(r^i, r^i)$
    $p^{i+1} = r^{i+1} + \beta_{i+1} p^i$
End do.
```

The directions p^i are computed from the previous direction and the new residual, and the steplength α_i is chosen to minimize the associated functional $J(\cdot)$ in the direction p^i. In other words, α_i is chosen to minimize $J(u^i + \alpha_i p^i)$, which is equivalent to solving the one-dimensional zero-point problem:

$$\frac{dJ(u^i + \alpha_i p^i)}{d\alpha_i} = 0.$$

Given the form of $J(\cdot)$ above, we have that

$$J(u^i + \alpha_i p^i) = \frac{1}{2}(A(u^i + \alpha_i p^i), u^i + \alpha_i p^i) + B(u^i + \alpha_i p^i) - (f, u^i + \alpha_i p^i)$$

A simple differentiation with respect to α_i (and some simplification) gives:

$$\frac{dJ(u^i + \alpha_i p^i)}{d\alpha_i} = \alpha_i (Ap^i, p^i) - (r^i, p^i) + (N(u^i + \alpha_i p^i) - N(u^i), p^i),$$

where $r^i = f - N(u^i) - Au^i$ is the nonlinear residual. The second derivative with respect to α_i will be useful also, which is easily seen to be:

$$\frac{d^2 J(u^i + \alpha_i p^i)}{d\alpha_i^2} = (Ap^i, p^i) + (N'(u^i + \alpha_i p^i) p^i, p^i).$$

Now, Newton's method for solving the zero-point problem for α_i takes the form:

$$\alpha_i^{m+1} = \alpha_i^m - \delta^m$$

where

$$\delta^m = \frac{dJ(u^i + \alpha_i^m p^i)/d\alpha_i}{d^2 J(u^i + \alpha_i^m p^i)/d\alpha_i^2} = \frac{\alpha_i^m (Ap^i, p^i) - (r^i, p^i) + (N(u^i + \alpha_i^m p^i) - N(u^i), p^i)}{(Ap^i, p^i) + (N'(u^i + \alpha_i^m p^i) p^i, p^i)}.$$

The quantities (Ap^i, p^i) and (r^i, p^i) can be computed once at the start of each line search for α_i, each requiring an inner-product (Ap^i is available from the CG iteration). Each Newton iteration for the new α_i^{m+1} then requires evaluation of the nonlinear term $N(u^i + \alpha_i^m p^i)$ and inner-product with p^i, as well as evaluation of the derivative mapping $N'(u^i + \alpha_i p^i)$, application to p^i, followed by inner-product with p^i.

In the case that $N(\cdot)$ arises from the discretization of a nonlinear partial differential equation and is of *diagonal form*, meaning that the j-th component function of the vector $N(\cdot)$ is a function of only the j-th component of the vector of nodal values u, or $N_j(u) = N_j(u_j)$, then the resulting Jacobian matrix $N'(\cdot)$ of $N(\cdot)$ is a diagonal matrix. This situation occurs with box-method discretizations of the nonlinear

Poisson-Boltzmann equation and similar equations. As a result, computing the term $(N'(u^i + \alpha_i p^i)p^i, p^i)$ can be performed with fewer operations than two inner-products.

The total cost for each Newton iteration (beyond the first) is then evaluation of $N(\cdot)$ and $N'(\cdot)$, and something less than three inner-products. Therefore, the line search can be performed fairly inexpensively in certain situations. If alternative methods are used to solve the one-dimensional problem defining α_i, then evaluation of the Jacobian matrix can be avoided altogether, although as we remarked earlier, the Jacobian matrix is cheaply computable in the particular applications we are interested in here.

Remark 4.4. Note that if the nonlinear term $N(\cdot)$ is absent, then the zero-point problem is linear and the associated energy functional is quadratic:

$$F(u) = Au - f = 0, \qquad J(u) = \frac{1}{2}(Au, u) - (f, u).$$

In this case, the Fletcher-Reeves CG algorithm reduces to exactly the Hestenes-Stiefel [93] linear conjugate gradient algorithm (Algorithm 3.2 of Chapter 3, with the preconditioner $B = I$). The exact solution to the linear problem $Au = f$, as well as to the associated minimization problem, can be reached in no more than n_k steps, where n_k is the dimension of the space $\mathcal{H}$ (see Theorem 8.6.1 in [158]). The calculation of the steplength α_i no longer requires the iterative solution of a one-dimensional minimization problem with Newton's method, since:

$$\frac{dJ(u^i + \alpha_i p^i)}{d\alpha_i} = \alpha_i(Ap^i, p^i) - (r^i, p^i) = 0$$

yields an explicit expression for the α_i which minimizes the functional J in the direction p^i:

$$\alpha_i = \frac{(r^i, p^i)}{(Ap^i, p^i)}.$$

4.2.3 Newton's method and inexact/quasi/truncated variants

Given the nonlinear operator $F : D \subset \mathcal{H} \mapsto \mathcal{H}$ for some finite-dimensional space $\mathcal{H}$, a generalization of the classical Newton's method for solving the problem $F(u) = 0$ is as follows:

$$\begin{aligned} F'(u^n)v^n &= -F(u^n) \\ u^{n+1} &= u^n + v^n. \end{aligned}$$

In other words, the Newton iteration is simply the fixed-point iteration:

$$u^{n+1} = G(u^n) = u^n - F'(u^n)^{-1}F(u^n). \tag{4.7}$$

By viewing the Newton iteration as a fixed-point iteration, a very general convergence theorem can be proven in the abstract space $\mathcal{H}$.

Theorem 4.5 (Newton-Kantorovich Theorem) *Given $u^0 \in D$ where D is convex, assume that $F : D \subset \mathcal{H} \mapsto \mathcal{H}$ is differentiable on D. If $\alpha = \gamma\beta\eta < \frac{1}{2}$, where:*

(1) $F'(u)$ is uniformly Lipschitz-continuous in D with Lipschitz constant γ,

(2) $F'(u^0)$ is nonsingular, with $\|F'(u^0)^{-1}\| \le \beta$,

(3) $\|u^1 - u^0\| = \|F'(u^0)^{-1}F(u^0)\| \le \eta$,

then the Newton iteration (4.7) converges to a unique $u^* \in D$.

Proof. See for example Theorem 8.2.6 in [157]. □

There are several variations of the standard Newton iteration (4.7) commonly used for nonlinear algebraic equations which we mention briefly. A *quasi*-Newton method refers to a method which uses an approximation to the true Jacobian matrix for solving the Newton equations. A *truncated*-Newton method uses the true Jacobian matrix in the Newton iteration, but solves the Jacobian system only approximately, using an iterative linear solver in which the iteration is stopped early or *truncated*. *Inexact-* or *approximate*-Newton methods refers to all of these types of methods collectively, where in the most general case an approximate Newton direction is produced in some unspecified fashion. It can be shown that the convergence behavior of these inexact-Newton methods is similar to the standard Newton's method, and theorems similar to (4.5) can be established (see Chapter 18 of [117], and the discussions below).

For our purposes here, the inexact-Newton approach will be of interest, for the following reasons. First, in the case of partial differential equations such as the Poisson-Boltzmann equation which consist of a leading linear term plus a nonlinear term which does not depend on derivatives of the solution, the nonlinear algebraic equations generated often have the form:

$$F(u) = Au + N(u) - f = 0.$$

The matrix A is SPD, and the nonlinear term $N(\cdot)$ is often simple, and in fact is often of *diagonal form*, meaning that the j-th component of the vector function $N(u)$ is a function of only the j-th entry of the vector u, or $N_j(u) = N_j(u_j)$; this occurs for example in the case of a box-method discretization of the Poisson-Boltzmann equation and similar equations. Further, it is often the case that the derivative $N'(\cdot)$ of the nonlinear term $N(\cdot)$, which will be a diagonal matrix due to the fact that $N(\cdot)$ is of diagonal form, can be computed (and applied to a vector) at low expense. If this is the case, then the true Jacobian matrix is available at low cost:

$$F'(u) = A + N'(u).$$

A second reason for our interest in the inexact-Newton approach is that the efficient multilevel methods developed in Chapter 3 for the linearized Poisson-Boltzmann equation and similar equations can be used effectively for the Jacobian systems; this is because the Jacobian $F'(u)$ is essentially the linearized Poisson-Boltzmann operator, where only the diagonal Helmholtz-like term $N'(\cdot)$ changes from one Newton iteration to the next. Our fast linear multilevel methods should be effective as inexact Jacobian system solvers, and this is demonstrated numerically in Chapter 6.

Remark 4.5. Regarding the assumptions on the function $F(\cdot)$ and the Jacobian $F'(\cdot)$ appearing in Theorem 4.5, although they may seem unnatural at first glance, they are essentially the minimal conditions necessary to show that the Newton iteration, viewed as a fixed-point iteration, is a *contraction*, so that a contraction argument may be employed (cf. page 286 in [37]). Since a contraction argument is used, no assumptions on the existence or uniqueness of a solution are required. A disadvantage of proving Newton convergence through the Contraction Mapping Theorem is that only Q-linear convergence is shown. If additional assumptions are made, such as the existence of a unique solution, then Q-quadratic convergence can be shown; examples can be found in [111, 120, 157].

4.2.4 Global Newton convergence through damping

As noted in the previous section, Newton-like methods converge if the initial approximation is "close" to the solution; different convergence theorems require different notions of closeness. If the initial approximation is close enough to the solution, then superlinear or Q-order(p) convergence occurs. However, the fact that these theorems require a good initial approximation is also indicated in practice: it is well known that Newton's method will converge slowly or fail to converge at all if the initial approximation is not good enough.

On the other hand, methods such as those used for unconstrained minimization can be considered to be "globally" convergent methods, although their convergence rates are often extremely poor. One approach to improving the robustness of a Newton iteration without loosing the favorable convergence properties close to the solution is to combine the iteration with a global minimization method. In other words, we can attempt to force global convergence of Newton's method by requiring that:

$$\|F(u^{n+1})\| < \|F(u^n)\|,$$

meaning that we require a decrease in the value of the function at each iteration. But this is exactly what global minimization methods, such as the nonlinear conjugate gradient method, attempt to achieve: progress toward the solution at each step.

More formally, we wish to define a minimization problem, such that the solution of the zero-point problem we are interested in also solves the associated minimization problem. Let us define the following two problems:

Problem 1: Find $u \in \mathcal{H}$ such that $F(u) = 0$.
Problem 2: Find $u \in \mathcal{H}$ such that $J(u) = \min_{v \in \mathcal{H}} J(v)$.

We assume that Problem 2 has been defined so that the unique solution to Problem 1 is also the unique solution to Problem 2; note that in general, there may not exist a *natural* functional $J(\cdot)$ for a given $F(\cdot)$, although we will see in a moment that it is always possible to construct an appropriate functional $J(\cdot)$.

A *descent direction* for the functional $J(\cdot)$ at the point u is any direction v such that the directional derivative of $J(\cdot)$ at u in the direction v is negative, or $J'(u)(v) = (J'(u), v) < 0$. If v is a descent direction, then it is not difficult to show (Theorem 8.2.1 in [158]) that there exists some $\lambda > 0$ such that:

$$J(u + \lambda v) < J(u). \tag{4.8}$$

This follows from a generalized Taylor expansion (cf. page 255 in [124]), since

$$J(u + \lambda v) = J(u) + \lambda(J'(u), v) + O(\lambda^2).$$

If λ is sufficiently small and $(J'(u), v) < 0$ holds (v is a descent direction), then clearly $J(u + \lambda v) < J(u)$. In other words, if a descent direction can be found at the current solution u^n, then an improved solution u^{n+1} can be found for some steplength in the descent direction v; i.e., by performing a one-dimensional line search for λ until (4.8) is satisfied.

Therefore, if we can show that the Newton direction is a descent direction, then performing a one-dimensional line search in the Newton direction will always guarantee progress toward the solution. In the case that we define the functional as:

$$J(u) = \frac{1}{2}\|F(u)\|^2 = \frac{1}{2}(F(u), F(u)),$$

we can show that the Newton direction is a descent direction. While the following result is easy to show for $\mathcal{H} = \mathbb{R}^n$, we showed in Lemma 2.7 that it is also true in the general case when $\|\cdot\| = (\cdot,\cdot)^{1/2}$:

$$J'(u) = F'(u)^T F(u).$$

Now, the Newton direction at u is simply $v = -F'(u)^{-1}F(u)$, so if $F(u) \neq 0$, then:

$$(J'(u), v) = -(F'(u)^T F(u), F'(u)^{-1}F(u)) = -(F(u), F(u)) < 0.$$

Therefore, the Newton direction is always a descent direction for this particular choice of $J(\cdot)$, and by the introduction of the damping parameter λ, the Newton iteration can be made globally convergent in the above sense.

4.3 Damped-inexact-Newton-multilevel methods

Given the problem of n_k nonlinear algebraic equations and n_k unknowns:

$$F(u) = Au + N(u) - f = 0,$$

for which we desire the solution u, the "holy grail" for this problem is an algorithm which (1) always converges, and (2) has optimal complexity, which in this case means $O(n_k)$.

As we have just seen, Newton's method can be made essentially globally convergent with the introduction of a damping parameter. In addition, close to the root, Newton's method has at least superlinear convergence properties. If a method with linear convergence properties is used to solve the Jacobian systems at each Newton iteration, and the complexity of the linear solver is the dominant cost of each Newton iteration, then the complexity properties of the linear method will determine the complexity of the resulting Newton iteration asymptotically.

We have discussed in detail in Chapter 3 the convergence and complexity properties of multilevel methods; in many situations they can be shown to have optimal complexity, and in many others this behavior can be demonstrated empirically. With an efficient inexact solver such as a multilevel method for the early damped iterations, employing a more stringent tolerance for the later iterations as the root is approached, a very efficient yet robust nonlinear iteration should result. The idea here, motivated by the work in [16, 17], is to combine the robust damped Newton methods with the fast linear multilevel solvers developed in Chapter 3 for the inexact Jacobian system solves.

The conditions for linear solve tolerance to insure superlinear convergence have been given in [47, 54, 55]. Guidelines for choosing damping parameters to ensure global convergence, and yet allow for superlinear convergence, have been established in [16]. Combination with linear multilevel iterative methods for the semiconductor problem has been considered in [17], along with questions of complexity. We outline the basic algorithm below, specializing it to the particular form of our nonlinear problems. We then give some results on damping and inexactness tolerance selection strategies.

4.3.1 Newton-multilevel iteration

We will restrict our discussion here to the following nonlinear problem, which has arisen for example from the discretization of a nonlinear elliptic problem:

$$F(u) = Au + N(u) - f = 0.$$

The derivative has the form:

$$F'(u) = A + N'(u).$$

The damped-inexact-Newton iteration for this problem takes the form:

Algorithm 4.2 *(Damped-Inexact-Newton Method)*

```
(1) Inexact Jacobian system solve:   $[A + N'(u^n)]\, v^n = f - Au^n - N(u^n)$
(2) Correction of the solution:      $u^{n+1} = u^n + \lambda_n v^n.$
```

We employ the linear multilevel methods of Chapter 3 in Step (1) of Algorithm 4.2. A convergence analysis of the undamped method is given in [85]. A detailed convergence analysis of the damped method is given in [17]. Below, we outline what guidelines exists for selection of the damping parameters and the linear iteration tolerance.

Remark 4.6. Note that due to the special form of the nonlinear operator, the damping step can be implemented in a surprisingly efficient manner. During the one-dimensional line search for the parameter λ_n, we continually check for satisfaction of the inequality:

$$\|F(u^n + \lambda_n v^n)\| < \|F(u^n)\|.$$

The term on the right is available from the previous Newton iteration. The term on the left, although it might appear to involve computing the full nonlinear residual, in fact can avoid the operator-vector product contributed by the linear term. Simply note that

$$F(u^n + \lambda_n v^n) = A[u^n + \lambda_n v^n] + N(u^n + \lambda_n v^n) - f = [Au^n - f] + \lambda_n[Av^n] + N(u^n + \lambda_n v^n).$$

The term $[Au^n - f]$ is available from the previous Newton iteration, and $[Av^n]$ need be computed only once at each Newton step. Computing $F(u^n + \lambda_n v^n)$ for each damping step beyond the first requires only the "saxpy" operation $[Au^n - f] + \lambda_n[Av^n]$ for the new damping parameter λ_n, and evaluation of the nonlinear term at the new damped solution, $N(u^n + \lambda_n v^n)$.

4.3.2 Linear iteration tolerance for local superlinear convergence

Quasi-Newton methods are studied in [54], and a "characterization" theorem is established for the sequence of approximate Jacobian systems. This theorem establishes sufficient conditions on the sequence $\{B_i\}$, where $B_i \approx F'$, to ensure superlinear convergence of a quasi-Newton method. An interesting result which they obtained is that the "consistency" condition is not required, meaning that the sequence $\{B_i\}$ need not converge to the true Jacobian $F'(\cdot)$ at the root of the equation $F(u) = 0$, and superlinear convergence can still be obtained.

In the review paper [55], the characterization theorem of [54] is rephrased in a geometric form, showing essentially that the full or true Newton step must be approached, asymptotically, in both length and direction, to attain superlinear convergence in a quasi-Newton iteration.

Inexact-Newton methods are studied directly in [47]. Their motivation is the use of iterative solution methods for approximate solution of the true Jacobian systems. They establish conditions on the accuracy of the inexact Jacobian solves at each Newton iteration which will ensure superlinear convergence. The inexact-Newton method is analyzed in the form:

$$\begin{aligned} F'(u^n)v^n &= -F(u^n) + r^n, \qquad \frac{\|r^n\|}{\|F(u^n)\|} \le \eta_n, \\ u^{n+1} &= u^n + v^n. \end{aligned}$$

In other words, the quantity r^n, which is simply the residual of the Jacobian linear system, indicates the inexactness allowed in the approximate linear solve, and is exactly what one would monitor in a linear iterative solver. It is established that if the *forcing sequence* $\eta_n < 1$ for all n, then the above method is locally convergent. Their main result is the following theorem.

Theorem 4.6 (Dembo-Eisenstat-Steihaug) *Assume that there exists a unique u^* such that $F(u^*) = 0$, that $F(\cdot)$ is continuously differentiable in a neighborhood of u^*, that $F'(u^*)$ is nonsingular, and that the inexact-Newton iterates $\{u^n\}$ converge to u^*. Then:*

(1) The convergence rate is superlinear if: $\lim_{n\to\infty} \eta_n = 0$.

(2) The convergence rate is Q-order at least $1 + p$ if $F'(u^*)$ is Hölder continuous with exponent p, and

$$\eta_n = O(\|F(u^n)\|^p), \text{ as } n \to \infty.$$

(3) The convergence rate is R-order at least $1 + p$ if $F'(u^*)$ is Hölder continuous with exponent p, and if $\{\eta_n\} \to 0$ with R-order at least $1 + p$.

Proof. See [47]. □

As a result of this theorem, they suggest the tolerance rule:

$$\eta_n = \min\left\{\frac{1}{2}, C\|F(u^n)\|^p\right\}, \qquad 0 < p \le 1, \tag{4.9}$$

which guarantees Q-order convergence of at least $1 + p$. In [149], the alternative criterion is suggested:

$$\eta_n = \min\left\{\frac{1}{n}, \|F(u^n)\|^p\right\}, \qquad 0 < p \le 1. \tag{4.10}$$

4.3.3 Necessary and sufficient conditions for inexact descent

Note the following subtle point regarding the combination of inexact Newton methods and damping procedures for obtaining global convergence properties: only the *exact* Newton direction is guaranteed to be a descent direction. Once inexactness is introduced into the Newton direction, there is no guarantee that damping will achieve global convergence in the sense outlined above. However, the following simple result gives a necessary and sufficient condition on the tolerance of the Jacobian system solve for the inexact Newton direction to be a descent direction.

Theorem 4.7 *The inexact Newton method (Algorithm 4.2) for $F(u) = 0$ yields a descent direction v at the point u if and only if the residual of the Jacobian system $r = F'(u)v + F(u)$ satisfies:*

$$(F(u), r) < (F(u), F(u)).$$

Proof. We remarked earlier that an equivalent minimization problem (appropriate for Newton's method) to associate with the zero point problem $F(u) = 0$ is given by $\min_{u\in\mathcal{H}} J(u)$, where $J(u) = (F(u), F(u))/2$. We also noted that the derivative of $J(u)$ can be written as $J'(u) = F'(u)^T F(u)$. Now, the direction v is a descent direction for $J(u)$ if and only if $(J'(u), v) < 0$. The exact Newton direction is $v = -F'(u)^{-1}F(u)$, and as shown earlier is always a descent direction. Consider now the inexact direction satisfying:

$$F'(u)v = -F(u) + r, \qquad \text{or} \qquad v = F'(u)^{-1}[r - F(u)].$$

This inexact direction is a descent direction if and only if:

$$\begin{aligned}(J'(u), v) &= (F'(u)^T F(u), F'(u)^{-1}[r - F(u)]) \\ &= (F(u), r - F(u)) \\ &= (F(u), r) - (F(u), F(u)) \\ &< 0,\end{aligned}$$

which is true if and only if the residual of the Jacobian system r satisfies:

$$(F(u), r) < (F(u), F(u)).$$

□

This leads to the following very simple sufficient condition for descent.

Corollary 4.8 *The inexact Newton method (Algorithm 4.2) for $F(u) = 0$ yields a descent direction v at the point u if the residual of the Jacobian system $r = F'(u)v + F(u)$ satisfies:*

$$\|r\| < \|F(u)\|.$$

Proof. From the proof of Theorem 4.7 we have:

$$(J'(u), v) = (F(u), r) - (F(u), F(u)) \leq \|F(u)\|\|r\| - \|F(u)\|^2,$$

where we have employed the Cauchy-Schwarz inequality. Therefore, if $\|r\| < \|F(u)\|$, then the rightmost term is clearly negative (unless $F(u) = 0$), so that v is a descent direction. □

Remark 4.7. The sufficient condition presented as Corollary 4.8 also appears as a lemma in [61]. Note that most stopping criteria for the Newton iteration involve evaluating $F(\cdot)$ at the previous Newton iterate u^n. The quantity $F(u^n)$ will have been computed during the computation of the previous Newton iterate u^n, and the tolerance for u^{n+1} which guarantees descent requires $(F(u^n), r) < (F(u^n), F(u^n))$ by Theorem 4.7. This involves only the inner-product of r and $F(u^n)$, so that enforcing this tolerance requires only an additional inner-product during the Jacobian linear system solve, which for n_k unknowns introduces an additional n_k multiplies and n_k additions. In fact, a scheme may be employed in which only a residual tolerance requirement for superlinear convergence is checked until an iteration is reached in which it is satisfied. At this point, the descent direction tolerance requirement can be checked, and additional iterations will proceed with this descent stopping criterion until it too is satisfied. If the linear solver reduces the norm of the residual monotonically (such as any of the linear methods of Chapter 3), then the first stopping criterion need not be checked again.

In other words, this adaptive Jacobian system stopping criterion, enforcing a tolerance on the residual for local superlinear convergence *and* ensuring a descent direction at each Newton iteration, can be implemented at the same computational cost as a simple check on the norm of the residual of the Jacobian system.

Alternatively, the sufficient condition given in Corollary 4.8 may be employed at no additional cost, since only the norm of the residual need be computed, which is also what is required to insure superlinear convergence using Theorem 4.6.

4.3.4 Global superlinear convergence

In [16], an analysis of inexact-Newton methods is performed, where a damping parameter has been introduced. Their goal was to establish selection strategies for both the linear solve tolerance and the damping parameters at each Newton iteration, in an attempt to achieve global superlinear convergence of the damped-inexact Newton iteration. It was established, similar to the result in [55], that the Jacobian system solve tolerance must converge to zero (exact solve in the limit), and the damping parameters must converge to one (the full Newton step in the limit), for superlinear convergence to be achieved. There are several technical assumptions on the function $F(\cdot)$ and the Jacobian $F'(\cdot)$ in their paper; we will summarize one of their main results in the following theorem, as it applies to the inexact-Newton framework we have constructed in this chapter.

Theorem 4.9 (Bank and Rose) *Suppose $F : D \subset \mathcal{H} \mapsto \mathcal{H}$ is a homeomorphism on $\mathcal{H}$. Assume also that $F(\cdot)$ is differentiable on closed bounded sets D, that $F'(u)$ is nonsingular and uniformly Lipschitz continuous on such sets D, and that closed level set*

$$S_o = \{u \mid \|F(u)\| \leq \|F(u^0)\|\}$$

is a bounded set. Suppose now that the forcing and damping parameters η_n and λ_n satisfy:

$$\eta_n \leq C\|F(x^n)\|^p, \qquad \eta_n \leq \eta_0, \qquad \eta_0 \in (0,1),$$

$$\lambda_n = \frac{1}{1 + K_n\|F(x^n)\|}, \qquad 0 \leq K_n \leq K_0, \qquad \text{so that} \qquad \lambda_n \leq 1.$$

Then, there exists $u^* \in \mathcal{H}$ such that $F(u^*) = 0$, and with any $u^0 \in \mathcal{H}$, the sequence generated by the damped-inexact-Newton method:

$$F'(u^n)v^n = -F(u^n) + r^n, \qquad \frac{\|r^n\|}{\|F(u^n)\|} \leq \eta_n, \tag{4.11}$$

$$u^{n+1} = u^n + \lambda_n v^n \tag{4.12}$$

converges to $u^* \in S_0 \subset \mathcal{H}$. In addition, on the set S_0, the sequence $\{u^n\}$ converges to u^* at rate Q-order at least $1 + p$.

Proof. See [17]. □

Note that by forcing $\eta_n \leq \eta_0 < 1$, it happens that the residual of the Jacobian system in Theorem 4.9 satisfies $\|r^n\| \leq \eta_n\|F(u^n)\| \leq \|F(u^n)\|$, which by Corollary 4.8 always ensures that the inexact Newton direction produced by their algorithm is a descent direction. The sequence $\{K_n\}$ is then selected so that each parameter is larger than a certain quantity (inequality 2.14 in [17]), which is a guarantee that an appropriate steplength for actual descent is achieved, without line search. We remark that there is also a weaker convergence result in [17] which essentially states that the convergence rate of the damped-inexact-Newton method above is R-linear or Q-order$(1 + p)$ on certain sets which are slightly more general than the set S_0. The parameter selection strategy suggested in [17] based on the above theorem is referred to as *Algorithm Global*, which appears on page 287 in [17]. The idea of the algorithm is to avoid the typical searching strategies required for other global methods by employing the sequence K_n above.

We now propose an alternative globally convergent inexact-Newton algorithm which is somewhat easier to understand and implement, motivated by the simple necessary and sufficient descent conditions established in the previous section.

Algorithm 4.3 *(Damped-Inexact-Newton method)*

```
(1) Inexact Jacobian system solve:  $F'(u^n)v^n = -F(u^n) + r^n$,    $TEST(r^n) = TRUE$,
(2) Correction of the solution:     $u^{n+1} = u^n + \lambda_n v^n$,
```

where the damping parameters λ_n and procedure $TEST(r^n)$ are defined by:

```
(1)  $TEST(r^n)$ guarantees both global descent and local Q-order$(1+p)$ convergence:
        IF:  $\|r^n\| \leq C\|F(u^n)\|^{p+1},\ C > 0,\ p > 0$,  (local Q-order$(1+p)$ convergence)
        AND:  $(F(u^n), r^n) < (F(u^n), F(u^n))$   (guaranteed descent for step n)
        THEN:  $TEST \equiv TRUE$;  ELSE:  $TEST \equiv FALSE$.
(2)  The damping parameters $\lambda_n$ satisfy: $\|F(u^n + \lambda_n v^n)\| \leq \|F(u^n)\|$,
     using any line search method; this is always possible if $TEST(r^n) = TRUE$.
     The full inexact-Newton step $\lambda = 1$ is always tried first.
```

An alternative less expensive procedure $TEST(r^n)$ is as follows:

(1') $TEST(r^n)$ guarantees both global descent and local Q-order$(1+p)$ convergence:
IF: $\|r^n\| \le C\|F(u^n)\|^{p+1}$, $C > 0$, $p > 0$, (local Q-order$(1+p)$ convergence)
AND: $\|r^n\| < \|F(u^n)\|$ (guaranteed descent for step n)
THEN: $TEST \equiv TRUE$; ELSE: $TEST \equiv FALSE$.

In Algorithm 4.3, the second condition in (1) is the necessary and sufficient condition for the inexact-Newton direction to be a descent direction, established in Theorem 4.7. The second condition in (1') of Algorithm 4.3 is the weaker sufficient condition established in Corollary 4.8. Note that, in early iterations when Q-order$(1+p)$ for $p > 0$ is not to be expected, just satisfying one of the descent conditions is (necessary and) sufficient for progress toward the solution. The condition $\eta_n < 1$ in Theorem 4.9 implies that the inexact-Newton directions produced by their algorithm are, by Corollary 4.8, descent directions. Algorithm 4.3 decouples the descent and superlinear convergence conditions, and would allow for the use of only the weakest possible test of $(F(u^n), r^n) < (F(u^n), F(u^n))$ far from the solution, ensuring progress toward the solution with the least amount of work per Newton step.

Note also that the Q-order$(1+p)$ condition

$$\|r^n\| \le C\|F(u^n)\|^{p+1}$$

does *not* guarantee a descent direction, so that it is indeed important to satisfy the descent condition separately. The Q-order$(1+p)$ condition *will* impose descent if

$$C\|F(u^n)\|^{p+1} < \|F(u^n)\|,$$

which does not always hold. If one is close to the solution, so that $\|F(u^n)\| < 1$, and if $C \le 1$, then the Q-order$(1+p)$ condition will imply descent. By this last comment, we see that if $\|F(u^n)\| < 1$ and $C \le 1$, then the full inexact-Newton step is a descent direction, and since we attempt this step first, we see that our algorithm reduces to the algorithm studied in [47] near the solution; therefore, Theorem 4.6 above applies to Algorithm 4.3 near the solution without modification.

4.3.5 Stopping criteria for Newton and other nonlinear iterations

As in a linear iteration, there are several quantities which can be monitored during a nonlinear iteration to determine whether a sufficiently accurate approximation u^{n+1} to the true solution u^* has been obtained. Possible choices, with respect to any norm $\|\cdot\|$, include:

(1) Nonlinear residual:	$\|F(u^{n+1})\|$	$\le FTOL$
(2) Relative residual:	$\|F(u^{n+1})\|/\|F(u^0)\|$	$\le RFTOL$
(3) Iterate change:	$\|u^{n+1} - u^n\|$	$\le UTOL$
(4) Relative change:	$\|u^{n+1} - u^n\|/\|u^{n+1}\|$	$\le RUTOL$
(5) Contraction estimate:	$\|u^{n+1} - u^n\|/\|u^n - u^{n-1}\|$	$\le CTOL.$

We also mention a sixth option, which attempts to obtain an error estimate from the Contraction Mapping Theorem 4.1 by estimating the contraction constant α of the nonlinear fixed-point mapping $G(\cdot)$ associated with the iteration. The constant is estimated as follows:

$$\alpha = \frac{\|u^{n+1} - u^n\|}{\|u^n - u^{n-1}\|} = \frac{\|G(u^n) - G(u^{n-1})\|}{\|u^n - u^{n-1}\|},$$

and the Contraction Mapping Theorem gives the error estimate-based criterion:

$$\text{(6) Error estimate : } \|u^* - u^{n+1}\| \le \frac{\alpha}{1-\alpha}\|u^{n+1} - u^n\| \le ETOL.$$

There are certain difficulties with employing any of these conditions alone. For example, if the iteration has temporarily stalled, then criteria (3) and (4) would prematurely halt the iteration. On the other hand, if the scaling of the function $F(\cdot)$ is such that $\|F(\cdot)\|$ is always very small, then criterion (1) could halt the iteration early. Criterion (2) attempts to alleviate this problem in much the same way as a relative stopping criteria in the linear case. However, if the initial approximation u^0 is such that $\|F(u^0)\|$ is extremely large, then (3) could falsely indicate that a good approximation has been reached. Criterion (5) cannot be used to halt the iteration alone, as it gives no information about the quality of the approximation; it would be useful in a Newton iteration to detect when the region of fast convergence has been entered.

Criterion (6) may be the most reliable stand-alone criteria, although it depends on the accuracy of the contraction number estimate. If the contraction number is constant (linear convergence) over many iterations or goes to zero monotonically (superlinear convergence), then this should be reliable; otherwise, the contraction estimate may have no bearing on the true contraction constant for the mapping $G(\cdot)$, and the error estimate may be poor.

Several dual criteria have been proposed in the literature. For example, the combination of (4) and (5) was suggested in [15], since (4) attempts to detect convergence has been reached, whereas (5) attempts to ensure that (4) has not been satisfied simply due to stalling of the iteration. In [56], the combination of (4) and (1) is suggested, where (1) attempts to prevent halting on (4) due to stalling. The idea of scaling the components of u^{n+1} in (1) and $F(u^{n+1})$ in (2) is also recommended in [56], along with use of the maximum norm $\|\cdot\|_\infty$. In [71], other combinations are suggested (with an optimization orientation, some combinations involving the associated functional $J(\cdot)$).

In the implementations of our nonlinear methods, including our implementations of the classical nonlinear methods, the nonlinear conjugate gradient method, the damped-inexact-Newton-multilevel methods, and the nonlinear multilevel methods discussed below, we provide all of (1) through (4) as options separately, using the maximum norm $\|\cdot\|_\infty$. In addition, we provide the combination of (1) and (4) as suggested in [56] as a fifth option, and (6) as a stand alone sixth option. In practice, for the Poisson-Boltzmann problem and similar problems, employing either (1) alone, or (1) and (4) together, seems to be reliable.

4.4 Nonlinear multilevel methods

Nonlinear multilevel methods were developed originally in [29, 80]. These methods attempt to avoid Newton-linearization by accelerating nonlinear relaxation methods with multiple coarse problems. We are again concerned with the problem:

$$F(u) = Au + N(u) - f = 0.$$

Let us introduce the notation $M(\cdot) = A + N(\cdot)$, which yields the equivalent problem:

$$M(u) = f.$$

While there is no direct analogue of the linear error equation in the case of a nonlinear operator $M(\cdot)$, a modified equation for e^n can be used. Given an approximation u^n to the true solution u at iteration n, the equations:

$$r^n = f - M(u^n), \qquad M(u) = M(u^n + e^n) = f,$$

where r^n and e^n are the residual and error, give rise to the expressions:

$$u^n = M^{-1}(f - r^n), \qquad e^n = M^{-1}(f) - u^n,$$

which together give an expression for the error:

$$e^n = (u^n + e^n) - u^n = M^{-1}(f) - M^{-1}(f - r^n).$$

This expression can be used to develop two- and multiple-level methods as in the linear case.

4.4.1 A nonlinear two-level method

Consider now the case of two nested finite-dimensional spaces $\mathcal{H}_{k-1} \subset \mathcal{H}_k$, where $\mathcal{H}_k$ is the fine space and $\mathcal{H}_{k-1}$ is a lower-dimensional coarse space, connected by a prolongation operator $I_{k-1}^k : \mathcal{H}_{k-1} \mapsto \mathcal{H}_k$ and a restriction operator $I_k^{k-1} : \mathcal{H}_k \mapsto \mathcal{H}_{k-1}$. These spaces may, for example, correspond to either the finite element spaces $\mathcal{M}_k$ or the grid function spaces $\mathcal{U}_k$ arising from the discretization of a nonlinear elliptic problem on two successively refined meshes as discussed in §4.1.4.

Assuming that the error can be smoothed efficiently as in the linear case, then the error equation can be solved in the coarser space. If the solution is transferred to the coarse space as $u_{k-1}^n = I_k^{k-1} u_k^n$, then the coarse space source function can be formed as $f_{k-1} = M_{k-1}(u_{k-1}^n)$. Transferring the residual r_k to the coarse space as $r_{k-1}^n = I_k^{k-1} r_k^n$, the error equation can then be solved in the coarse space as:

$$e_{k-1}^n = I_k^{k-1} u_k^n - M_{k-1}^{-1}(M_{k-1}(I_k^{k-1} u_k^n) - I_k^{k-1} r_k^n).$$

The solution is corrected as:

$$\begin{aligned} u_k^{n+1} &= u_k^n + I_{k-1}^k e_{k-1}^n \\ &= u_k^n + I_{k-1}^k [I_k^{k-1} u_k^n - M_{k-1}^{-1}(M_{k-1}(I_k^{k-1} u_k^n) - I_k^{k-1}[f_k - M_k(u_k^n)])] \\ &= K_k(u_k^n, f_k). \end{aligned}$$

Therefore, the nonlinear coarse space correction can be viewed as a fixed-point iteration.

The algorithm implementing the nonlinear error equation is known as the *full approximation scheme* [29] or the *nonlinear multigrid method* [85]. The two-level version of this iteration can be formulated as:

Algorithm 4.4 *(Nonlinear Two-level Method)*

```
(1) Coarse level correction:   $v_k = K_k(u_k^n, f_k)$
(2) Post-smoothing:            $u_k^{n+1} = S_k(v_k, f_k)$.
```

Algorithm 4.4 will require a nonlinear relaxation operator $S_k(\cdot)$ in Step (2), and restriction and prolongation operators as in the linear case, as well as the solution of the nonlinear coarse space equations, to apply the mapping $K_k(\cdot)$ in Step (1).

4.4.2 Nonlinear multilevel methods

We consider now a nested sequence of finite-dimensional spaces $\mathcal{H}_1 \subset \mathcal{H}_2 \subset \cdots \subset \mathcal{H}_J \equiv \mathcal{H}$, where $\mathcal{H}_J$ is the finest space and $\mathcal{H}_1$ the coarsest space, each space being connected to the others via prolongation and restriction operators, as discussed in §4.1.4.

The *multi*-level version of Algorithm 4.4 would employ another two-level method to solve the coarse space problem in Step (1), and can be described recursively as follows:

Algorithm 4.5 *(Nonlinear Multilevel Method)*

$$u^{n+1} = NML(J, u^n, f)$$

where the operation $u_k^{\text{NEW}} = NML(k, u_k^{\text{OLD}}, f_k)$ *is defined recursively:*

```
IF $(k = 1)$ THEN:
    (1) Solve directly:                    $u_1^{\text{NEW}} = M_1^{-1}(f_1)$.
ELSE:
    (1) Restrict residual and solution:    $r_{k-1} = I_k^{k-1}(f_k - M_k(u_k^{\text{OLD}}))$,   $u_{k-1} = I_k^{k-1} u_k^{\text{OLD}}$
    (2) Form coarse source term:           $f_{k-1} = M_{k-1}(u_{k-1}) - r_{k-1}$
    (3) Solve coarse problem:              $w_{k-1} = u_{k-1} - NML(k-1, u_{k-1}, f_{k-1})$
    (4) Prolongate correction:             $w_k = I_{k-1}^k w_{k-1}$
    (5) Find damping parameter:            $\lambda =$ (see below)
    (6) Coarse level correction:           $v_k = u_k^{\text{OLD}} + \lambda w_k$
    (7) Post-smoothing:                    $u_k^{\text{NEW}} = S_k(v_k, f_k)$.
END.
```

The practical aspects of this algorithm and variations are discussed in [29]. A convergence theory has been discussed in [85], and more recently in the sequence of papers [87, 88, 163, 164].

4.4.3 The damping parameter

Note that we have introduced a damping parameter λ in the coarse space correction step of Algorithm 4.5, analogous to the damped-inexact-Newton-multilevel method discussed earlier. In fact, without this damping parameter, the algorithm fails for difficult problems such as those with exponential or rapid nonlinearities (this is also true for the Newton iteration without damping).

To explain how the damping parameter is chosen, we refer back to our discussion of nonlinear conjugate gradient methods. We begin with the following energy functional:

$$J_k(u_k) = \frac{1}{2}(A_k u_k, u_k)_k + B_k(u_k) - (f_k, u_k)_k.$$

As we have seen, the resulting minimization problem:

$$\text{Find } u_k \in \mathcal{H}_k \text{ such that } J_k(u_k) = \min_{v_k \in \mathcal{H}_k} J_k(v_k)$$

is equivalent to the associated zero-point problem:

$$\text{Find } u_k \in \mathcal{H}_k \text{ such that } F_k(u_k) = A_k u_k + N_k(u_k) - f_k = 0,$$

where $N_k(u_k) = B_k'(u_k)$. In other words, $F_k(\cdot)$ is a gradient mapping of the associated energy functional $J_k(\cdot)$, where we assume that both problems above are uniquely solvable.

In [88], it is shown (with suitable conditions on the nonlinear term $B_k(\cdot)$ and satisfaction of a nonlinear form of the variational conditions) that the prolongated coarse space correction $w_k = I_{k-1}^k w_{k-1}$ is a descent direction for the functional $J_k(\cdot)$. Therefore, there exists some $\lambda > 0$ such that

$$J_k(u_k + \lambda w_k) < J_k(u_k).$$

Minimization of $J_k(\cdot)$ along the descent direction w_k is equivalent to solving the following one-dimensional problem:

$$\frac{dJ(u_k + \lambda w_k)}{d\lambda} = 0.$$

As in the discussion of the nonlinear conjugate gradient method, the one-dimensional problem can be solved with Newton's method:

$$\lambda^{m+1} = \lambda^m - \frac{\lambda^m (A_k w_k, w_k)_k - (r_k, w_k)_k + (N_k(u_k + \lambda^m w_k) - N_k(u_k), w_k)_k}{(A_k w_k, w_k)_k + (N_k'(u_k + \lambda^m w_k) w_k, w_k)_k}.$$

Now, recall that the "direction" from the coarse space correction has the form: $w_k = I_{k-1}^k w_{k-1}$. The Newton correction for λ then takes the form:

$$\frac{\lambda^m (A_k I_{k-1}^k w_{k-1}, I_{k-1}^k w_{k-1})_k - (r_k, I_{k-1}^k w_{k-1})_k + (N_k(u_k + \lambda^m I_{k-1}^k w_{k-1}) - N_k(u_k), I_{k-1}^k w_{k-1})_k}{(A_k I_{k-1}^k w_{k-1}, I_{k-1}^k w_{k-1})_k + (N_k'(u_k + \lambda^m I_{k-1}^k w_{k-1}) I_{k-1}^k w_{k-1}, I_{k-1}^k w_{k-1})_k}.$$

If the linear variational conditions are satisfied:

$$A_{k-1} = I_k^{k-1} A_k I_{k-1}^k, \qquad I_k^{k-1} = (I_{k-1}^k)^T, \tag{4.13}$$

then this expression becomes:

$$\frac{\lambda^m (A_{k-1} w_{k-1}, w_{k-1})_{k-1} - (r_{k-1}, w_{k-1})_{k-1} + (I_k^{k-1}(N_k(u_k + \lambda^m I_{k-1}^k w_{k-1}) - N_k(u_k)), w_{k-1})_{k-1}}{(A_{k-1} w_{k-1}, w_{k-1})_{k-1} + (I_k^{k-1} N_k'(u_k + \lambda^m I_{k-1}^k w_{k-1}) I_{k-1}^k w_{k-1}, w_{k-1})_{k-1}}.$$

It can be shown [88] that as in the linear case, a conforming finite element discretization of the nonlinear elliptic problem we are considering, on two successively refined meshes, satisfies the following so-called *nonlinear variational conditions*:

$$A_{k-1} + N_{k-1}(\cdot) = I_k^{k-1} A_k I_{k-1}^k + I_k^{k-1} N_k(I_{k-1}^k \cdot), \qquad I_k^{k-1} = (I_{k-1}^k)^T. \tag{4.14}$$

As in the linear case, these conditions are usually required [87, 88] to show theoretical convergence results about nonlinear multilevel methods. Unfortunately, unlike the linear case, there does not appear to be a way to enforce these conditions algebraically (at least for the strictly nonlinear term $N_k(\cdot)$) in an efficient way. Therefore, if we employ discretization methods other than finite element methods, or cannot approximate the integrals accurately (such as if discontinuities occur within elements on coarser levels) for assembling the discrete nonlinear system, then the variational conditions will be violated. With our algebraic approach, we will have to be satisfied with violation of the nonlinear variational conditions, at least for the strictly nonlinear term $N_k(\cdot)$, in the case of the nonlinear multilevel method.

Our comments earlier with regard to the computational details of this one-dimensional minimization are also valid here. In short, in the case that the a discretization of a nonlinear elliptic partial differential equation gives rise to a nonlinear term $N(\cdot)$ is of diagonal form, meaning that the j-th component is a function of only the j-th component of vector of nodal values u, or $N_j(u) = N_j(u_j)$, then the Jacobian matrix $N'(\cdot)$ is diagonal. The computation of the Newton correction for λ^{m+1} then requires the equivalent of less than three inner-products plus function evaluations (and two inner-products for initialization). Therefore, this calculation is reasonably cheap if few iterations are required.

In [87, 88], an expression is given for λ in an attempt to avoid solving the one-dimensional minimization problem. Certain norm estimates are required in their expression for λ, which depends on the particular nonlinearity; therefore, the full line search approach may be more robust, although more costly.

Remark 4.8. There is an interesting recent result regarding the damping parameter in the linear case, first noticed in [87]. If the nonlinear term $N(\cdot)$ is absent, the zero-point problem is linear and the associated energy functional is quadratic:

$$F_k(u_k) = A_k u_k - f_k = 0, \qquad J_k(u_k) = \frac{1}{2}(A_k u_k, u_k)_k - (f_k, u_k)_k.$$

As in the conjugate gradient algorithm, the calculation of the steplength λ no longer requires the iterative solution of a one-dimensional minimization problem with Newton's method, since:

$$\frac{dJ(u_k + \lambda w_k)}{d\lambda} = \lambda(A_k w_k, w_k)_k - (r_k, w_k)_k = 0$$

yields an explicit expression for λ which minimizes the functional $J_k(\cdot)$ in the direction w_k:

$$\lambda = \frac{(r_k, w_k)_k}{(A_k w_k, w_k)_k}.$$

Since $w_k = I_{k-1}^k w_{k-1}$, we have that:

$$\lambda = \frac{(r_k, w_k)_k}{(A_k w_k, w_k)_k} = \frac{(r_k, I_{k-1}^k w_{k-1})_k}{(A_k I_{k-1}^k w_{k-1}, I_{k-1}^k w_{k-1})_k} = \frac{((I_{k-1}^k)^T r_k, w_{k-1})_{k-1}}{((I_{k-1}^k)^T A_k I_{k-1}^k w_{k-1}, w_{k-1})_{k-1}}.$$

Therefore, if the variational conditions (4.13) are satisfied, the damping parameter can be computed cheaply with only coarse space quantities:

$$\lambda = \frac{(I_k^{k-1} r_k, w_{k-1})_{k-1}}{(I_k^{k-1} A_k I_{k-1}^k w_{k-1}, w_{k-1})_{k-1}} = \frac{(r_{k-1}, w_{k-1})_{k-1}}{(A_{k-1} w_{k-1}, w_{k-1})_{k-1}}.$$

Note that in the two-level case, $w_{k-1} = A_{k-1}^{-1} r_{k-1}$, so that always $\lambda = 1$. Otherwise, numerical experiments show that $\lambda \geq 1$, and it is argued [87] that this is always the case. Adding the parameter λ to the linear multilevel algorithms of Chapter 3 guarantees that the associated functional $J_k(\cdot)$ is minimized along the direction defined by the coarse space correction. A simple numerical example in [87] illustrates that in fact the convergence rate of the linear algorithm can be improved to a surprising degree by employing the damping parameter. Our experience bears this out, and we have incorporated the damping parameter into our linear algorithms as well.

4.5 Nonlinear operator-based prolongation

To explain this attempt at improving on the usual linear prolongation in the case of problems with coefficient discontinuities and rapid nonlinearities, we will use a one-dimensional example as in Chapter 3, where we now consider the nonlinear case:

$$-\frac{d}{dx}\left(a(x)\frac{d}{dx}u(x)\right) + b(x,u(x)) = f(x) \text{ in } (c,d), \qquad u(c) = u(d) = 0. \tag{4.15}$$

The functions $a(x)$ and $b(x,\cdot)$ are positive for all x in $[c,d]$, and $a(x), b(x,\cdot)$, and $f(x)$ are continuously differentiable everywhere, except that one or more of the three may be discontinuous at the *interface* point $x = \xi \in (c,d)$.

Define a discrete mesh $c = x_0 < x_1 < \ldots < x_{n+1} = d$, with $x_{i+1} = x_i + h_i$ for $h_i > 0$, such that the point of discontinuity coincides with some mesh point $x_i = \xi$. For a box-method discretization, we consider the interval $[x_i - h_{i-1}/2, x_i + h_i/2]$ containing the point x_i and integrate (4.15) over the interval. Let us denote the half-mesh points as $x_{i-1/2} = x_i - h_{i-1}/2$ and $x_{i+1/2} = x_i + h_i/2$. After performing the integration of the first term of (4.15) separately over the half-intervals $[x_{i-1/2}, x_i]$ and $[x_i, x_{i+1/2}]$, and enforcing the continuity condition at the interface point $x_i = \xi$

$$\lim_{x\to x_i-} a(x)\frac{d}{dx}u(x) = \lim_{x\to x_i+} a(x)\frac{d}{dx}u(x), \tag{4.16}$$

the following expression is obtained, which is exact for the solution $u(x)$ in the interval:

$$\left(a(x_{i-1/2})\frac{d}{dx}u(x_{i-1/2})\right) - \left(a(x_{i+1/2})\frac{d}{dx}u(x_{i+1/2})\right) + \int_{x_{i-1/2}}^{x_{i+1/2}} b(x,u(x))dx = \int_{x_{i-1/2}}^{x_{i+1/2}} f(x)dx.$$

An algebraic expression is then obtained for an approximation $u_h(x_i)$ to $u(x_i)$ by replacing the derivatives with differences, and replacing the integrals with quadrature formulas separately over the half intervals. Denoting the discretized functions as $u_h(x_i)$, we can for example use centered differences and the rectangle rule:

$$\begin{aligned} a_h(x_{i-1/2})\left(\frac{u_h(x_i) - u_h(x_{i-1})}{h_{i-1}}\right) - a_h(x_{i+1/2})\left(\frac{u_h(x_{i+1}) - u_h(x_i)}{h_i}\right) \\ + \left(\frac{h_{i-1}b_h(x_i^-, u_h(x_i)) + h_i b_h(x_i^+, u_h(x_i))}{2}\right) = \left(\frac{h_{i-1}f_h(x_i^-) + h_i f_h(x_i^+)}{2}\right). \end{aligned} \tag{4.17}$$

If we assume that the nonlinear term $b(\cdot,\cdot)$ is *separable* in the sense that:

$$b(x,u(x)) = \eta(x)\beta(u(x)),$$

such as in the case of the Poisson-Boltzmann equation, then the discrete equations can be written as:

$$\begin{aligned} a_h(x_{i-1/2})\left(\frac{u_h(x_i) - u_h(x_{i-1})}{h_{i-1}}\right) - a_h(x_{i+1/2})\left(\frac{u_h(x_{i+1}) - u_h(x_i)}{h_i}\right) \\ + \left(\frac{h_{i-1}\eta_h(x_i^-) + h_i\eta_h(x_i^+)}{2}\right)\beta_h(u_h(x_i)) = \left(\frac{h_{i-1}f_h(x_i^-) + h_i f_h(x_i^+)}{2}\right). \end{aligned} \tag{4.18}$$

In the stencil form of Chapter 3, this can be written as:

$$\left[\begin{array}{ccc} -W_i & C_i & -E_i \end{array}\right]_h^h \left[\begin{array}{c} u_h(x_{i-1}) \\ u_h(x_i) \\ u_h(x_{i+1}) \end{array}\right] + [D_i]\,\beta_h(u_h(x_i)) = \left[\tilde{f}_h(x_i)\right]. \tag{4.19}$$

In the linear case ($D_i = 0$), the operator-based prolongation was derived in Chapter 3 by attempting to enforce flux conservation at box boundaries when a coarse mesh function was prolongated to a fine mesh. Alternatively, we also noted that this prolongation could be derived by solving (4.19) for a new fine mesh point $u_h(x_i)$, where $u_h(x_{i-1})$ and $u_h(x_{i+1})$ correspond to coarse mesh points (having first been injected to the fine mesh). If the *correction* from the coarse mesh is being prolongated then a zero source function should be used, as this can be interpreted as requiring that the prolongated correction not contribute to increasing the residual of the linear system on the fine mesh. Otherwise, if the *solution* on the coarse mesh is being prolongated, as during a nested iteration, then the true source function should be used.

Using the stencil notation of Chapter 3, the linear prolongation has the form:

$$I_H^h = [\ PE_{i-1} \quad 0 \quad PW_{i+1}\ ]_{H(h)}^h \vee [\ PC_i\ ]_{H(H)}^h,$$

where:

$$PC_i = 1, \qquad PE_{i-1} = \frac{W_i}{C_i}, \qquad PW_{i+1} = \frac{E_i}{C_i}.$$

In the nonlinear case, we cannot simply solve directly for the interpolated fine mesh point $u_h(x_i)$, and instead must solve the following one-dimensional nonlinear problem for $u_h(x_i)$:

$$C_i u_h(x_i) + D_i \beta_h(u_h(x_i)) = F_i + W_i u_h(x_{i-1}) + E_i u_h(x_{i+1}),$$

where $F_i = \tilde{f}_h(x_i)$. We can write this in stencil form by abusing the notation from Chapter 3 a little:

$$I_H^h(u_h) = [\ PN_i(u_h)\ ]_{H(h)}^h \vee [\ PC_i\ ]_{H(H)}^h,$$

where:

$$PC_i = 1,$$

$$PN_i(u_h) = u_h(x_i) + \frac{D_i}{C_i}\beta_h(u_h(x_i)) - \frac{W_i}{C_i}u_h(x_{i-1}) - \frac{E_i}{C_i}u_h(x_{i+1}) - \frac{F_i}{C_i} = 0.$$

To apply the stencil, all coarse mesh points coincident with fine mesh points are first injected, and then for all fine mesh points x_i not lying on a coarse mesh, the one-dimensional zero-point problem $PN_i(u_h) = 0$ must be solved for $u_h(x_i)$. If $D_i = 0$, then the one-dimensional problem is linear, and the prolongation reduces to the linear case.

Remark 4.9. The extensions to two and three dimensions are immediate, by employing the stencil compression ideas of Chapter 3. Since a box-method discretization produces diagonal-type nonlinearities for problems such as the Poisson-Boltzmann equation, the nonlinear problems which must be solved for the prolongation remain one-dimensional; the stencil compression involves only the linear terms.

5. Multilevel Convergence Theory

In this chapter, we examine briefly how multilevel methods are analyzed theoretically. We summarize the thirty year progression of multilevel convergence theory, beginning with the paper of Federenko in 1961, and finishing with the recent work of Bramble, Pasciak, Wang, and Xu in 1991 (the BPWX theory). We review some of the early approaches, identify the two fundamental assumptions used in most of the contemporary theories, and examine when these assumptions are valid. We prove weak two- and multilevel regularity-free results using algebraic generalizations of the existing two- and multilevel theories from the finite element setting, and indicate some possible stronger results for the multilevel case, motivated by some numerical evidence presented later in Chapter 6. We then outline a generalization of the most recently developed finite element-based Schwarz method theories, which we refer to as *product and sum splitting theory*, or *algebraic Schwarz theory*. This theory is an algebraic subspace splitting theory for additive and multiplicative Schwarz methods, representatives of which include both algebraic multilevel and domain decomposition methods.

This chapter provides an overview of multilevel theory, and outlines an adaptation of some results found in the current literature, so as to provide a theory for the algebraic Galerkin methods described in Chapter 3. In particular, what we have done here is the following.[1]

- We develop an operator-based framework for multilevel methods, employing the generalized recursive and product forms of the multilevel error propagation operator derived earlier, applying to both algebraic and finite element-based multilevel methods.
- We give (weak) algebraic two- and multilevel convergence proofs based on this framework. These proofs are equivalent to some existing results for algebraic multilevel methods.
- We outline an algebraic *product and sum splitting theory* for multiplicative and additive Schwarz methods, derived from the Schwarz theory arising in the finite element setting. This theory requires few assumptions, and the convergence results are reasonably good in the algebraic domain decomposition case. The theory can also be used to design effective algebraic multigrid methods.

In addition to the references cited directly in the text below, the material in this chapter owes much to the following sources: [14, 24, 26, 141, 142, 143, 184, 185].

5.1 Classical and modern convergence theories

The modern multilevel convergence theories involve the separation of the partial differential equation from an abstract algorithm in a Hilbert space. A *regularity and approximation assumption* then underlies most theories, which ties the partial differential equation and the abstract algorithm together through a regularity assumption on the solution to the partial differential equation. The verification of this assumption uses duality arguments originating in the finite element literature; these duality arguments require elliptic regularity assumptions (smoothness assumptions) on the solution of the partial differential equation. Unfortunately, solutions to problems such as the Poisson-Boltzmann equation often do not satisfy these regularity assumptions, hence the usual multilevel theories are not valid for these problems. It is only quite recently that a

[1] This material appears in expanded form in [98].

partial theory has emerged for linear equations with discontinuous coefficients and other irregularities.

In this first section, we will review some of the early approaches to multilevel convergence theory, briefly summarize some of the results obtained in the last ten years, discuss in detail the two assumptions which are fundamental to most of these multilevel theories, and then look in detail at a new regularity-free theory which has appeared only very recently. For an explanation of the notation we will use here for multilevel methods, refer to the detailed discussion in Chapter 3.

5.1.1 Variational conditions and regularity-approximation assumptions

Most modern multilevel theories require the so-called *variational conditions*:

$$A_{k-1} = I_k^{k-1} A_k I_{k-1}^k, \qquad I_k^{k-1} = (I_{k-1}^k)^T. \tag{5.1}$$

As we noted in Chapter 3, the variational conditions imply that correction on the coarsest level corresponds to A-orthogonal projection of the error onto the complement of the coarse level space. The variational conditions are satisfied naturally with nested finite element discretizations, and many proofs rely either explicitly on the variational conditions by stating them as assumptions [84, 136, 146], or implicitly by performing the analysis in nested finite element spaces [14, 153].

The linear multilevel framework we constructed in Chapter 3 does not rely on these conditions; however, both the recursive and product forms of the error propagation operator which we derived require that the variational conditions hold. In [85], a convergence theory is given which allows for the violation of (5.1) by a small perturbation, which then appears in the analysis (equation 6.3.27, and pages 147-149, in [85]).

Finally, we remark that even in the case of nested finite element discretizations, the variational conditions (5.1) will hold for the resulting algebraic equations only with *exact* evaluation of the integrals which must be evaluated for the components of the stiffness matrices; see [74] for a detailed and discussion of how quadrature effects multilevel convergence theory.

Most multilevel convergence theories require the so-called regularity and approximation assumption, which in our notation can be written in the form:

$$(A_k(I - P_{k;k-1})u_k, u_k)_k \le C_1 \left(\lambda_k^{-1} \|A_k u_k\|_k^2\right)^{\alpha} (A_k u_k, u_k)^{1-\alpha}, \qquad \forall u_k \in \mathcal{H}_k, \tag{5.2}$$

where $\alpha \in (0, 1]$, and where λ_k is the maximal eigenvalue of A_k. The case of $\alpha = 1$ is called the *full regularity and approximation assumption*:

$$(A_k(I - P_{k;k-1})u_k, u_k)_k \le C_1 \lambda_k^{-1} \|A_k u_k\|_k^2, \qquad \forall u_k \in \mathcal{H}_k. \tag{5.3}$$

The proofs that these assumptions are valid for elliptic partial differential equations hinge on certain duality arguments originating in the finite element literature; refer for example to [36] for a discussion of the *Aubin-Nitsche trick*, also called L^2-lifting. These duality arguments require certain *elliptic regularity assumptions* which we discussed briefly in Chapter 2, as well as assumptions involving the approximation properties of the (finite element) subspaces involved. As discussed in Chapter 2, these elliptic regularity assumptions take the form: there exists a constant C and some $\alpha \in (0, 1]$ such that

$$\|u\|_{H^{1+\alpha}(\Omega)} \le C \|f\|_{H^{\alpha-1}(\Omega)}, \tag{5.4}$$

where $\|\cdot\|_{H^s(\Omega)}$ denotes the usual Sobolev norm of order $s \in \mathbb{R}$. In several papers, in particular [14, 24, 46], the regularity and approximation assumptions have been analyzed in detail, and it is shown that in certain situations, the assumption (5.4) implies assumption (5.2).

To prove that the regularity assumption (5.4) implies the regularity and approximation assumption (5.2) requires a certain discrete norm equivalence result, first given as Lemma 1 in [14], which can be shown for finite element spaces based on quasi-uniform meshes. By combining this norm equivalence result with the standard approximation properties of finite element spaces, and using duality arguments similar to the Aubin-Nitsche trick, it can be shown (cf. Proposition 5.1 in [24]) that (5.4) implies (5.2) for the same α. We note that also in [24], examples are given which imply that the two results are strongly related, in that if (5.4) is violated, then so is (5.2).

Until recently, all multilevel theories required assumption (5.2) for p-cycle results, and the full regularity and approximation assumption (5.3) for v-cycle results. Unfortunately, in the case of problems with discontinuous coefficients such as the Poisson-Boltzmann equation, the elliptic regularity inequalities like (5.4) either do not hold at all, or hold only with extremely large constants C. If it is known only that $u \in H^1(\Omega)$, then the proof techniques relying on the regularity estimates cannot be used. Note that it can be demonstrated numerically that even in the case of discontinuous coefficients, multilevel methods can often be made to yield optimal order behavior, or nearly so. Therefore, it has long been debated whether the regularity assumption is really required to prove optimal multigrid convergence results [46, 141]. We remark that two-level convergence theories, not requiring elliptic regularity assumptions, have been known for some time [30, 85, 138, 166]; we will outline a two-level theory using the framework of Chapter 3 later in this chapter.

Recently, a new theory (the BPWX theory) of multilevel methods has appeared in [26], avoiding the use of elliptic regularity assumptions by requiring only approximation assumptions. To apply the theory to a multigrid algorithm for an elliptic equation, one must establish the existence of certain projection operators relating the finite element subspaces $\mathcal{M}_J$, which satisfy two simple properties. The key to this theory is a certain product form of the multilevel error propagation operator, which is a special case of the generalized product form we derived in Chapter 3, arising when the prolongation and restriction operators are taken to be the natural inclusion and orthogonal projection, respectively.

As we will discuss in more detail shortly, in the case of elliptic equations with discontinuous coefficients with discontinuities lying along element boundaries of a finite element mesh, it can be shown that there exists operators which satisfy these two properties, and hence the convergence theory is applicable. The resulting convergence property that can be proven has the following form:

$$\|E^s\|_A \leq \delta_J = 1 - \frac{C}{J^\nu} < 1,$$

where E^s is the symmetric multilevel error propagator, J is the number of levels in the algorithm, and $\nu = 1$ for simple discontinuities, or more generally $\nu = 2$. Thus, the contraction number decays with J, which gives rise to a logarithmic factor in the complexity estimates.

5.1.2 Brief summary of early approaches

The formulation of a multilevel method appeared first in the Russian literature in [64], although block relaxation methods which are similar in some respects are described earlier in the west [173, 180]. In his 1961 paper Fedorenko described a two-level method for solving elliptic equations, and in a second paper from 1964 [65] proved convergence of a multilevel method for Poisson's equation on the square. In a paper from 1966 [11], Bakhvalov extended these results to more general elliptic operators on rectangles, and showed that the resulting methods were of optimal order. Astrakhantsev was the first to consider the multilevel method in a variational setting in 1971 [8], and showed convergence for more general elliptic operators on fairly general domains with a finite element discretization.

After 1975, papers on multilevel methods began appearing in Europe and the United States. In 1977, Nicolaides [153] gave another convergence proof for a finite element discretization, but was unaware of the work of Astrakhantsev. Brandt began to use the methods in the context of fluid dynamics applications in 1972 [28], and published the first comprehensive paper on multilevel methods in 1977 [29]. In 1980, Bank and Dupont gave convergence proofs of two different multilevel methods using finite element discretizations [13, 14]. Hackbusch began a series of papers on convergence theory in 1976 [78, 79], and continued with various generalizations [23, 81, 82, 83, 84], which culminated with the publication of his comprehensive book in 1985 [85].

For model problems such as Poisson's equation on the unit square, a discrete Fourier analysis can be used to construct a similarity transformation which block-diagonalizes the multilevel error propagator $E_k = I - B_k A_k$; the size of the diagonal blocks reflect the dimension of invariant subspaces under multiplication by E_k. The convergence rate can then be computed directly from the diagonal blocks. The early convergence proofs were generally of this form [11, 65, 78, 175], and as a result were restricted to model problems.

Since 1980, many multilevel convergence proofs have been published (an *incomplete* list numbers over forty). The earliest approaches for more general problems [8, 14, 57, 84, 136, 143, 146, 153, 182] were for the p-cycle only ($p > 1$), and convergence was shown only for "sufficiently many" smoothings per iteration. The proof techniques involved analyzing the two-level scheme and deriving multilevel estimates indirectly from two-level estimates. The p-cycle proofs typically require only the weaker regularity and approximation assumption (5.2) for some $\alpha > 0$.

The first v-cycle proof for any number of smoothings per iteration was [23], and involved a direct analysis of the multilevel method rather than the extension of two-level results. More recent v-cycle proofs for any positive number of smoothings per iteration include [12, 23, 24, 137, 142, 144, 145, 188]. The v-cycle proofs typically require the full regularity and approximation assumption (5.3).

5.1.3 Modern approaches based on product error propagators

In this section, we outline a simple abstract operator-based approach, based on the product multilevel operator framework constructed in Chapter 3. This framework was derived as a generalization of the BPWX-theory discussed at the end of the chapter. Implicit in this approach are the variational conditions (5.1); these conditions are essentially the only assumptions, so that the theory applies to the algebraic Galerkin two-level and multilevel methods of Chapter 3, as well as to finite element-based multilevel methods.

Recall the operator-form of the multilevel algorithm for solving the operator equation $Au = f$ in the finest space of a nested sequence of spaces $\mathcal{H}_1 \subset \mathcal{H}_2 \subset \cdots \subset \mathcal{H}_J \equiv \mathcal{H}$, presented as Algorithm 3.6 in Chapter 3. We repeat the algorithm here for reference; in this chapter, we will generally leave off the subscripts J for quantities in the finest space $\mathcal{H} \equiv \mathcal{H}_J$, without danger of confusion.

Algorithm 5.1 *(Nonsymmetric Multilevel Method – Operator Form)*

$$u^{n+1} = u^n + B(f - Au^n)$$

where the multilevel operator $B \equiv B_J$ is defined by the recursion:

(1) Let $B_1 = A_1^{-1}$, and assume B_{k-1} has been defined.
(2) $B_k = I_{k-1}^k B_{k-1} I_k^{k-1} + R_k - R_k A_k I_{k-1}^k B_{k-1} I_k^{k-1}$.

In the algorithm, the operator R_k is the smoothing operator, I_{k-1}^k and I_k^{k-1} are the prolongation and restriction operators, and B_k is the approximate inverse of the operator A_k which is implicitly defined by the multilevel algorithm; for details, refer to the discussion and derivation in Chapter 3.

From this algorithm, we derived in Lemma 3.21 of Chapter 3 (based on [26]) the generalized recursive form of the multilevel error propagator at each level k, which is:

$$E_k = I - B_k A_k = (I - R_k A_k)(I - P_{k;k-1} + I_{k-1}^k E_{k-1} \tilde{P}_k^{k-1}), \tag{5.5}$$

where the operator $P_{k;k-1}$ is the A_k-orthogonal projector from $\mathcal{H}_k$ onto $I_{k-1}^k \mathcal{H}_{k-1}$, and the operator $\tilde{P}_k^{k-1}$ is a related operator, given by $P_{k;k-1} = I_{k-1}^k \tilde{P}_k^{k-1}$.

We also derived in Lemma 3.22 of Chapter 3 the generalized product form for the multilevel error propagator:

$$E = I - BA = (I - T_J)(I - T_{J-1}) \cdots (I - T_1), \tag{5.6}$$

where

$$I_J = I, \qquad I_k = I_{J-1}^J I_{J-2}^{J-1} \cdots I_{k+1}^{k+2} I_k^{k+1}, \qquad k = 1, \ldots, J-1,$$

$$T_1 = I_1 A_1^{-1} I_1^T A, \qquad T_k = I_k R_k I_k^T A, \qquad k = 2, \ldots, J.$$

Note that in the derivations of the forms in equations (5.5) and (5.6), it was required that the variational conditions (5.1) hold. We remarked in Chapter 3 that the error propagator of the symmetric multilevel algorithm E^s can be written as $E^s = EE^*$, where E^* is the A-adjoint of E.

Many of the simple results we showed in Chapter 3 which related the properties of A_k and B_k to those of the error propagator $E_k = I - B_k A_k$ required not only that A_k be SPD, but that B_k be SPD as well. Therefore, we wish to establish a simple condition on the smoothing operator R_k that will ensure that the multilevel operators B_k be SPD. The following result was given in [184], along with an abbreviated proof; the proof we give below is original.

Lemma 5.1 *If the smoothing operator R_k corresponds to any convergent linear method, then the symmetric multilevel operator B_k is also positive.*

Proof. Consider the symmetric two-level error propagator from Chapter 3:

$$E_k^s = I - B_k A_k = (I - R_k A_k)(I - C_k A_k)(I - R_k^T A_k),$$

for some approximate coarse level inverse operator C_k. In the multilevel case, following our discussion in Chapter 3 for the nonsymmetric case, the symmetric multilevel error propagator is defined in terms of the sequence of recursively defined multilevel operators B_k:

$$E_k^s = I - B_k A_k = (I - R_k A_k)(I - I_{k-1}^k B_{k-1} I_k^{k-1} A_k)(I - R_k^T A_k). \tag{5.7}$$

Now, for arbitrary $v_k \neq 0$, consider:

$$(A_k E_k^s v_k, v_k)_k = (A_k (I - B_k A_k) v_k, v_k)_k = (A_k v_k, v_k)_k - (B_k A_k v_k, A_k v_k)_k.$$

Using this result, and employing the product expression (5.7) for E_k^s, we have:

$$\begin{aligned}
(B_k A_k v_k, A_k v_k)_k &= (A_k v_k, v_k)_k - (A_k E_k^s v_k, v_k)_k \\
&= (A_k v_k, v_k)_k - (A_k (I - R_k A_k)(I - I_{k-1}^k B_{k-1} I_k^{k-1} A_k)(I - R_k^T A_k) v_k, v_k)_k \\
&= (A_k v_k, v_k)_k - (A_k (I - I_{k-1}^k B_{k-1} I_k^{k-1} A_k)(I - R_k^T A_k) v_k, (I - R_k^T A_k) v_k)_k \\
&= (A_k v_k, v_k)_k - (A_k (I - R_k^T A_k) v_k, (I - R_k^T A_k) v_k)_k + (I_{k-1}^k B_{k-1} I_k^{k-1} A_k w_k, A_k w_k)_k \\
&= (A_k v_k, v_k)_k - (A_k S_k^* v_k, S_k^* v_k)_k + (B_{k-1} I_k^{k-1} A_k w_k, I_k^{k-1} A_k w_k)_{k-1},
\end{aligned}$$

where $S_k = I - R_k A_k$, and $w_k = (I - R_k^T A_k) v_k$, and where we have assumed the variational conditions with the restriction I_k^{k-1} equal to the adjoint of the prolongation I_{k-1}^k, taken with respect to the inner-products $(\cdot, \cdot)_k$ as discussed in Chapter 3. Now, if R_k corresponds to a convergent linear method with error propagator $S_k = I - R_k A_k$, we have that $\rho(S_k) < 1$. For simplicity, we assume the stronger sufficient condition as well, $\|S_k\|_{A_k} = \|S_k^*\|_{A_k} < 1$. This implies that:

$$(A_k S_k^* v_k, S_k^* v_k)_k < (A_k v_k, v_k)_k, \quad \forall v_k \in \mathcal{H}_k, \quad v_k \neq 0.$$

From above we then have:

$$(B_k A_k v_k, A_k v_k)_k > (B_{k-1} v_{k-1}, v_{k-1})_{k-1}, \quad \forall v_k \in \mathcal{H}_k,$$

where $v_{k-1} = I_k^{k-1} A_k (I - R_k^T A_k) v_k$. Since A_k is SPD, A_k^{-1} is as well, and the proof now follows by induction on B_k, since $B_1 = A_1^{-1}$ is SPD. □

5.1.4 A two-level convergence theorem

We outline a two-level theory which does not require regularity assumptions. Although our notation is somewhat different, this result for a nonsymmetric two-level method was derived from a result in [184] for symmetric two-level methods. Both results are equivalent to results that have appeared in [30, 126, 139, 140, 166].

To begin, we assume that the variational conditions (5.1) hold. Let $S_k = I - R_k A_k$, where R_k is the smoothing operator. The following two assumptions are required:

Assumption 5.1 *(Approximation assumption) There exists $C_1 > 0$ such that:*

$$\inf_{w_{k-1} \in \mathcal{H}_{k-1}} \|v_k - I_{k-1}^k w_{k-1}\|_k^2 \le C_1 \lambda_k^{-1} \|v_k\|_{A_k}^2, \quad \forall v_k \in \mathcal{H}_k.$$

Assumption 5.2 *(Smoothing assumption) There exists $C_2 > 0$ such that:*

$$C_2 \|A_k v_k\|_k^2 \le \lambda_k (A_k (I - S_k^* S_k) v_k, v_k)_k, \quad \forall v_k \in \mathcal{H}_k.$$

Assumption 5.2 can be verified for Richardson's iteration and other classical smoothing operators for finite element and box-method discretizations of elliptic equations; see for example [25, 26] for detailed discussions. Assumption 5.1 can be verified in the case that A_k has "1-regularity" as described in [85]; this will always be true for a two-level method and a discretization on the fine mesh. However, it is unreasonable to expect this condition to hold, independent of the number of levels, in a general variational definition of the operators A_k. The discussions in this section are therefore limited to two levels.

Now, recall the nonsymmetric two-level error propagation operator:

$$E_k = I - B_k A_k = (I - R_k A_k)(I - C_k A_k) = S_k(I - P_{k;k-1}),$$

where $S_k = I - R_k A_k$, and where $P_{k;k-1}$ is the A_k-orthogonal projector from the fine space $\mathcal{H}_k$ onto the coarse space $I_{k-1}^k \mathcal{H}_{k-1}$. To establish the two-level result, we must employ the following simple lemma, which is found in [184]. This result establishes an inequality similar to the full regularity and approximation assumption (5.3) using only the weak (approximation) Assumption 6.1 above. Note that the inequality below differs from (5.3) in that the right hand side is restricted to the A_k-orthogonal projected component $(I - P_{k;k-1})u_k$ rather than u_k defined over the whole space $\mathcal{H}_k$.

Lemma 5.2 *Under Assumption 5.1, it holds that:*

$$(A_k(I - P_{k;k-1})u_k, u_k)_k \leq C_1 \lambda_k^{-1} \|A_k(I - P_{k;k-1})u_k\|_k^2, \qquad \forall u_k \in \mathcal{H}_k.$$

Proof. First, note that by the Cauchy-Schwarz inequality, we have:

$$\begin{aligned}(A_k(I - P_{k;k-1})u_k, u_k)_k &= (A_k(I - P_{k;k-1})u_k, (I - P_{k;k-1})u_k)_k \\ &= (A_k(I - P_{k;k-1})u_k, (I - P_{k;k-1})u_k - I_{k-1}^k w_{k-1})_k \\ &\leq \|A_k(I - P_{k;k-1})u_k\|_k \|(I - P_{k;k-1})u_k - I_{k-1}^k w_{k-1}\|_k.\end{aligned}$$

Now, by Assumption 5.1 we have that

$$\begin{aligned}(A_k(I - P_{k;k-1})u_k, u_k)_k &\leq \|A_k(I - P_{k;k-1})u_k\|_k [C_1 \lambda_k^{-1} (A_k(I - P_{k;k-1})u_k, (I - P_{k;k-1})u_k)_k]^{1/2} \\ &= C_1^{1/2} \lambda_k^{-1/2} \|A_k(I - P_{k;k-1})u_k\|_k (A_k(I - P_{k;k-1})u_k, u_k)_k^{1/2},\end{aligned}$$

which, after division and a squaring of both sides, yields:

$$(A_k(I - P_{k;k-1})u_k, u_k)_k \leq C_1 \lambda_k^{-1} \|A_k(I - P_{k;k-1})u_k\|_k^2.$$

□

A convergence result for the nonsymmetric two-level method is stated in the following theorem. Again, although our notation is somewhat different, this result is derived directly from a two-level result in [184] for symmetric two-level methods.

Theorem 5.3 *Under Assumption 5.1 and Assumption 5.2, the error propagator of the nonsymmetric two-level algorithm (Algorithm 5.1 for $J = k = 2$) satisfies:*

$$\|E_k\|_{A_k}^2 \leq \delta_k = 1 - \frac{C_2}{C_1} < 1.$$

Proof. Since we must have that

$$\|E_k\|_{A_k}^2 = \max_{0 \neq v_k \in \mathcal{H}_k} \frac{\|E_k v_k\|_{A_k}^2}{\|v_k\|_{A_k}^2} \leq \delta,$$

it suffices to show that

$$\|E_k v_k\|_{A_k}^2 \leq \delta_k \|v_k\|_{A_k}^2, \qquad \forall v_k \in \mathcal{H}_k,$$

or equivalently,

$$(A_k E_k v_k, E_k v_k)_k \leq \delta_k (A_k v_k, v_k)_k, \qquad \forall v_k \in \mathcal{H}_k.$$

We begin by noting that:

$$(A_k E_k v_k, E_k v_k)_k = (A_k E_k^* E_k v_k, v_k)_k$$
$$= (A_k(I - P_{k;k-1})S_k^* S_k(I - P_{k;k-1})v_k, v_k)_k = (A_k S_k^* S_k(I - P_{k;k-1})v_k, (I - P_{k;k-1})v_k)_k$$
$$= (A_k(I - P_{k;k-1})v_k, (I - P_{k;k-1})v_k)_k - (A_k(I - S_k^* S_k)(I - P_{k;k-1})v_k, (I - P_{k;k-1})v_k)_k$$
$$= (A_k(I - P_{k;k-1})v_k, v_k)_k - (A_k(I - S_k^* S_k)(I - P_{k;k-1})v_k, (I - P_{k;k-1})v_k)_k.$$

By Assumption 5.2, the above result implies that

$$(A_k E_k v_k, E_k v_k)_k \leq (A_k(I - P_{k;k-1})v_k, v_k)_k - C_2\lambda_k^{-1}\|A_k(I - P_{k;k-1})v_k\|_k^2.$$

By Lemma 5.2 (which employed only Assumption 5.1),

$$\|A_k(I - P_{k;k-1})u_k\|_k^2 \geq \frac{\lambda_k}{C_1}(A_k(I - P_{k;k-1})u_k, u_k)_k,$$

so that

$$(A_k E_k v_k, E_k v_k)_k \leq (A_k(I - P_{k;k-1})v_k, v_k)_k - \frac{C_2}{C_1}(A_k(I - P_{k;k-1})v_k, v_k)_k$$
$$= \left(1 - \frac{C_2}{C_1}\right)(A_k(I - P_{k;k-1})v_k, v_k)_k.$$

Since $I - P_{k;k-1}$ is the A_k-orthogonal projector, $\|I - P_{k;k-1}\|_{A_k} = 1$, so that

$$\|I - P_{k;k-1}\|_{A_k} = \max_{v_k \neq 0} \frac{(A_k(I - P_{k;k-1})v_k, (I - P_{k;k-1})v_k)_k}{(A_k v_k, v_k)_k} = 1.$$

But this implies that

$$(A_k(I - P_{k;k-1})v_k, v_k)_k = (A_k(I - P_{k;k-1})v_k, (I - P_{k;k-1})v_k)_k \leq (A_k v_k, v_k)_k, \qquad \forall v_k \in \mathcal{H}_k,$$

which gives

$$(A_k E_k v_k, E_k v_k)_k \leq \left(1 - \frac{C_2}{C_1}\right)(A_k v_k, v_k)_k, \qquad \forall v_k \in \mathcal{H}_k,$$

which is what we needed to show. □

5.1.5 A multilevel convergence theorem

We can show the following multilevel result, and although it gives no indication of the rate of convergence, it shows the robustness which is gained by enforcing the variational conditions. The simple proof relies on the general product form of the multilevel error propagator, and on Lemma 3.8 from Chapter 3. Similar conclusions are reached in [166, 184] by different approaches.

Theorem 5.4 *If variational conditions (5.1) hold and the smoothing iteration is convergent, then the error propagator of the nonsymmetric multilevel algorithm (Algorithm 5.1 for $J \geq 2$) satisfies:*

$$\|E\|_A^2 < 1.$$

Proof. Since the variational conditions hold, the coarse level product term is the A- orthogonal projector as in the two-level case:

$$I - T_1 = I - I_1 A_1^{-1} I_1^T A = I - P_1.$$

Therefore, $I - T_1 = (I - T_1)^2 = (I - T_1)(I - T_1)^*$, and:

$$E^s = (I - T_J)\cdots(I - T_2)(I - T_1)(I - T_2)^* \cdots (I - T_J)^*$$
$$= (I - T_J)\cdots(I - T_2)(I - T_1)(I - T_1)^*(I - T_2)^* \cdots (I - T_J)^*$$
$$= EE^*.$$

Since $E^s = EE^*$, and since A is SPD, we have that:

$$(AE^s v, v) = (AE^* v, E^* v) = (Aw, w) \geq 0 \quad \forall w \in \mathcal{H},$$

which implies that $E^s = I - BA$ is A-non-negative. Since R_k corresponds to a convergent linear method with $S_k = I - R_k A_k$, by Lemma 5.1 the recursively defined B is SPD, and so by Lemma 3.8 of Chapter 3 the A-non-negativity of E^s implies that $\|E^s\|_A < 1$. Finally, we have that

$$\|E\|_A^2 = \|EE^*\|_A = \|E^s\|_A < 1.$$

□

Remark 5.1. It is shown in [184] that E is A-positive under the same assumptions as above; however the sufficiency of this condition for convergence through the use of a result as in Lemma 3.8 was not mentioned.

In §5.1.6, we will discuss the BPWX theory developed in [26], which can be seen as a special case of the recursive multilevel operator approach discussed in the previous sections. The BPWX theory yields contraction number bounds of the form:

$$\|E\|_A^2 \leq \delta_J = 1 - \frac{C}{J^\nu}, \tag{5.8}$$

for some constant C, where J is the number of levels in the multilevel algorithm, and $\nu = 1$ if the discontinuities are somewhat simple, or in the worst case $\nu = 2$. This gives convergence rates that decay as the number of levels increases, which implies a logarithmic term in the overall complexity of the algorithm, since the number of levels J is related to the number of unknowns n_J through $J = O(\ln n_J)$. The computational evidence in [26] suggests that this type of a bound is of the correct form for finite element-based multilevel methods applied to problems with discontinuous coefficients, the only restriction being that the discontinuities lie along element boundaries on all coarse meshes.

In this work, we have been concerned with algorithms for problems with discontinuous coefficients, and we are faced with the situation that discontinuities may not lie along element boundaries on coarse meshes. We described in detail in Chapter 3, as well as in Appendix A, how the variational conditions (5.1) can be enforced algebraically in an efficient way for matrices arising from box-method or finite element method discretization of second order elliptic problems on non-uniform Cartesian meshes in one-, two-, and three-dimensional meshes. Unfortunately, while our multilevel algorithms therefore satisfy the variational conditions (5.1), the coarse level spaces do not in general satisfy the approximation assumptions required for the BPWX theory of the next section.

However, the computational evidence we will present later in Chapter 6 for a difficult jump discontinuity test problem suggests that even in the case that discontinuities do not lie along element boundaries on coarse meshes, our algorithms in fact also demonstrate the same type of contraction number decay given in (5.8), for some $\nu \leq 1$.

Therefore, we suspect that it should be possible to show a similar result for a completely algebraic multilevel approach, in which only the variational conditions hold, along with weak smoothing and approximation assumptions similar to Assumptions 6.1 and 6.2.

5.1.6 Bramble-Pasciak-Wang-Xu regularity-free theory

We outline the main ideas and assumptions in the theory, and give the main convergence result appearing in [26].

The framework which we constructed in Chapter 3 was based on that developed in [26], with a generalization regarding the restriction and prolongation operators. By selecting the restriction and prolongation operators to be orthogonal projection and inclusion, respectively, the BPWX theory results. By choosing the restriction as the orthogonal projection:

$$I_k^T \equiv Q_k,$$

it is not difficult to show that that: $Q_k A = A_k P_k$. The product form then becomes:

$$E = I - BA = (I - T_J)(I - T_{J-1}) \cdots (I - T_1),$$

where

$$T_1 = P_1, \qquad T_k = R_k A_k P_k, \qquad k = 2, \ldots, J.$$

With this form of the product operator, a very general result can be shown.

The convergence result requires three assumptions on the operators appearing above. The first two assumptions are on the orthogonal projectors Q_k, $k = 1, \ldots, J$, and are as follows.

$$\|(Q_k - Q_{k-1})u\|_k^2 \leq C_1 \lambda_k^{-1} (Au, u), \qquad \forall u \in \mathcal{H}, \tag{5.9}$$

$$(A_k Q_k u, Q_k u)_k \leq C_2 (Au, u), \qquad \forall u \in \mathcal{H}. \tag{5.10}$$

The third assumption is on the smoothing operators R_k, $k = 1, \ldots, J$.

$$\|u_k\|_k^2 \leq \lambda_k C_3 (R_k u_k, u_k)_k, \qquad \forall u_k \in \mathcal{V}_k \subseteq \mathcal{H}_k. \tag{5.11}$$

The main convergence result is given in the following theorem.

Theorem 5.5 *(Bramble-Pasciak-Wang-Xu) Assume (5.9), (5.10), and (5.11) hold; then*

$$\|E\|_A^2 \leq \delta_J = 1 - \frac{C}{J^\nu} < 1,$$

where $C = [1 + C_2^{1/2} + (C_3 C_1)^{1/2}]^{-2}$, and where $\nu \geq 1$.

Proof. See Theorem 1, page 29 in [26]. □

To apply the theorem to finite element-based multilevel methods, allowing for discontinuous coefficients, the existence of projection operators Q_k satisfying assumptions (5.9) and (5.10) must be established. In the case that the coefficient discontinuities lie only along element boundaries, the following result can be shown; it is stated in [26], and a proof is given in [184].

Lemma 5.6 *(Bramble-Pasciak-Wang-Xu) Assume $\mathcal{H}_k = \mathcal{M}_k$, the finite element spaces. If all coefficient discontinuities lie along element boundaries on all levels, then there exists L^2-like projectors Q_k for which (5.9) and (5.10) hold.*

Proof. See Lemma 6.1, page 40 in [26]. □

Finally, the third assumption (5.11) on the smoothing operators R_k can be shown very generally for many of the classical linear iterations, as discussed in detail in [25].

Lemma 5.7 *(Bramble-Pasciak-Wang-Xu) Assume $\mathcal{H}_k = \mathcal{M}_k$, the finite element spaces. The classical iterations Richardson, weighted Jacobi, and Gauss-Seidel as smoothing operators are such that (5.11) holds.*

Proof. See [25]. □

5.1.7 An algebraic Schwarz (product/sum-splitting) theory

For the remainder of the chapter, we discuss the general theory of additive and multiplicative Schwarz methods for self-adjoint positive linear operator equations, representative methods of particular interest here being multigrid and domain decomposition. We examine closely one of the most useful and elegant modern convergence theories for these methods, following closely the recent work of Dryja and Widlund, Xu, and their colleagues. Our motivation is to fully understand this theory, and then to develop a variation of the theory in a slightly more general setting, which will be useful in the analysis of algebraic multigrid and domain decomposition methods, when little or no finite element structure is available. Using this approach we can show some convergence results for a very broad class of fully algebraic domain decomposition methods, without regularity assumptions about the continuous problem. Although we cannot at this time use the theory to provide a "good" convergence theory for algebraic multigrid methods, we believe that with additional analysis it may be possible to do so using this framework, as well as to use the framework to guide the design of the coarse problems. The language we employ throughout is algebraic, and can be interpreted abstractly in terms of operators on Hilbert spaces, or in terms of matrix operators.

Our approach in the following sections is quite similar (and owes much) to [185], with the following exceptions. We first develop a separate and complete theory for products and sums of operators, without reference to subspaces, and then use this theory to formulate a Schwarz theory based on subspaces. In addition, we develop the Schwarz theory allowing for completely general prolongation and restriction operators, so that the theory is not restricted to the use of inclusion and projection as the transfer operators (a similar Schwarz framework with general transfer operators was constructed recently by Hackbusch [86]). The resulting theoretical framework is useful for analyzing specific algebraic methods, such as algebraic multigrid and algebraic domain decomposition, without requiring the use of finite element spaces (and their associated transfer operators of inclusions and projection). The framework may also be useful for analyzing methods based on transforms to other spaces not naturally thought of as subspaces, such as methods based on successive wavelet or other transforms. Finally, we show quite clearly how the basic product/sum and Schwarz theories must be modified and refined to analyze the effects of using a global operator, or of using additional nested spaces as in the case of multigrid-type methods. We also present (adding somewhat to the length of an already lengthy discussion) a number of (albeit simple but useful) results in the product/sum and Schwarz theory frameworks which are commonly used in the literature, the proofs of which are often difficult to locate. The result is a consistent and self-contained theoretical framework for analyzing abstract linear methods for self-adjoint positive linear operator equations, based on subspace-decomposition ideas.

Outline of the remainder of the chapter

We now give a more detailed outline of the material which follows. We will use the notation from §3.1, and assume familiarity with the material on linear operators, linear methods, and conjugate gradient methods presented there.

In §5.2, we present a unified approach for bounding the norms and condition numbers of products and sums of self-adjoint operators on a Hilbert space, derived from work due to Dryja and Widlund [59], Bramble et al. [27], and Xu [185]. Our particular approach is quite general in that we establish the main norm and condition number bounds without reference to subspaces; each of the three required assumptions for the theory involve only the operators on the original Hilbert space. Therefore, this product/sum operator theory may find use in other applications without natural subspace decompositions. Later, we will apply the product and sum operator theory to the case when the operators correspond to corrections in subspaces of the original space, as in multigrid and domain decomposition methods.

In §5.3, we consider abstract Schwarz methods based on subspaces, and apply the general product and sum operator theory to these methods. The resulting theory, which is a variation of that presented in [185] and [59], rests on the notion of a stable subspace splitting of the original Hilbert space (cf. [159, 165]). Although our derivation here is presented in a somewhat different, algebraic language, many of the intermediate results we use have appeared previously in the literature in other forms (we provide references at the appropriate points). In contrast to earlier approaches, we develop the entire theory employing general prolongation and restriction operators; the use of inclusion and projection as prolongation and restriction are represented in our approach as a special case.

In §5.4 and §5.5, we apply the theory derived earlier to domain decomposition methods and to multigrid methods, and in particular to their algebraic forms. Since our theoretical framework allows for general prolongation and restriction operators, the theory is applicable to methods for general algebraic equations (coming from finite difference or finite volume discretization of elliptic equations) for which strong theories are currently lacking. For algebraic domain decomposition, we are able to derive useful (although not optimal) convergence estimates. Although the algebraic multigrid results are not as interesting, the theory does provide yet another proof of the robustness of the algebraic multigrid approach. We also indicate how the convergence estimates for multigrid and domain decomposition methods may be improved (giving optimal estimates), following the recent work of Dryja and Widlund, Bramble et al., and Xu, which requires some of the additional structure provided in the finite element setting.

In addition to the references cited directly in the text below, the material here owes much to the following sources: [14, 24, 26, 58, 59, 86, 141, 142, 143, 184, 185].

5.2 The theory of products and sums of operators

In this section, we present an approach for bounding the norms and condition numbers of products and sums of self-adjoint operators on a Hilbert space, derived from work due to Dryja and Widlund [59], Bramble et al. [27], and Xu [185]. This particular approach is quite general in that we establish the main norm and condition number bounds without reference to subspaces; each of the three required assumptions for the theory involve only the operators on the original Hilbert space. Therefore, this product/sum operator theory may find use in other applications without natural subspace decompositions. Later in the paper, the product and sum operator theory is applied to the case when the operators correspond to corrections in subspaces of the original space, as in multigrid and domain decomposition methods.

5.2.1 Basic product and sum operator theory

Let $\mathcal{H}$ be a real Hilbert space equipped with the inner-product $(\cdot,\cdot)$ inducing the norm $\|\cdot\| = (\cdot,\cdot)^{1/2}$. Let there be given an SPD operator $A \in \mathbf{L}(\mathcal{H},\mathcal{H})$ defining another inner-product on $\mathcal{H}$, which we denote as $(\cdot,\cdot)_A = (A\cdot,\cdot)$. This second inner-product also induces a norm $\|\cdot\|_A = (\cdot,\cdot)_A^{1/2}$. We are interested in general product and sum operators of the form

$$E = (I - T_J)(I - T_{J-1}) \cdots (I - T_1), \tag{5.12}$$

$$P = T_1 + T_2 + \cdots + T_J, \tag{5.13}$$

for some A-self-adjoint operators $T_k \in \mathbf{L}(\mathcal{H},\mathcal{H})$. If E is the error propagation operator of some linear method, then the convergence rate of this linear method will be governed by the norm of E. Similarly, if a preconditioned linear operator has the form of P, then the convergence rate of a conjugate gradient method employing this system operator will be governed by the condition number of P.

The A-norm is convenient here, as it is not difficult to see that P is A-self-adjoint, as well as $E^s = EE^*$. Therefore, we will be interested in deriving bounds of the form:

$$\|E\|_A^2 \leq \delta < 1, \qquad \kappa_A(P) = \frac{\lambda_{\max}(P)}{\lambda_{\min}(P)} \leq \gamma. \tag{5.14}$$

The remainder of this section is devoted to establishing some minimal assumptions on the operators T_k in order to derive bounds of the form in equation (5.14). If we define $E_k = (I - T_k)(I - T_{k-1}) \cdots (I - T_1)$, and define $E_0 = I$ and $E_J = E$, then we have the following relationships.

Lemma 5.8 *The following relationships hold for $k = 1, \ldots, J$:*

(1) $E_k = (I - T_k)E_{k-1}$

(2) $E_{k-1} - E_k = T_k E_{k-1}$

(3) $I - E_k = \sum_{i=1}^k T_i E_{i-1}$

Proof. The first relationship is obvious from the definition of E_k, and the second follows easily from the first. Taking $E_0 = I$, and summing the second relationship from $i = 1$ to $i = k$ gives the third. □

Regarding the operators T_k, we make the following assumption:

Assumption 5.3 *The operators $T_k \in \mathbf{L}(\mathcal{H},\mathcal{H})$ are A-self-adjoint, A-non-negative, and*

$$\rho(T_k) = \|T_k\|_A \leq \omega < 2, \quad k = 1, \ldots, J.$$

Note that this implies that $0 \leq \lambda_i(T_k) \leq \omega < 2,\ k = 1, \ldots, J$.

The following simple lemma, first appearing in [27], will often be useful at various points in the analysis of the product and sum operators.

Lemma 5.9 *Under Assumption 5.3, it holds that*

$$(AT_k u, T_k u) \leq \omega (AT_k u, u), \quad \forall u \in \mathcal{H}.$$

Proof. Since T_k is A-self-adjoint, we know that $\rho(T_k) = \|T_k\|_A$, so that

$$\rho(T_k) = \max_{v \neq 0} \frac{(AT_k v, v)}{(Av, v)} \leq \omega < 2,$$

so that $(AT_k v, v) \leq \omega(Av, v)$, $\forall v \in \mathcal{H}$. But this gives $(AT_k u, T_k u) = (AT_k^{1/2} T_k u, T_k^{1/2} u) = (AT_k T_k^{1/2} u, T_k^{1/2} u)$ $= (AT_k v, v) \leq \omega(Av, v) = \omega(AT_k^{1/2} u, T_k^{1/2} u) = \omega(AT_k u, u)$, $\forall u \in \mathcal{H}$. □

The next lemma, also appearing first in [27], will be a key tool in the analysis of the product operator.

Lemma 5.10 *Under Assumption 5.3, it holds that*

$$(2-\omega)\sum_{k=1}^{J}(AT_k E_{k-1} v, E_{k-1} v) \leq \|v\|_A^2 - \|E_J v\|_A^2.$$

Proof. Employing the relationships in Lemma 5.8, we can rewrite the following difference as

$$\begin{aligned} \|E_{k-1}v\|_A^2 - \|E_k v\|_A^2 &= (AE_{k-1}v, E_{k-1}v) - (AE_k v, E_k v) \\ &= (AE_{k-1}v, E_{k-1}v) - (A[I-T_k]E_{k-1}v, [I-T_k]E_{k-1}v) \\ &= 2(AT_k E_{k-1}v, E_{k-1}v) - (AT_k E_{k-1}v, T_k E_{k-1}v) \end{aligned}$$

By Lemma 5.9 we have $(AT_k E_{k-1}v, T_k E_{k-1}v) \leq \omega(AT_k E_{k-1}v, E_{k-1}v)$, so that

$$\|E_{k-1}v\|_A^2 - \|E_k v\|_A^2 \geq (2-\omega)(AT_k E_{k-1}v, E_{k-1}v).$$

With $E_0 = I$, by summing from $k = 1$ to $k = J$ we have:

$$\|v\|_A^2 - \|E_J v\|_A^2 \geq (2-\omega)\sum_{k=1}^{J}(AT_k E_{k-1}v, E_{k-1}v).$$

□

We now state four simple assumptions which will, along with Assumption 5.3, allow us to give norm and condition number bounds by employing the previous lemmas. These four assumptions form the basis for the product and sum theory, and the remainder of our work will chiefly involve establishing conditions under which these assumptions are satisfied.

Assumption 5.4 *(Splitting assumption) There exists $C_0 > 0$ such that*

$$\|v\|_A^2 \leq C_0 \sum_{k=1}^{J}(AT_k v, v), \quad \forall v \in \mathcal{H}.$$

Assumption 5.5 *(Composite assumption) There exists $C_1 > 0$ such that*

$$\|v\|_A^2 \leq C_1 \sum_{k=1}^{J}(AT_k E_{k-1}v, E_{k-1}v), \quad \forall v \in \mathcal{H}.$$

Assumption 5.6 *(Product assumption) There exists $C_2 > 0$ such that*

$$\sum_{k=1}^{J}(AT_k v, v) \leq C_2 \sum_{k=1}^{J}(AT_k E_{k-1}v, E_{k-1}v), \quad \forall v \in \mathcal{H}.$$

Assumption 5.7 *(Sum assumption) There exists $C_3 > 0$ such that*

$$\sum_{k=1}^{J}(AT_k v, v) \leq C_3 \|v\|_A^2, \quad \forall v \in \mathcal{H}.$$

Lemma 5.11 *Under Assumptions 5.4 and 5.6, Assumption 5.5 holds with $C_1 = C_0C_2$.*

Proof. This is immediate, since

$$\|v\|_A^2 \le C_0 \sum_{k=1}^{J}(AT_kv, v) \le C_0C_2 \sum_{k=1}^{J}(AT_kE_{k-1}v, E_{k-1}v), \quad \forall v \in \mathcal{H}.$$

□

Remark 5.2. In what follows, it will be necessary to satisfy Assumption 5.5 for some constant C_1. Lemma 5.11 provides a technique for verifying Assumption 5.5 by verifying Assumptions 5.4 and 5.6 separately. In certain cases it will still be necessary to verify Assumption 5.5 directly.

The following theorems provide a fundamental framework for analyzing product and sum operators, employing only the five assumptions previously stated. A version of the product theorem similar to the one below first appeared in [27]. Theorems for sum operators were established early by Dryja and Widlund [58] and Björstad and Mandel [22].

Theorem 5.12 *Under Assumptions 5.3 and 5.5, the product operator (5.12) satisfies:*

$$\|E\|_A^2 \le 1 - \frac{2-\omega}{C_1}.$$

Proof. To prove the result, it suffices to show that

$$\|Ev\|_A^2 \le \left(1 - \frac{2-\omega}{C_1}\right)\|v\|_A^2, \quad \forall v \in \mathcal{H},$$

or that

$$\|v\|_A^2 \le \frac{C_1}{2-\omega}\left(\|v\|_A^2 - \|Ev\|_A^2\right), \quad \forall v \in \mathcal{H}.$$

By Lemma 5.10 (which required only Assumption 5.3), it is enough to show

$$\|v\|_A^2 \le C_1 \sum_{k=1}^{J}(AT_kE_{k-1}v, E_{k-1}v), \quad \forall v \in \mathcal{H}.$$

But, by Assumption 5.5 this result holds, and the theorem follows. □

Corollary 5.13 *Under Assumptions 5.3, 5.4, and 5.6, the product operator (5.12) satisfies:*

$$\|E\|_A^2 \le 1 - \frac{2-\omega}{C_0C_2}.$$

Proof. This follows from Theorem 5.12 and Lemma 5.11. □

Theorem 5.14 *Under Assumptions 5.3, 5.4, and 5.7, the sum operator (5.13) satisfies:*

$$\kappa_A(P) \le C_0C_3.$$

Proof. This result follows immediately from Assumptions 5.4 and 5.7, since $P = \sum_{k=1}^{J} T_k$ is A-self-adjoint by Assumption 5.3, and since

$$\frac{1}{C_0}(Av, v) \le \sum_{k=1}^{J}(AT_kv, v) = (APv, v) \le C_3(Av, v), \quad \forall v \in \mathcal{H}.$$

This implies that $C_0^{-1} \le \lambda_i(P) \le C_3$, and by Lemma 3.12 it holds that $\kappa_A(P) \le C_0C_3$. □

The constants C_0 and C_1 in Assumptions 5.4 and 5.5 will depend on the specific application; we will discuss estimates for C_0 and C_1 in the following sections. The constants C_2 and C_3 in Assumptions 5.6 and 5.7 will also depend on the specific application; however, we can derive bounds which grow with the number of operators J, which will always hold without additional assumptions. Both of these default or worst case results appear essentially in [27]. First, we recall the Cauchy-Schwarz inequality in $\mathbb{R}^n$, and state a useful corollary.

Lemma 5.15 *If $a_k, b_k \in \mathbb{R}$, $k = 1, \ldots, n$, then it holds that*

$$\left(\sum_{k=1}^n a_k b_k\right)^2 \le \left(\sum_{k=1}^n a_k^2\right)\left(\sum_{k=1}^n b_k^2\right).$$

Proof. See for example [125]. □

Corollary 5.16 *If $a_k \in \mathbb{R}$, $k = 1, \ldots, n$, then it holds that*

$$\left(\sum_{k=1}^n a_k\right)^2 \le n \sum_{k=1}^n a_k^2.$$

Proof. This follows easily from Lemma 5.15 by taking $b_k = 1$ for all k. □

Lemma 5.17 *Under only Assumption 5.3, we have that Assumption 5.6 holds, where:*

$$C_2 = 2 + \omega^2 J(J-1).$$

Proof. We must show that

$$\sum_{k=1}^J (AT_k v, v) \le [2 + \omega^2 J(J-1)] \sum_{k=1}^J (AT_k E_{k-1} v, E_{k-1} v), \quad \forall v \in \mathcal{H}.$$

By Lemma 5.8, we have that

$$(AT_k v, v) = (AT_k v, E_{k-1} v) + (AT_k v, [I - E_{k-1}] v) = (AT_k v, E_{k-1} v) + \sum_{i=1}^{k-1} (AT_k v, T_i E_{i-1} v)$$

$$\le (AT_k v, v)^{1/2} (AT_k E_{k-1} v, E_{k-1} v)^{1/2} + \sum_{i=1}^{k-1} (AT_k v, T_k v)^{1/2} (AT_i E_{i-1} v, T_i E_{i-1} v)^{1/2}.$$

By Lemma 5.9, we have

$$(AT_k v, v) \le (AT_k v, v)^{1/2} (AT_k E_{k-1} v, E_{k-1} v)^{1/2} + \omega (AT_k v, v)^{1/2} \sum_{i=1}^{k-1} (AT_i E_{i-1} v, E_{i-1} v)^{1/2},$$

or finally

$$(AT_k v, v) \le \left[(AT_k E_{k-1} v, E_{k-1} v)^{1/2} + \omega \sum_{i=1}^{k-1} (AT_i E_{i-1} v, E_{i-1} v)^{1/2}\right]^2. \tag{5.15}$$

Employing Corollary 5.16 for the two explicit terms in the inequality (5.15) yields:

$$(AT_k v, v) \le 2 \left[(AT_k E_{k-1} v, E_{k-1} v) + \omega^2 \left[\sum_{i=1}^{k-1} (AT_i E_{i-1} v, E_{i-1} v)^{1/2}\right]^2\right].$$

Employing Corollary 5.16 again for the $k-1$ terms in the sum yields

$$(AT_k v, v) \le 2 \left[(AT_k E_{k-1} v, E_{k-1} v) + \omega^2 (k-1) \sum_{i=1}^{k-1} (AT_i E_{i-1} v, E_{i-1} v)\right].$$

Summing the terms, and using the fact that the T_k are A-non-negative, we have

$$\sum_{k=1}^{J}(AT_kv,v) \leq 2\left[\sum_{k=1}^{J}\left\{(AT_kE_{k-1}v,E_{k-1}v)+\omega^2(k-1)\sum_{i=1}^{k-1}(AT_iE_{i-1}v,E_{i-1}v)\right\}\right]$$

$$\leq 2\left[1+\omega^2\sum_{i=1}^{J}(i-1)\right]\sum_{k=1}^{J}(AT_kE_{k-1}v,E_{k-1}v).$$

Since $\sum_{i=1}^{J} i=(J+1)J/2$, we have that the lemma follows. □

Lemma 5.18 *Under only Assumption 5.3, we have that Assumption 5.7 holds, where:*

$$C_3=\omega J.$$

Proof. By Assumption 5.3, we have

$$\sum_{k=1}^{J}(AT_kv,v) \leq \sum_{k=1}^{J}(AT_kv,T_kv)^{1/2}(Av,v)^{1/2} \leq \sum_{k=1}^{J}\omega(Av,v)=\omega J\|v\|_A^2,$$

so that $C_3=\omega J$. □

Remark 5.3. Note that since Lemmas 5.17 and 5.18 provide default (worst case) estimates for C_2 and C_3 in Assumptions 5.6 and 5.7, due to Lemma 5.11 it suffices to estimate only C_0 in Assumption 5.4 in order to employ the general product and sum operator theorems (namely Corollary 5.13 and Theorem 5.14).

5.2.2 The interaction hypothesis

We now consider an additional assumption, which will be natural in multigrid and domain decomposition applications, regarding the "interaction" of the operators T_k. This assumption brings together more closely the theory for the product and sum operators. The constants C_2 and C_3 in Assumptions 5.6 and 5.7 can both be estimated in terms of the constants C_4 and C_5 appearing below, which will be determined by the interaction properties of the operators T_k. We will further investigate the interaction properties more precisely in a moment. This approach to quantifying the interaction of the operators T_k is similar to that taken in [185].

Assumption 5.8 *(Interaction assumption - weak) There exists $C_4>0$ such that*

$$\sum_{k=1}^{J}\sum_{i=1}^{k-1}(AT_ku_k,T_iv_i) \leq C_4\left(\sum_{k=1}^{J}(AT_ku_k,u_k)\right)^{1/2}\left(\sum_{i=1}^{J}(AT_iv_i,v_i)\right)^{1/2}, \ \forall u_k,v_i\in\mathcal{H}.$$

Assumption 5.9 *(Interaction assumption - strong) There exists $C_5>0$ such that*

$$\sum_{k=1}^{J}\sum_{i=1}^{J}(AT_ku_k,T_iv_i) \leq C_5\left(\sum_{k=1}^{J}(AT_ku_k,u_k)\right)^{1/2}\left(\sum_{i=1}^{J}(AT_iv_i,v_i)\right)^{1/2}, \ \forall u_k,v_i\in\mathcal{H}.$$

Remark 5.4. We introduce the terminology "weak" and "strong" because in the *weak* interaction assumption above, the interaction constant C_4 is defined by considering the interaction of a particular operator T_k *only* with operators T_i with $i<k$; note that this implies an ordering of the operators T_k, and different orderings may produce different values for C_4. In the *strong* interaction assumption above, the interaction constant C_5 is defined by considering the interaction of a particular operator T_k with *all* operators T_i (the ordering of the operators T_k is now unimportant).

The interaction assumptions can be used to bound the constants C_2 and C_3 in Assumptions 5.6 and 5.7.

Lemma 5.19 *Under Assumptions 5.3 and 5.8, we have that Assumption 5.6 holds, where:*

$$C_2 = (1 + C_4)^2.$$

Proof. Consider

$$\sum_{k=1}^{J}(AT_k v, v) = \sum_{k=1}^{J}\{(AT_k v, E_{k-1}v) + (AT_k v, [I - E_{k-1}]v)\} \tag{5.16}$$

$$= \sum_{k=1}^{J}(AT_k v, E_{k-1}v) + \sum_{k=1}^{J}\sum_{i=1}^{k-1}(AT_k v, T_i E_{i-1}v).$$

For the first term, the Cauchy-Schwarz inequalities give

$$\sum_{k=1}^{J}(AT_k v, E_{k-1}v) \leq \sum_{k=1}^{J}(AT_k v, v)^{1/2}(AT_k E_{k-1}v, E_{k-1}v)^{1/2}$$

$$\leq \left(\sum_{k=1}^{J}(AT_k v, v)\right)^{1/2}\left(\sum_{k=1}^{J}(AT_k E_{k-1}v, E_{k-1}v)\right)^{1/2}.$$

For the second term, we have by Assumption 5.8 that

$$\sum_{k=1}^{J}\sum_{i=1}^{k-1}(AT_k v, T_i E_{i-1}v) \leq C_4\left(\sum_{k=1}^{J}(AT_k v, v)\right)^{1/2}\left(\sum_{k=1}^{J}(AT_k E_{k-1}v, E_{k-1}v)\right)^{1/2}.$$

Thus, together we have

$$\sum_{k=1}^{J}(AT_k v, v) \leq (1 + C_4)\left(\sum_{k=1}^{J}(AT_k v, v)\right)^{1/2}\left(\sum_{k=1}^{J}(AT_k E_{k-1}v, E_{k-1}v)\right)^{1/2},$$

which yields

$$\sum_{k=1}^{J}(AT_k v, v) \leq (1 + C_4)^2\sum_{k=1}^{J}(AT_k E_{k-1}v, E_{k-1}v).$$

□

Lemma 5.20 *Under Assumptions 5.3 and 5.9, we have that Assumption 5.7 holds, where:*

$$C_3 = C_5.$$

Proof. Consider first that $\forall v \in \mathcal{H}$, Assumption 5.9 implies

$$\|\sum_{k=1}^{J}T_k v\|_A^2 = \sum_{k=1}^{J}\sum_{i=1}^{J}(AT_k v, T_i v) \leq C_5\left(\sum_{k=1}^{J}(AT_k v, v)\right)^{1/2}\left(\sum_{i=1}^{J}(AT_i v, v)\right)^{1/2}$$

$$= C_5\sum_{k=1}^{J}(AT_k v, v).$$

If $P = \sum_{k=1}^{J} T_k$, then we have shown that $(APv, Pv) \leq C_5(APv, v)$, $\forall v \in \mathcal{H}$, so that

$$(APv, v) \leq (APv, Pv)^{1/2}(Av, v)^{1/2} \leq C_5^{1/2}(APv, v)^{1/2}(Av, v)^{1/2}, \forall v \in \mathcal{H}.$$

This implies that $(APv, v) \leq C_5\|v\|_A^2, \forall v \in \mathcal{H}$, which proves the lemma. □

The constants C_4 and C_5 can be further estimated, in terms of the following two *interaction matrices*. An early approach employing an interaction matrix appears in [27]; the form appearing below is most closely related to that used in [86] and [185]. The idea of employing a strictly upper-triangular interaction matrix to improve the bound for the weak interaction property is due to Hackbusch [86]. The default bound for the strictly upper-triangular matrix is also due to Hackbusch [86].

Definition 5.1 *Let Ξ be the strictly upper-triangular part of the interaction matrix $\Theta \in \mathbf{L}(\mathbb{R}^J, \mathbb{R}^J)$, which is defined to have as entries Θ_{ij} the smallest constants satisfying:*

$$|(AT_i u, T_j v)| \le \Theta_{ij}(AT_i u, T_i u)^{1/2}(AT_j v, T_j v)^{1/2}, \quad 1 \le i,j \le J, \qquad \forall u, v \in \mathcal{H}.$$

The matrix Θ is symmetric, and $0 \le \Theta_{ij} \le 1, \forall i,j$. Also, we have that $\Theta = I + \Xi + \Xi^T$.

Lemma 5.21 *It holds that $\|\Xi\|_2 \le \rho(\Theta)$. Also, $\|\Xi\|_2 \le \sqrt{J(J-1)/2}$ and $1 \le \rho(\Theta) \le J$.*

Proof. Since Θ is symmetric, we know that $\rho(\Theta) = \|\Theta\|_2 = \max_{\mathbf{x}\neq 0} \|\Theta\mathbf{x}\|_2/\|\mathbf{x}\|_2$. Now, given any $\mathbf{x} \in \mathbb{R}^J$, define $\bar{\mathbf{x}} \in \mathbb{R}^J$ such that $\bar{x}_i = |x_i|$. Note that $\|\mathbf{x}\|_2^2 = \sum_{i=1}^J |x_i|^2 = \|\bar{\mathbf{x}}\|_2^2$, and since $0 \le \Theta_{ij} \le 1$, we have that

$$\|\Theta\mathbf{x}\|_2^2 = \sum_{i=1}^J \left(\sum_{j=1}^J \Theta_{ij} x_j\right)^2 \le \sum_{i=1}^J \left(\sum_{j=1}^J \Theta_{ij}|x_j|\right)^2 = \|\Theta\bar{\mathbf{x}}\|_2^2.$$

Therefore, it suffices to consider only $\mathbf{x} \in \mathbb{R}^J$ with $x_i \ge 0$. For such an $\mathbf{x} \in \mathbb{R}^J$, it is clear that $\|\Xi\mathbf{x}\|_2 \le \|\Theta\mathbf{x}\|_2$, so we must have that

$$\|\Xi\|_2 = \max_{\mathbf{x}\neq 0} \frac{\|\Xi\mathbf{x}\|_2}{\|\mathbf{x}\|_2} \le \max_{\mathbf{x}\neq 0} \frac{\|\Theta\mathbf{x}\|_2}{\|\mathbf{x}\|_2} = \|\Theta\|_2 = \rho(\Theta).$$

The worst case estimate $\|\Xi\|_2 \le \sqrt{J(J-1)/2}$ follows easily, since $0 \le \Xi_{ij} \le 1$, and since:

$$[\Xi^T\Xi]_{ij} = \sum_{k=1}^J [\Xi^T]_{ik}\Xi_{kj} = \sum_{k=1}^J \Xi_{ki}\Xi_{kj} = \sum_{k=1}^{\min\{i-1,j-1\}} \Xi_{ki}\Xi_{kj} \le \min\{i-1, j-1\}.$$

Thus, we have that

$$\|\Xi\|_2^2 = \rho(\Xi^T\Xi) \le \|\Xi^T\Xi\|_1 = \max_j \left\{\sum_{i=1}^J |\ [\Xi^T\Xi]_{ij}\ |\right\} \le \sum_{i=1}^J (i-1) = \frac{J(J-1)}{2}.$$

It remains to show that $1 \le \rho(\Theta) \le J$. The upper bound follows easily since we know that $0 \le \Theta_{ij} \le 1$, and so that $\rho(\Theta) \le \|\Theta\|_1 = \max_j\{\sum_i |\Theta_{ij}|\} \le J$. Regarding the lower bound, recall that the trace of a matrix is equal to the sum of it's eigenvalues. Since all diagonal entries of Θ are unity, the trace is simply equal to J. If all the eigenvalues of Θ are unity, we are done. If we suppose there is at least one eigenvalue $\lambda_i < 1$ (possibly negative), then in order for the J eigenvalues of Θ to sum to J, there must be a corresponding eigenvalue $\lambda_j > 1$. Therefore, $\rho(\Theta) \ge 1$. □

We now have the following lemmas.

Lemma 5.22 *Under Assumption 5.3 we have that Assumption 5.8 holds, where:*

$$C_4 \le \omega\|\Xi\|_2.$$

Proof. Consider

$$\sum_{k=1}^J \sum_{i=1}^{k-1} (AT_k u_k, T_i v_i) \le \sum_{k=1}^J \sum_{i=1}^J \Xi_{ik}\|T_k u_k\|_A \|T_i v_i\|_A = (\Xi\mathbf{x}, \mathbf{y})_2,$$

where $\mathbf{x}, \mathbf{y} \in \mathbb{R}^J$, $x_k = \|T_k u_k\|_A$, $y_i = \|T_i v_i\|_A$, and $(\cdot,\cdot)_2$ is the usual Euclidean inner-product in $\mathbb{R}^J$. Now, we have that

$$(\Xi\mathbf{x}, \mathbf{y})_2 \le \|\Xi\|_2\|\mathbf{x}\|_2\|\mathbf{y}\|_2 = \|\Xi\|_2 \left(\sum_{k=1}^J (AT_k u_k, T_k u_k)\right)^{1/2} \left(\sum_{i=1}^J (AT_i v_i, T_i v_i)\right)^{1/2}$$

$$\leq \omega \|\Xi\|_2 \left(\sum_{k=1}^{J} (AT_k u_k, u_k) \right)^{1/2} \left(\sum_{i=1}^{J} (AT_i v_i, v_i) \right)^{1/2}.$$

Finally, this gives

$$\sum_{k=1}^{J} \sum_{i=1}^{k-1} (AT_k u_k, T_i v_i) \leq \omega \|\Xi\|_2 \left(\sum_{k=1}^{J} (AT_k u_k, u_k) \right)^{1/2} \left(\sum_{i=1}^{J} (AT_i v_i, v_i) \right)^{1/2}, \quad \forall u_k, v_i \in \mathcal{H}.$$

□

Lemma 5.23 *Under Assumption 5.3 we have that Assumption 5.9 holds, where:*

$$C_5 \leq \omega \rho(\Theta).$$

Proof. Consider

$$\sum_{k=1}^{J} \sum_{i=1}^{J} (AT_k u_k, T_i v_i) \leq \sum_{k=1}^{J} \sum_{i=1}^{J} \Theta_{ik} \|T_k u_k\|_A \|T_i v_i\|_A = (\Theta \mathbf{x}, \mathbf{y})_2,$$

where $\mathbf{x}, \mathbf{y} \in \mathbb{R}^J$, $x_k = \|T_k u_k\|_A$, $y_i = \|T_i v_i\|_A$, and $(\cdot, \cdot)_2$ is the usual Euclidean inner-product in $\mathbb{R}^J$. Now, since Θ is symmetric, we have that

$$(\Theta \mathbf{x}, \mathbf{y})_2 \leq \rho(\Theta) \|\mathbf{x}\|_2 \|\mathbf{y}\|_2 = \rho(\Theta) \left(\sum_{k=1}^{J} (AT_k u_k, T_k u_k) \right)^{1/2} \left(\sum_{i=1}^{J} (AT_i v_i, T_i v_i) \right)^{1/2}$$

$$\leq \omega \rho(\Theta) \left(\sum_{k=1}^{J} (AT_k u_k, u_k) \right)^{1/2} \left(\sum_{i=1}^{J} (AT_i v_i, v_i) \right)^{1/2}.$$

Finally, this gives

$$\sum_{k=1}^{J} \sum_{i=1}^{J} (AT_k u_k, T_i v_i) \leq \omega \rho(\Theta) \left(\sum_{k=1}^{J} (AT_k u_k, u_k) \right)^{1/2} \left(\sum_{i=1}^{J} (AT_i v_i, v_i) \right)^{1/2}, \quad \forall u_k, v_i \in \mathcal{H}.$$

□

This leads us finally to

Lemma 5.24 *Under Assumption 5.3 we have that Assumption 5.6 holds, where:*

$$C_2 = (1 + \omega \|\Xi\|_2)^2.$$

Proof. This follows from Lemmas 5.19 and 5.22. □

Lemma 5.25 *Under Assumption 5.3 we have that Assumption 5.7 holds, where:*

$$C_3 = \omega \rho(\Theta).$$

Proof. This follows from Lemmas 5.20 and 5.23. □

Remark 5.5. Note that Lemmas 5.24 and 5.21 reproduce the worst case estimate for C_2 given in Lemma 5.17, since:

$$C_2 = (1 + \omega \|\Xi\|_2)^2 \leq 2(1 + \omega^2 \|\Xi\|_2^2) \leq 2 + \omega^2 J(J-1).$$

In addition, Lemmas 5.25 and 5.21 reproduce the worst case estimate of $C_3 = \omega \rho(\Theta) \leq \omega J$ given in Lemma 5.18.

5.2.3 Allowing for a global operator

Consider the product and sum operators

$$E = (I - T_J)(I - T_{J-1}) \cdots (I - T_0), \tag{5.17}$$

$$P = T_0 + T_1 + \cdots + T_J, \tag{5.18}$$

where we now include a special operator T_0, which we assume may interact with *all* of the other operators. For example, T_0 might later represent some "global" coarse space operator in a domain decomposition method. Note that if such a global operator is included directly in the analysis of the previous section, then the bounds on $\|\Xi\|_2$ and $\rho(\Theta)$ necessarily depend on the number of operators; thus, to develop an optimal theory, we must exclude T_0 from the interaction hypothesis. This was recognized early in the domain decomposition community, and the modification of the theory in the previous sections to allow for such a global operator has been achieved mainly by Widlund and his co-workers. We will follow essentially their approach in this section.

In the following, we will use many of the results and assumptions from the previous section, where we now explicitly require that the $k = 0$ term *always* be included; the only exception to this will be the interaction assumption, which will still involve only the $k \neq 0$ terms. Regarding the minor changes to the results of the previous sections, note that we must now define $E_{-1} = I$, which modifies Lemma 5.8 in that

$$I - E_k = \sum_{i=0}^{k} T_i E_{i-1},$$

the sum beginning at $k = 0$. We make the usual Assumption 5.3 on the operators T_k (now including T_0 also), and we then have the results from Lemmas 5.9 and 5.10. The main assumptions for the theory are as in Assumptions 5.4, 5.6, and 5.7, with the additional term $k = 0$ included in each assumption. The two main results in Theorems 5.12 and 5.14 are unchanged. The default bounds for C_2 and C_3 given in Lemmas 5.17 and 5.18 now must take into account the additional operator T_0:

$$C_2 = 2 + \omega^2 J(J+1), \qquad C_3 = \omega(J+1).$$

The remaining analysis becomes now somewhat different from the case when T_0 is not present. First, we will quantify the interaction properties of the remaining operators T_k for $k \neq 0$ exactly as was done earlier, except that we must now employ the strong interaction assumption (Assumption 5.9) for both the product and sum theories. (In the previous section, we were able to use only the weak interaction assumption for the product operator.) This leads us to the following two lemmas.

Lemma 5.26 *Under Assumptions 5.3 (including T_0), 5.8 (excluding T_0), and 5.9 (excluding T_0), we have that Assumption 5.6 (including T_0) holds, where:*

$$C_2 = [1 + \omega^{1/2} C_5^{1/2} + C_4]^2.$$

Proof. Beginning with Lemma 5.8 we have that

$$\sum_{k=0}^{J} (AT_k v, v) = (AT_0 v, v) + \sum_{k=1}^{J} \{(AT_k v, E_{k-1} v) + (AT_k v, [I - E_{k-1}]v)\}$$

$$= \sum_{k=0}^{J} (AT_k v, E_{k-1} v) + \sum_{k=1}^{J} \sum_{i=0}^{k-1} (AT_k v, T_i E_{i-1} v)$$

$$= \sum_{k=0}^{J} (AT_k v, E_{k-1} v) + \sum_{k=1}^{J} (AT_k v, T_0 v) + \sum_{k=1}^{J} \sum_{i=1}^{k-1} (AT_k v, T_i E_{i-1} v) = \mathbf{S}_1 + \mathbf{S}_2 + \mathbf{S}_3. \tag{5.19}$$

We now estimate $\mathbf{S}_1$, $\mathbf{S}_2$, and $\mathbf{S}_3$ separately. For $\mathbf{S}_1$, we employ the Cauchy-Schwarz inequality to obtain

$$\mathbf{S}_1 = \sum_{k=0}^{J} (AT_k v, E_{k-1} v) \leq \sum_{k=0}^{J} (AT_k v, v)^{1/2} (AT_k E_{k-1} v, E_{k-1} v)^{1/2}$$

$$\leq \left(\sum_{k=0}^{J}(AT_k v, v)\right)^{1/2} \left(\sum_{k=0}^{J}(AT_k E_{k-1} v, E_{k-1} v)\right)^{1/2}.$$

To bound $\mathbf{S}_2$, we employ Assumption 5.9 as follows:

$$\mathbf{S}_2 = \sum_{k=1}^{J}(AT_k v, T_0 v) \leq \| \sum_{k=1}^{J} T_k v\|_A \|T_0 v\|_A = \left(\sum_{k=1}^{J}\sum_{i=1}^{J}(AT_k v, T_i v)\right)^{1/2} (AT_0 v, T_0 v)^{1/2}$$

$$\leq \omega^{1/2} C_5^{1/2} \left(\sum_{k=1}^{J}(AT_k v, v)\right)^{1/2} (AT_0 v, v)^{1/2}$$

$$\leq \omega^{1/2} C_5^{1/2} \left(\sum_{k=0}^{J}(AT_k v, v)\right)^{1/2} \left(\sum_{k=0}^{J}(AT_k E_{k-1} v, E_{k-1} v)\right)^{1/2}.$$

We now bound $\mathbf{S}_3$, employing Assumption 5.8 as

$$\mathbf{S}_3 = \sum_{k=1}^{J}\sum_{i=1}^{k-1}(AT_k v, T_i E_{i-1} v) \leq C_4 \left(\sum_{k=1}^{J}(AT_k v, v)\right)^{1/2} \left(\sum_{k=1}^{J}(AT_k E_{k-1} v, E_{k-1} v)\right)^{1/2}$$

$$\leq C_4 \left(\sum_{k=0}^{J}(AT_k v, v)\right)^{1/2} \left(\sum_{k=0}^{J}(AT_k E_{k-1} v, E_{k-1} v)\right)^{1/2}.$$

Putting the bounds for $\mathbf{S}_1$, $\mathbf{S}_2$, and $\mathbf{S}_3$ together, dividing (5.19) by $\sum_{k=1}^{J}(AT_k v, v)$ and squaring, yields

$$\sum_{k=0}^{J}(AT_k v, v) \leq [1 + \omega^{1/2} C_5^{1/2} + C_4]^2 \sum_{k=0}^{J}(AT_k E_{k-1} v, E_{k-1} v).$$

Therefore, Assumption 5.6 holds, where:

$$C_2 = [1 + \omega^{1/2} C_5^{1/2} + C_4]^2.$$

□

Results similar to the next lemma are used in several recent papers on domain decomposition [59]; the proof is quite simple once the proof of Lemma 5.20 is available.

Lemma 5.27 *Under Assumptions 5.3 (including T_0) and 5.9 (excluding T_0), we have that Assumption 5.7 (including T_0) holds, where:*

$$C_3 = \omega + C_5.$$

Proof. The proof of Lemma 5.20 gives immediately $\sum_{k=1}^{J}(AT_k v, v) \leq C_5 \|v\|_A^2$. Now, since $(AT_0 v, v) \leq \omega \|v\|_A^2$, we simply add in the $k = 0$ term, yielding

$$\sum_{k=0}^{J}(AT_k v, v) \leq (\omega + C_5)\|v\|_A^2.$$

□

We finish the section by relating the constants C_2 and C_3 (required for Corollary 5.13 and Theorem 5.14) to the interaction matrices. The constants C_4 and C_5 are estimated by using the interaction matrices exactly as before, since the interaction conditions still involve only the operators T_k for $k \neq 0$.

Lemma 5.28 *Under Assumption 5.3 we have that Assumption 5.6 holds, where:*

$$C_2 \leq 6[1+\omega^2\rho(\Theta)^2].$$

Proof. From Lemma 5.26 we have that

$$C_2 = [1+\omega^{1/2}C_5^{1/2}+C_4]^2.$$

Now, from Lemmas 5.22 and 5.23, and since $\omega < 2$, it follows that

$$C_2 = [1+\omega^{1/2}C_5^{1/2}+C_4]^2 \leq [1+\sqrt{2}(\omega\rho(\Theta))^{1/2}+\omega\|\Xi\|_2]^2.$$

Employing first Lemma 5.21 and then Corollary 5.16 twice, we have

$$C_2 \leq [1+\sqrt{2}(\omega\rho(\Theta))^{1/2}+\omega\rho(\Theta)]^2 \leq 3[1+2\omega\rho(\Theta)+\omega^2\rho(\Theta)^2]$$
$$= 3[1+\omega\rho(\Theta)]^2 \leq 6[1+\omega^2\rho(\Theta)^2].$$

□

Lemma 5.29 *Under Assumption 5.3 we have that Assumption 5.7 holds, where:*

$$C_3 \leq \omega(\rho(\Theta)+1).$$

Proof. From Lemmas 5.27 and 5.23 it follows that

$$C_3 = \omega + C_5 \leq \omega + \omega\rho(\Theta) = \omega(\rho(\Theta)+1).$$

□

Remark 5.6. It is apparently possible to establish a sharper bound [34, 59] than the one given above in Lemma 5.28, the improved bound having the form

$$C_2 = 1+2\omega^2\rho(\Theta)^2.$$

This result is stated and used in several recent papers on domain decomposition, e.g., in [59], but the proof of the result has apparently not been published. A proof of a similar result is established for some related nonsymmetric problems in [34].

5.2.4 Main results of the theory

The main theory may be summarized in the following way. We are interested in norm and condition number bounds of the product and sum operators:

$$E = (I-T_J)(I-T_{J-1})\cdots(I-T_0), \tag{5.20}$$

$$P = T_0+T_1+\cdots+T_J. \tag{5.21}$$

The necessary assumptions for the theory are as follows.

Assumption 5.10 *(Operator norms) The operators $T_k \in \mathbf{L}(\mathcal{H},\mathcal{H})$ are A-self-adjoint, A-non-negative, and*

$$\rho(T_k) = \|T_k\|_A \leq \omega < 2, \quad k=0,\ldots,J.$$

Assumption 5.11 *(Splitting constant) There exists $C_0 > 0$ such that*

$$\|v\|_A^2 \leq C_0\sum_{k=0}^{J}(AT_kv,v), \quad \forall v \in \mathcal{H}.$$

Definition 5.2 *(Interaction matrices) Let Ξ be the strictly upper-triangular part of the interaction matrix $\Theta \in \mathbf{L}(\mathbb{R}^J, \mathbb{R}^J)$, which is defined to have as entries Θ_{ij} the smallest constants satisfying:*

$$|(AT_i u, T_j v)| \leq \Theta_{ij}(AT_i u, T_i u)^{1/2}(AT_j v, T_j v)^{1/2}, \quad 1 \leq i, j \leq J.$$

The main theorems are as follows.

Theorem 5.30 *(Product operator) Under Assumptions 5.10 and 5.11, the product operator (5.20) satisfies:*

$$\|E\|_A^2 \leq 1 - \frac{2-\omega}{C_0(6 + 6\omega^2 \rho(\Theta)^2)}.$$

Proof. Assumptions 5.10 and 5.11 are clearly equivalent to Assumptions 5.3 and 5.4, and by Lemma 5.28 we know that Assumption 5.6 must hold with $C_2 = [6 + 6\omega^2\rho(\Theta)^2]$. The theorem then follows by application of Corollary 5.13. □

Theorem 5.31 *(Sum operator) Under Assumptions 5.10 and 5.11, the sum operator (5.21) satisfies:*

$$\kappa_A(P) \leq C_0\omega(\rho(\Theta) + 1).$$

Proof. Assumptions 5.10 and 5.11 are clearly equivalent to Assumptions 5.3 and 5.4, and by Lemma 5.29 we know that Assumption 5.7 must hold with $C_3 = \omega(1 + \rho(\Theta))$. The theorem then follows by application of Theorem 5.14. □

For the case when there is *not* a global operator T_0 present, set $T_0 \equiv 0$ in the above definitions and assumptions. Note that this implies that all $k = 0$ terms in the assumptions and definitions are ignored. The main theorems are now modified as follows.

Theorem 5.32 *(Product operator) If $T_0 \equiv 0$, then under Assumptions 5.10 and 5.11, the product operator (5.20) satisfies:*

$$\|E\|_A^2 \leq 1 - \frac{2-\omega}{C_0(1 + \omega\|\Xi\|_2)^2}.$$

Proof. Assumptions 5.10 and 5.11 are clearly equivalent to Assumptions 5.3 and 5.4, and by Lemma 5.24 we know that Assumption 5.6 must hold with $C_2 = (1 + \omega\|\Xi\|)^2$. The theorem then follows by application of Corollary 5.13. □

Theorem 5.33 *(Sum operator) If $T_0 \equiv 0$, then under Assumptions 5.10 and 5.11, the sum operator (5.21) satisfies:*

$$\kappa_A(P) \leq C_0\omega\rho(\Theta).$$

Proof. Assumptions 5.10 and 5.11 are clearly equivalent to Assumptions 5.3 and 5.4, and by Lemma 5.25 we know that Assumption 5.7 must hold with $C_3 = \omega\rho(\Theta)$. The theorem then follows by application of Theorem 5.14. □

Remark 5.7. We see that the product and sum operator theory now rests completely on the estimation of the constant C_0 in Assumption 5.11 and the bounds on the interaction matrices. (The bound involving ω in Assumption 5.10 always holds for any reasonable method based on product and sum operators.) We will further reduce the estimate of C_0 to simply the estimate of a "splitting" constant, depending on the particular splitting of the main space $\mathcal{H}$ into subspaces $\mathcal{H}_k$, and to an estimate of the effectiveness of the approximate solver in the subspaces.

Remark 5.8. Note that if we cannot estimate $\|\Xi\|_2$ or $\rho(\Theta)$, then we can still use the above theory since we have worst case estimates from Lemmas 5.22 and 5.23, namely:

$$\|\Xi\|_2 \leq \sqrt{J(J-1)/2} < J, \qquad \rho(\Theta) \leq J.$$

In the case of the nested spaces in multigrid methods, it may be possible to analyze $\|\Xi\|_2$ through the use of *strengthened Cauchy-Schwarz inequalities*, showing in fact that $\|\Xi\|_2 = O(1)$. In the case of domain decomposition methods, it will *always* be possible to show that $\|\Xi\|_2 = O(1)$ and $\rho(\Theta) = O(1)$, due to the local nature of the domain decomposition projection operators.

5.3 Abstract Schwarz theory

In this section, we consider abstract Schwarz methods based on subspaces, and apply the general product and sum operator theory to these methods. The resulting theory, which is a variation of that presented in [185] and [59], rests on the notion of a stable subspace splitting of the original Hilbert space (cf. [159, 165]). Although the derivation here is presented in a somewhat different, algebraic language, many of the intermediate results we use have appeared previously in the literature in other forms (we provide references at the appropriate points). In contrast to earlier approaches, we develop the entire theory employing general prolongation and restriction operators; the use of inclusion and projection as prolongation and restriction are represented in this approach as a special case.

5.3.1 The Schwarz methods

Consider now a Hilbert space $\mathcal{H}$, equipped with an inner-product $(\cdot,\cdot)$ inducing a norm $\|\cdot\| = (\cdot,\cdot)^{1/2}$. Let there be given an SPD operator $A \in \mathbf{L}(\mathcal{H},\mathcal{H})$ defining another inner-product on $\mathcal{H}$, which we denote as $(\cdot,\cdot)_A = (A\cdot,\cdot)$. This second inner-product also induces a norm $\|\cdot\|_A = (\cdot,\cdot)_A^{1/2}$. We are also given an associated set of spaces

$$\mathcal{H}_1, \mathcal{H}_2, \ldots, \mathcal{H}_J, \qquad \dim(\mathcal{H}_k) \leq \dim(\mathcal{H}), \qquad I_k\mathcal{H}_k \subseteq \mathcal{H}, \qquad \mathcal{H} = \sum_{k=1}^{J} I_k\mathcal{H}_k,$$

for some operators $I_k : \mathcal{H}_k \mapsto \mathcal{H}$, where we assume that $\text{null}(I_k) = \{0\}$. This defines a splitting of $\mathcal{H}$ into the subspaces $I_k\mathcal{H}_k$, although the spaces $\mathcal{H}_k$ alone may not relate to the largest space $\mathcal{H}$ in any natural way without the operator I_k. No requirements are made on the associated spaces $\mathcal{H}_k$ beyond the above, so that they are not necessarily nested, disjoint, or overlapping.

Associated with each space $\mathcal{H}_k$ is an inner-product $(\cdot,\cdot)_k$ inducing a norm $\|\cdot\|_k = (\cdot,\cdot)_k^{1/2}$, and an SPD operator $A_k \in \mathbf{L}(\mathcal{H}_k,\mathcal{H}_k)$, defining a second inner-product $(\cdot,\cdot)_{A_k} = (A_k\cdot,\cdot)_k$ and norm $\|\cdot\|_{A_k} = (\cdot,\cdot)_{A_k}^{1/2}$. The spaces $\mathcal{H}_k$ are related to the finest space $\mathcal{H}$ through the *prolongation* I_k defined above, and also through the *restriction* operator, defined as the adjoint of I_k relating the inner-products in $\mathcal{H}$ and $\mathcal{H}_k$:

$$(I_k v_k, v) = (v_k, I_k^T v)_k, \qquad I_k^T : \mathcal{H} \mapsto \mathcal{H}_k.$$

It will always be completely clear from the arguments of the inner-product (or norm) which particular inner-product (or norm) is implied; i.e., if the arguments lie in $\mathcal{H}$ then either $(\cdot,\cdot)$ or $(A\cdot,\cdot)$ is to be used, whereas if the arguments lie in $\mathcal{H}_k$, then either $(\cdot,\cdot)_k$ or $(A_k\cdot,\cdot)_k$ is to be used. Therefore, we will leave off the implied subscript k from the inner-products and norms in all of the following discussions, without danger of confusion. Finally, we assume the existence of SPD linear operators $R_k \in \mathbf{L}(\mathcal{H}_k,\mathcal{H}_k)$, such that $R_k \approx A_k^{-1}$.

Definition 5.3 *The operator $A_k \in \mathbf{L}(\mathcal{H}_k,\mathcal{H}_k)$ is called variational with respect to $A \in \mathbf{L}(\mathcal{H},\mathcal{H})$ if, for a fixed operator $I_k \in \mathbf{L}(\mathcal{H}_k,\mathcal{H})$, it holds that:*

$$A_k = I_k^T A I_k.$$

If the operators A_k are each variational with A, then the operator A_k in space $\mathcal{H}_k$ is in some sense a representation of the operator A in the space $\mathcal{H}_k$. For example, in a multigrid or domain decomposition algorithm, the operator I_k^T may correspond to an orthogonal projector, and I_k to the natural inclusion of a subspace into the whole space.

Regarding the operators R_k, a natural condition to impose is that they correspond to some convergent linear methods in the associated spaces, the necessary and sufficient condition for which would be (by Theorem 3.7):

$$\rho(I - R_kA_k) = \|I - R_kA_k\|_A < 1, \qquad k = 1, \cdots, J.$$

Note that if $R_k = A_k^{-1}$, this is trivially satisfied. More generally, $R_k \approx A_k^{-1}$, corresponding to some classical linear smoothing method (in the case of multigrid), or some other linear solver.

An abstract multiplicative Schwarz method, employing associated space corrections in the spaces $\mathcal{H}_k$, has the form:

Algorithm 5.2 *(Abstract Multiplicative Schwarz Method – Implementation Form)*

$$u^{n+1} = MS(u^n, f)$$

where the operation $u^{\mathrm{NEW}} = MS(u^{\mathrm{OLD}}, f)$ *is defined as:*

```
Do $k = 1, \ldots, J$
    $r_k = I_k^T(f - Au^{\mathrm{OLD}})$
    $e_k = R_k r_k$
    $u^{\mathrm{NEW}} = u^{\mathrm{OLD}} + I_k e_k$
    $u^{\mathrm{OLD}} = u^{\mathrm{NEW}}$
End do.
```

Note that the first step through the loop in $MS(\cdot,\cdot)$ gives:

$$u^{\mathrm{NEW}} = u^{\mathrm{OLD}} + I_1 e_1 = u^{\mathrm{OLD}} + I_1 R_1 I_1^T(f - Au^{\mathrm{OLD}}) = (I - I_1 R_1 I_1^T A)u^{\mathrm{OLD}} + I_1 R_1 I_1^T f.$$

Continuing in this fashion, and by defining $T_k = I_k R_k I_k^T A$, we see that after the full loop in $MS(\cdot,\cdot)$ the solution transforms according to:

$$u^{n+1} = (I - T_J)(I - T_{J-1}) \cdots (I - T_1)u^n + Bf,$$

where B is a quite complicated combination of the operators R_k, I_k, I_k^T, and A. By defining $E_k = (I - T_k)(I - T_{k-1}) \cdots (I - T_1)$, we see that $E_k = (I - T_k)E_{k-1}$. Therefore, since $E_{k-1} = I - B_{k-1}A$ for some (implicitly defined) B_{k-1}, we can identify the operators B_k through the recursion $E_k = I - B_k A = (I - T_k)E_{k-1}$, giving

$$B_k A = I - (I - T_k)E_{k-1} = I - (I - B_{k-1}A) + T_k(I - B_{k-1}A) = B_{k-1}A + T_k - T_k B_{k-1}A$$

$$= B_{k-1}A + I_k R_k I_k^T A - I_k R_k I_k^T A B_{k-1} A = \left[B_{k-1} + I_k R_k I_k^T - I_k R_k I_k^T A B_{k-1}\right] A,$$

so that $B_k = B_{k-1} + I_k R_k I_k^T - I_k R_k I_k^T A B_{k-1}$. But this means the above algorithm is equivalent to:

Algorithm 5.3 *(Abstract Multiplicative Schwarz Method – Operator Form)*

$$u^{n+1} = u^n + B(f - Au^n) = (I - BA)u^n + Bf$$

where the multiplicative Schwarz error propagator E is defined by:

$$E = I - BA = (I - T_J)(I - T_{J-1}) \cdots (I - T_1), \qquad T_k = I_k R_k I_k^T A, \quad k = 1, \ldots, J.$$

The operator $B \equiv B_J$ is defined implicitly, and obeys the recursion:

$$B_1 = I_1 R_1 I_1^T, \quad B_k = B_{k-1} + I_k R_k I_k^T - I_k R_k I_k^T A B_{k-1}, \quad k = 2, \ldots, J.$$

An abstract additive Schwarz method, employing corrections in the spaces $\mathcal{H}_k$, has the form:

Algorithm 5.4 *(Abstract Additive Schwarz Method – Implementation Form)*

$$u^{n+1} = MS(u^n, f)$$

where the operation $u^{\mathrm{NEW}} = MS(u^{\mathrm{OLD}}, f)$ *is defined as:*

```
$r = f - Au^{\mathrm{OLD}}$
Do $k = 1, \ldots, J$
    $r_k = I_k^T r$
    $e_k = R_k r_k$
    $u^{\mathrm{NEW}} = u^{\mathrm{OLD}} + I_k e_k$
End do.
```

Since each loop iteration depends only on the original approximation u^{OLD}, we see that the full correction to the solution can be written as the sum:

$$u^{n+1} = u^n + B(f - Au^n) = u^n + \sum_{k=1}^{J} I_k R_k I_k^T (f - Au^n),$$

where the preconditioner B has the form $B = \sum_{k=1}^{J} I_k R_k I_k^T$, and the error propagator is $E = I - BA$. Therefore, the above algorithm is equivalent to:

Algorithm 5.5 *(Abstract Additive Schwarz Method – Operator Form)*

$$u^{n+1} = u^n + B(f - Au^n) = (I - BA)u^n + Bf$$

where the additive Schwarz error propagator E is defined by:

$$E = I - BA = I - \sum_{k=1}^{J} T_k, \qquad T_k = I_k R_k I_k^T A, \quad k = 1, \ldots, J.$$

The operator B is defined explicitly as $B = \sum_{k=1}^{J} I_k R_k I_k^T$.

5.3.2 Subspace splitting theory

We now consider the framework of §5.3.1, employing the abstract results of §5.2.4. First, we prove some simple results about projectors, and the relationships between the operators R_k on the spaces $\mathcal{H}_k$ and the resulting operators $T_k = I_k R_k I_k^T A$ on the space $\mathcal{H}$. We then consider the "splitting" of the space $\mathcal{H}$ into subspaces $I_k \mathcal{H}_k$, and the verification of the assumptions required to apply the abstract theory of §5.2.4 is reduced to deriving an estimate of the "splitting constant".

Recall that an orthogonal projector is an operator $P \in \mathbf{L}(\mathcal{H}, \mathcal{H})$ having a closed subspace $\mathcal{V} \subseteq \mathcal{H}$ as its range (on which P acts as the identity), and having the orthogonal complement of $\mathcal{V}$, denoted as $\mathcal{V}^\perp \subseteq \mathcal{H}$, as its null space. By this definition, the operator $I - P$ is also clearly a projector, but having the subspace $\mathcal{V}^\perp$ as range and $\mathcal{V}$ as null space. In other words, a projector P splits a Hilbert space $\mathcal{H}$ into a direct sum of a closed subspace and its orthogonal complement as follows:

$$\mathcal{H} = \mathcal{V} \oplus \mathcal{V}^\perp = P\mathcal{H} \oplus (I - P)\mathcal{H}.$$

The following lemma gives a useful characterization of a projection operator; note that this characterization is often used as an equivalent alternative definition of a projection operator.

Lemma 5.34 *Let $A \in \mathbf{L}(\mathcal{H}, \mathcal{H})$ be SPD. Then the operator $P \in \mathbf{L}(\mathcal{H}, \mathcal{H})$ is an A-orthogonal projector if and only if P is A-self-adjoint and idempotent ($P^2 = P$).*

Proof. See [129], Theorem 9.5-1, page 481. □

Lemma 5.35 *Assume* $\dim(\mathcal{H}_k) \le \dim(\mathcal{H})$*,* $I_k : \mathcal{H}_k \mapsto \mathcal{H}$*,* $\mathrm{null}(I_k) = \{0\}$*, and that A is SPD. Then*

$$Q_k = I_k (I_k^T I_k)^{-1} I_k^T, \qquad P_k = I_k (I_k^T A I_k)^{-1} I_k^T A,$$

are the unique orthogonal and A-orthogonal projectors onto $I_k \mathcal{H}_k$.

Proof. By assuming that $\mathrm{null}(I_k) = \{0\}$, we guarantee that both $\mathrm{null}(I_k^T I_k) = \{0\}$ and $\mathrm{null}(I_k^T A I_k) = \{0\}$, so that both Q_k and P_k are well-defined. It is easily verified that Q_k is self-adjoint and P_k is A-self-adjoint, and it is immediate that $Q_k^2 = Q_k$ and that $P_k^2 = P_k$. Clearly, $Q_k : \mathcal{H} \mapsto I_k \mathcal{H}_k$, and $P_k : \mathcal{H} \mapsto I_k \mathcal{H}_k$, so that by Lemma 5.34 these operators are orthogonal and A-orthogonal projectors onto $I_k \mathcal{H}_k$. All that remains is to show that these operators are unique. By definition, a projector onto a subspace $I_k \mathcal{H}_k$ acts as the identity on $I_k \mathcal{H}_k$, and as the zero operator on $(I_k \mathcal{H}_k)^\perp$. Therefore, any two projectors P_k and $\tilde{P}_k$ onto $I_k \mathcal{H}_k$ must act identically on the entire space $\mathcal{H} = I_k \mathcal{H}_k \oplus (I_k \mathcal{H}_k)^\perp$, and therefore $P_k = \tilde{P}_k$. Similarly, Q_k is unique. □

We now make the following natural assumption regarding the operators $R_k \approx A_k^{-1}$.

Assumption 5.12 *The operators $R_k \in \mathbf{L}(\mathcal{H}_k, \mathcal{H}_k)$ are SPD. Further, there exists a subspace $\mathcal{V}_k \subseteq \mathcal{H}_k$, and parameters $0 < \omega_0 \leq \omega_1 < 2$, such that*

$$\begin{array}{llcll} (a) & \omega_0(A_k v_k, v_k) & \leq & (A_k R_k A_k v_k, v_k), \quad \forall v_k \in \mathcal{V}_k \subseteq \mathcal{H}_k, & k = 1, \ldots, J, \\ (b) & (A_k R_k A_k v_k, v_k) & \leq & \omega_1(A_k v_k, v_k), \quad \forall v_k \in \mathcal{H}_k, & k = 1, \ldots, J. \end{array}$$

This implies that on the subspace $\mathcal{V}_k \subseteq \mathcal{H}_k$, it holds that $0 < \omega_0 \leq \lambda_i(R_k A_k)$, $k = 1, \ldots, J$, whereas on the entire space $\mathcal{H}_k$, it holds that $\lambda_i(R_k A_k) \leq \omega_1 < 2$, $k = 1, \ldots, J$.

There are several consequences of the above assumption which will be useful later.

Lemma 5.36 *Assumption 5.12(b) implies that $0 < \lambda_i(R_k A_k) \leq \omega_1$, and $\rho(I - R_k A_k) = \|I - R_k A_k\|_{A_k} < 1$.*

Proof. Since R and A are SPD by assumption, we have by Lemma 3.6 that RA is A-SPD. By Assumption 5.12(b), the Rayleigh quotients are bounded above by ω_1, so that

$$0 < \lambda_i(RA) \leq \omega_1.$$

Thus,

$$\rho(I - RA) = \max_i |\lambda_i(I - RA)| = \max_i |1 - \lambda_i(RA)|.$$

Clearly then $\rho(I - RA) < 1$ since $0 < \omega_1 < 2$. □

Lemma 5.37 *Assumption 5.12(b) implies that $(A_k v_k, v_k) \leq \omega_1(R_k^{-1} v_k, v_k), \forall v_k \in \mathcal{H}_k$.*

Proof. We drop the subscripts for ease of exposition. By Assumption 5.12(b), $(ARAv, v) \leq \omega_1(Av, v)$, so that ω_1 bounds the Raleigh quotients generated by RA. Since RA is similar to $R^{1/2} A R^{1/2}$, we must also have that

$$(R^{1/2} A R^{1/2} v, v) \leq \omega_1(v, v).$$

But this implies

$$(AR^{1/2} v, R^{1/2} v) \leq \omega_1(R^{-1} R^{1/2} v, R^{1/2} v),$$

or $(Aw, w) \leq \omega_1(R^{-1} w, w), \forall w \in \mathcal{H}$. □

Lemma 5.38 *Assumption 5.12(b) implies that $T_k = I_k R_k I_k^T A$ is A-self-adjoint and A-non-negative, and*

$$\rho(T_k) = \|T_k\|_A \leq \omega_1 < 2.$$

Proof. That $T_k = I_k R_k I_k^T A$ is A-self-adjoint and A-non-negative follows immediately from the symmetry of R_k and A_k. To show the last result, we employ Lemma 5.37 to obtain

$$\begin{aligned} (AT_k v, T_k v) = (AI_k R_k I_k^T Av, I_k R_k I_k^T Av) &= (I_k^T A I_k R_k I_k^T Av, R_k I_k^T Av) \\ = (A_k R_k I_k^T Av, R_k I_k^T Av) \leq \omega_1(R_k^{-1} R_k I_k^T Av, R_k I_k^T Av) &= \omega_1(I_k^T Av, R_k I_k^T Av) \\ = \omega_1(AI_k R_k I_k^T Av, v) &= \omega_1(AT_k v, v). \end{aligned}$$

Now, from the Schwarz inequality, we have

$$(AT_k v, T_k v) \leq \omega_1(AT_k v, v) \leq \omega_1(AT_k v, T_k v)^{1/2}(Av, v)^{1/2},$$

or that

$$(AT_k v, T_k v)^{1/2} \leq \omega_1(Av, v)^{1/2},$$

which implies that $\|T_k\|_A \leq \omega_1 < 2$. □

The key idea in all of the following theory involves the splitting of the original Hilbert space $\mathcal{H}$ into a collection of subspaces $I_k \mathcal{V}_k \subseteq I_k \mathcal{H}_k \subseteq \mathcal{H}$. It will be important for the splitting to be *stable* in a certain sense, which we state as the following assumption.

Assumption 5.13 *Given any $v \in \mathcal{H} = \sum_{k=1}^J I_k \mathcal{H}_k$, $I_k \mathcal{H}_k \subseteq \mathcal{H}$, there exists subspaces $I_k \mathcal{V}_k \subseteq I_k \mathcal{H}_k \subseteq \mathcal{H} = \sum_{k=1}^J I_k \mathcal{V}_k$, and a particular splitting $v = \sum_{k=1}^J I_k v_k$, $v_k \in \mathcal{V}_k$, such that*

$$\sum_{k=1}^J \|I_k v_k\|_A^2 \leq S_0 \|v\|_A^2,$$

for some splitting constant $S_0 > 0$.

The following key lemma (in the case of inclusion and projection as prolongation and restriction) is sometimes referred to as *Lions' Lemma* [133], although the multiple-subspace case is essentially due to Widlund [183].

Lemma 5.39 *Under Assumption 5.13 it holds that*

$$\left(\frac{1}{S_0}\right) \|v\|_A^2 \leq \sum_{k=1}^J (AP_k v, v), \quad \forall v \in \mathcal{H}.$$

Proof. Given any $v \in \mathcal{H}$, we employ the splitting of Assumption 5.13 to obtain

$$\|v\|_A^2 = \sum_{k=1}^J (Av, I_k v_k) = \sum_{k=1}^J (I_k^T Av, v_k) = \sum_{k=1}^J (I_k^T A(I_k(I_k^T A I_k)^{-1} I_k^T A)v, v_k) = \sum_{k=1}^J (AP_k v, I_k v_k).$$

Now, let $\tilde{P}_k = (I_k^T A I_k)^{-1} I_k^T A$, so that $P_k = I_k \tilde{P}_k$. Then

$$\|v\|_A^2 = \sum_{k=1}^J (I_k^T A I_k \tilde{P}_k v, v_k) = \sum_{k=1}^J (A_k \tilde{P}_k v, v_k) \leq \sum_{k=1}^J (A_k v_k, v_k)^{1/2} (A_k \tilde{P}_k v, \tilde{P}_k v)^{1/2}$$

$$\leq \left(\sum_{k=1}^J (A_k v_k, v_k)\right)^{1/2} \left(\sum_{k=1}^J (A_k \tilde{P}_k v, \tilde{P}_k v)\right)^{1/2} = \left(\sum_{k=1}^J (A I_k v_k, I_k v_k)\right)^{1/2} \left(\sum_{k=1}^J (A_k \tilde{P}_k v, \tilde{P}_k v)\right)^{1/2}$$

$$= \left(\sum_{k=1}^J \|I_k v_k\|_A^2\right)^{1/2} \left(\sum_{k=1}^J (A_k \tilde{P}_k v, \tilde{P}_k v)\right)^{1/2} \leq S_0^{1/2} \|v\|_A \left(\sum_{k=1}^J (A I_k \tilde{P}_k v, I_k \tilde{P}_k v)\right)^{1/2}$$

$$= S_0^{1/2} \|v\|_A \left(\sum_{k=1}^J (AP_k v, P_k v)\right)^{1/2}, \quad \forall v \in \mathcal{H}.$$

Since $(AP_k v, P_k v) = (AP_k v, v)$, dividing the above by $\|v\|_A$ and squaring yields the result. □

The next intermediate result will be useful in the case that the subspace solver R_k is effective on only the part of the subspace $\mathcal{H}_k$, namely $\mathcal{V}_k \subseteq \mathcal{H}_k$.

Lemma 5.40 *Under Assumptions 5.12(a) and 5.13 (for the same subspaces $I_k \mathcal{V}_k \subseteq I_k \mathcal{H}_k$) it holds that*

$$\sum_{k=1}^J (R_k^{-1} v_k, v_k) \leq \left(\frac{S_0}{\omega_0}\right) \|v\|_A^2, \quad \forall v = \sum_{k=1}^J I_k v_k \in \mathcal{H}, \ v_k \in \mathcal{V}_k \subseteq \mathcal{H}_k.$$

Proof. With $v = \sum_{k=1}^J I_k v_k$, where we employ the splitting in Assumption 5.13, we have

$$\sum_{k=1}^J (R_k^{-1} v_k, v_k) = \sum_{k=1}^J (A_k A_k^{-1} R_k^{-1} v_k, v_k) = \sum_{k=1}^J (A_k v_k, v_k) \frac{(A_k A_k^{-1} R_k^{-1} v_k, v_k)}{(A_k v_k, v_k)}$$

$$\leq \sum_{k=1}^J (A_k v_k, v_k) \max_{v_k \neq 0} \frac{(A_k A_k^{-1} R_k^{-1} v_k, v_k)}{(A_k v_k, v_k)} \leq \sum_{k=1}^J \omega_0^{-1} (A_k v_k, v_k)$$

$$= \sum_{k=1}^J \omega_0^{-1} (A I_k v_k, I_k v_k) = \sum_{k=1}^J \omega_0^{-1} \|I_k v_k\|_A^2 \leq \left(\frac{S_0}{\omega_0}\right) \|v\|_A^2,$$

which proves the lemma. □

The following lemma relates the constant appearing in the "splitting" Assumption 5.11 of the product and sum operator theory to the subspace splitting constant appearing in Assumption 5.13 above.

Lemma 5.41 *Under Assumptions 5.12(a) and 5.13 (for the same subspaces $I_k\mathcal{V}_k \subseteq I_k\mathcal{H}_k$) it holds that*

$$\|v\|_A^2 \leq \left(\frac{S_0}{\omega_0}\right)\sum_{k=1}^J (AT_kv, v), \quad \forall v \in \mathcal{H}.$$

Proof. Given any $v \in \mathcal{H}$, we begin with the splitting in Assumption 5.13 as follows

$$\|v\|_A^2 = (Av, v) = \sum_{k=1}^J (Av, I_kv_k) = \sum_{k=1}^J (I_k^TAv, v_k) = \sum_{k=1}^J (R_kI_k^TAv, R_k^{-1}v_k).$$

We employ now the Cauchy-Schwarz inequality in the R_k inner-product, yielding

$$\|v\|_A^2 \leq \left(\sum_{k=1}^J (R_kR_k^{-1}v_k, R_k^{-1}v_k)\right)^{1/2}\left(\sum_{k=1}^J (R_kI_k^TAv, I_k^TAv)\right)^{1/2}$$

$$\leq \left(\frac{S_0}{\omega_0}\right)^{1/2}\|v\|_A\left(\sum_{k=1}^J (AI_kR_kI_k^TAv, Av)\right)^{1/2} = \left(\frac{S_0}{\omega_0}\right)^{1/2}\|v\|_A\left(\sum_{k=1}^J (AT_kv, v)\right)^{1/2},$$

where we have employed Lemma 5.40 for the last inequality. Dividing the inequality above by $\|v\|_A$ and squaring yields the lemma. □

In order to employ the product and sum theory, we must quantify the interaction of the operators T_k. As the T_k involve corrections in subspaces, we will see that the operator interaction properties will be determined completely by the interaction of the subspaces. Therefore, we introduce the following notions to quantify the interaction of the subspaces involved.

Definition 5.4 *(Strong interaction matrix) The interaction matrix $\Theta \in \mathbf{L}(\mathbb{R}^J, \mathbb{R}^J)$ is defined to have as entries Θ_{ij} the smallest constants satisfying:*

$$|(AI_iu_i, I_jv_j)| \leq \Theta_{ij}(AI_iu_i, I_iu_i)^{1/2}(AI_jv_j, I_jv_j)^{1/2}, \; 1 \leq i, j \leq J, \; u_i \in \mathcal{H}_i, v_j \in \mathcal{H}_j.$$

Definition 5.5 *(Weak interaction matrix) The strictly upper-triangular interaction matrix $\Xi \in \mathbf{L}(\mathbb{R}^J, \mathbb{R}^J)$ is defined to have as entries Ξ_{ij} the smallest constants satisfying:*

$$|(AI_iu_i, I_jv_j)| \leq \Xi_{ij}(AI_iu_i, I_iu_i)^{1/2}(AI_jv_j, I_jv_j)^{1/2}, \; 1 \leq i < j \leq J, \; u_i \in \mathcal{H}_i, v_j \in \mathcal{V}_j \subseteq \mathcal{H}_j.$$

The following lemma relates the interaction properties of the subspaces specified by the strong interaction matrix to the interaction properties of the associated subspace correction operators $T_k = I_kR_kI_k^TA$.

Lemma 5.42 *For the strong interaction matrix Θ given in Definition 5.4, it holds that*

$$|(AT_iu, T_jv)| \leq \Theta_{ij}(AT_iu, T_iu)^{1/2}(AT_jv, T_jv)^{1/2}, \quad 1 \leq i, j \leq J, \quad \forall u, v \in \mathcal{H}.$$

Proof. Since $T_ku = I_kR_kI_k^TAu = I_ku_k$, where $u_k = R_kI_k^TAu$, the lemma follows simply from the definition of Θ in Definition 5.4 above. □

Remark 5.9. Note that the weak interaction matrix in Definition 5.5 involves a subspace $\mathcal{V}_k \subseteq \mathcal{H}_k$, which will be necessary in the analysis of multigrid-like methods. Unfortunately, this will preclude the simple application of the product operator theory of the previous sections. In particular, we cannot estimate the constant C_2 required for the use of Corollary 5.13, because we cannot show Lemma 5.22 for arbitrary T_k. In order to prove Lemma 5.22, we would need to employ the upper-triangular portion of the strong interaction matrix Θ in Definition 5.4, involving the entire space $\mathcal{H}_k$, which is now different from the upper-triangular weak interaction matrix Ξ (employing only the subspace $\mathcal{V}_k$) defined as above in Definition 5.5. There was no such distinction between the weak and strong interaction matrices in the product and sum operator theory of the previous sections; the weak interaction matrix was defined simply as the strictly upper-triangular portion of the strong interaction matrix.

We can, however, employ the original Theorem 5.12 by attempting to estimate C_1 directly, rather than employing Corollary 5.13 and estimating C_1 indirectly through C_0 and C_2. The following result will allow us to do this, and still employ the weak interaction property above in Definition 5.5.

Lemma 5.43 *Under Assumptions 5.12 and 5.13 (for the same subspaces $I_k\mathcal{V}_k \subseteq I_k\mathcal{H}_k$), it holds that*

$$\|v\|_A^2 \le \left(\frac{S_0}{\omega_0}\right)[1+\omega_1\|\Xi\|_2]^2 \sum_{k=1}^J (AT_kE_{k-1}v, E_{k-1}v), \qquad \forall v \in \mathcal{H},$$

where Ξ is the weak interaction matrix of Definition 5.5.

Proof. We employ the splitting of Assumption 5.13, namely $v=\sum_{k=1}^J I_kv_k$, $v_k \in \mathcal{V}_k \subseteq \mathcal{H}_k$, as follows:

$$\|v\|_A^2 = \sum_{k=1}^J (Av, I_kv_k) = \sum_{k=1}^J (AE_{k-1}v, I_kv_k) + \sum_{k=1}^J (A[I-E_{k-1}]v, I_kv_k)$$

$$= \sum_{k=1}^J (AE_{k-1}v, I_kv_k) + \sum_{k=1}^J \sum_{i=1}^{k-1} (AT_iE_{i-1}v, I_kv_k) = \mathbf{S}_1 + \mathbf{S}_2.$$

We now estimate $\mathbf{S}_1$ and $\mathbf{S}_2$ separately. For the first term, we have:

$$\mathbf{S}_1 = \sum_{k=1}^J (AE_{k-1}v, I_kv_k) = \sum_{k=1}^J (I_k^TAE_{k-1}v, v_k) = \sum_{k=1}^J (R_kI_k^TAE_{k-1}v, R_k^{-1}v_k)$$

$$\le \sum_{k=1}^J (R_kI_k^TAE_{k-1}v, I_k^TAE_{k-1}v)^{1/2}(R_k^{-1}v_k, v_k)^{1/2} = \sum_{k=1}^J (AT_kE_{k-1}v, E_{k-1}v)^{1/2}(R_k^{-1}v_k, v_k)^{1/2}$$

$$\le \left(\sum_{k=1}^J (AT_kE_{k-1}v, E_{k-1}v)\right)^{1/2} \left(\sum_{k=1}^J (R_k^{-1}v_k, v_k)\right)^{1/2}.$$

where we have employed the Cauchy-Schwarz inequality in the R_k inner-product for the first inequality and in $\mathbb{R}^J$ for the second. Employing now Lemma 5.40 (requiring Assumptions 5.12 and 5.13) to bound the right-most term, we have

$$\mathbf{S}_1 \le \left(\frac{S_0}{\omega_0}\right)^{1/2} \|v\|_A \left(\sum_{k=1}^J (AT_kE_{k-1}v, E_{k-1}v)\right)^{1/2}.$$

We now bound the term $\mathbf{S}_2$, employing the weak interaction matrix given in Definition 5.5 above, as follows:

$$\mathbf{S}_2 = \sum_{k=1}^J \sum_{i=1}^{k-1} (AT_iE_{i-1}v, I_kv_k) = \sum_{k=1}^J \sum_{i=1}^{k-1} (AI_i[R_iI_i^TAE_{i-1}v], I_kv_k)$$

$$\le \sum_{k=1}^J \sum_{i=1}^J \Xi_{ik}\|I_i[R_iI_i^TAE_{i-1}v]\|_A\|I_kv_k\|_A = \sum_{k=1}^J \sum_{i=1}^J \Xi_{ik}\|T_iE_{i-1}v\|_A\|I_kv_k\|_A = (\Xi\mathbf{x},\mathbf{y})_2,$$

where $\mathbf{x},\mathbf{y} \in \mathbb{R}^J$, $x_k = \|I_kv_k\|_A$, $y_i = \|T_iE_{i-1}v\|_A$, and $(\cdot,\cdot)_2$ is the usual Euclidean inner-product in $\mathbb{R}^J$. Now, we have that

$$\mathbf{S}_2 \le (\Xi\mathbf{x},\mathbf{y})_2 \le \|\Xi\|_2\|\mathbf{x}\|_2\|\mathbf{y}\|_2 = \|\Xi\|_2 \left(\sum_{k=1}^J (AT_kE_{k-1}v, T_kE_{k-1}v)\right)^{1/2} \left(\sum_{k=1}^J (AI_kv_k, I_kv_k)\right)^{1/2}$$

$$\le \omega_1^{1/2}\|\Xi\|_2 \left(\sum_{k=1}^J (AT_kE_{k-1}v, E_{k-1}v)\right)^{1/2} \left(\sum_{k=1}^J (A_kv_k, v_k)\right)^{1/2},$$

since $A_k = I_k^T A I_k$, and by Lemma 5.9, which may be applied because of Lemma 5.38. By Lemma 5.37, we have $(A_k v_k, v_k) \leq \omega_1 (R_k^{-1} v_k, v_k)$, and employing this result along with Lemma 5.40 gives

$$\mathbf{S}_2 \leq \omega_1 \|\Xi\|_2 \left(\sum_{k=1}^{J} (A T_k E_{k-1} v, E_{k-1} v) \right)^{1/2} \left(\sum_{k=1}^{J} (R_k^{-1} v_k, v_k) \right)^{1/2}$$

$$\leq \left(\frac{S_0}{\omega_0} \right)^{1/2} \|v\|_A \omega_1 \|\Xi\|_2 \left(\sum_{k=1}^{J} (A T_k E_{k-1} v, E_{k-1} v) \right)^{1/2} .$$

Combining the two results gives finally

$$\|v\|_A^2 \leq \mathbf{S}_1 + \mathbf{S}_2 \leq \left(\frac{S_0}{\omega_0} \right)^{1/2} \|v\|_A \left[1 + \omega_1 \|\Xi\|_2 \right] \left(\sum_{k=1}^{J} (A T_k E_{k-1} v, E_{k-1} v) \right)^{1/2} , \qquad \forall v \in \mathcal{H}.$$

Dividing by $\|v\|_A$ and squaring yieldings the result. □

Remark 5.10. Although our language and notation is quite different, the proof we have given above for Lemma 5.43 is similar to results in [189] and [86]. Similar ideas and results appear [181]. The main ideas and techniques underlying proofs of this type were originally developed in [26, 27, 185].

5.3.3 Product and sum splitting theory for non-nested Schwarz methods

The main theory for Schwarz methods based on non-nested subspaces, as in the case of overlapping domain decomposition-like methods, may be summarized in the following way. We still consider an abstract method, but we assume it satisfies certain assumptions common to real overlapping Schwarz domain decomposition methods. In particular, due to the local nature of the operators T_k for $k \neq 0$ arising from subspaces associated with overlapping subdomains, it will be important to allow for a special global operator T_0 for global communication of information (the need for T_0 will be demonstrated later). Therefore, we use the analysis framework of the previous sections which includes the use of a special global operator T_0. Note that the local nature of the remaining T_k will imply that $\rho(\Theta) \leq N_c$, where N_c is the number of maximum number of subdomains which overlap any subdomain in the region.

The analysis of domain decomposition-type algorithms is in most respects a straightforward application of the theory of products and sums of operators, as presented earlier. The theory for multigrid-type algorithms is more subtle; we will discuss this in the next section.

Let the operators E and P be defined as:

$$E = (I - T_J)(I - T_{J-1}) \cdots (I - T_0), \tag{5.22}$$

$$P = T_0 + T_1 + \cdots + T_J, \tag{5.23}$$

where the operators $T_k \in \mathbf{L}(\mathcal{H}, \mathcal{H})$ are defined in terms of the approximate corrections in the spaces $\mathcal{H}_k$ as:

$$T_k = I_k R_k I_k^T A, \qquad k = 0, \ldots, J, \tag{5.24}$$

where

$$I_k : \mathcal{H}_k \mapsto \mathcal{H}, \qquad \text{null}(I_k) = \{0\}, \qquad I_k \mathcal{H}_k \subseteq \mathcal{H}, \qquad \mathcal{H} = \sum_{k=1}^{J} I_k \mathcal{H}_k.$$

The following assumptions are required; note that the following theory employs many of the assumptions and lemmas of the previous sections, for the case that $\mathcal{V}_k \equiv \mathcal{H}_k$.

Assumption 5.14 *(Subspace solvers) The operators $R_k \in \mathbf{L}(\mathcal{H}_k, \mathcal{H}_k)$ are SPD. Further, there exists parameters $0 < \omega_0 \leq \omega_1 < 2$, such that*

$$\omega_0 (A_k v_k, v_k) \leq (A_k R_k A_k v_k, v_k) \leq \omega_1 (A_k v_k, v_k), \quad \forall v_k \in \mathcal{H}_k, \quad k = 0, \ldots, J.$$

Assumption 5.15 *(Splitting constant) Given any $v \in \mathcal{H}$, there exists $S_0 > 0$ and a particular splitting $v = \sum_{k=0}^{J} I_k v_k$, $v_k \in \mathcal{H}_k$, such that*

$$\sum_{k=0}^{J} \|I_k v_k\|_A^2 \leq S_0 \|v\|_A^2.$$

Definition 5.6 *(Interaction matrix) The interaction matrix $\Theta \in \mathbf{L}(\mathbb{R}^J, \mathbb{R}^J)$ is defined to have as entries Θ_{ij} the smallest constants satisfying:*

$$|(AI_i u_i, I_j v_j)| \leq \Theta_{ij}(AI_i u_i, I_i u_i)^{1/2}(AI_j v_j, I_j v_j)^{1/2}, \ 1 \leq i, j \leq J, \ u_i \in \mathcal{H}_i, v_j \in \mathcal{H}_j.$$

Theorem 5.44 *(Multiplicative method) Under Assumptions 5.14 and 5.15, it holds that*

$$\|E\|_A^2 \leq 1 - \frac{\omega_0(2-\omega_1)}{S_0(6 + 6\omega_1^2 \rho(\Theta)^2)}.$$

Proof. By Lemma 5.38, Assumption 5.14 implies that Assumption 5.10 holds, with $\omega = \omega_1$. By Lemma 5.41, we know that Assumptions 5.14 and 5.15 imply that Assumption 5.11 holds, with $C_0 = S_0/\omega_0$. By Lemma 5.42, we know that Definition 5.6 is equivalent to Definition 5.2 for Θ. Therefore, the theorem follows by application of Theorem 5.30. □

Theorem 5.45 *(Additive method) Under Assumptions 5.14 and 5.15, it holds that*

$$\kappa_A(P) \leq \frac{S_0(\rho(\Theta)+1)\omega_1}{\omega_0}.$$

Proof. By Lemma 5.38, Assumption 5.14 implies that Assumption 5.10 holds, with $\omega = \omega_1$. By Lemma 5.41, we know that Assumptions 5.14 and 5.15 imply that Assumption 5.11 holds, with $C_0 = S_0/\omega_0$. By Lemma 5.42, we know that Definition 5.6 is equivalent to Definition 5.2 for Θ. Therefore, the theorem follows by application of Theorem 5.31. □

Remark 5.11. Note that Assumption 5.14 is equivalent to

$$\kappa_A(R_k A_k) \leq \frac{\omega_1}{\omega_0}, \qquad k = 0, \ldots, J,$$

or $\max_k\{\kappa_A(R_k A_k)\} \leq \omega_1/\omega_0$. Thus, the result in Theorem 5.45 can be written as:

$$\kappa_A(P) \leq S_0(\rho(\Theta)+1)\max_k\{\kappa_A(R_k A_k)\}.$$

Therefore, the *global* condition number is completely determined by the *local* condition numbers, the splitting constant, and the interaction property.

Remark 5.12. We have the default estimate for $\rho(\Theta)$:

$$\rho(\Theta) \leq J.$$

For use of the theory above, we must also estimate the splitting constant S_0, and the subspace solver spectral bounds ω_0 and ω_1, for each particular application.

Remark 5.13. Note that if a coarse space operator T_0 is not present, then the alternate bounds from the previous sections could have been employed. However, the advantage of the above approach is that the additional space $\mathcal{H}_0$ does not adversely effect the bounds, while it provides an additional space to help satisfy the splitting assumption. In fact, in the finite element case, it is exactly this coarse space which allows one to show that S_0 does not depend on the number of subspaces, yielding optimal algorithms when a coarse space is involved.

Remark 5.14. The theory in this section was derived mainly from work in the domain decomposition community, due chiefly to Widlund and his co-workers. In particular, our presentation owes much to [185] and [59].

5.3.4 Product and sum splitting theory for nested Schwarz methods

The main theory for Schwarz methods based on nested subspaces, as in the case of multigrid-like methods, is summarized in this section. By "nested" subspaces, we mean here that there are additional subspaces $\mathcal{V}_k \subseteq \mathcal{H}_k$ of importance, and we refine the analysis to consider these addition nested subspaces $\mathcal{V}_k$. Of course, we must still assume that $\sum_{k=1}^J I_k\mathcal{V}_k = \mathcal{H}$. Later, when analyzing multigrid methods, we will consider in fact a nested sequence $I_1\mathcal{H}_1 \subseteq I_2\mathcal{H}_2 \subseteq \cdots \subseteq \mathcal{H}_J \equiv \mathcal{H}$, with $\mathcal{V}_k \subseteq \mathcal{H}_k$, although this assumption is not necessary here. We will however assume here that one space $\mathcal{H}_1$ automatically performs the role of a "global" space, and hence it will not be necessary to include a special global space $\mathcal{H}_0$ as in the non-nested case. Therefore, we will employ the analysis framework of the previous sections which does not specifically include a special global operator T_0. (By working with the subspaces $\mathcal{V}_k$ rather than the $\mathcal{H}_k$ we will be able to avoid the problems encountered with a global operator interacting with all other operators, as in the previous sections.)

The analysis of multigrid-type algorithms is more subtle than analysis for overlapping domain decomposition methods, in that the efficiency of the method comes from the effectiveness of simple linear methods (e.g., Gauss-Seidel iteration) at reducing the error in a certain sub-subspace $\mathcal{V}_k$ of the "current" space $\mathcal{H}_k$. The overall effect on the error is not important; just the effectiveness of the linear method on error subspace $\mathcal{V}_k$. The error in the remaining space $\mathcal{H}_k \backslash \mathcal{V}_k$ is handled by subspace solvers in the other subspaces, since we assume that $\mathcal{H} = \sum_{k=1}^J I_k\mathcal{V}_k$. Therefore, in the analysis of the nested space methods to follow, the spaces $\mathcal{V}_k \subseteq \mathcal{H}_k$ introduced earlier will play a key role. This is in contrast to the non-nested theory of the previous section, where it was taken to be the case that $\mathcal{V}_k \equiv \mathcal{H}_k$. Roughly speaking, nested space algorithms "split" the error into components in $\mathcal{V}_k$, and if the subspace solvers in each space $\mathcal{H}_k$ are good at reducing the error in $\mathcal{V}_k$, then the overall method will be good.

Let the operators E and P be defined as:

$$E = (I - T_J)(I - T_{J-1}) \cdots (I - T_1), \tag{5.25}$$

$$P = T_1 + T_2 + \cdots + T_J, \tag{5.26}$$

where the operators $T_k \in \mathbf{L}(\mathcal{H}, \mathcal{H})$ are defined in terms of the approximate corrections in the spaces $\mathcal{H}_k$ as:

$$T_k = I_k R_k I_k^T A, \qquad k = 1, \ldots, J, \tag{5.27}$$

where

$$I_k : \mathcal{H}_k \mapsto \mathcal{H}, \qquad \text{null}(I_k) = \{0\}, \qquad I_k\mathcal{H}_k \subseteq \mathcal{H}, \qquad \mathcal{H} = \sum_{k=1}^J I_k\mathcal{H}_k.$$

The following assumptions are required.

Assumption 5.16 *(Subspace solvers) The operators $R_k \in \mathbf{L}(\mathcal{H}_k, \mathcal{H}_k)$ are SPD. Further, there exists subspaces $I_k\mathcal{V}_k \subseteq I_k\mathcal{H}_k \subseteq \mathcal{H} = \sum_{k=1}^J I_k\mathcal{V}_k$, and parameters $0 < \omega_0 \le \omega_1 < 2$, such that*

$$\omega_0(A_k v_k, v_k) \le (A_k R_k A_k v_k, v_k), \quad \forall v_k \in \mathcal{V}_k \subseteq \mathcal{H}_k, \quad k = 1, \ldots, J,$$

$$(A_k R_k A_k v_k, v_k) \le \omega_1(A_k v_k, v_k), \quad \forall v_k \in \mathcal{H}_k, \quad k = 1, \ldots, J.$$

Assumption 5.17 *(Splitting constant) Given any $v \in \mathcal{H}$, there exists subspaces $I_k\mathcal{V}_k \subseteq I_k\mathcal{H}_k \subseteq \mathcal{H} = \sum_{k=1}^J I_k\mathcal{V}_k$ (the same subspaces $\mathcal{V}_k$ as in Assumption 5.16 above) and a particular splitting $v = \sum_{k=1}^J I_k v_k$, $v_k \in \mathcal{V}_k$, such that*

$$\sum_{k=1}^J \|I_k v_k\|_A^2 \le S_0 \|v\|_A^2, \quad \forall v \in \mathcal{H},$$

for some splitting constant $S_0 > 0$.

Definition 5.7 *(Strong interaction matrix) The interaction matrix $\Theta \in \mathbf{L}(\mathbb{R}^J, \mathbb{R}^J)$ is defined to have as entries Θ_{ij} the smallest constants satisfying:*

$$|(AI_i u_i, I_j v_j)| \le \Theta_{ij}(AI_i u_i, I_i u_i)^{1/2}(AI_j v_j, I_j v_j)^{1/2}, \ 1 \le i, j \le J, \ u_i \in \mathcal{H}_i, v_j \in \mathcal{H}_j.$$

Definition 5.8 *(Weak interaction matrix) The strictly upper-triangular interaction matrix $\Xi \in \mathbf{L}(\mathbb{R}^J, \mathbb{R}^J)$ is defined to have as entries Ξ_{ij} the smallest constants satisfying:*

$$|(AI_iu_i, I_jv_j)| \leq \Xi_{ij}(AI_iu_i, I_iu_i)^{1/2}(AI_jv_j, I_jv_j)^{1/2}, \ 1 \leq i < j \leq J, \ u_i \in \mathcal{H}_i, v_j \in \mathcal{V}_j \subseteq \mathcal{H}_j.$$

Theorem 5.46 *(Multiplicative method) Under Assumptions 5.16 and 5.17, it holds that*

$$\|E\|_A^2 \leq 1 - \frac{\omega_0(2-\omega_1)}{S_0(1+\omega_1\|\Xi\|_2)^2}.$$

Proof. The proof of this result is more subtle than the additive method, and requires more work than a simple application of the product operator theory. This is due to the fact that the weak interaction matrix of Definition 5.8 specifically involves the subspace $\mathcal{V}_k \subseteq \mathcal{H}_k$. Therefore, rather than employing Theorem 5.32, which employs Corollary 5.13 indirectly, we must do a more detailed analysis, and employ the original Theorem 5.12 directly. (See the remarks preceding Lemma 5.43.)

By Lemma 5.38, Assumption 5.16 implies that Assumption 5.3 holds, with $\omega = \omega_1$. Now, to employ Theorem 5.12, it suffices to realize that Assumption 5.5 holds with with $C_1 = S_0(1 + \omega_1\|\Xi\|_2)^2/\omega_0$. This follows from Lemma 5.43. □

Theorem 5.47 *(Additive method) Under Assumptions 5.16 and 5.17, it holds that*

$$\kappa_A(P) \leq \frac{S_0\rho(\Theta)\omega_1}{\omega_0}.$$

Proof. By Lemma 5.38, Assumption 5.16 implies that Assumption 5.10 holds, with $\omega = \omega_1$. By Lemma 5.41, we know that Assumptions 5.16 and 5.17 imply that Assumption 5.11 holds, with $C_0 = S_0/\omega_0$. By Lemma 5.42, we know that Definition 5.7 is equivalent to Definition 5.2 for Θ. Therefore, the theorem follows by application of Theorem 5.33. □

Remark 5.15. We have the default estimates for $\|\Xi\|_2$ and $\rho(\Theta)$:

$$\|\Xi\|_2 \leq \sqrt{J(J-1)/2} < J, \qquad \rho(\Theta) \leq J.$$

For use of the theory above, we must also estimate the splitting constant S_0, and the subspace solver spectral bounds ω_0 and ω_1, for each particular application.

Remark 5.16. The theory in this section was derived from several sources; in particular, our presentation owes much to [185], [86], and to [189].

5.4 Applications to domain decomposition

Domain decomposition methods were first proposed by H.A. Schwarz as a theoretical tool for studying elliptic problems on complicated domains, constructed as the union of simple domains. An interesting early reference not often mentioned is [118], containing both analysis and numerical examples, and references to the original work by Schwarz. In this section, we briefly describe the fundamental overlapping domain decomposition methods, and apply the theory of the previous sections to give convergence rate bounds.

5.4.1 Variational formulation and subdomain-based subspaces

Given a domain Ω and coarse triangulation by J regions $\{\Omega_k\}$ of mesh size H_k, we refine (several times) to obtain a fine mesh of size h_k. The regions defined by the initial triangulation Ω_k are then extended by δ_k to form the "overlapping subdomains" Ω_k'. Now, let V and V_0 denote the finite element spaces associated with the h_k and H_k triangulation of Ω, respectively. The variational problem in V has the form:

$$\text{Find } u \in V \text{ such that } a(u,v) = f(v), \quad \forall v \in V.$$

The form $a(\cdot,\cdot)$ is bilinear, symmetric, coercive, and bounded, whereas $f(\cdot)$ is linear and bounded. Therefore, through the Riesz representation theorem we can associate with the above problem an abstract operator equation $Au = f$, where A is SPD.

Domain decomposition methods can be seen as iterative methods for solving the above operator equation, involving approximate projections of the error onto subspaces of V associated with the overlapping subdomains Ω'_k. To be more specific, let $V_k = H^1_0(\Omega'_k) \cap V$, $k = 1, \ldots, J$; it is not difficult to show that $V = V_1 + \cdots + V_J$, where the coarse space V_0 may also be included in the sum.

5.4.2 The multiplicative and additive Schwarz methods

We denote as A_k the restriction of the operator A to the space V_k, corresponding to (any) discretization of the original problem restricted to the subdomain Ω'_k. Algebraically, it can be shown that $A_k = I_k^T A I_k$, where I_k is the natural inclusion in $\mathcal{H}$ and I_k^T is the corresponding projection. The property that I_k is the natural inclusion and I_k^T is the corresponding projection holds if either $\mathcal{V}_k$ is a finite element space or the Euclidean space $\mathbb{R}^{n_k}$ (in the case of multigrid, I_k and I_k^T are inclusion and projection only in the finite element space case). In other words, domain decomposition methods automatically satisfy the variational condition, Definition 5.3, in the subspaces V_k, $k \neq 0$, for *any* discretization method.

Now, if $R_k \approx A_k^{-1}$, we can define the approximate A-orthogonal projector from V onto V_k as $T_k = I_k R_k I_k^T A$. An overlapping domain decomposition method can be written as the basic linear method, Algorithm 3.1, where the *multiplicative Schwarz* error propagator E is:

$$E = (I - T_J)(I - T_{J-1}) \cdots (I - T_0).$$

The *additive Schwarz* preconditioned system operator P is:

$$P = T_0 + T_1 + \cdots + T_J.$$

Therefore, the overlapping multiplicative and additive domain decomposition methods fit exactly into the framework of abstract multiplicative and additive Schwarz methods discussed in the previous sections.

5.4.3 Algebraic domain decomposition methods

As remarked above, for domain decomposition methods it automatically holds that $A_k = I_k^T A I_k$, where I_k is the natural inclusion, I_k^T is the corresponding projection, and V_k is either a finite element space or $\mathbb{R}^{n_k}$. While this *variational condition* holds for multigrid methods only in the case of finite element discretizations, or when directly enforced as in algebraic multigrid methods (see the next section), the condition holds naturally and automatically for domain decomposition methods employing any discretization technique.

We see that the Schwarz method framework then applies equally well to domain decomposition methods based on other discretization techniques (box-method or finite differences), or to algebraic equations having a block-structure which can be viewed as being associated with the discretization of an elliptic equation over a domain. The Schwarz framework can be used to provide a convergence analysis even in the algebraic case, although the results may be suboptimal compared to the finite element case when more information is available about the continuous problem.

5.4.4 Convergence theory for the algebraic case

For domain decomposition methods, the local nature of the projection operators will allow for a simple analysis of the interaction properties required for the Schwarz theory. To quantify the local nature of the projection operators, assume that we are given $\mathcal{H} = \sum_{k=0}^J I_k \mathcal{H}_k$ along with the subspaces $I_k \mathcal{H}_k \subseteq \mathcal{H}$, and denote as P_k the A-orthogonal projector onto $I_k \mathcal{H}_k$. We now make the following definition.

Definition 5.9 *For each operator P_k, $1 \le k \le J$, define $N_c^{(k)}$ to be the number of operators P_i such that $P_k P_i \neq 0$, $1 \le i \le J$, and let $N_c = \max_{1 \le k \le J}\{N_c^{(k)}\}$.*

Remark 5.17. This is a natural condition for domain decomposition methods, where $N_c^{(k)}$ represents the number of subdomains which overlap a given domain associated with P_k, excluding a possible coarse space

$I_0\mathcal{H}_0$. By treating the projector P_0 separately in the analysis, we allow for a global space $\mathcal{H}_0$ which may in fact interact with all of the other spaces. Note that $N_c \le J$ in general with Schwarz methods; with domain decomposition, we can show that $N_c = O(1)$. Our use of the notation N_c comes from the idea that N_c represents essentially the minimum number of colors required to color the subdomains so that no two subdomains sharing interior mesh points have the same color. (If the domains were non-overlapping, then this would be a case of the four-color problem, so that in two dimensions it would always hold that $N_c \le 4$.)

The following splitting is the basis for applying the theory of the previous sections. Note that this splitting is well-defined in a completely algebraic setting without further assumptions.

Lemma 5.48 *Given any $v \in \mathcal{H} = \sum_{k=0}^J I_k\mathcal{H}_k$, $I_k\mathcal{H}_k \subseteq \mathcal{H}$, there exists a particular splitting $v = \sum_{k=0}^J I_k v_k$, $v_k \in \mathcal{H}_k$, such that*

$$\sum_{k=0}^J \|I_k v_k\|_A^2 \le S_0 \|v\|_A^2,$$

for the splitting constant $S_0 = \sum_{k=0}^J \|Q_k\|_A^2$.

Proof. Let $Q_k \in \mathbf{L}(\mathcal{H}, \mathcal{H}_k)$ be the orthgonal projectors onto the subspaces $\mathcal{H}_k$. We have that $\mathcal{H}_k = Q_k\mathcal{H}$, and any $v \in \mathcal{H}$ can be represented uniquely as

$$v = \sum_{k=0}^J Q_k v = \sum_{k=0}^J I_k v_k, \qquad v_k \in \mathcal{H}_k.$$

We have then that

$$\sum_{k=0}^J \|I_k v_k\|_A^2 = \sum_{k=0}^J \|Q_k v\|_A^2 \le \sum_{k=0}^J \|Q_k\|_A^2 \|v\|_A^2 = S_0\|v\|_A^2,$$

where $S_0 = \sum_{k=0}^J \|Q_k\|_A^2$. □

Lemma 5.49 *It holds that $\rho(\Theta) \le N_c$.*

Proof. This follows easily, since $\rho(\Theta) \le \|\Theta\|_1 = \max_j\{\sum_i |\Theta_{ij}|\} \le N_c$. □

We make the following assumption on the subspace solvers.

Assumption 5.18 *Assume there exists SPD operators $R_k \in \mathbf{L}(\mathcal{H}_k, \mathcal{H}_k)$ and parameters $0 < \omega_0 \le \omega_1 < 2$, such that*

$$\omega_0(A_k v_k, v_k) \le (A_k R_k A_k v_k, v_k) \le \omega_1(A_k v_k, v_k), \quad \forall v_k \in \mathcal{H}_k, \quad k = 1, \ldots, J.$$

Theorem 5.50 *Under Assumption 5.18, the multiplicative Schwarz domain decomposition method has an error propagator which satisfies:*

$$\|E\|_A^2 \le 1 - \frac{\omega_0(2-\omega_1)}{S_0(6 + 6\omega_1^2 N_c^2)}.$$

Proof. By Assumption 5.18, we have that Assumption 5.14 holds. By Lemma 5.48, we have that Assumption 5.15 holds, with $S_0 = \sum_{k=0}^J \|Q_k\|_A^2$. By Lemma 5.49, we have that for Θ as in Definition 5.6, it holds that $\rho(\Theta) \le N_c$. The proof now follows from Theorem 5.44. □

Theorem 5.51 *Under Assumption 5.18, the additive Schwarz domain decomposition method as a preconditioner gives a condition number bounded by:*

$$\kappa_A(P) \le S_0(1 + N_c)\frac{\omega_1}{\omega_0}.$$

Proof. By Assumption 5.18, we have that Assumption 5.14 holds. By Lemma 5.48, we have that Assumption 5.15 holds, with $S_0 = \sum_{k=0}^J \|Q_k\|_A^2$. By Lemma 5.49, we have that for Θ as in Definition 5.6, it holds that $\rho(\Theta) \le N_c$. The proof now follows from Theorem 5.45. □

5.4.5 Improved results through finite element theory

If a coarse space is employed, and the overlap of the subdomains δ_k is on the order of the subdomain size H_k, i.e., $\delta_k = cH_k$, then one can bound the splitting constant S_0 to be independent of the mesh size and the number of subdomains J. Required to prove such a result is some elliptic regularity or smoothness on the solution to the original continuous problem:

$$\text{Find } u \in H_0^1(\Omega) \text{ such that } a(u,v) = (f,v), \quad \forall v \in H_0^1(\Omega).$$

The regularity assumption is stated as an apriori estimate or regularity inequality of the following form: The solution to the continuous problem satisfies $u \in H^{1+\alpha}(\Omega)$ for some real number $\alpha > 0$, and there exists a constant C such that

$$\|u\|_{H^{1+\alpha}(\Omega)} \leq C\|f\|_{H^{\alpha-1}(\Omega)}.$$

If this regularity inequality holds for the continuous solution, one can show the following result by employing some results from interpolation theory and finite element approximation theory.

Lemma 5.52 *There exists a splitting $v = \sum_{k=0}^J I_k v_k$, $v_k \in \mathcal{H}_k$ such that*

$$\sum_{k=0}^J \|I_k v_k\|_A^2 \leq S_0 \|v\|_A^2, \quad \forall v \in \mathcal{H},$$

where S_0 is independent of J (and h_k and H_k).

Proof. Refer for example to the proof in [185] and the references therein to related results. □

5.5 Applications to multigrid

Multigrid methods were first developed by Federenko in the early 1960's, and have been extensively studied and developed since they became widely known in the late 1970's. In this section, we briefly describe the linear multigrid method as a Schwarz method, and apply the theory of the previous sections to give convergence rate bounds.

5.5.1 Recursive multigrid and nested subspaces

Consider a set of finite-dimensional Hilbert spaces $\mathcal{H}_k$ of increasing dimension:

$$\dim(\mathcal{H}_1) < \dim(\mathcal{H}_2) < \cdots < \dim(\mathcal{H}_J).$$

The spaces $\mathcal{H}_k$, which may for example be finite element function spaces, or simply $\mathbf{R}^{n_k}$ (where $n_k = \dim(\mathcal{H}_k)$), are assumed to be connected by prolongation operators $I_{k-1}^k \in \mathbf{L}(\mathcal{H}_{k-1}, \mathcal{H}_k)$, and restriction operators $I_k^{k-1} \in \mathbf{L}(\mathcal{H}_k, \mathcal{H}_{k-1})$. We can use these various operators to define mappings I_k that provide a nesting structure for the set of spaces $\mathcal{H}_k$ as follows:

$$I_1\mathcal{H}_1 \subset I_2\mathcal{H}_2 \subset \cdots \subset I_J\mathcal{H}_J \equiv \mathcal{H},$$

where

$$I_J = I, \quad I_k = I_{J-1}^J I_{J-2}^{J-1} \cdots I_{k+1}^{k+2} I_k^{k+1}, \quad k = 1, \ldots, J-1.$$

We assume that each space $\mathcal{H}_k$ is equipped with an inner-product $(\cdot,\cdot)_k$ inducing the norm $\|\cdot\|_k = (\cdot,\cdot)_k^{1/2}$. Also associated with each $\mathcal{H}_k$ is an operator A_k, assumed to be SPD with respect to $(\cdot,\cdot)_k$. It is assumed that the operators satisfy *variational conditions*:

$$A_{k-1} = I_k^{k-1} A_k I_{k-1}^k, \quad I_k^{k-1} = (I_{k-1}^k)^T. \tag{5.28}$$

These conditions hold naturally in the finite element setting, and are imposed directly in algebraic multigrid methods.

Given $B \approx A^{-1}$ in the space $\mathcal{H}$, the *basic linear method* constructed from the preconditioned system $BAu = Bf$ has the form:

$$u^{n+1} = u^n - BAu^n + Bf = (I - BA)u^n + Bf. \tag{5.29}$$

Now, given some B, or some procedure for applying B, we can either formulate a linear method using $E = I - BA$, or employ a CG method for $BAu = Bf$ if B is SPD.

5.5.2 Variational multigrid as a multiplicative Schwarz method

The recursive formulation of multigrid methods has been well-known for more than fifteen years; mathematically equivalent forms of the method involving product error propagators have been recognized and exploited theoretically only very recently. In particular, it can be shown [26, 94, 147] that if the variational conditions (5.28) hold, then the multigrid error propagator can be factored as:

$$E = I - BA = (I - T_J)(I - T_{J-1}) \cdots (I - T_1), \tag{5.30}$$

where:

$$I_J = I, \qquad I_k = I_{J-1}^{J} I_{J-2}^{J-1} \cdots I_{k+1}^{k+2} I_k^{k+1}, \qquad k = 1, \ldots, J-1, \tag{5.31}$$

$$T_1 = I_1 A_1^{-1} I_1^T A, \qquad T_k = I_k R_k I_k^T A, \qquad k = 2, \ldots, J, \tag{5.32}$$

where $R_k \approx A_k^{-1}$ is the "smoothing" operator employed in each space $\mathcal{H}_k$. It is not difficult to show that with the definition of I_k in equation (5.31), the variational conditions (5.28) imply that additional variational conditions hold between the finest space and each of the subspaces separately, as required for the Schwarz theory:

$$A_k = I_k^T A I_k. \tag{5.33}$$

5.5.3 Algebraic multigrid methods

Equations arising in various application areas often contain complicated discontinuous coefficients, the shapes of which may not be resolvable on all coarse mesh element boundaries as required for accurate finite element approximation (and as required for validity of finite element error estimates). Multigrid methods typically perform badly, and even the regularity-free multigrid convergence theory [26] is invalid.

Possible approaches include coefficient averaging methods (cf. [2]) and the explicit enforcement of the conditions (5.28) (cf. [2, 51, 166]). By introducing a symbolic stencil calculus and employing MAPLE or MATHEMATICA, the conditions (5.28) can be enforced algebraically in an efficient way for certain types of sparse matrices; details may be found for example in the appendix of [94].

If one imposes the variational conditions (5.28) algebraically, then from our comments in the previous section we know that algebraic multigrid methods can be viewed as multiplicative Schwarz methods, and we can attempt to analyze the convergence rate of algebraic multigrid methods using the Schwarz theory framework.

5.5.4 Convergence theory for the algebraic case

The following splitting is the basis for applying the theory of the previous sections. Note that this splitting is well-defined in a completely algebraic setting without further assumptions.

Lemma 5.53 *Given any $v \in \mathcal{H} = \sum_{k=0}^{J} I_k \mathcal{H}_k$, $I_{k-1}\mathcal{H}_{k-1} \subseteq I_k \mathcal{H}_k \subseteq \mathcal{H}$, there exists subspaces $I_k \mathcal{V}_k \subseteq I_k \mathcal{H}_k \subseteq \mathcal{H} = \sum_{k=1}^{J} I_k \mathcal{V}_k$, and a particular splitting $v = \sum_{k=0}^{J} I_k v_k$, $v_k \in \mathcal{V}_k$, such that*

$$\sum_{k=0}^{J} \|I_k v_k\|_A^2 \equiv \|v\|_A^2.$$

The subspaces are $I_k \mathcal{V}_k = (P_k - P_{k-1})\mathcal{H}$, and the splitting is $v = \sum_{k=1}^{J} (P_k - P_{k-1}) v$.

Proof. We have the projectors $P_k : \mathcal{H} \mapsto I_k \mathcal{H}_k$ as defined in Lemma 5.35, where we take the convention that $P_J = I$, and that $P_0 = 0$. Since $I_{k-1}\mathcal{H}_{k-1} \subset I_k \mathcal{H}_k$, we know that $P_k P_{k-1} = P_{k-1} P_k = P_{k-1}$. Now, let us define:

$$\hat{P}_1 = P_1, \qquad \hat{P}_k = P_k - P_{k-1}, \qquad k = 2, \ldots, J.$$

By Theorem 9.6-2 in [129] we have that each $\hat{P}_k$ is a projection. (It is easily verified that $\hat{P}_k$ is idempotent and A-self-adjoint.) Define now

$$I_k \mathcal{V}_k = \hat{P}_k \mathcal{H} = (P_k - P_{k-1})\mathcal{H} = (I_k A_k^{-1} I_k^T A - I_{k-1} A_{k-1}^{-1} I_{k-1}^T A)\mathcal{H}$$

$$= I_k(A_k^{-1} - I_{k-1}^k A_{k-1}^{-1}(I_{k-1}^k)^T)I_k^T A\mathcal{H}, \qquad k = 1, \ldots, J,$$

where we have used the fact that two forms of variational conditions hold, namely those of equation (5.28) and equation (5.33). Note that

$$\hat{P}_k\hat{P}_j = (P_k - P_{k-1})(P_j - P_{j-1}) = P_kP_j - P_kP_{j-1} - P_{k-1}P_j + P_{k-1}P_{j-1}.$$

Thus, if $k > j$, then

$$\hat{P}_k\hat{P}_j = P_j - P_{j-1} - P_j + P_{j-1} = 0.$$

Similarly, if $k < j$, then

$$\hat{P}_k\hat{P}_j = P_k - P_k - P_{k-1} + P_{k-1} = 0.$$

Thus,

$$\mathcal{H} = I_1\mathcal{V}_1 \oplus I_2\mathcal{V}_2 \oplus \cdots \oplus I_J\mathcal{V}_J = \hat{P}_1\mathcal{H} \oplus \hat{P}_2\mathcal{H} \oplus \cdots \oplus \hat{P}_J\mathcal{H},$$

and $P = \sum_{k=1}^J \hat{P}_k = I$ defines a splitting (an A-orthogonal splitting) of $\mathcal{H}$. We then have that

$$\|v\|_A^2 = (APv, v) = \sum_{k=1}^J (A\hat{P}_kv, v) = \sum_{k=1}^J (A\hat{P}_kv, \hat{P}_kv) = \sum_{k=1}^J \|\hat{P}_kv\|_A^2 = \sum_{k=1}^J \|I_kv_k\|_A^2.$$

□

For the particular splitting employed above, the weak interaction property is quite simple.

Lemma 5.54 *The (strictly upper-triangular) interaction matrix* $\Xi \in \mathbf{L}(\mathbb{R}^J, \mathbb{R}^J)$*, having entries* Ξ_{ij} *as the smallest constants satisfying:*

$$|(AI_iu_i, I_jv_j)| \leq \Xi_{ij}(AI_iu_i, I_iu_i)^{1/2}(AI_jv_j, I_jv_j)^{1/2}, \ 1 \leq i < j \leq J, \ u_i \in \mathcal{H}_i, v_j \in \mathcal{V}_j \subseteq \mathcal{H}_j,$$

satisfies $\Xi \equiv 0$ *for the subspace splitting* $I_k\mathcal{V}_k = \hat{P}_k\mathcal{H} = (P_k - P_{k-1})\mathcal{H}$*.*

Proof. Since $\hat{P}_jP_i = (P_j - P_{j-1})P_i = P_jP_i - P_{j-1}P_i = P_i - P_i = 0$ for $i < j$, we have that $I_j\mathcal{V}_j = \hat{P}_j\mathcal{H}$ is orthogonal to $I_i\mathcal{H}_i = P_i\mathcal{H}$, for $i < j$. Thus, it holds that

$$(AI_iu_i, I_jv_j) = 0, \ 1 \leq i < j \leq J, \ u_i \in \mathcal{H}_i, v_j \in \mathcal{V}_j \subseteq \mathcal{H}_j.$$

□

The most difficult assumption to verify will be the following one.

Assumption 5.19 *There exists SPD operators* R_k *and parameters* $0 < \omega_0 \leq \omega_1 < 2$ *such that*

$$\omega_0(A_kv_k, v_k) \leq (A_kR_kA_kv_k, v_k), \quad \forall v_k \in \mathcal{V}_k, \ I_k\mathcal{V}_k = (P_k - P_{k-1})\mathcal{H} \subseteq I_k\mathcal{H}_k, \ k = 1, \ldots, J,$$

$$(A_kR_kA_kv_k, v_k) \leq \omega_1(A_kv_k, v_k), \quad \forall v_k \in \mathcal{H}_k, \quad k = 1, \ldots, J.$$

With this single assumption, we can state the main theorem.

Theorem 5.55 *Under Assumption 5.19, the multigrid method has an error propagator which satisfies:*

$$\|E\|_A^2 \leq 1 - \omega_0(2 - \omega_1).$$

Proof. By Assumption 5.19, Assumption 5.16 holds. The splitting in Lemma 5.53 shows that Assumption 5.17 holds, with $S_0 = 1$. Lemma 5.54 shows that for Ξ as in Definition 5.8, it holds that $\Xi \equiv 0$. The theorem now follows by Theorem 5.46. □

Remark 5.18. In order to analyze the convergence rate of an algebraic multigrid method, we now see that we must be able to estimate the two parameters ω_0 and ω_1 in Assumption 5.19. However, in an algebraic multigrid method, we are free to choose the prolongation operator I_k, which of course also influences $A_k = I_k^TAI_k$. Thus, we can attempt to select the prolongation operator I_k and the subspace solver R_k together, so that Assumption 5.19 will hold, independent of the number of levels J employed. In other words, the Schwarz theory framework can be used to help design an effective algebraic multigrid method. Whether it will be possible to select R_k and I_k satisfying the above requirements is the subject of future work.

5.5.5 Improved results through finite element theory

It can be shown that Assumption 5.19 holds for parameters ω_0 and ω_1 independent of the mesh size and number of levels J, if one assumes some elliptic regularity or smoothness on the solution to the original continuous problem:

$$\text{Find } u \in H_0^1(\Omega) \text{ such that } a(u,v) = (f,v), \quad \forall v \in H_0^1(\Omega).$$

This regularity assumption is stated as an apriori estimate or regularity inequality of the following form: The solution to the continuous problem satisfies $u \in H^{1+\alpha}(\Omega)$ for some real number $\alpha > 0$, and there exists a constant C such that

$$\|u\|_{H^{1+\alpha}(\Omega)} \leq C\|f\|_{H^{\alpha-1}(\Omega)}.$$

If this regularity inequality holds with $\alpha = 1$ for the continuous solution, one can show the following result by employing some results from interpolation theory and finite element approximation theory.

Lemma 5.56 *There exists SPD operators R_k and parameters $0 < \omega_0 \leq \omega_1 < 2$ such that*

$$\omega_0(A_k v_k, v_k) \leq (A_k R_k A_k v_k, v_k), \quad \forall v_k \in \mathcal{V}_k,\ I_k\mathcal{V}_k = (P_k - P_{k-1})\mathcal{H} \subseteq I_k\mathcal{H}_k,\ k = 1,\ldots,J,$$

$$(A_k R_k A_k v_k, v_k) \leq \omega_1(A_k v_k, v_k), \quad \forall v_k \in \mathcal{H}_k, \quad k = 1,\ldots,J.$$

Proof. See for example the proof in [189]. □

More generally, assume only that $u \in H^1(\Omega)$ (so that the regularity inequality holds only with $\alpha = 0$), and that there exists $L^2(\Omega)$-like orthogonal projectors Q_k onto the finite element spaces $\mathcal{M}_k$, where we take the convention that $Q_J = I$ and $Q_0 = 0$. This defines the splitting

$$v = \sum_{k=1}^{J}(Q_k - Q_{k-1})v,$$

which is central to the BPWX theory [26]. Employing this splitting along with results from finite element approximation theory, it is shown in [26], using a similar Schwarz theory framework, that

$$\|E\|_A^2 \leq 1 - \frac{C}{J^{1+\nu}}, \quad \nu \in \{0,1\}.$$

This result holds even in the presence of coefficient discontinuities (the constants being independent of the jumps in the coefficients). The restriction is that all discontinuities lie along all element boundaries on all levels. The constant ν depends on whether coefficient discontinuity "cross-points" are present.

6. Application to the Linearized PBE

Numerical experiments are performed to investigate the effectiveness of the linear multilevel methods when applied to the linearized Poisson-Boltzmann equation with several test molecules and to a test problem with very large jump discontinuities in the coefficients. A detailed comparison to other methods is presented, including comparisons to diagonally scaled CG, ICCG, vectorized ICCG and MICCG, and to SOR provided with an optimal relaxation parameter. For a broad range of molecule sizes and types it is shown that the multilevel methods are superior to the other methods, and this superiority grows with the problem size. We perform numerical experiments to investigate the relationship between the multilevel convergence rates and various parameters, including the number of levels, and the magnitude of the coefficient discontinuities in the interface problem. We attempt to determine empirically the overall complexity of the algorithms.[1]

6.1 Three linearized PBE methods

Of recent investigations into numerical solution of linearized PBE, the two most efficient methods appear to be the adaptive SOR procedure described by Nicholls and Honig [152], and the incomplete Cholesky preconditioned conjugate gradient method of Davis and McCammon [42]. Consequently, we will focus on these two methods for the comparisons with multilevel methods to follow. We first briefly describe what results were obtained with these methods, and then describe the multilevel method we have developed.

6.1.1 Successive Over-Relaxation

In [152], an adaptive SOR procedure is developed for the linearized Poisson-Boltzmann equation, employing a power method to estimate the largest eigenvalue of the Jacobi iteration matrix, which enables estimation of the optimal relaxation parameter for SOR using Young's formula (page 110 in [179]). The eigenvalue estimation technique employed is similar to the power method approach discussed on page 284 in [179]. In the implementation of the method in the computer program DELPHI, several additional techniques are employed to increase the efficiency of the method. In particular, a red/black ordering is employed allowing for vectorization, and array-oriented data structures (as opposed to three-dimensional grid data structures) are employed to maximize vector lengths. The implementation is also specialized to the linearized Poisson-Boltzmann equation, with constants hard-coded into the loops rather than loaded as vectors to reduce vector loads.

In our comparisons with the multilevel methods, we use an SOR method provided with the optimal relaxation parameter, implemented with a red/black ordering and array oriented data structures, yielding maximal vector lengths and, as will be apparent, very high performance on both the Convex C240 and the Cray Y-MP. We will also remark on the exceptional efficiency of the DELPHI implementation, and compare it to our implementations.

[1] The material in this chapter also appears in [106, 107].

6.1.2 (Modified) Incomplete Cholesky conjugate gradient methods

The application of conjugate gradient methods to the Poisson-Boltzmann equation is discussed by Davis and McCammon [42], including comparisons with some classical iterative methods such as SOR. The conclusions of their study were that the conjugate gradient methods were substantially more efficient than relaxation methods including SOR, and that incomplete factorizations were effective preconditioning techniques for the linearized Poisson-Boltzmann equation. We will see below that in fact for the problem sizes typically considered, the advantage of conjugate gradient methods over SOR is not so clear if an efficient SOR procedure is implemented, and if a near optimal parameter is available. Of course, if larger problem sizes are consider, then the superior complexity properties of the conjugate gradient methods (as summarized at the end of Chapter 3) will eventually yield a more efficient technique than SOR.

We will also consider below several more advanced preconditioners than considered in [42]. Among the most effective preconditioners for linear systems arising from the discretization of partial differential equations are the incomplete factorizations, as considered in [42]; unfortunately, the very implicitness which gives these preconditioners their effectiveness also makes them difficult to vectorize on vector computers. However, the incomplete Cholesky factorizations for symmetric problems on non-uniform Cartesian meshes developed by van der Vorst and others [178] employ special orderings to improve vectorization during the back substitutions.

We present experiments with a preconditioned conjugate gradient method (implemented so as to yield maximal vector lengths and high performance), provided with four different preconditioners: (1) diagonal scaling; (2) an incomplete Cholesky factorization (the method for which Davis and McCammon present results [42]); (3) the same factorization but with a *plane-diagonal-wise ordering* [178] allowing for some vectorization of the backsolves; and (4) a vectorized *modified* incomplete Cholesky factorization [178] with modification parameter $\alpha = 0.95$, which has an improved convergence rate over standard ICCG.

6.1.3 A multilevel method for the Linearized PBE

We present results for the linearized Poisson-Boltzmann equation for a single multilevel method, which was selected from several multilevel methods as the most efficient for these types of problems. We will compare several different multilevel methods for the jump discontinuity test problem in a section which follows later in the chapter.

The particular multilevel method we have chosen for the linearized Poisson-Boltzmann equation is constructed from the following components which we have discussed in previous chapters (we have also presented this method in [107]). The harmonic averaging technique as described in Chapter 3 is used to create coefficients for the coarser mesh problems, and a standard box method is used to discretize the problem on the coarse mesh using the averaged coefficients. Operator-based prolongation is also employed, using the stencil compression ideas of Chapter 3. The full expressions for the prolongation operator stencil components are given in Appendix A. We use as the restriction operator the adjoint of trilinear interpolation as described in Appendix A; since we are not using the Galerkin expressions for this method, it is not essential to take the restriction to be the adjoint of the prolongation operator, and numerical experiments indicated that the adjoint of trilinear interpolation was superior for this particular method. It will be important for experiments later in the chapter to take the restriction to be the adjoint of the prolongation.

The pre- and post-smoothing operators employed correspond to red/black Gauss-Seidel iterations, where each smoothing step consisting of ν sweeps, with each sweep consisting of one sub-sweep with the red points followed by one sub-sweep with the black points. A *variable v-cycle* [24] approach to accelerating multilevel convergence is employed, so that the number of pre- and post-smoothing sweeps changes on each level; in our implementation, the number of pre- and post-smoothing sweeps at level k is given by $\nu = 2^{J-k}$, so that one pre- and post-smoothing is performed on the finest level $k = J$, and $\nu = 2^{J-1}$ sweeps on the coarsest level $k = 1$, with the number increasing geometrically on coarser levels. The coarse problem is solved with the conjugate gradient method.

We have also performed experiments with the *linear* damping parameter as described in Chapter 4; it appears to improve the contraction properties of a v-cycle method to the same degree that the variable v-cycle approach accelerates the method, at roughly the same cost. However, using the two techniques together does not seem to improve the contraction properties further, and the cost is increased. Therefore, the method presented in this section employs only the variable v-cycle acceleration.

6.2 Some test problems

We describe the Poisson-Boltzmann problems which we use to numerically evaluate and compare the SOR, preconditioned conjugate gradient methods, and multilevel methods. We also describe a test problem which has very large jump discontinuities in the coefficients, which will be used to evaluate some of the multilevel techniques.

6.2.1 The linearized Poisson-Boltzmann equation

Consider a very broad range of temperatures $T \in [200K, 400K]$, a broad range of ionic strengths $I_s \in [0, 10]$, and the following representative polygonal domain:

$$\Omega = [\mathbf{x}_{\min} \overset{o}{\mathrm{A}}, \mathbf{x}_{\max} \overset{o}{\mathrm{A}}] \times [\mathbf{y}_{\min} \overset{o}{\mathrm{A}}, \mathbf{y}_{\max} \overset{o}{\mathrm{A}}] \times [\mathbf{z}_{\min} \overset{o}{\mathrm{A}}, \mathbf{z}_{\max} \overset{o}{\mathrm{A}}].$$

We assume that the set of discrete charges $\{\mathbf{x}_1, \ldots, \mathbf{x}_{N_m}\}$ representing the molecule lie well within the domain, and hence far from the boundary Γ of Ω. The linearized Poisson-Boltzmann equation for the dimensionless potential $u(\mathbf{x})$ then has the form:

$$-\nabla \cdot (\bar{\mathbf{a}}(\mathbf{x}) \nabla u(\mathbf{x})) + b(\mathbf{x}) u(\mathbf{x}) = f(\mathbf{x}) \text{ in } \Omega \subset \mathbb{R}^3, \qquad u(\mathbf{x}) = g(\mathbf{x}) \text{ on } \Gamma. \tag{6.1}$$

From the discussion in Chapter 1, the problem coefficients are of the following forms, and satisfy the following bounds for the given temperature and ionic strength ranges:

(1) $\bar{\mathbf{a}} : \Omega \mapsto \mathbf{L}(\mathbb{R}^3, \mathbb{R}^3),\ a_{ij}(\mathbf{x}) = \delta_{ij} \epsilon(\mathbf{x}),\ 2 \leq \epsilon(\mathbf{x}) \leq 80,\ \forall \mathbf{x} \in \Omega.$

(2) $b : \Omega \mapsto \mathbb{R},\ b(\mathbf{x}) = \bar{\kappa}^2(\mathbf{x}),\ 0 \leq \bar{\kappa}^2(\mathbf{x}) \leq 127.0,\ \forall \mathbf{x} \in \Omega.$

(3) $f : \Omega \mapsto \mathbb{R},\ f(\mathbf{x}) = C \cdot \sum_{i=1}^{N_m} z_i \delta(\mathbf{x} - \mathbf{x}_i),\ 5249.0 \leq C \leq 10500.0,\ -1 \leq z_i \leq 1,\ \forall \mathbf{x} \in \Omega.$

(4) $g : \Gamma \mapsto \mathbb{R},\ g(\mathbf{x}) = [C/(4\pi\epsilon_w)] \cdot \sum_{i=1}^{N_m} [z_i e^{-\bar{\kappa}(\mathbf{x})|\mathbf{x}-\mathbf{x}_i|/\sqrt{\epsilon_w}}]/|\mathbf{x} - \mathbf{x}_i|,\ \epsilon_w = 80,\ \forall \mathbf{x} \in \Gamma.$

The linearized Poisson-Boltzmann problem will then be completely defined by specifying the following quantities:

- $\mathbf{x}_{\min}, \mathbf{x}_{\max}, \mathbf{y}_{\min}, \mathbf{y}_{\max}, \mathbf{z}_{\min}, \mathbf{z}_{\max}$; the domain geometry.
- $\epsilon(\mathbf{x})$; the electrostatic surface of the molecule.
- $\bar{\kappa}(\mathbf{x})$; defined by the ionic strength I_s and the exclusion layer around the molecule.
- C; a constant which depends only on the temperature T.
- $\{\mathbf{x}_1, \ldots, \mathbf{x}_{N_m}\}$; charge locations, and associated fractional charges $\{z_1, \ldots, z_{N_m}\}$.

For all of our molecule test problems, we use $T = 298$ which determines the constant C; this is a common parameter setting for these types of problems. The domain geometry will be defined by the particular molecule, as well as the parameters $\epsilon(\mathbf{x})$ and $\bar{\kappa}(\mathbf{x})$, although we must specify also the ionic strength I_s to completely determine $\bar{\kappa}(\mathbf{x})$. The charge locations and corresponding fractional charges will also be determined by the particular molecule.

6.2.2 The Brookhaven Protein Databank and existing biophysics software

We have connected the software implementations of our methods to both the DELPHI and UHBD electrostatics programs, and we will use data provided by these packages. These codes are designed to begin with a protein data bank (pdb) file description of the protein or enzyme in question, obtained from the protein data bank at Brookhaven National Laboratory. The pdb files contain the coordinates of all of the atoms in a particular structure, obtained from X-ray crystallography pictures of the structure. The UHBD and DELPHI programs begin with the atom coordinates, and then construct both the electrostatic surface and the exclusion layer by moving a probe around the molecule which has the radius of a representative ion. We remark that quite sophisticated algorithms are now being employed for surfacing [151].

Both UHBD and DELPHI are designed around Cartesian meshes (both implementations are actually restricted to uniform Cartesian meshes), and the electrostatic surface and exclusion layer information are represented as three-dimensional discrete grid functions $\epsilon_h(\mathbf{x})$ and $\bar{\kappa}_h(\mathbf{x})$. The mesh function $\bar{\kappa}_h(\mathbf{x})$ is produced at the same mesh-points where the unknowns $u_h(\mathbf{x})$ are located, whereas the mesh function $\epsilon_h(\mathbf{x})$ is produced at half-mesh-points in each coordinate direction as needed for a box-method discretization (employed in both UHBD and DELPHI). The atoms themselves, which will most likely not lie on a uniform Cartesian mesh, must be mapped to the uniform Cartesian coordinates, and their corresponding charges distributed to the neighboring mesh points. Several approaches are possible; a trilinear interpolation approach is taken in both packages.

Note that the selection of the domain completely determines the boundary conditions for a given problem, as we have specified the boundary function $g(\mathbf{x})$ above. Several different approaches have been proposed to approximate $g(\mathbf{x})$, since it is clear that to evaluate $g(\mathbf{x})$ at each boundary point of the three-dimensional domain will require all pair-wise interactions of the charges and the boundary points; efficient versions are offered as options for example in UHBD, all of which appear to give similarly good approximations of the true boundary condition $u(\infty) = 0$ (when the molecule is taken to lie well within the domain Ω). In both UHBD and DELPHI, the problem domain Ω is constructed around the selected molecule so that no more than thirty percent of Ω in each coordinate direction is taken up by the molecule, which is centered in the domain. This appears to give very good approximation of the true boundary conditions in most situations.

The elliptic solver component of our software, which has been connected to the biophysics software, is designed to solve general problems of the form (6.1) on arbitrary logically non-uniform Cartesian, three-dimensional meshes; the nonlinear capabilities of the software are discussed in Chapter 7 and Appendix B. The available methods in the solver include the multilevel and conjugate gradient methods discussed in Chapter 3, plus a few more. The domain geometry is specified by the user with the coordinates $\mathbf{x}_{\min}$, $\mathbf{x}_{\max}$, $\mathbf{y}_{\min}$, $\mathbf{y}_{\max}$, $\mathbf{z}_{\min}$, and $\mathbf{z}_{\max}$, and the non-uniform Cartesian mesh which tessellates the domain is also specified by the user, allowing for arbitrary mesh spacings in any direction. There is only the restriction that the mesh be *logically* non-uniform Cartesian, or *axi-parallel*, so that the algebraic Galerkin multilevel methods (described in Chapter 3) are well-defined; these methods work only with matrices representable as stencils. The problem coefficients $b(\mathbf{x})$ and $f(\mathbf{x})$, which are allowed to be wildly discontinuous at arbitrary points, are specified in arrays at the non-uniform Cartesian mesh points coinciding with the mesh provided by the user, whereas the coefficient $\bar{\mathbf{a}}(\mathbf{x})$, also allowed to have extremely large jump discontinuities at arbitrary points, is provided at half-mesh-points as required for a box method discretization. The boundary coefficient $g(\mathbf{x})$ is specified by the user in arrays which match the dimensions of the six surrounding discretized faces of the domain Ω. The parameter settings for the various methods are provided by the user in a single array of flags.

A more complete description of the software may be found in Appendix B.

6.2.3 A collection of molecule test problems

We will focus on three test molecules, at varying ionic strengths, which represent a wide range of difficulty and size. The first two data sets were obtained from DELPHI, and the third from UHBD.

- Acetamide (CH_3CONH_2) at 0.1 molar, a small molecule (few angstroms in diameter).
- Lysozyme at 0.1 molar, often used as a test problem for the linearized PBE.
- SOD at 0.1 molar, a problem we study in Chapter 7 as well.

6.2.4 A test problem with large jump discontinuities

The following test problem, which is essentially a three-dimensional version of the two-dimensional test problem appearing on page 42 in [26], will be used to explore the convergence behavior of the methods as a function of the difficulty of the problem, represented by the problem size as well as the magnitudes of the jump discontinuities in the coefficients. The domain is the unit cube:

$$\Omega = [0,1] \times [0,1] \times [0,1].$$

The linear equation has the form:

$$-\nabla \cdot (\bar{\mathbf{a}}(\mathbf{x}) \nabla u(\mathbf{x})) + b(\mathbf{x}) u(\mathbf{x}) = f(\mathbf{x}) \text{ in } \Omega \subset \mathbb{R}^3, \qquad u(\mathbf{x}) = g(\mathbf{x}) \text{ on } \Gamma. \tag{6.2}$$

where the coefficients in equation (6.2) are taken to be:

(1) $\bar{\mathbf{a}} : \Omega \mapsto \mathbf{L}(\mathbb{R}^3, \mathbb{R}^3),\ a_{ij}(\mathbf{x}) = \delta_{ij}\epsilon(\mathbf{x}),\ 1 \leq \epsilon(\mathbf{x}) \leq 1.0 \times 10^8,\ \forall \mathbf{x} \in \Omega.$

Table 6.1: Linearized Poisson-Boltzmann equation methods.

Method	Description
MH	Multilevel (Harmonic ave., op-prolongation, red/black GS, var. v-cycle)
OSOR	successive over-relaxation method with optimal relaxation parameter ω
DSCG	diagonally scaled conjugate gradient method
ICCG1	incomplete Cholesky preconditioned conjugate gradient method
ICCG2	same as ICCG1 but with plane-diagonal-wise ordering during backsolves
MICCG	same as ICCG2 but with a modification parameter $\alpha = 0.95$

(2) $b : \Omega \mapsto \mathbb{R},\ b(\mathbf{x}) = 0,\ \forall \mathbf{x} \in \Omega.$
(3) $f : \Omega \mapsto \mathbb{R},\ -1 \le f(\mathbf{x}) \le 1,\ \forall \mathbf{x} \in \Omega.$
(4) $g : \Gamma \mapsto \mathbb{R},\ g(\mathbf{x}) = 0,\ \forall \mathbf{x} \in \Gamma.$

We will construct $\epsilon(\mathbf{x})$ to be piecewise constant, taking one value in a subdomain $\Omega_1 \subset \Omega$, and a second value in the region $\Omega \backslash \Omega_1$, so that $\epsilon(\mathbf{x})$ is defined as follows:

$$\epsilon(\mathbf{x}) = \left\{ \begin{array}{l} 1 \le \epsilon_1 \le 1.0 \times 10^8 \text{ if } \mathbf{x} \in \Omega_1, \\ 1 \le \epsilon_2 \le 1.0 \times 10^8 \text{ if } \mathbf{x} \in \Omega \backslash \Omega_1. \end{array} \right\}$$

We will let ϵ_1 and ϵ_2 vary for different test runs so that their ratio:

$$D = \frac{\epsilon_1}{\epsilon_2}$$

can be as large as 10^8 or as small as 10^{-8} for a particular run, and we will then observe the resulting convergence behavior of the multilevel methods. We define the subdomain $\Omega_1 \subset \Omega$ to consist of the following two smaller cubes:

$$\Omega_1 = [0.25, 0.50] \times [0.25, 0.50] \times [0.25, 0.50] \quad \bigcup \quad [0.50, 0.75] \times [0.50, 0.50] \times [0.50, 0.75].$$

For this simple problem, it would of course be possible to construct all coarse meshes as needed for the multilevel methods to align with Ω_1; this would not be possible with problems such as the linearized Poisson-Boltzmann equation and a complex molecule. Therefore, since we wish to simulate the case that the discontinuities in $\epsilon(\mathbf{x})$ cannot be resolved on coarser meshes, the multiple levels of tessellations of Ω into discrete meshes Ω_k are constructed so that the discontinuities in $\epsilon(\mathbf{x})$ lie along mesh lines *only on the finest mesh.*

Note that if $\epsilon_1 = \epsilon_2 \equiv 1$, then problem (6.2) with the above coefficients is Poisson's equation on the unit cube.

6.3 Numerical results for the Linearized PBE

Table 6.1 provides a key to the plots and tables to follow.

Unless otherwise indicated, all data in the plots and tables to follow *include* the pre-processing costs incurred by the various methods. In other words, the multilevel method times include the additional time required to set up the problem on coarse grids, and the times for the conjugate gradient methods employing incomplete factorizations include the initial costs of performing the factorizations. This gives a complete and fair assessment of the total time required to reach the solution.

An initial approximation of zero was taken to start each method, and each method used a relative residual stopping criterion:

$$\frac{\|r_k^n\|}{\|f_k\|} = \frac{\|f_k - A_k u_k^n\|}{\|f_k\|} < \text{ TOL } = 1.0e-6,$$

where u_k^n represents the n^{th} iterate. Normally, $\|r_k^n\|$ is not available in the preconditioned conjugate gradient iteration (the quantity $< Cr_k^n, r_k^n >^{1/2}$ is available, where C is the preconditioner), and must be computed

Table 6.2: Megaflops with [without] matrix construction and factorization setup.

Machine	Method					
(1 Processor)	MH	MICCG	OSOR	ICCG2	DSCG	ICCG1
Convex C240	12.3 [12.7]	13.1 [13.6]	18.9 [20.4]	13.6 [14.1]	18.1 [18.5]	7.24 [7.05]
Cray Y-MP	118 [120]	135 [158]	215 [220]	142 [158]	215 [218]	36.8 [35.5]

Table 6.3: Total time to reach TOL=1.0e-6 with $65 \times 65 \times 65$ grid (CPU seconds).

Machine	Method					
(1 Processor)	MH	MICCG	OSOR	ICCG2	DSCG	ICCG1
Convex C240	13.2	23.1	35.9	35.7	56.8	69.9
Cray Y-MP	1.40	2.45	3.22	3.62	4.76	13.8

at extra cost; however, this additional cost was not included in the conjugate gradient method timings in order to avoid unfairly penalizing the conjugate gradient methods.

Timings, operation counts, and megaflops (one million floating point operations per second) figures on the Cray Y-MP were obtained from the performance monitoring hardware accessed through *perftrace* and *perfview*. Timing figures on the Convex C240 were obtained from the system timing routine `getrusage`, and megaflop rates were computed from the exact operation counts provided earlier by the Cray.

Table 6.2 gives the *performance in megaflops* for each method when applied to any of the test molecules, where we have listed the performance statistics with and without pre-processing costs such as matrix construction and Cholesky factorizations. A more detailed performance analysis on several more sequential as well as some parallel machines can be found in Chapter 8.

6.3.1 Results for acetamide

Figure 6.1 gives the reduction in the relative residual per CPU second for each method on the Convex C240. Figure 6.2 gives the corresponding information on the Cray Y-MP. In Table 6.3, the information from Figures 6.1 and 6.2 is translated into a single number for each method, representing the *total time required to reach the acetamide solution* on a given architecture.

These graphs and tables show that multilevel method is nearly two times faster than the next best method, MICCG. It is interesting to note from Table 6.3 that optimal SOR is in fact equal or superior to all of the conjugate gradient methods for this problem, except for MICCG. Table 6.2 indicates that our implementation of the optimal SOR method is exceptionally efficient, operating at near the peak rate available from FORTRAN of matrix-vector operations on the Cray Y-MP. In addition, the vectorized incomplete Cholesky preconditioned conjugate gradient methods execute with very high rates, consistent with the earlier reports [178] for these methods on the Cray X-MP.

We remark that while we have presented the results above for a $65 \times 65 \times 65$ mesh, the multilevel method becomes more and more efficient compared to the other methods as the problem size is increased; we will demonstrate this later in the chapter.

6.3.2 Results for lysozyme

Figure 6.3 gives the reduction in the relative residual per time work unit for each method on the Convex C240, when the molecule is taken to be the larger lysozyme molecule.

This graph shows that the multilevel method is even more efficient (approximately three times faster) than the next best method for this more complicated problem. The conjecture here would be that multilevel methods are more effectively at moving global information around, and the less homogeneous the problem is, the more advantage a multilevel method will have. Again, the separation between the multilevel method and the other methods increases as the problem size is increased.

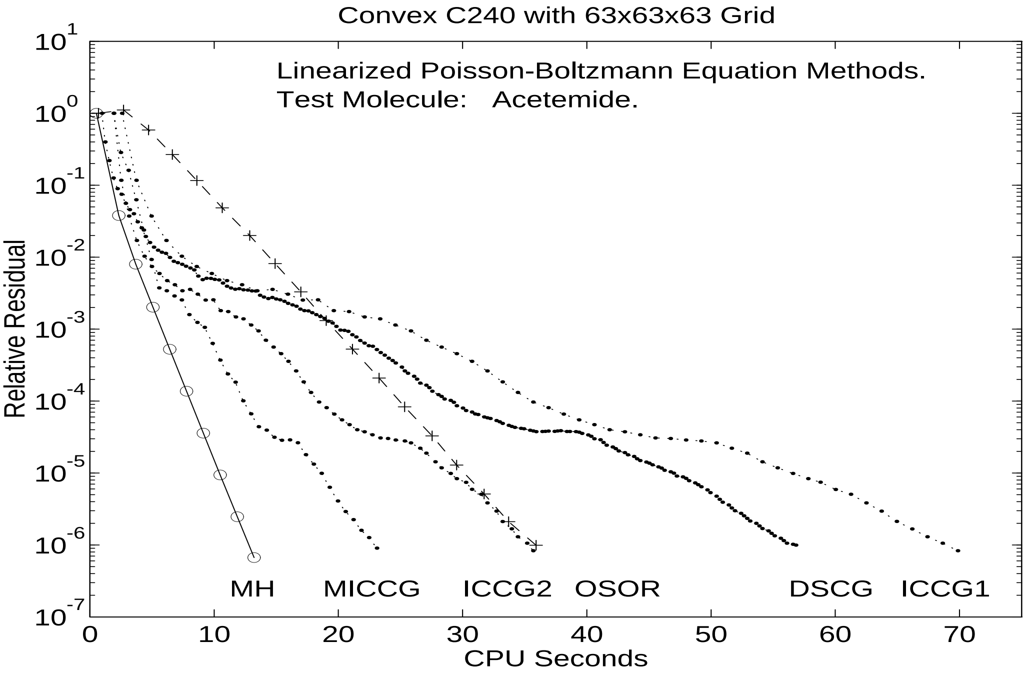


Figure 6.1: Comparison of various methods for the linear acetamide problem.

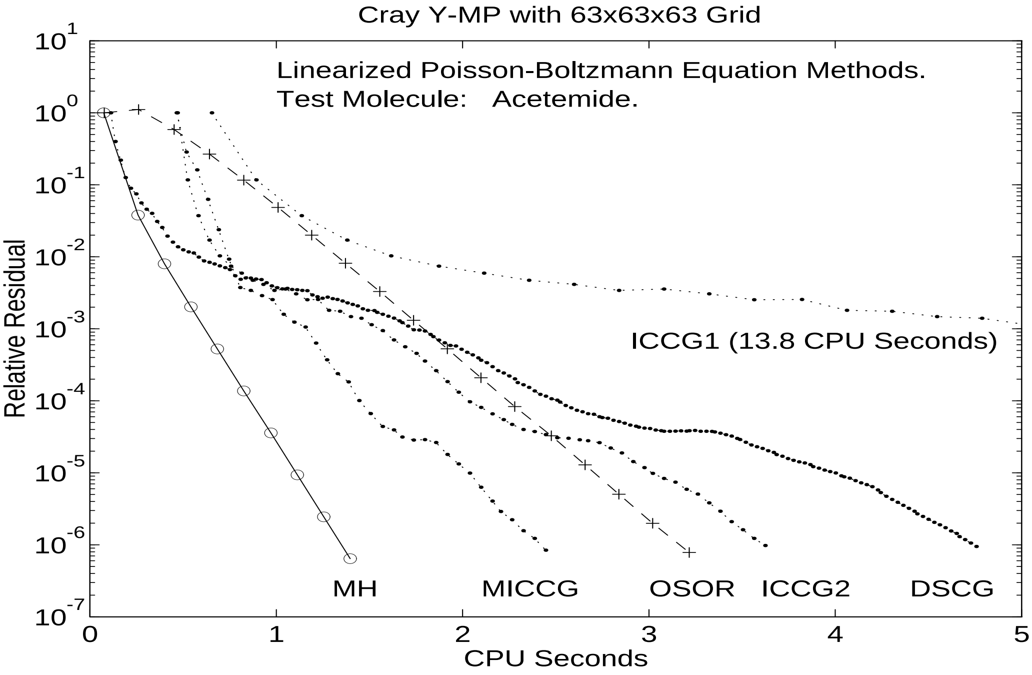


Figure 6.2: Comparison of various methods for the linear acetamide problem.

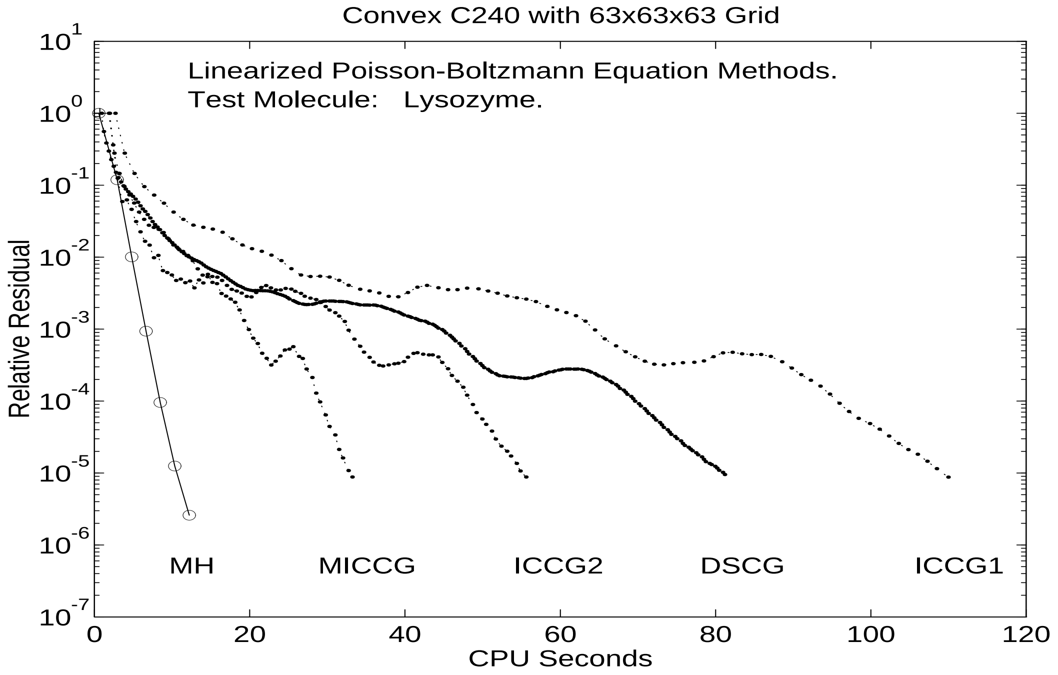


Figure 6.3: Comparison of various methods for the linear lysozyme problem.

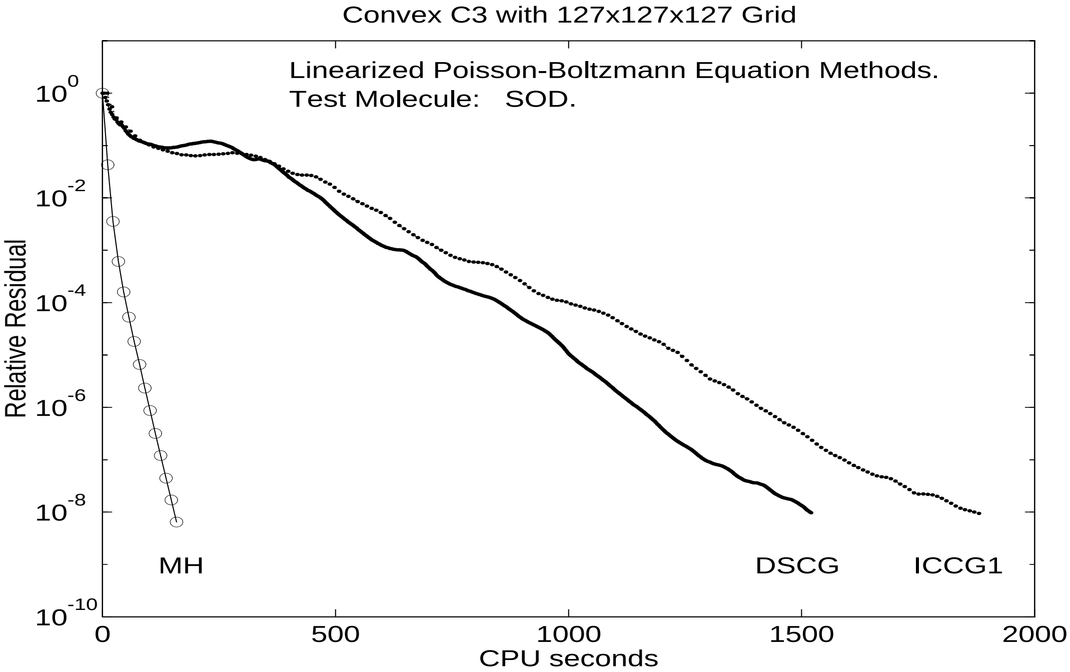


Figure 6.4: Comparison of various methods for the linear SOD problem.

6.3.3 Results for SOD

Figure 6.4 gives the reduction in the relative residual per time work unit for each method on the Convex C3, when the molecule is taken to be the large SOD molecule. We remark that only three methods appear on the plot because only these three linear methods have been connected to the UHBD software at this time. We have used a $127 \times 127 \times 127$ mesh, and for this larger problem size, the advantage of the multilevel solver becomes more clear. The multilevel method appears to be about fifteen more efficient than the diagonally scaled conjugate gradient method, compared to a factor of three to five for the $65 \times 65 \times 65$ mesh in the case of acetamide and lysozyme.

Remark 6.1. Our results are consistent with an earlier study by Dendy and Hyman [53], who compared multigrid methods to non-vectorized forms of ICCG and MICCG for two-dimensional interface problems, including the multi-group neutron diffusion problem. Their conclusions for the two-dimensional case were that the multigrid method developed by Alcouffe et al. [2] was superior to both ICCG and MICCG. One observation they made, which we did not take advantage of in this study, was the following: *multigrid reaches discretization error accuracy very rapidly* – with far fewer iterations than that required to reach a small residual tolerance. One cannot usually make this statement for other methods.

While the implementations presented here may also be used for general second order problems in three dimensions, in the case of the Poisson-Boltzmann equation special techniques may be used to increase solver efficiency. In particular, the highly optimized SOR method of Nicholls and Honig [152], using a technique referred to as *stripping*, along with a novel procedure for determining the optimal relaxation parameter adaptively, achieves a factor of two improvement over our "unstripped" optimal SOR method (35.9 CPU seconds, from Table 6.3 for the acetamide problem). Their code solves the acetamide problem on the Convex C240 in 17.3 CPU seconds, compared to 13.2 CPU seconds for our implementation of the multilevel method.

It should be stressed that their optimization techniques may be used to equal advantage with the multilevel method presented here, as it is based on a red/black Gauss-Seidel smoothing iteration; therefore, we would expect a similar (factor of two) improvement in the efficiency of the multilevel method. However, it is unclear how to take advantage of their *stripping* technique in the preconditioning phases of the incomplete Cholesky conjugate gradients methods, which in our experiments made up more than sixty percent of the total execution times of these methods (more than eighty-five percent in the non-vectorized ICCG case).

A final remark is that for higher ionic strengths, which results in a larger "Helmholtz-like" term $\bar{\kappa}(\mathbf{r})$ in the linearized PBE, the resulting discrete linear systems are better conditioned, and the preconditioned conjugate gradient methods in particular do appear to benefit in these situations by requiring fewer iterations.

6.4 Multilevel behavior for the jump discontinuity problem

The pre-smoothing operator employed here is a standard point Gauss-Seidel iteration, where the pre-smoothing step consists of ν sweeps. Note that for the multilevel methods not employing the Galerkin coarse problem formulation, the matrices on each level have seven point stencils, allowing for a red/black coloring of the unknowns and the use of efficient vectorizable red/black Gauss-Seidel pre- and post-smoothing. This is also the case on the finest mesh for the Galerkin methods, so that a vectorizable red-black smoothing may be employed on at least the finest mesh. On coarser meshes, either point Gauss-Seidel or vectorizable weighted Jacobi smoothing may be used for the Galerkin methods. The post-smoothing step we employ here is as the pre-smoothing step, where the order of the sweeps is reversed so that the resulting post-smoothing operator is the adjoint of the pre-smoothing operator, to yield a symmetric multilevel operator as discussed in detail in Chapter 3. A *variable v-cycle* [24] is employed, so that the number of pre- and post-smoothing sweeps changes on each level; in our implementation, the number of pre- and post-smoothing sweeps at level k is given by $\nu = 2^{J-k}$, so that one pre- and post-smoothing is performed on the finest level, with the number increasing geometrically on coarser levels. A linear damping parameter was employed in each of the multilevel methods to improve their convergence behavior, exactly as described in Chapter 4. The coarse problem is solved with banded LINPACK.

Table 6.4 provides a key to the plots and tables to follow.

Table 6.4: Three-dimensional elliptic equation methods.

Method	Description
MV	vanilla multigrid (linear-prolongation, red/black GS)
MH	multilevel (harmonic ave., operator-prolongation, red/black GS)
MG	multilevel (Galerkin expressions, linear-prolongation, red/black GS)
MVCG	MV accelerated with the conjugate gradient method
MHCG	MH accelerated with the conjugate gradient method
MGCG	MG accelerated with the conjugate gradient method
CG	vanilla conjugate gradient method
MICCG	incomplete Cholesky PCG, plane-diagonal-wise ordering, $\alpha = 0.95$

6.4.1 Convergence vs. the discontinuity size and the number of levels

We now present a series of plots giving the convergence behavior of the methods in Table 6.4 as the discontinuity ratio

$$D = \frac{\epsilon_1}{\epsilon_2}$$

is varied from $D = 1.0 \times 10^{-8}$ to $D = 1.0 \times 10^{8}$. Recall that ϵ_1 represents the value of the constant inside the inner domain $\Omega_1 \subset \Omega$, and ϵ_2 represents the constant outside the inner domain, in the region $\Omega \backslash \Omega_1$. We begin with just Poisson's equation ($D = 1$) in Figure 6.5.

Figure 6.6 through Figure 6.11 shows the behavior of each method in Table 6.4 as the discontinuity ratio is taken from $D = 1.0e - 1$ to $D = 1.0e - 8$; this is the same type of discontinuity appearing in the PBE, with the interior dielectric smaller than the exterior dielectric. For this type of discontinuity, the methods based on coefficient averaging are quite effective, and are the most efficient methods for these types of problems.

Figure 6.12 through Figure 6.17 shows the behavior of each method in Table 6.4 as the discontinuity ratio is taken from $D = 1.0e + 1$ to $D = 1.0e + 8$. For this type of discontinuity, the methods based on coefficient averaging are only slightly more effective than the standard multigrid approach; they eventually break down as the discontinuity becomes large. Note that the methods based on enforcing the variational conditions, MG and MGCG, remain effective for all ranges of discontinuity, and in fact the convergence behavior appears to be almost independent of the discontinuity.

We now present some experiments which investigate the convergence behavior of the method MG in a little more detail. The method MG employed is exactly as described earlier, except that a variable V-cycle is not used; a single pre- and post-smoothing iteration is employed on all levels. In addition, we have not employed the linear damping parameter here for improving the convergence rate of the method. As a result, by constructing the method MG to employ the adjoint of the pre-smoothing operator as the post-smoothing operator, and by imposing the variational conditions exactly, it is clear that the framework constructed in Chapter 3 applies to the resulting method. In particular, by Theorem 3.7 of Chapter 3, we know that

$$\|E^s\|_A = \|I - BA\|_A = \rho(I - BA),$$

where $E^s = I - BA$ is the symmetric multilevel error propagator for this method. Therefore, while it may be difficult to compute the quantity $\|E^s\|_A$ numerically, which is the quantity bounded in the theory described in Chapter 5, we can easily compute the largest eigenvalue with the standard power method, which is equivalent. Further, for this method we have that $\|E^s\|_A < 1$ for any SPD operator A defining the problem, which holds by Theorem 5.4 of Chapter 5.

Tables 6.5 and 6.6 contain the numerically computed spectral radii of the multilevel error propagator for the MG method described above, applied to the jump discontinuity problem; the smoothing operator employed is Gauss-Seidel, again with the adjoint of the pre-smoothing operator taken as the post-smoothing operator. This set of experiments was modeled after §7, Table 7.2, in [26]. It was our intention to attempt to determine numerically, as in [26], exactly what type of contraction number deterioration occurs as the number of levels is increased, and as the discontinuity becomes more severe.

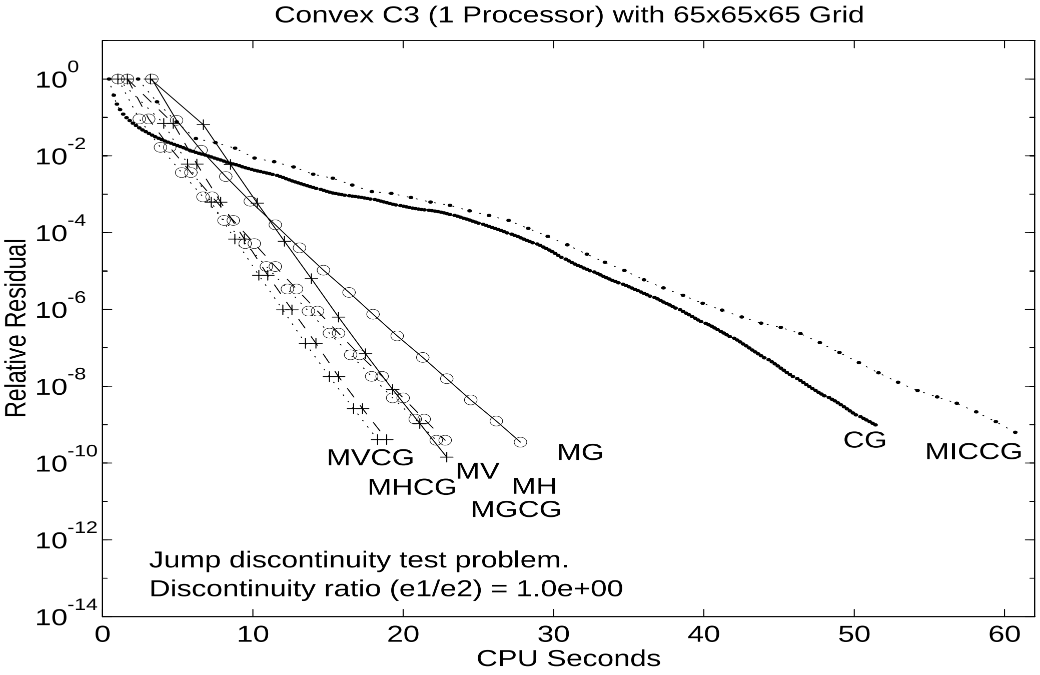


Figure 6.5: Behavior for Poisson's equation.

Table 6.5: Spectral radii behavior for the Gauss-Seidel-based MG method.

$J(n_J^{1/3})$	$D = 1$	$D = 10^1$	$D = 10^2$	$D = 10^3$	$D = 10^4$	$D = 10^8$	$1 - .52/J^{.75}$
2(17)	.30	.43	.62	.68	.69	.69	.69
3(33)	.32	.55	.72	.76	.78	.79	.77
4(65)	.33	.57	.76	.79	.80	.83	.82
5(129)	.34	.59	.78	.83	.83	.84	.84

The last column of each table shows the values of a function which has been fitted to the data, for $D = 10^8$ and $D = 10^{-8}$. The motivation for this form of the fitted function was the BPWX-based theory discussed in Chapter 5. In particular, in the special case that a finite element discretization is employed, with the coefficient discontinuity lying along element boundaries on all coarse meshes, then the BPWX Theory discussed in Chapter 5 would yield a contraction bound decaying as

$$\|E^s\|_A \leq \delta_J = 1 - \frac{C}{J^\nu},$$

with $\nu = 1$ with simple discontinuities, or with $\nu = 2$ in the presence of cross points. In our particular case of the completely algebraic method MG, without any further structure such as discontinuities lying along element boundaries on all coarse meshes, the method still demonstrates contraction numbers which decay only with $\nu = 0.75$ and $\nu = 0.39$. This would seem to indicate that it might be possible to show a similar bound on the contraction number for the completely algebraic method MG.

Tables 6.7 and 6.8 contain the same set of experiments, the computed spectral radii of the multilevel error propagator for method MG, but now the smoothing operator is taken to be weighted Jacobi, with the weight $\omega = 0.8$. Again, the last column of each table shows the values of a function which has been fitted to the data, for $D = 10^8$ and $D = 10^{-8}$. In this case, while the contraction numbers decay worse than

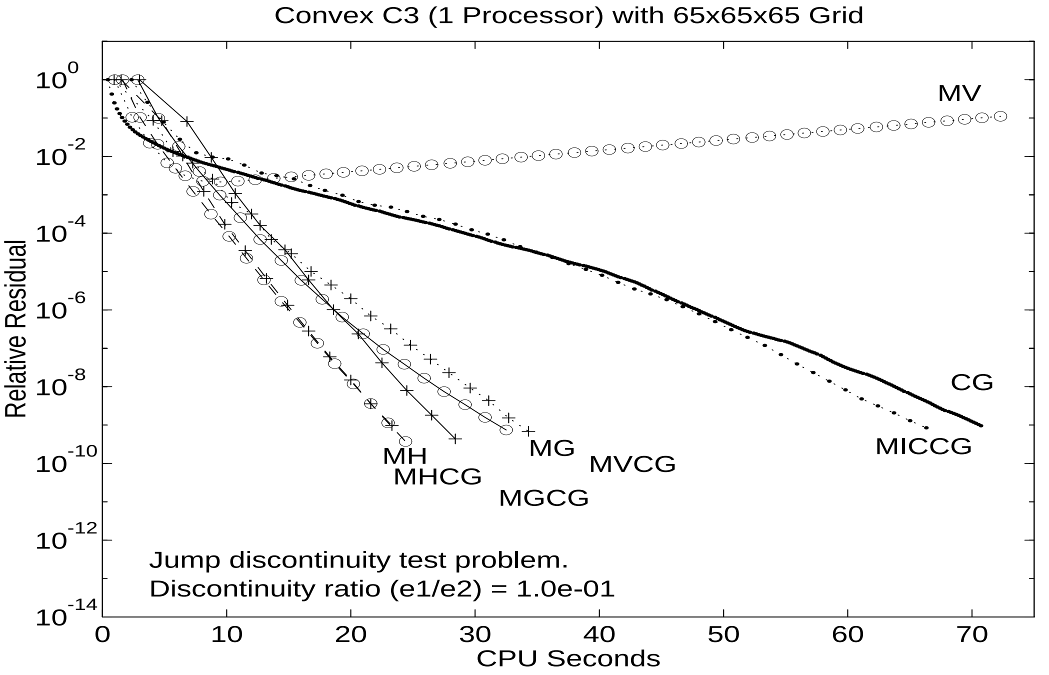


Figure 6.6: The jump discontinuity problem with D=1.0e-1.

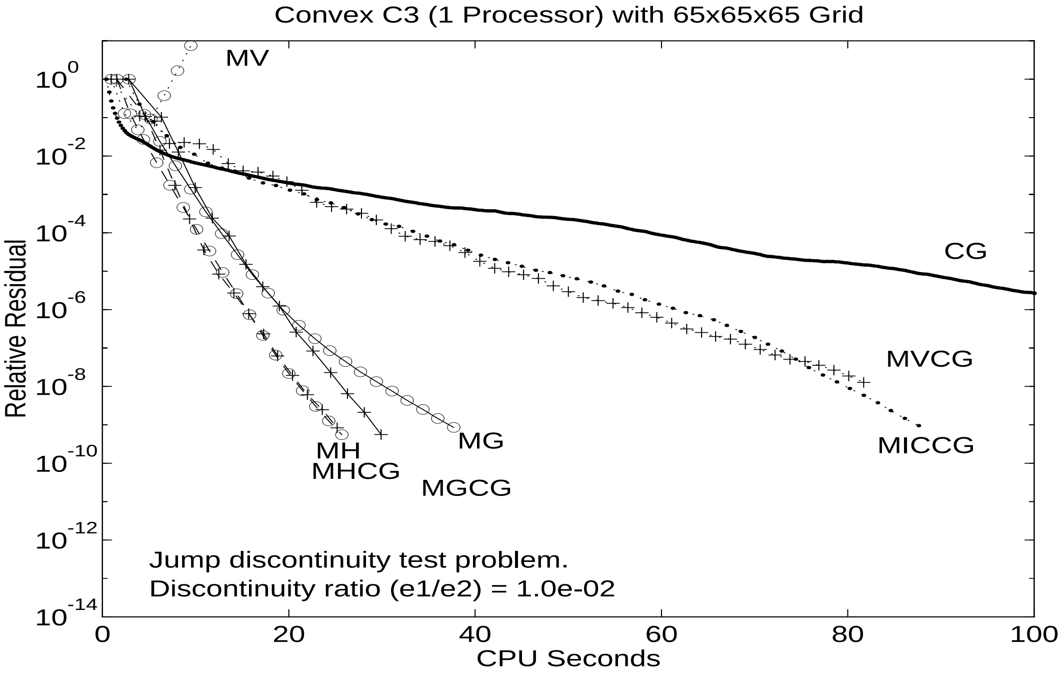


Figure 6.7: The jump discontinuity problem with D=1.0e-2.

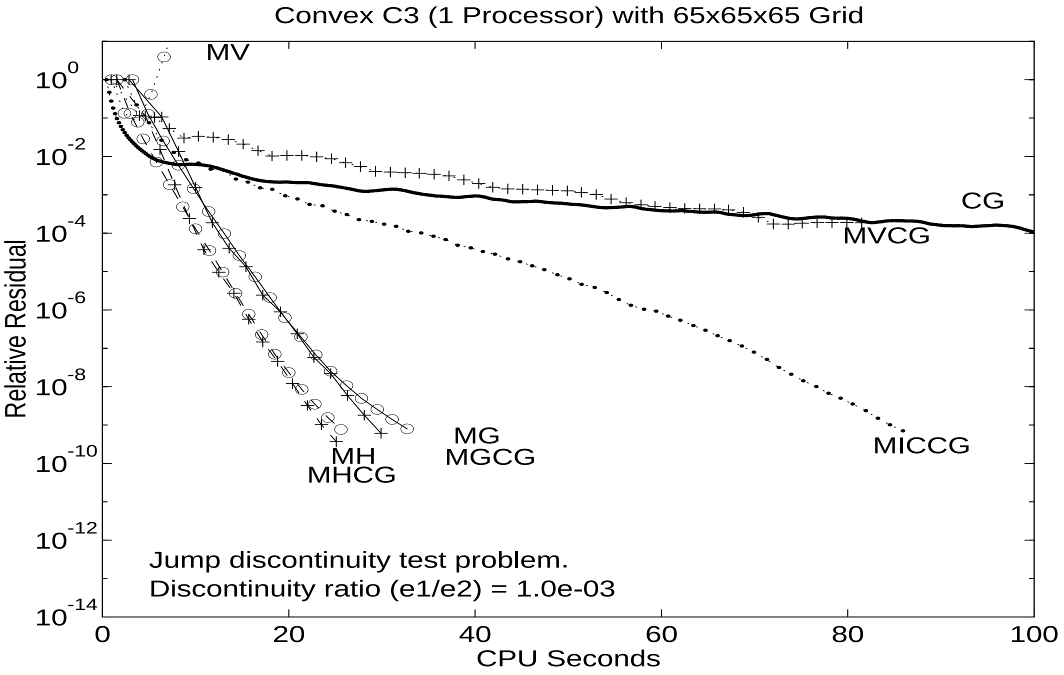


Figure 6.8: The jump discontinuity problem with D=1.0e-3.

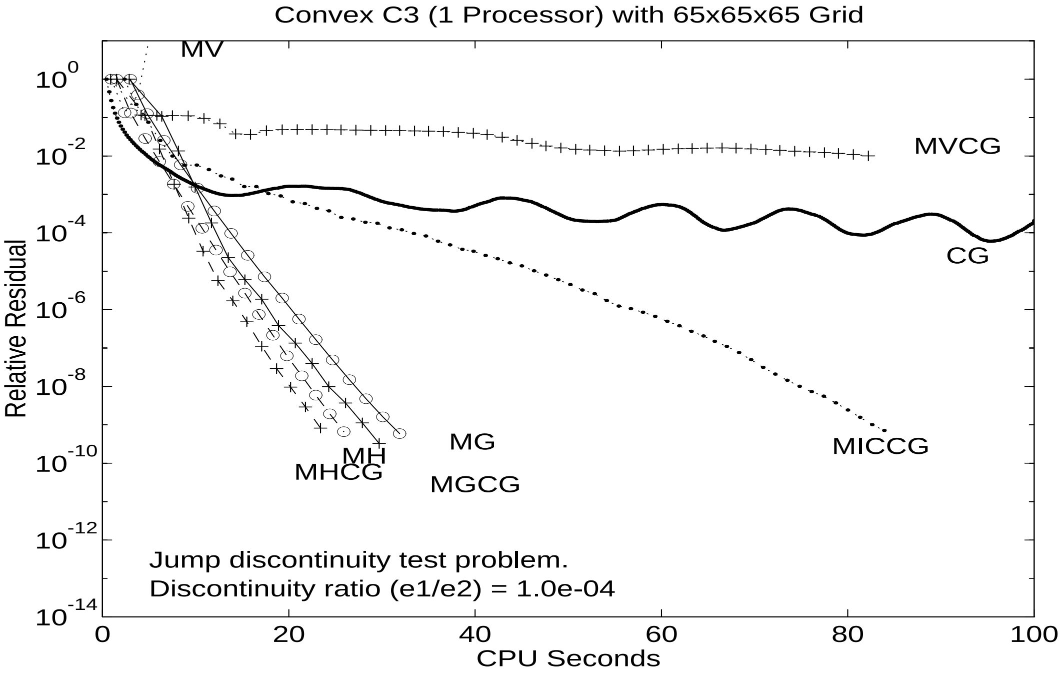


Figure 6.9: The jump discontinuity problem with D=1.0e-4.

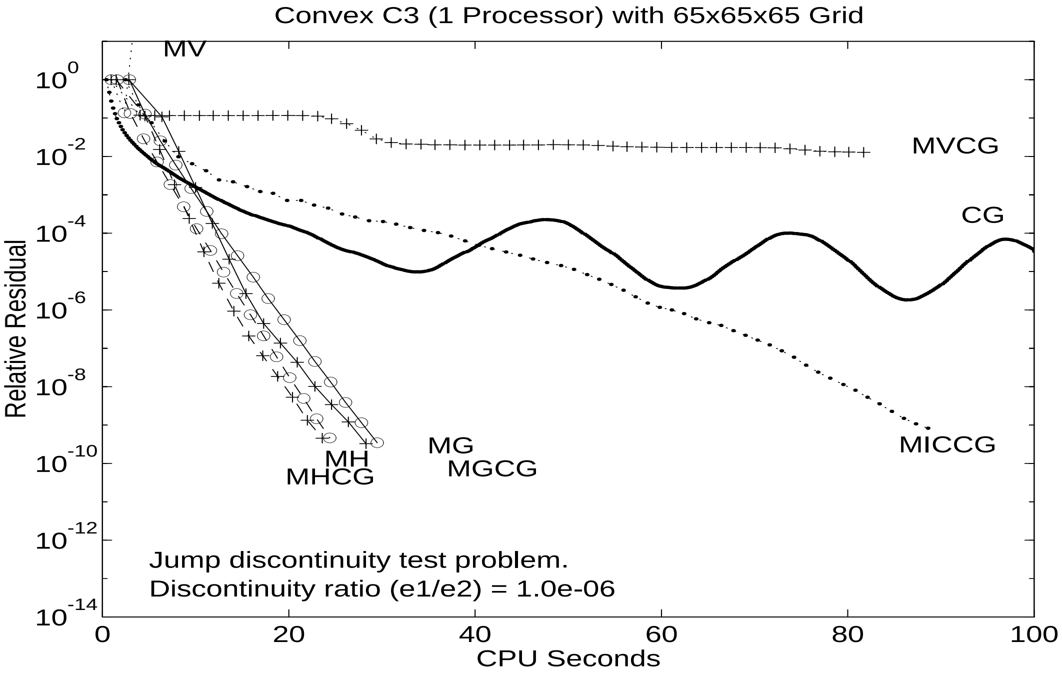


Figure 6.10: The jump discontinuity problem with D=1.0e-6.

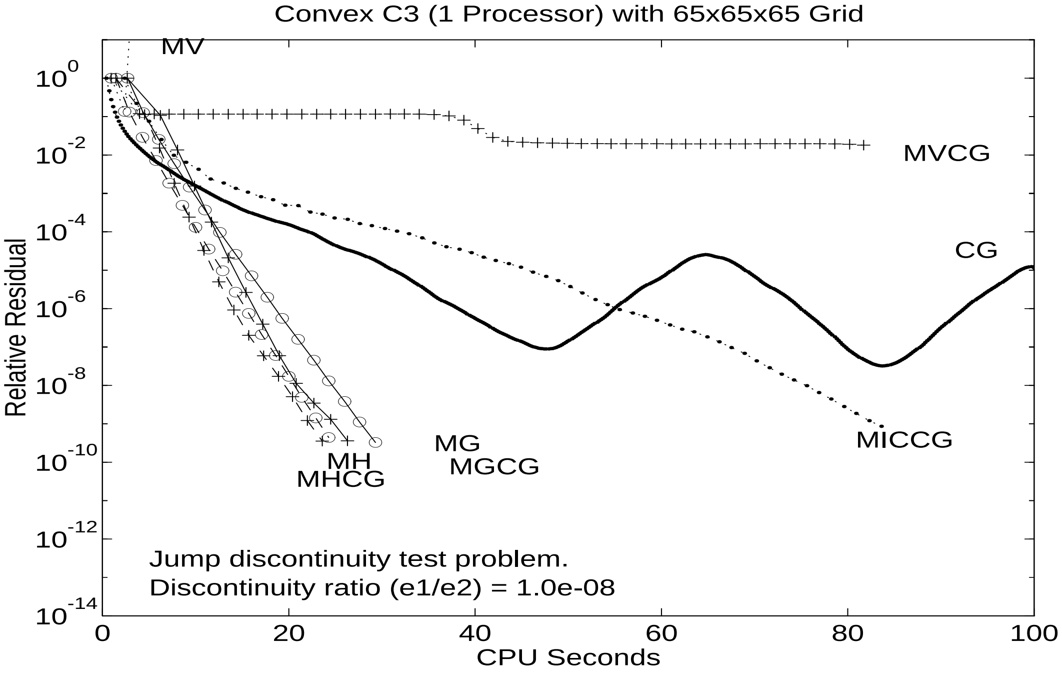


Figure 6.11: The jump discontinuity problem with D=1.0e-8.

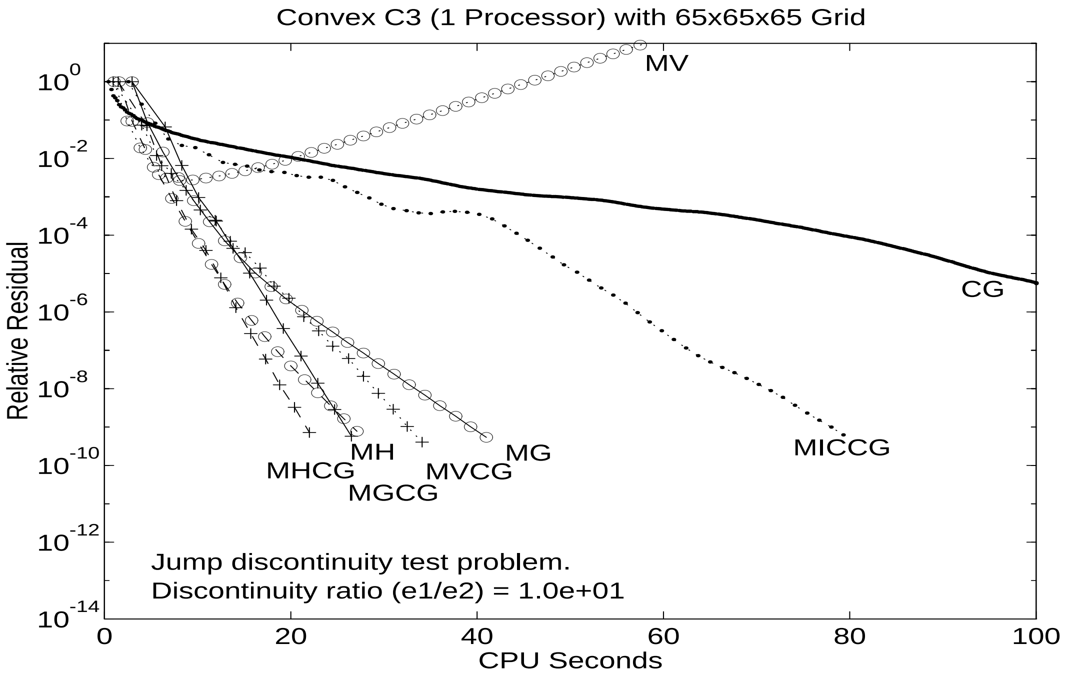


Figure 6.12: The jump discontinuity problem with D=1.0e+1.

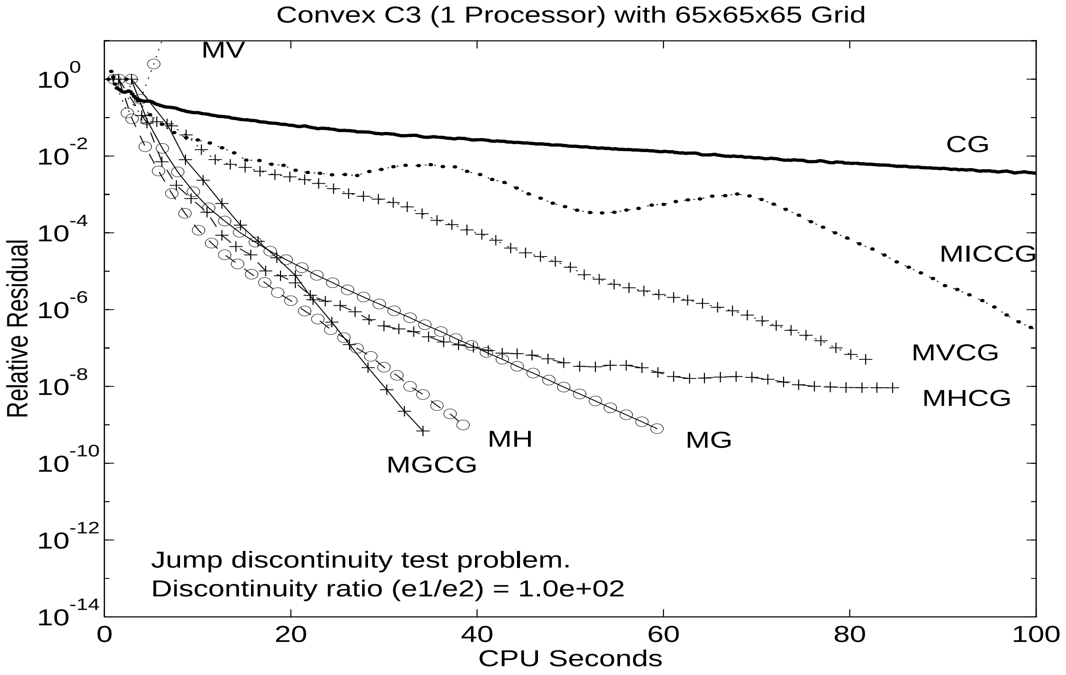


Figure 6.13: The jump discontinuity problem with D=1.0e+2.

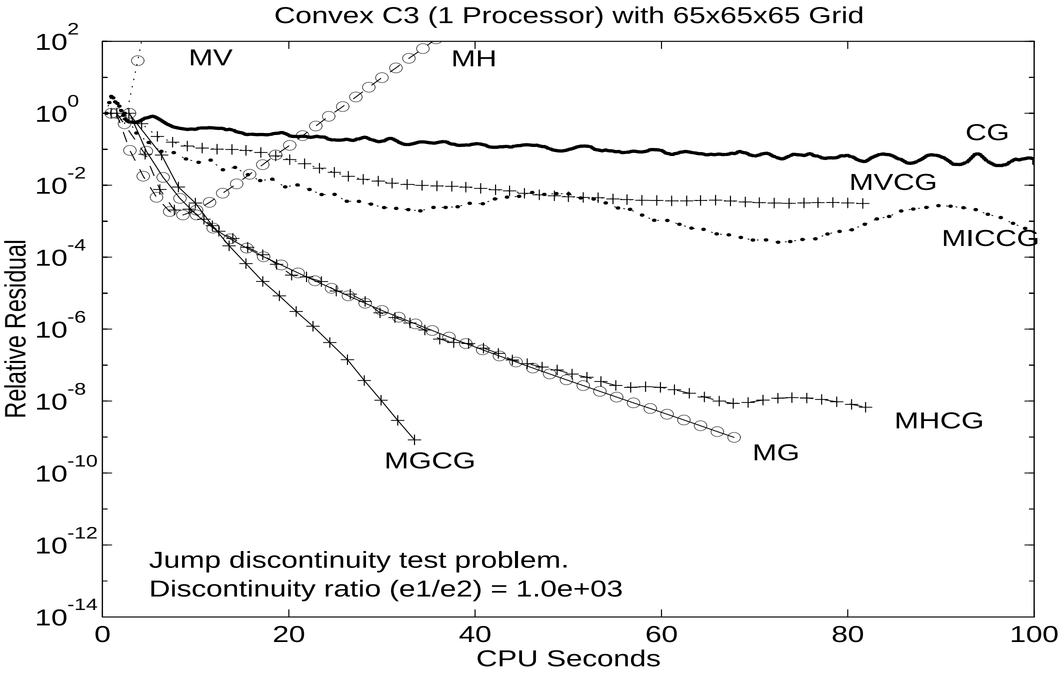


Figure 6.14: The jump discontinuity problem with D=1.0e+3.

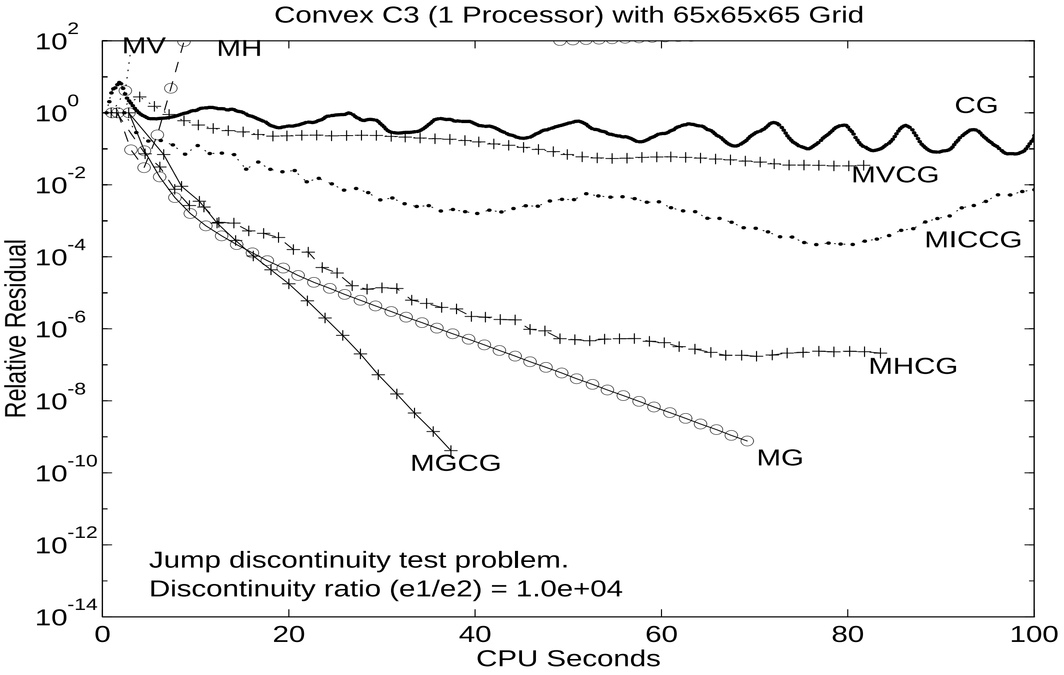


Figure 6.15: The jump discontinuity problem with D=1.0e+4.

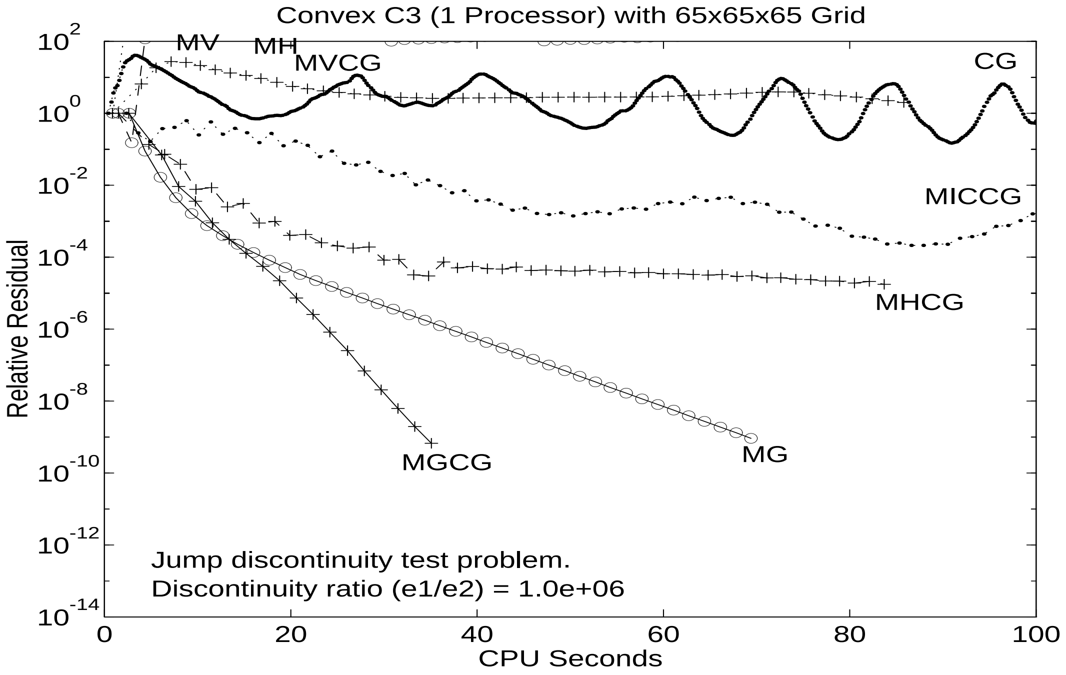


Figure 6.16: The jump discontinuity problem with D=1.0e+6.

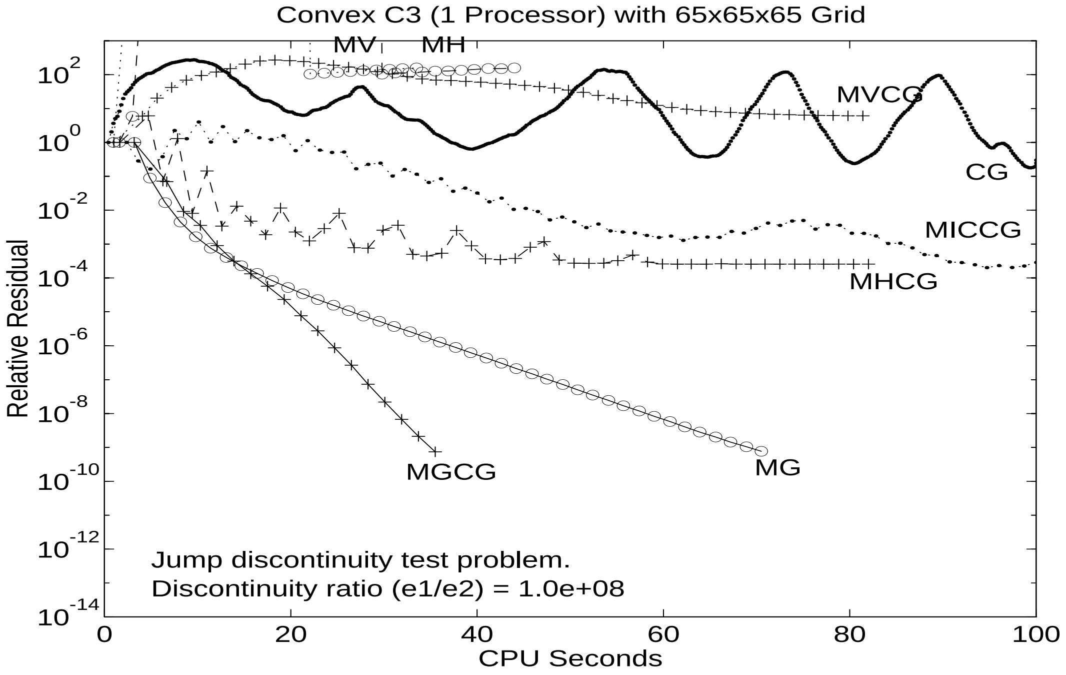


Figure 6.17: The jump discontinuity problem with D=1.0e+8.

Table 6.6: Spectral radii behavior for the Gauss-Seidel-based MG method.

$J(n_J^{1/3})$	$D=1$	$D=10^{-1}$	$D=10^{-2}$	$D=10^{-3}$	$D=10^{-4}$	$D=10^{-8}$	$1-.63/J^{.39}$
2(17)	.30	.41	.50	.52	.52	.52	.52
3(33)	.32	.48	.59	.60	.61	.61	.59
4(65)	.33	.52	.62	.64	.64	.64	.63
5(129)	.34	.54	.64	.66	.66	.66	.66

Table 6.7: Spectral radii behavior for the weighted Jacobi-based MG method.

$J(n_J^{1/3})$	$D=1$	$D=10^{1}$	$D=10^{2}$	$D=10^{3}$	$D=10^{4}$	$D=10^{8}$	$1-.32/J^{.9}$
2(17)	.52	.66	.78	.82	.83	.83	.83
3(33)	.55	.71	.84	.87	.88	.89	.88
4(65)	.56	.73	.86	.89	.90	.91	.91
5(129)	.56	.74	.87	.90	.90	.92	.92

the Gauss-Seidel-based MG method, they are still better than the bound given by the BPWX theory, with $\nu = 0.8$ and $\nu = 0.44$.

6.4.2 Convergence as a function of the problem size

Multilevel methods are provably optimal order for a broad class of problems, meaning that the cost to solve a problem with N unknowns is proportional to N. Unfortunately, as we have outlined in detail in Chapter 5, the discontinuities of $\epsilon(\mathbf{r})$ in the linearized PBE preclude the use of much of the existing theory, which requires strong smoothness assumptions on the problem coefficients. The most recent BPWX theory discussed in Chapter 5 applies in the case of discontinuous coeficients, but requires that coefficient discontinuities lie along element boundaries on all coarse meshes, a condition which must be violated for problems such as the Poisson-Boltzmann equation.

Note that in the previous section, we showed numerically that the contraction numbers of the algebraic methods decay with the number of levels raised to some power, and an analysis of the resulting complexity (as in Chapter 3) would yield a logarithmic term in the complexity bounds. In the case of smaller discontinuities, as in the Poisson-Boltzmann equation, a glance at tables 6.5 through 6.8 shows that the decay is less strong, and the resulting complexity is nearly optimal. We now demonstrate this explicitly with some experiments.

Figure 6.18 gives the cost of each method to solve the jump discontinuity test problem with $D = \epsilon_1/\epsilon_2 = 2/80$ (similar to the Poisson-Boltzmann discontinuity type and magnitude), in time work units on the Convex C240, as the problem size is increased by a factor of two beginning with a $17 \times 17 \times 17$ grid, and ending with a $129 \times 129 \times 129$ grid. Note that in this figure, the *time per unknown* is being plotted as a function of the problem size. The fact that the multilevel method curve is virtually horizontal reflects the optimal order behavior of multilevel methods. In particular, we can see that the superiority of the multilevel method increases as we move to larger grids. This behavior can often be demonstrated for the multilevel method even when the existing theory is no longer applicable.

6.5 Storage requirements

We make a few remarks about the storage required for the multilevel methods as well as some of the other methods appearing in the chapter. We are faced with the discrete problem of the form:

$$Au = f,$$

where A is an $N \times N$ SPD matrix, u is the $N \times 1$ vector of unknowns, and f is the $N \times 1$ vector of source function values. The number of unknowns N is related to the original discrete mesh as $N = I \cdot J \cdot K$,

Table 6.8: Spectral radii behavior for the weighted Jacobi-based MG method.

$J(n_J^{1/3})$	$D=1$	$D=10^{-1}$	$D=10^{-2}$	$D=10^{-3}$	$D=10^{-4}$	$D=10^{-8}$	$1-.38/J^{.44}$
2(17)	.52	.62	.70	.72	.72	.72	.72
3(33)	.55	.69	.76	.78	.78	.78	.77
4(65)	.56	.72	.79	.80	.80	.81	.80
5(129)	.56	.73	.80	.81	.81	.81	.81

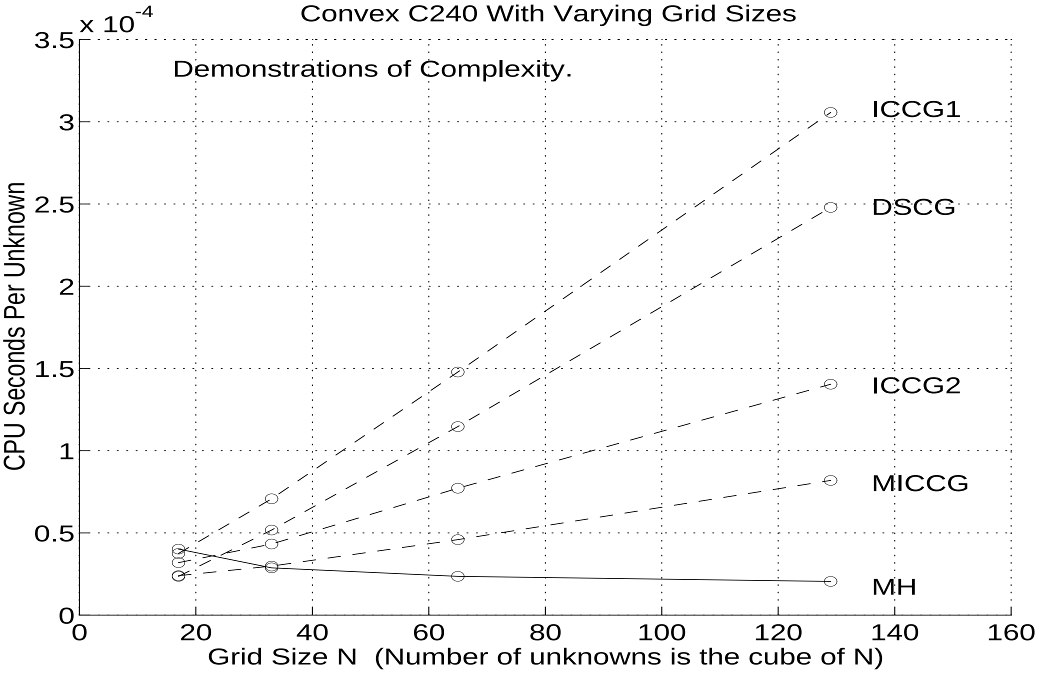


Figure 6.18: Cost per unknown as a function of the grid size.

where I, J, and K are the number of mesh-points in each direction of the non-uniform Cartesian mesh. Employing the box-method on the non-uniform Cartesian mesh, the matrix A can be represented by seven diagonals, only four of which need be stored in arrays of length N, due to the symmetry of A. Therefore, simply to store the matrix problem on the finest desired non-uniform Cartesian mesh requires approximately $4N+1N+1N = 6N$. The iterative algorithms we have considered here require various amounts of additional storage for implementation.

With regard to multilevel methods, since the number of unknowns drops by a factor of eight as one moves to a coarser mesh in three dimensions if standard successively refined non-uniform Cartesian meshes are used, we see that the storage required to represent on all meshes a vector having length N on the finest mesh is:

$$NA = N + \frac{N}{8} + \frac{N}{64} + \cdots = N \cdot \left(\frac{1}{8} + \frac{1}{64} + \cdots \right) \leq \frac{8}{7} \cdot N = N + \frac{N}{7}.$$

We will assume that with the multilevel methods, enough levels are always used so that not only is the coarse problem computational cost negligible, but also the storage requirement (including possibly direct factorization of the matrix) is negligible due to the size of the coarse problem.

Table 6.9 gives the required storage for a selection of methods. We remark that these reflect the storage requirements in our implementations; in particular, while the SOR and CG storage requirements are minimal

Table 6.9: Storage required by various linear elliptic solvers.

Method	Storage Requirements					
Name	A	u	f	I_{k-1}^k	WORK	TOTAL
OSOR	$4N$	$1N$	$1N$	$0N$	$1N$	$7N$
(DS)CG	$4N$	$1N$	$1N$	$0N$	$3N$	$9N$
(M)ICCG	$4N$	$1N$	$1N$	$0N$	$4N$	$10N$
MV	$4N+\frac{4}{7}N$	$1N+\frac{1}{7}N$	$1N+\frac{1}{7}N$	$0N$	$4N+\frac{4}{7}N$	$\approx 11.4N$
MH	$4N+\frac{4}{7}N$	$1N+\frac{1}{7}N$	$1N+\frac{1}{7}N$	$\frac{27}{7}N$	$4N+\frac{4}{7}N$	$\approx 15.3N$
MG	$4N+\frac{14}{7}N$	$1N+\frac{1}{7}N$	$1N+\frac{1}{7}N$	$\frac{27}{7}N$	$4N+\frac{4}{7}N$	$\approx 16.7N$
MVCG	$4N+\frac{4}{7}N$	$1N+\frac{1}{7}N$	$1N+\frac{1}{7}N$	$0N$	$7N+\frac{7}{7}N$	$\approx 14.8N$
MHCG	$4N+\frac{4}{7}N$	$1N+\frac{1}{7}N$	$1N+\frac{1}{7}N$	$\frac{27}{7}N$	$7N+\frac{7}{7}N$	$\approx 18.7N$
MGCG	$4N+\frac{14}{7}N$	$1N+\frac{1}{7}N$	$1N+\frac{1}{7}N$	$\frac{27}{7}N$	$7N+\frac{7}{7}N$	$\approx 20.1N$

or close to minimal, the storage requirements for our multilevel methods could probably be reduced somewhat. However, to maintain a modular structure in the implementation, which enables using the modules together in a straightforward fashion, we have allowed some redundant storage in the implementations. For example, in the methods MH and MHCG, it is possible to implement the operator-based prolongation I_{k-1}^k completely in terms of the matrix A, without requiring explicit storage of I_{k-1}^k. This can save $27N/7 \approx 4N$, which makes these methods almost equivalent to the (M)ICCG methods in terms of storage requirements, with MH and MHCG requiring the same storage as for MV and MVCG, which is approximately $11.4N$ and $14.8N$, respectively. In addition, if standard linear or trilinear prolongations are used with the methods MG and MGCG, which as we have seen in the experiments earlier in this chapter can be very robust for discontinuous coefficient problems such as the PBE, the same savings of approximately $4N$ can be seen for both MG and MGCG, bringing their storage requirements down to approximately $12.8N$ and $16.2N$.

Therefore, not only do the multilevel methods discussed here demonstrate superior complexity properties, we see that they can be implemented with very efficient memory use, requiring the same or only slightly more storage than that required for competing methods, such as preconditioned conjugate gradient methods.

6.6 Pre-processing costs

In this final section, we wish to quantify the cost of some of the multilevel components. In particular, the setup cost to form the algebraic Galerkin coarse level matrices is not trivial, and we would like to make a statement as to its cost in relation to the setup costs of standard multilevel methods and other methods. To begin, we will assume that all methods begin with a box-method discretization of the following problem on the finest mesh:

$$-\nabla \cdot (\bar{\mathbf{a}}(\mathbf{x})\nabla u(\mathbf{x})) + b(\mathbf{x})u(\mathbf{x}) = f(\mathbf{x}) \text{ in } \Omega \subset \mathbb{R}^3, \qquad u(\mathbf{x}) = g(\mathbf{x}) \text{ on } \Gamma,$$

where $\mathbf{x} = (x, y, z)$, and where the tensor $\bar{\mathbf{a}} : \Omega \mapsto \mathbf{L}(\mathbb{R}^3, \mathbb{R}^3)$ has the diagonal form:

$$\bar{\mathbf{a}}(\mathbf{x}) = \left(\begin{array}{ccc} a^{(11)}(\mathbf{x}) & 0 & 0 \\ 0 & a^{(22)}(\mathbf{x}) & 0 \\ 0 & 0 & a^{(22)}(\mathbf{x}) \end{array} \right).$$

This yields a seven-point stencil operator. Our implementation of a general three-dimensional box-method discretization on a non-uniform Cartesian mesh, employing vectorizable MIN0 statements combined with multiplies rather than IF statements for the boundary condition cases, requires approximately $60N$ multiplies and $32N$ additions for a mesh with N mesh points.

Since the methods MV and MVCG employ a standard box-method discretization on coarser meshes, along with standard trilinear prolongation, the only additional setup costs for these methods is the discretization on coarser meshes. From our discussion earlier in this chapter, these costs are approximately:

$$MULTS \approx 60N + \frac{60N}{7}, \qquad ADDS \approx 32N + \frac{32N}{7}.$$

Table 6.10: Setup costs for various methods.

Method	Fine Setup		Coarse Setup, Form Preconditioner, Etc.		SETUP
Name	Mults	Adds	Mults	Adds	TOTAL
OSOR	$60N$	$32N$	$0N$	$0N$	$92N$
(DS)CG	$60N$	$32N$	$5N$	$0N$	$97N$
(M)ICCG	$60N$	$32N$	$18N$	$9N$	$119N$
MV(CG)	$60N$	$32N$	$(60/7)N$	$(32/7)N$	$\approx 105N$
MH(CG)	$60N$	$32N$	$([60+51]/7)N$	$([32+27]/7)N$	$\approx 116N$
MG(CG)	$60N$	$32N$	$([581+1455/7]/8)N$	$([416+1256/7]/8)N$	$\approx 265N$

Methods MH and MHCG incur some additional setup costs, since the coarse problem requires first an averaging of the coefficients before discretization, and then calculation of the prolongation operators once the coarse level matrices are formed. The methods MH and MHCG average the coefficients $a^{(ii)}$ by combining the arithmetic and harmonic average as discussed in Chapter 3, where the harmonic average is defined as $H(x,y) = 2xy/(x+y)$. For example, the coefficient $a^{(11)}$ is averaged as:

$$a_h^{(11)}(\mathbf{x}_{ijk}) = \frac{1}{2} H\left(a_h^{(11)}(\mathbf{x}_{i-1/2,j,k}), a_h^{(11)}(\mathbf{x}_{i+1/2,j,k})\right)$$

$$+\frac{1}{8}\left[H\left(a_h^{(11)}(\mathbf{x}_{i-1/2,j-1,k}), a_h^{(11)}(\mathbf{x}_{i+1/2,j-1,k})\right) + H\left(a_h^{(11)}(\mathbf{x}_{i-1/2,j+1,k}), a_h^{(11)}(\mathbf{x}_{i+1/2,j+1,k})\right)\right.$$

$$\left.+H\left(a_h^{(11)}(\mathbf{x}_{i-1/2,j,k-1}), a_h^{(11)}(\mathbf{x}_{i+1/2,j,k-1})\right) + H\left(a_h^{(11)}(\mathbf{x}_{i-1/2,j,k+1}), a_h^{(11)}(\mathbf{x}_{i+1/2,j,k+1})\right)\right].$$

Since each application of the harmonic average costs 1 addition and 3 multiplies, the total cost to form the average of the coefficient $a_h^{(11)}$ at one coarse mesh point is clearly 2 multiplies and 4 additions, plus the cost of five harmonic averagings, for a total of 17 multiplies and 9 additions. Each of the three coefficients $a^{(ii)}$, $i = 1,2,3$ is averaged at approximately $N/8$ points. Therefore, the cost of forming the coefficients on all of the coarser meshes is $3(17N)/7 = 51N/7$ multiplies and $3(9N)/7 = 27N/7$ additions, where N is the number of mesh points on the finest mesh.

For the Galerkin methods, a simple count of the multiplies and additions required to form the twenty-seven-point Galerkin coarse level operator from a seven-point fine level operator and a general twenty-seven-point prolongation operator is $581N/8$ multiplies and $416N/8$ additions (symmetry is exploited, so only fourteen coefficients are computed). This is approximately $125N$ total floating point operations; the cost of our implementation is slightly higher than the $116N$ quoted in [51] for the same type of method. This is probably due to the fact that our implementation breaks long sequences of continuation lines into the calculation of temporary quantities which are later combined. This is necessary for transportability across machines having compilers with continuation line limitations, since some of the longer Galerkin expressions require approximately *four-hundred* continuation lines unless temporary quantities are employed.

To form a twenty-seven-point Galerkin coarse level operator from a twenty-seven-point fine level operator and a general twenty-seven-point prolongation operator costs $1455N/8$ multiplies and $1256N/8$ additions in our implementation (again, only fourteen coefficients are computed due to symmetry). This is approximately $339N$ total floating point operations (the implementation in [51] requires $309N$, where the additional cost of our implementation is again probably due to the use of temporary quantities).

Table 6.10 summarizes the setup cost information for the methods considered in this chapter. The cost of estimating the optimal relaxation parameter for SOR using some type of power method was not included. The setup cost for DSCG is the scaling of the matrix and source function by a diagonal matrix, and the cost for (M)ICCG applies to the implementation we obtained from NETLIB, which includes the cost of diagonal scaling.

Note that while the setup cost of the method MH is comparable to (M)ICCG in terms of operation count, in fact the actual setup time required for MH is typically much less than that for (M)ICCG, since the averaging and discretization procedures vectorize and parallelize quite readily, whereas the incomplete factorization setup for (M)ICCG does not. This difference is illustrated quite graphically on a vector-processor machine such as the Cray Y-MP; see Figure 6.2.

If a matrix-vector operation (such as applying a smoothing operator) costs approximately $7N$ multiplies and $6N$ additions, and prolongation and restriction cost roughly $27/8N \approx 3.4N$ multiplies and the same number of additions each, then it is not difficult to see that a single iteration of a multilevel method, employing a V-cycle with one pre- and post-smoothing, is approximately $8(13 + 13 + 6.8 + 6.8)N/7 \approx 45N$ floating point operations. Therefore, the additional setup costs of $265N - 105N = 160N$ floating point operations for MG compared to MV is equivalent to approximately three multilevel iterations. Referring to Figure 6.5, we see that this does appear to be the case in practice, as both MV and MH complete three iterations in about the same time that MG completes its first iteration.

In the case that nested iteration is employed, it is often claimed [29] that a single nested iteration culminating with one V-cycle on the finest mesh is enough to solve the problem to discretization error accuracy. If this is the case, then we see that the methods MV and MH should solve the problem in less time than required to simply set up the problem for method MG. Unfortunately, for the more difficult problems having discontinuous coefficients, the method MG is the only reliable approach, and the additional setup costs must be incurred. It should be noted, however, that forming the Galerkin equations is a logically uniform and highly vectorizable/parallelizable computation.

For some of the Poisson-Boltzmann test problems we considered earlier, for which the method MH was very effective, if the same argument applies regarding solving the problem to truncation error with a single V-cycle on the finest mesh, then the method MH solves the problem with fewer operations than that required for setup for the method MG.

6.7 Conclusions

We have shown numerically that the linear multilevel methods discussed in this work are generally more efficient and robust than existing methods for the linearized PBE for a range of test molecules, as well as for a very degenerate test problem with huge coefficient discontinuities. Storage requirements and pre-processing costs were also considered, and the multilevel-based methods are comparable to more traditional methods in each category, with the exception of the Galerkin expression-based method MG which is somewhat more expensive in both setup and storage. However, it was observed that for the more difficult test problems, the additional overhead of MG was easily amortized by improved robustness.

We also presented an in-depth study of the convergence behavior of the Galerkin method MG, motivated by the discussions in Chapter 5. In particular, we observed the convergence rate decay as the number of levels in the method was increased, and as the magnitude of the coefficient discontinuity was increased. The decay rate was seen to behave as a function of the number of levels J in each case, where the function took the form of the bound arising in the BPWX theory for finite element-based multilevel methods discussed in Chapter 5. The decay was better than the worst case predicted by the BPWX theory for each situation considered, which would seem to suggest that it might be possible to show a similar bound on the contraction number for the completely algebraic method MG.

7. Application to the Nonlinear PBE

The two nonlinear multilevel methods presented earlier are investigated numerically when applied to the nonlinear Poisson-Boltzmann equation and to a nonlinear test problem with large jump discontinuities in the coefficients and exponential nonlinearity. A detailed comparison to other methods is presented, including comparisons to nonlinear relaxation methods and to the Fletcher-Reeves nonlinear conjugate gradient method. Our results indicate that the two multilevel methods are superior to the relaxation and conjugate gradient methods, and that the advantage of the multilevel methods grows with the problem size. In addition, experiments indicate that the inexact Newton-multilevel approach is the most efficient and robust method for the test problems.[1]

7.1 Three nonlinear Poisson-Boltzmann methods

Investigations into numerical solution of nonlinear PBE have employed nonlinear Gauss-Seidel [3], nonlinear SOR [152], and a nonlinear conjugate gradient method [135]. Therefore, we will focus on these methods for the comparisons to multilevel methods in following sections. We first briefly describe what results were obtained with these methods, and then describe the nonlinear multilevel implementations based on the material presented earlier in Chapter 4.

7.1.1 Nonlinear Gauss-Seidel and SOR

In Chapter 4 we briefly discussed nonlinear extensions of the classical iterations. These methods are used in [3], where a nonlinear Gauss-Seidel procedure is developed for the full nonlinear Poisson-Boltzmann equation, employing a continuation method to handle the numerical difficulties created by the exponential nonlinearity. Polynomial approximations of the exponential function are employed, and the degree of the polynomial is continued from degree one (linearized PBE) to degree nineteen. At each continuation step, the nonlinear Poisson-Boltzmann equation employing the weaker nonlinearity is solved with nonlinear Gauss-Seidel iteration. The final degree nineteen solution is then used as an initial approximation for the full exponential nonlinearity PBE, and nonlinear Gauss-Seidel is used to resolve the final solution. This procedure, while perhaps one of the first numerical solutions produced for the full nonlinear problem, is extremely time-consuming.

An improvement is, as in the linear case, to employ a nonlinear SOR iteration. The procedure works very well in many situations and is extremely efficient [152]; unfortunately, there are cases where the iteration diverges [151, 152]. In particular, it is noted on page 443 of [152] that if the potential in the solvent (where the exponential term is evaluated) passes a threshold value of seven or eight, then the nonlinear SOR method they propose diverges. We will present some experiments with a nonlinear SOR iteration, provided with an experimentally determined near optimal relaxation parameter, and implemented with a red/black ordering and array oriented data structures for high performance.

[1] The material in this chapter also appears in [100].

7.1.2 A nonlinear conjugate gradient method

In a very recent paper [135], a nonlinear conjugate gradient method is applied to the nonlinear Poisson-Boltzmann equation. The conclusions of their study were that the Fletcher-Reeves variant of the nonlinear conjugate gradient method, which is the natural extension of the Hestenes-Steifel algorithm they had employed for the linearized PBE in an earlier study [42], was an effective technique for the nonlinear PBE. We note that it is remarked on page 1117 of [135] that solution time for the nonlinear conjugate gradient method on the full nonlinear problem is five times greater than for the linear method applied to the linearized problem. We will present experiments with the standard Fletcher-Reeves nonlinear conjugate gradient method which they employed; we described this algorithm in detail in Chapter 4. Our implementation is aggressively optimized for high performance.

7.1.3 Newton-multilevel and nonlinear multilevel methods

We present results for the nonlinear Poisson-Boltzmann equation using two nonlinear multilevel methods; these were selected from several as the most efficient for these types of problems. We compare several different multilevel methods for the nonlinear jump discontinuity test problem in a section which follows later in the chapter.

The first method we employ is the damped inexact-Newton method we presented as Algorithm 4.3 in Chapter 4. We have designed the tolerance and damping strategies to guarantee both global convergence and local superlinear convergence; this is discussed in detail in Chapter 4. The Jacobian system is solved inexactly at each step to the residual tolerance specified in Algorithm 4.3 by employing the linear multilevel we designed for the linearized Poisson-Boltzmann equation, as described in Chapter 3. Refer to §6.1.3 for a detailed description of this method. We take $p = 1$ and $C = 1.0 \times 10^{-2}$ in the algorithm $TEST$.

The second method we employ is the nonlinear multilevel method presented as Algorithm 4.5 in Chapter 4. All components required for this nonlinear method are as in the linear method described in §6.1.3 of Chapter 6, except for the following required modifications. The pre- and post-smoothing iterations correspond to nonlinear Gauss-Seidel, where each smoothing step consisting of ν sweeps; as in the linear case, we employ a variable v-cycle so that ν increases as coarser levels are reached. Nonlinear operator-based prolongation is also employed for nested iteration, as described in Chapter 4; otherwise, linear operator-based prolongation is used. The coarse problem is solved with the nonlinear conjugate gradient method.

For the nonlinear multilevel method, the damping parameter as described in Chapter 4 is required; otherwise, the method is not robust, and does not converge for rapid nonlinearities such as those present in the nonlinear PBE.

7.2 Some test problems

We describe briefly the nonlinear PBE test problems which we use to numerically evaluate and compare the methods which have been proposed for the nonlinear PBE. We also describe a test problem which has a rapid nonlinearity and very large jump discontinuities in the coefficients, which will be used to evaluate some of the multilevel techniques.

7.2.1 The nonlinear Poisson-Boltzmann equation

Consider a very broad range of temperatures $T \in [200K, 400K]$, a broad range of ionic strengths $I_s \in [0, 10]$, and the following representative polygonal domain:

$$\Omega = [\mathbf{x}_{\min} \overset{o}{\mathrm{A}}, \mathbf{x}_{\max} \overset{o}{\mathrm{A}}] \times [\mathbf{y}_{\min} \overset{o}{\mathrm{A}}, \mathbf{y}_{\max} \overset{o}{\mathrm{A}}] \times [\mathbf{z}_{\min} \overset{o}{\mathrm{A}}, \mathbf{z}_{\max} \overset{o}{\mathrm{A}}].$$

We assume that the set of discrete charges $\{\mathbf{x}_1, \ldots, \mathbf{x}_{N_m}\}$ representing the molecule lie well within the domain, and hence far from the boundary Γ of Ω. The nonlinear Poisson-Boltzmann equation for the dimensionless potential $u(\mathbf{x})$ then has the form:

$$-\nabla \cdot (\bar{\mathbf{a}}(\mathbf{x}) \nabla u(\mathbf{x})) + b(\mathbf{x}, u(\mathbf{x})) = f(\mathbf{x}) \text{ in } \Omega \subset \mathbb{R}^3, \qquad u(\mathbf{x}) = g(\mathbf{x}) \text{ on } \Gamma. \tag{7.1}$$

From the discussion in Chapter 1, the problem coefficients are of the following forms, and satisfy the following bounds for the given temperature and ionic strength ranges:

(1) $\bar{\mathbf{a}} : \Omega \mapsto \mathbf{L}(\mathbb{R}^3, \mathbb{R}^3),\ a_{ij}(\mathbf{x}) = \delta_{ij}\epsilon(\mathbf{x}),\ 2 \le \epsilon(\mathbf{x}) \le 80,\ \forall \mathbf{x} \in \Omega.$

(2) $b : \Omega \times \mathbb{R} \mapsto \mathbb{R},\ b(\mathbf{x}, u(\mathbf{x})) = \bar{\kappa}^2(\mathbf{x}) \sinh(\mathbf{x})),\ 0 \le \bar{\kappa}^2(\mathbf{x}) \le 127.0,\ \forall \mathbf{x} \in \Omega.$

(3) $f : \Omega \mapsto \mathbb{R},\ f(\mathbf{x}) = C \cdot \sum_{i=1}^{N_m} z_i \delta(\mathbf{x} - \mathbf{x}_i),\ 5249.0 \le C \le 10500.0,\ -1 \le z_i \le 1,\ \forall \mathbf{x} \in \Omega.$

(4) $g : \Gamma \mapsto \mathbb{R},\ g(\mathbf{x}) = [C/(4\pi\epsilon_w)] \cdot \sum_{i=1}^{N_m} [z_i e^{-\bar{\kappa}(\mathbf{x})|\mathbf{x}-\mathbf{x}_i|/\sqrt{\epsilon_w}}]/|\mathbf{x} - \mathbf{x}_i|,\ \epsilon_w = 80,\ \forall \mathbf{x} \in \Gamma.$

The nonlinear Poisson-Boltzmann problem will then be completely defined by specifying the following quantities:

- $\mathbf{x}_{\min}, \mathbf{x}_{\max}, \mathbf{y}_{\min}, \mathbf{y}_{\max}, \mathbf{z}_{\min}, \mathbf{z}_{\max}$; the domain geometry.
- $\epsilon(\mathbf{x})$; the electrostatic surface of the molecule.
- $\bar{\kappa}(\mathbf{x})$; defined by the ionic strength I_s and the exclusion layer around the molecule.
- C; a constant which depends only on the temperature T.
- $\{\mathbf{x}_1, \ldots, \mathbf{x}_{N_m}\}$; charge locations, and associated fractional charges $\{z_1, \ldots, z_{N_m}\}$.

For all of our molecule test problems, we use $T = 298$ which determines the constant C; this is a common parameter setting for these types of problems. The domain geometry will be defined by the particular molecule, as well as the parameters $\epsilon(\mathbf{x})$ and $\bar{\kappa}(\mathbf{x})$, although we must specify also the ionic strength I_s to completely determine $\bar{\kappa}(\mathbf{x})$. The charge locations and corresponding fractional charges will also be determined by the particular molecule.

7.2.2 A collection of molecule test problems

The test data is taken from the DELPHI and UHBD programs; refer to §6.2.2 of Chapter 6 for details of how these software packages function, and how our software is connected to these codes.

The test molecules chosen for our study of the nonlinear PBE are the following:

- Acetamide (CH_3CONH_2) at 1.0 molar, a small molecule (few angstroms in diameter).
- Crambin at 0.001 molar, a medium size molecule.
- tRNA at 0.2 molar, a large highly charged molecular creating numerical difficulties.
- SOD at 0.1 molar, a large enzyme being intensively studied in biophysics.

7.2.3 Polynomial approximations to the PBE nonlinearity

It has been common in the literature to use low-degree polynomial approximations to the hyperbolic sine function, avoiding the difficulties which occur with the exponential terms in the true sinh function. For example, in [112], three term polynomials are used. However, Figure 7.1 illustrates how poor such approximations are in situations (which frequently occur) when the argument becomes on the order of 10 or more. Note that the units on the vertical axis are 1×10^{12}. In the figure, the true hyperbolic function is plotted with the dotted line; polynomial approximations of degree five and twenty-five are plotted with the solid lines. It seems clear that the full exponential terms must be included in the nonlinear equation in these situations, which occur even in the case of lysozyme [151]. In some sense it is a mute point, since our multilevel methods control the numerical problems of the exponential nonlinearity well, and for implementation reasons (the intrinsic exponential functions are much faster than a loop which evaluates a polynomial) the polynomial nonlinearity solution actually takes longer to compute numerically with our methods (and other methods, when they converge for the exponential case) than the full exponential case. Therefore, we will consider only the more correct exponential model.

7.2.4 A nonlinear elliptic equation with large jump discontinuities

The following test problem will be used to explore the convergence behavior of the multilevel methods. The domain is the unit cube:

$$\Omega = [0,1] \times [0,1] \times [0,1].$$

The nonlinear equation has the form:

$$-\nabla \cdot (\bar{\mathbf{a}}(\mathbf{x}) \nabla u(\mathbf{x})) + b(\mathbf{x}, u(\mathbf{x})) = f(\mathbf{x}) \text{ in } \Omega \subset \mathbb{R}^3, \qquad u(\mathbf{x}) = g(\mathbf{x}) \text{ on } \Gamma. \tag{7.2}$$

where the coefficients in equation (7.2) are taken to be:

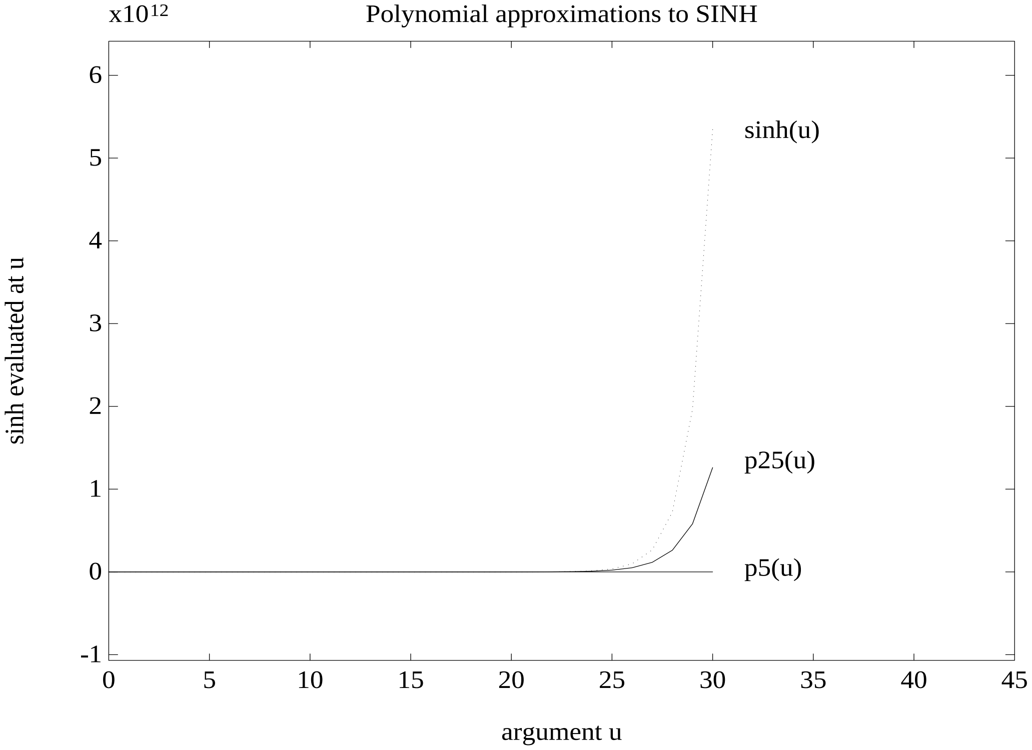


Figure 7.1: Accuracy of polynomial approximations to the hyperbolic sine function.

(1) $\bar{\mathbf{a}} : \Omega \mapsto \mathbf{L}(\mathbb{R}^3, \mathbb{R}^3),\ a_{ij}(\mathbf{x}) = \delta_{ij}\epsilon(\mathbf{x}),\ 1 \leq \epsilon(\mathbf{x}) \leq 1.0 \times 10^8,\ \forall \mathbf{x} \in \Omega.$
(2) $b : \Omega \times \mathbb{R} \mapsto \mathbb{R},\ b(\mathbf{x}, u(\mathbf{x})) = \lambda e^{u(\mathbf{x})},\ \forall \mathbf{x} \in \Omega.$
(3) $f : \Omega \mapsto \mathbb{R},\ -1 \leq f(\mathbf{x}) \leq 1,\ \forall \mathbf{x} \in \Omega.$
(4) $g : \Gamma \mapsto \mathbb{R},\ g(\mathbf{x}) = 0,\ \forall \mathbf{x} \in \Gamma.$

We will construct $\epsilon(\mathbf{x})$ to be piecewise constant, taking one value in a subdomain $\Omega_1 \subset \Omega$, and a second value in the region $\Omega \backslash \Omega_1$, so that $\epsilon(\mathbf{x})$ is defined as follows:

$$\epsilon(\mathbf{x}) = \left\{ \begin{array}{l} 1 \leq \epsilon_1 \leq 1.0 \times 10^8 \text{ if } \mathbf{x} \in \Omega_1, \\ 1 \leq \epsilon_2 \leq 1.0 \times 10^8 \text{ if } \mathbf{x} \in \Omega \backslash \Omega_1. \end{array} \right\}$$

We will take ϵ_1 and ϵ_2 to be quite different in magnitude, so that their ratio:

$$D = \frac{\epsilon_1}{\epsilon_2}$$

can be as large as 10^8 or as small as 10^{-8} for a particular run, and we will observe the resulting convergence behavior of the multilevel methods. We define the subdomain $\Omega_1 \subset \Omega$ to consist of the following two smaller cubes:

$$\Omega_1 = [0.25, 0.50] \times [0.25, 0.50] \times [0.25, 0.50] \quad \bigcup \quad [0.50, 0.75] \times [0.50, 0.50] \times [0.50, 0.75].$$

For this simple problem, it would of course be possible to construct all coarse meshes as needed for the multilevel methods to align with Ω_1; this would not possible with problems such as the nonlinear Poisson-Boltzmann equation and a complex molecule. Therefore, since we wish to simulate the case that the discontinuities in $\epsilon(\mathbf{x})$ cannot be resolved on coarser meshes, the multiple levels of tessellations of Ω into discrete meshes Ω_k are constructed so that the discontinuities in $\epsilon(\mathbf{x})$ lie along mesh lines *only on the finest mesh.*

Table 7.1: Nonlinear Poisson-Boltzmann equation methods.

Method	Description
DINMH	damped-inexact-Newton-method (linear MH as the Jacobian solver)
DFNMH	damped-full-Newton-method (linear MH as the Jacobian solver)
DINMG	damped-inexact-Newton-method (linear MG as the Jacobian solver)
NMH	nonlinear multilevel (nonlinear extension of linear MH)
NCG	nonlinear conjugate gradient method (Fletcher-Reeves variant)
NSOR	nonlinear successive over-relaxation (one-dim. Newton solves)
NGS	nonlinear Gauss-Seidel (one-dim. Newton solves)

Note that if $\epsilon_1 = \epsilon_2 \equiv 1$, then problem (7.2) with the above coefficients is the Bratu problem (see page 432 in [40] for information about this interesting problem) on the unit cube.

7.3 Numerical results for the nonlinear PBE

Table 7.1 provides a key to the plots and tables to follow.

Unless otherwise indicated, all data in the plots and tables to follow *include* the pre-processing costs incurred by the various methods. In other words, the multilevel methods times include the additional time required to set up the problem on coarse grids. This gives a complete and fair assessment of the total time required to reach the solution.

An initial approximation of zero was taken to start each method, and each method used a stopping criteria based on the norm of the nonlinear function:

$$\|F(u^n)\| < \text{ TOL } = 1.0e-9,$$

where u^n represents the n^{th} iterate, and $F(\cdot)$ is the discrete nonlinear algebraic operator for the equation $F(u) = 0$ which we are trying to solve. Of course, this is not the most appropriate stopping criteria for nonlinear iterations (more appropriate stopping tests were discussed in Chapter 4), but for our test problems this test does indicate well when the solution is approached, and it is the best approach for comparing different methods since it guarantees that each method is producing a solution of the same quality.

We remark that it was required to perform all computations in double precision; this is necessitated by the rapid nonlinearities present in the equations, which result an extreme loss in precision. Note that calculations in double precision are more costly than single precision calculations, and so the execution times reported here for some of the methods will be somewhat longer than some of the single precision times reported in Chapter 6.

Timing figures on the Convex C240 and the Convex C3 were obtained from the system timing routine `getrusage`. A more detailed performance analysis on several more sequential as well as some parallel machines can be found in Chapter 8.

7.3.1 Results for acetamide

Figure 7.2 compares the methods in Table 7.1 for the acetamide problem. For this problem, all of the methods converge, and the two multilevel-based algorithms are superior. The nonlinear conjugate gradient and nonlinear SOR methods have comparable performance. The method DINMH is extremely efficient, representing an improvement of more than a factor of fifty over the nonlinear SOR and nonlinear conjugate gradient methods.

7.3.2 Results for crambin

Figure 7.3 compares the methods in Table 7.1 for the crambin problem. Again, all of the methods converge, and the two multilevel-based algorithms are superior. The nonlinear conjugate gradient shows superiority

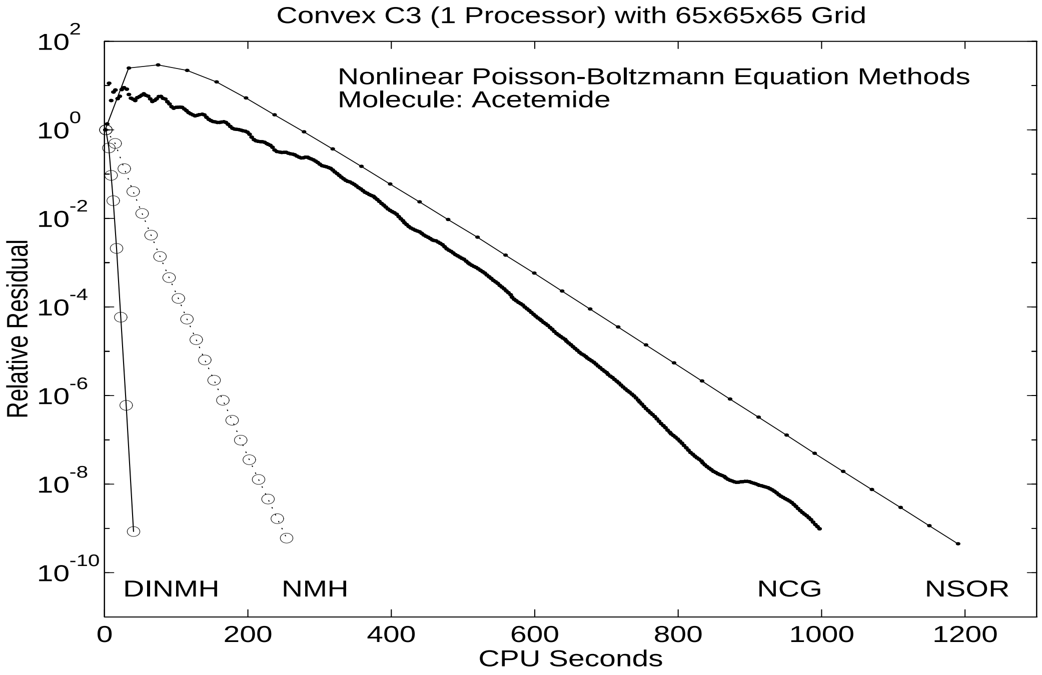


Figure 7.2: Comparison of various methods for the nonlinear acetamide problem.

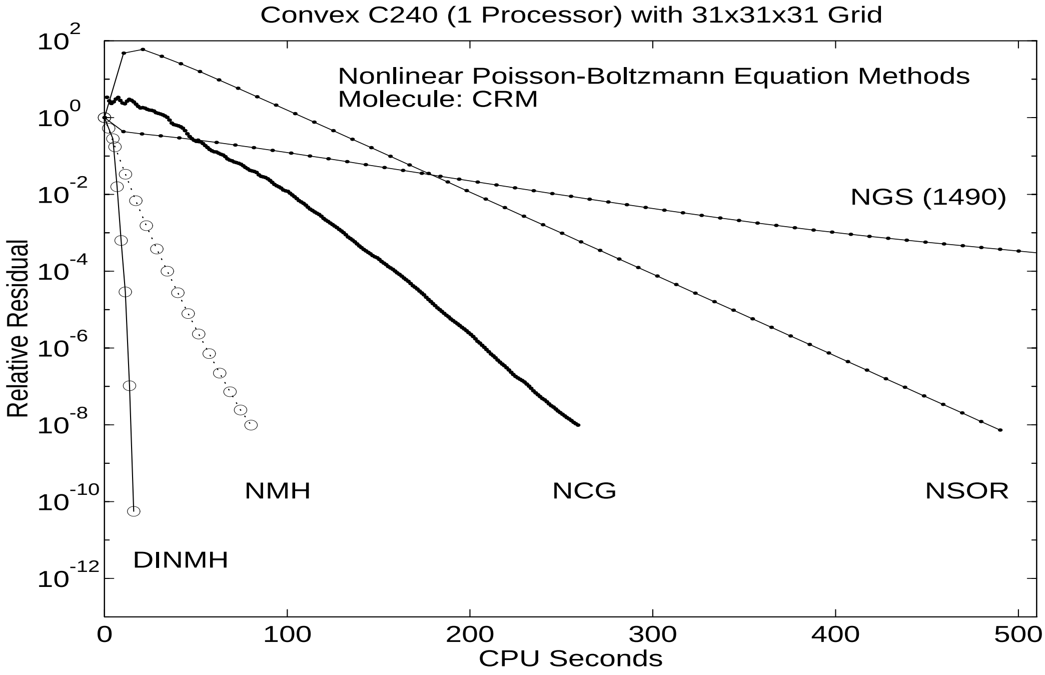


Figure 7.3: Comparison of various methods for the nonlinear crambin problem.

to the nonlinear relaxation methods. The method DINMH is again extremely efficient, representing an improvement of more than a factor of fifty over the nonlinear conjugate gradient method.

7.3.3 Results for tRNA

We included this test problem because it appears to cause severe difficulties for other methods which have been tried; in fact, the nonlinear SOR procedure proposed in [152] was known to diverge for this problem [151]. However, we note that their method was not a true SOR-Newton iteration, and was instead a fixed-point iteration based on a certain splitting of the operator (see page 443 of [152]). When a true SOR-Newton iteration is employed, the method converges for this problem. Figure 7.4 shows the relative performance of the various methods. Again, the method DINMH is the most efficient by far of the methods presented, representing a factor of fifty improvement over the next best method.

Note that for this problem, the nonlinear multilevel method diverges, even with linesearch for a damping parameter. Since we do not enforce the nonlinear variational conditions exactly, as outlined in Chapter 4, we have no guarantee that the coarse level correction is a descent direction, and so this method is not a global method; this particular test problem illustrates this fact.

7.3.4 Results for SOD

Figure 7.5 shows only two methods applied to the SOD test problem: the method DINMH applied to the full nonlinear PBE; and the linear DSCG method applied to the linearized PBE. All other nonlinear methods studied here diverged for this test problem. Again, the method DINMH converges very rapidly, and the superlinear convergence is clearly visible.

We have included the plot of the linear method DSCG to show clearly that the DINMH method, solving the full nonlinear problem, is more than a factor of two times more efficient than one of the best available methods in the literature for only the linearized problem.

7.4 Multilevel behavior for the jump discontinuity problem

Figure 7.6 shows the behavior of the five methods in Table 7.1 when applied to the jump discontinuity dest problem, with $D = \epsilon_1/\epsilon_2 = 1.0e-3$. The three multilevel-based methods are substantially superior to the nonlinear relaxation and conjugate gradient methods. More interestingly, the comparison between the full Newton method (DFNMH) and the inexact Newton method (DINMH) shows at least a factor of four improvement gained by employing the inexactness strategy outlined in Chapter 4.

Figure 7.7 shows the first 200 CPU seconds of Figure 7.6 expanded to the whole axis. We have included the linear methods MH, MICCG, and DSCG on the plot to illustrate more graphically how efficient the method DINMH is; it requires less than a factor of two times more CPU seconds than the linear method MH for the linearized problem, and is a factor of two times more efficient than the next best *linear* method, MICCG.

7.5 Storage requirements

We make a few remarks about the storage required for the multilevel methods as well as some of the other methods appearing in the chapter. We are faced with the discrete problem of the form:

$$Au + B(u) = f,$$

where A is an $N \times N$ SPD matrix, $B(\cdot)$ is a nonlinear function mapping $\mathbb{R}^N$ into $\mathbb{R}^N$, u is the $N \times 1$ vector of unknowns, and f is the $N \times 1$ vector of source function values. The number of unknowns N is related to the original discrete mesh as $N = I \cdot J \cdot K$, where I, J, and K are the number of mesh-points in each direction of the non-uniform Cartesian mesh. Employing the box-method on the non-uniform Cartesian mesh, the matrix A can be represented by seven diagonals, only four of which need be stored in arrays of length N, due to the symmetry of A. The box-method produces "diagonal" nonlinear functions $B(\cdot)$ from the types of nonlinear partial differential equations we consider in this work, and $B(\cdot)$ can be represented by a single real nonlinear function and a coefficient array of length N. Therefore, simply to store the nonlinear algebraic

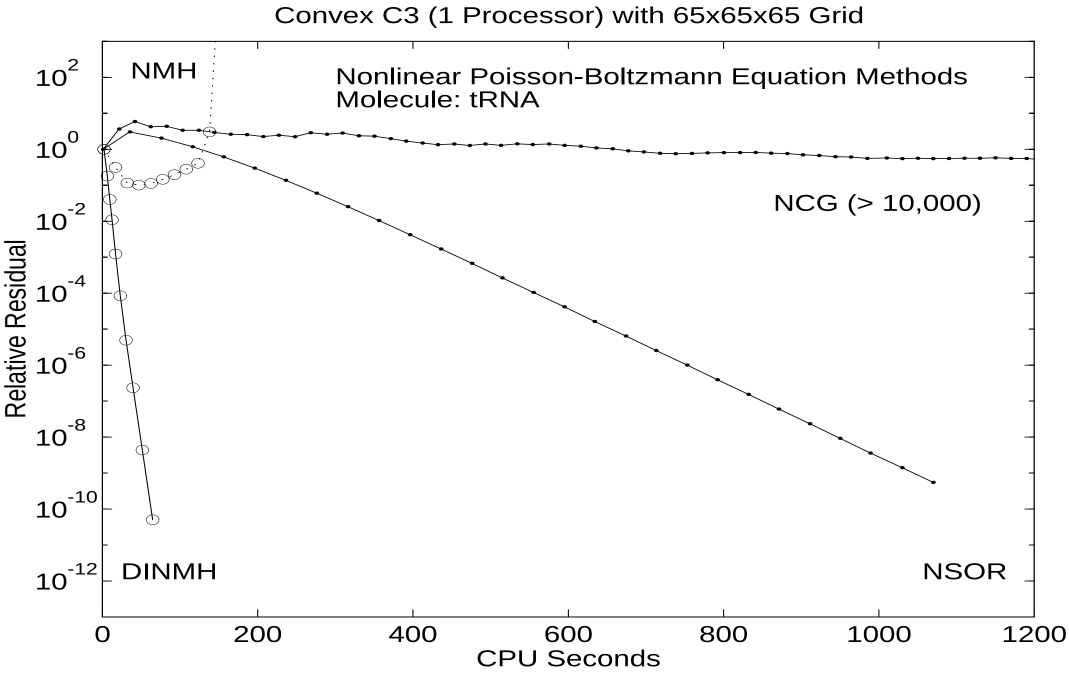


Figure 7.4: Comparison of various methods for the nonlinear tRNA problem.

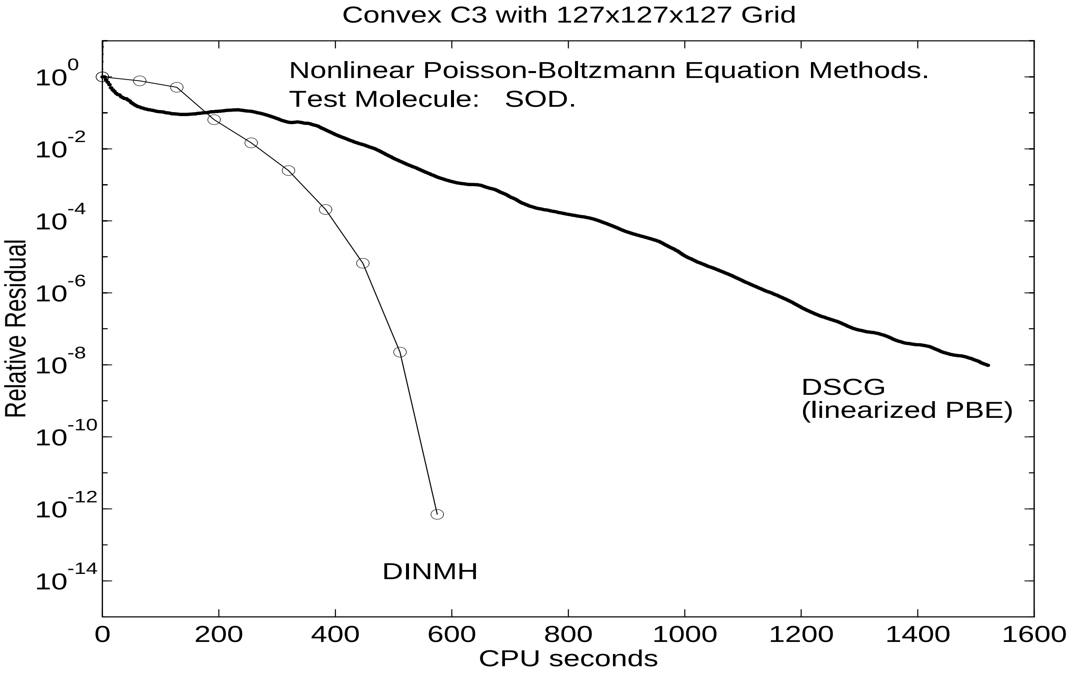


Figure 7.5: Comparison of various methods for the nonlinear SOD problem.

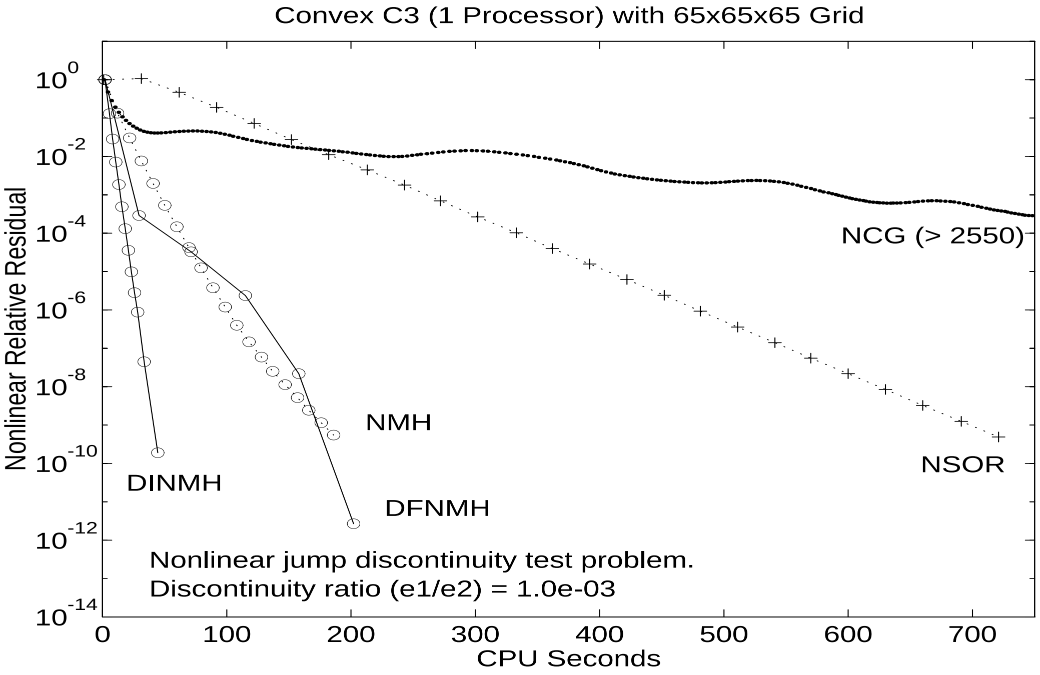


Figure 7.6: Comparison of methods for the nonlinear discontinuity problem.

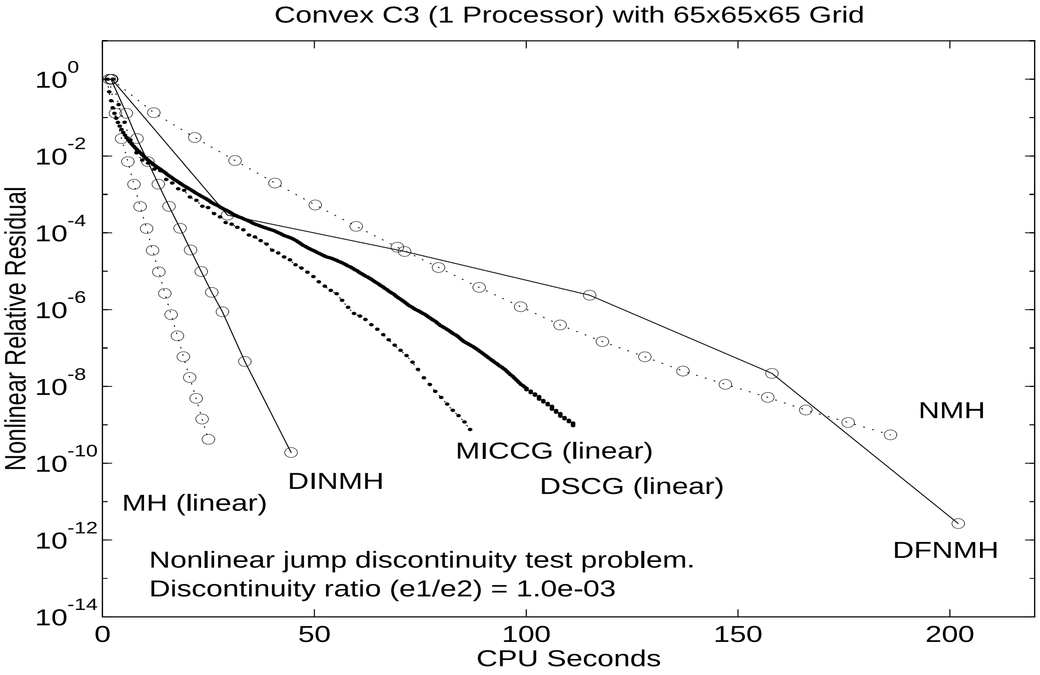


Figure 7.7: Enlargement of previous figure, with linear methods added.

Table 7.2: Storage required by various nonlinear elliptic solvers.

Method	Storage Requirements						
Name	A	$B(\cdot)$	u	f	I_{k-1}^{k}	WORK	TOTAL
NGS	$4N$	$1N$	$1N$	$1N$	$0N$	$1N$	$8N$
NSOR	$4N$	$1N$	$1N$	$1N$	$0N$	$1N$	$8N$
NCG	$4N$	$1N$	$1N$	$1N$	$0N$	$6N$	$13N$
NMH	$4N + \frac{4}{7}N$	$1N + \frac{1}{7}N$	$1N + \frac{1}{7}N$	$1N + \frac{1}{7}N$	$\frac{27}{7}N$	$5N + \frac{5}{7}N$	$\approx 17.6N$
DINMH	$4N + \frac{4}{7}N$	$1N + \frac{1}{7}N$	$1N + \frac{1}{7}N$	$1N + \frac{1}{7}N$	$\frac{27}{7}N$	$7N + \frac{7}{7}N$	$\approx 19.9N$
DINMG	$4N + \frac{14}{7}N$	$1N + \frac{1}{7}N$	$1N + \frac{1}{7}N$	$1N + \frac{1}{7}N$	$\frac{27}{7}N$	$7N + \frac{7}{7}N$	$\approx 21.3N$

problem on the finest desired non-uniform Cartesian mesh requires approximately $4N+1N+1N+1N = 7N$. The nonlinear iterative algorithms we have considered here require various amounts of additional storage for implementation.

With regard to multilevel methods, since the number of unknowns drops by a factor of eight as one moves to a coarser mesh in three dimensions if standard successively refined non-uniform Cartesian meshes are used, we see that the storage required to represent on all meshes a vector having length N on the finest mesh is:

$$NA = N + \frac{N}{8} + \frac{N}{64} + \cdots = N \cdot \left(\frac{1}{8} + \frac{1}{64} + \cdots \right) \leq \frac{8}{7} \cdot N = N + \frac{N}{7}.$$

We will assume that with the multilevel methods, enough levels are always used so that not only is the coarse problem computational cost negligible, but also the storage requirement (including possibly direct factorization of the matrix) is negligible due to the size of the coarse problem.

Table 7.2 gives the required storage for a selection of methods. As in our discussion in Chapter 6, these reflect the storage requirements in our implementations; in particular, while the NGS, NSOR, and NCG storage requirements are minimal or close to minimal, the storage requirements for our multilevel methods could be reduced somewhat. To maintain a logically modular structure in our implementations, we have allowed some redundant storage in the implementations. In the methods NMH and DINMH, it is possible to implement the (linear or nonlinear) operator-based prolongation I_{k-1}^{k} completely in terms of the matrix A (and the nonlinearity $B(\cdot)$), without requiring explicit storage of I_{k-1}^{k}. This can save $27N/7 \approx 4N$, which makes these methods almost equivalent to NCG in terms of storage requirements, with NMH and DINMH requiring approximately $13.7N$ and $16N$, respectively. If standard linear or trilinear prolongations are used with the method DINMG, which is the most robust approach for nonlinear problems having discontinuous coefficient such as the nonlinear PBE, the same savings of approximately $4N$ can be seen, bringing their storage requirements for DINMG down to approximately $17.4N$.

Therefore, as in the case of the linear multilevel methods of Chapter 3, not only do the multilevel methods discussed here demonstrate superior complexity properties, we see that they can be implemented with very efficient memory use, requiring the same or only slightly more storage than that required for competing methods such as the Fletcher-Reeves nonlinear conjugate gradient methods.

7.6 Conclusions

We have shown numerically that the multilevel methods discussed in this work are generally more efficient and robust than existing methods for the nonlinear PBE for a range of test molecules, and for a difficult test problem with huge coefficient discontinuities and rapid nonlinearity. In addition, our results indicate that the damped-inexact-Newton-multilevel approach is not only the most efficient approach for these problems, but is also the most robust of all the methods considered. It converged in all situations, and for the SOD test problem was the only nonlinear method to converge.

This shows that the damped-inexact-Newton-multilevel method not only makes the nonlinear model feasible by providing a robust solution technique, but it actually improves on the efficiency of available linear algorithms which are currently used for the less accurate linear model. We remark that initial numerical experiments with larger mesh sizes show that the improvement of the damped-inexact-Newton-multilevel approach over linear methods such as diagonally scaled conjugate gradients grows with the problem size [100].

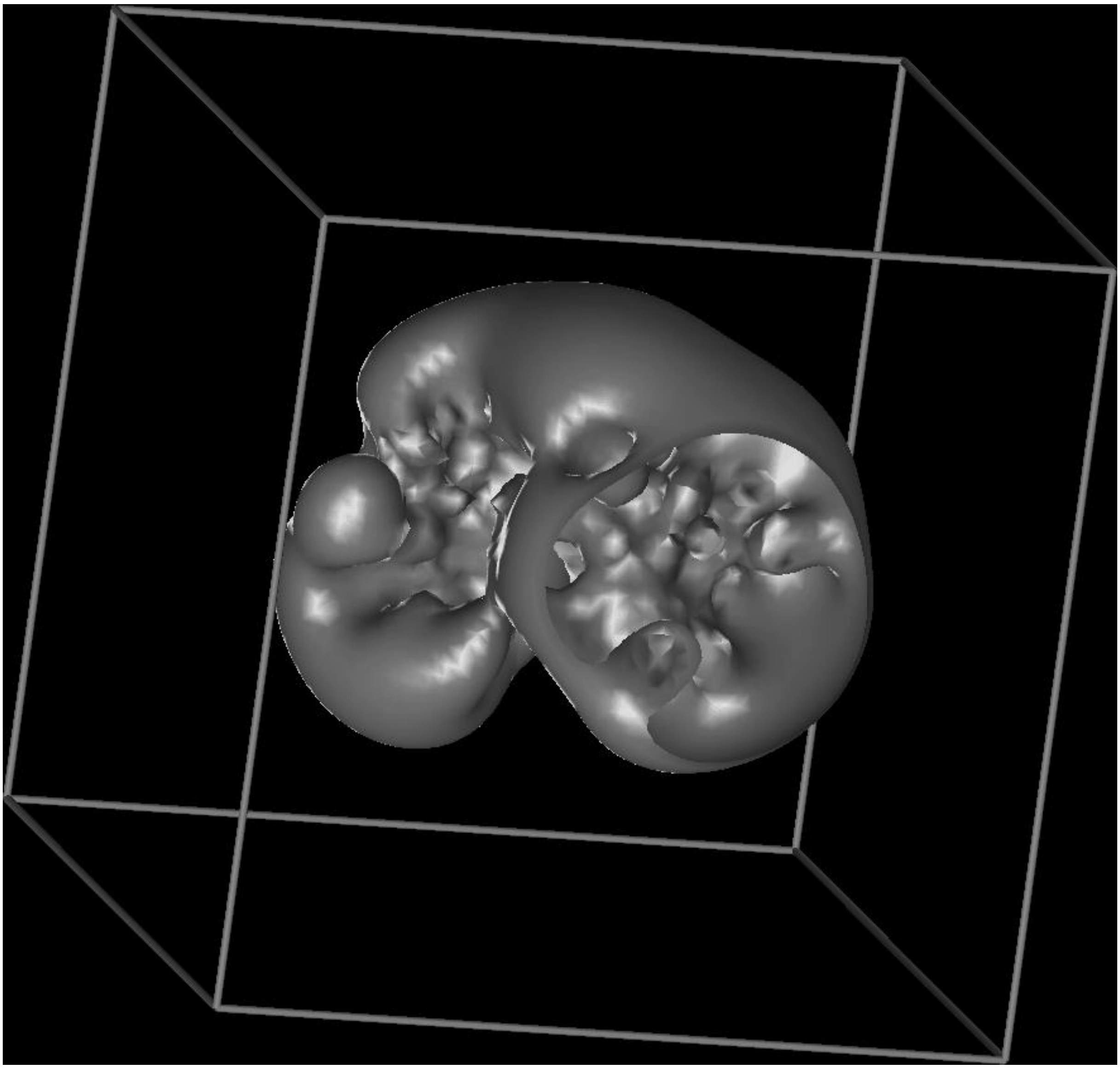


Figure 7.8: The potential isosurface of an SOD enzyme at roughly $-1.2e_c/[k_BT]$. The potential field was obtained by solving the nonlinear Poisson-Boltzmann equation numerically, using the damped-inexact-Newton-multilevel method. The front of the surface has been sliced off, to give a better view of the behavior of the electrostatic potential inside the surface, near the binding sites. The SOD (SuperOxide Dismutase) enzyme is an *antiradical* or *antioxident*, meaning that it moves around the body binding to and then deactivating free radicals in the body, preventing them from causing cancer or other cell damage in the body. It is believed that antiradicals such as SOD can help prevent cancer, heart disease, and a host of other diseases, and perhaps even delay the aging process. The electrostatic *steering effect* of the SOD enzyme, which enables it to attract, bind to, and deactivate free radicals in the human body, is vividly seen in the above graph. The negatively charged surface surrounding the postively charged binding sites has the effect of "steering" the radical into the site, from the upper left of the graph into the "hole" in the electrostatic surface.

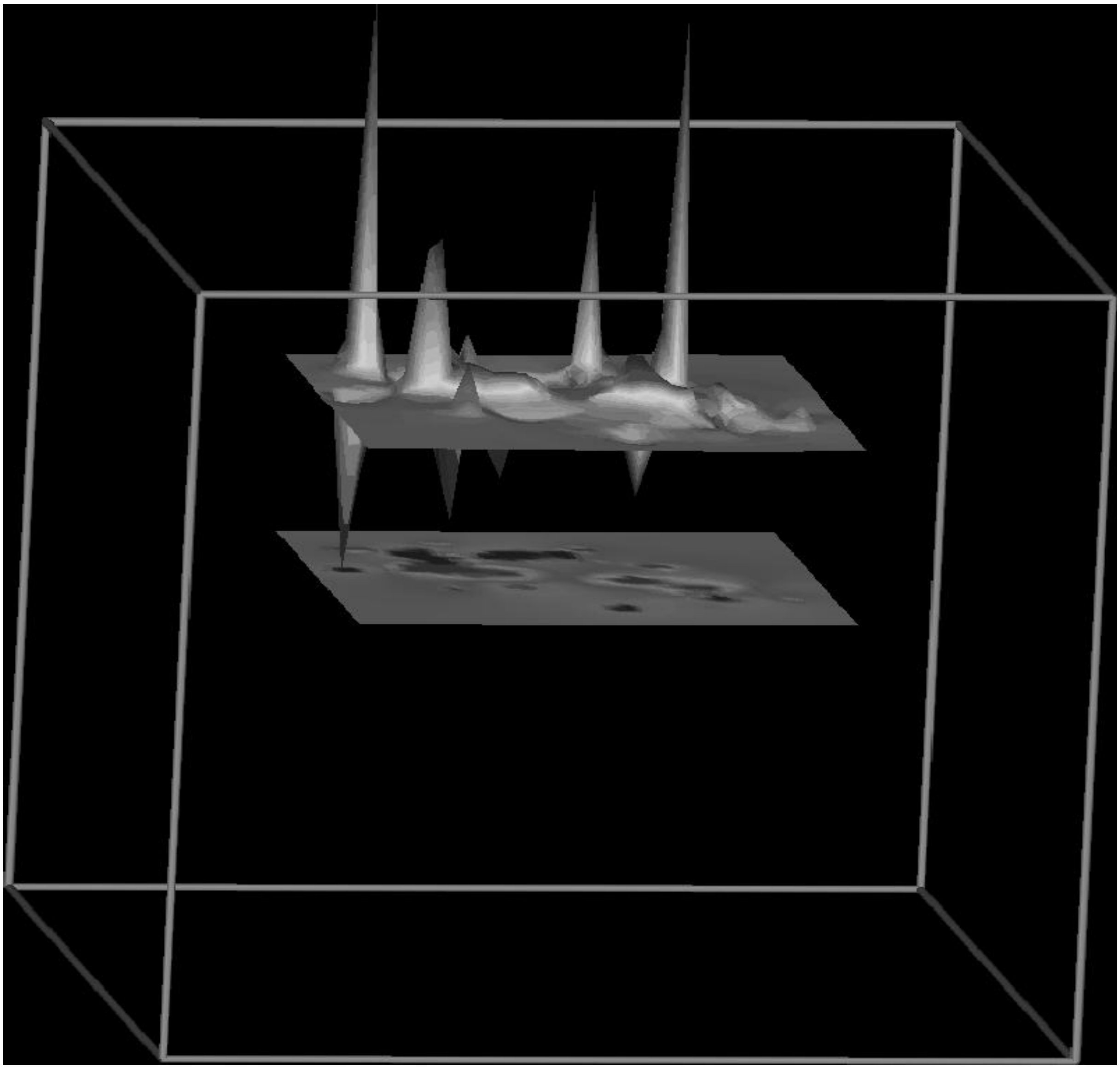


Figure 7.9: A potential slice of SOD passing through the two binding sites. The linearized and nonlinear Poisson-Boltzmann models yield substantially different electrostatic potential values near the binding sites, leading to correspondingly different reaction rates predicted by Brownian dynamics simulations. This implies that the full nonlinear model is important for certain modeling situations. In the March 4, 1993 issue of the New York times, it was announced that researchers at MIT had discovered that the gene responsible for generating the antiradical SOD in the human body is defective in patients with amytrophic lateral sclerosis (ALS), also known as Lou Gehrig's disease. The discovery was deemed so important that the MIT report was accepted for publication by the journal Nature within 36 hours of submission. Using models such as the nonlinear Poisson-Boltmann model to gain a better understanding of the function of antiradical enzymes such as SOD, it is hoped that new drug therapies will be developed for diseases such as ALS based on synthetic antiradical agents.

8. Performance Benchmarks

In this chapter, we present a collection of performance statistics and benchmarks for the multilevel solvers on a number of sequential and parallel computers. The software has been developed specifically for high performance on vector-processor computers; the non-uniform Cartesian meshes give rise to fast diagonal-form matrix-vector operations in all components of the multilevel iteration. High vector and parallel performance is demonstrated on several vector and shared memory parallel computers with a series of numerical experiments, and we give a table of benchmark figures for the software, taken from a large sample of commonly used workstations and supercomputers.

8.1 Performance on shared memory parallel computers

We consider parallel performance of the multilevel software described in Chapters 6 and 7; we consider only shared-memory, coarse-grain parallel machines, such as the Convex C240, Convex C3, and the Cray Y-MP.

8.1.1 Execution times

Timings, operation counts, and megaflops (one million floating point operations per second) figures on the Cray Y-MP were obtained from the performance monitoring hardware accessed through *perftrace* and *perfview*. Timing figures on the Convex C240 and the Convex C3 were obtained from the system timing routine `getrusage`, and megaflop rates were computed from the exact operation counts provided by the Cray.

Figures 8.1 and 8.2 give the required execution times to solve the jump discontinuity test problem on the Convex C240 and the Convex C3, respectively, using method MH described in Chapter 6. The problem was solved when the relative residual was reduced below $1.0e-9$. The plots display the execution time as the number of processors is increased from one to four on the Convex C240, and from one to eight on the Convex C3.

8.1.2 Parallel speedup and efficiency

We first recall the definitions of parallel speedup and efficiency:

$$S_P \equiv \frac{\text{Execution time on 1 processor}}{\text{Execution time on P processors}},$$

$$E_P \equiv S_P \times \frac{1}{P} \times 100\%.$$

Figures 8.3 and 8.4 display the parallel speedup of the software on the Convex C240 and the Convex C3, as the number of processors is increased from one to four, and one to eight, respectively. In addition, figures 8.5 and 8.6 display the parallel efficiency of the software on the Convex C240 and the Convex C3, as the number of processors is increased from one to four, and one to eight, respectively.

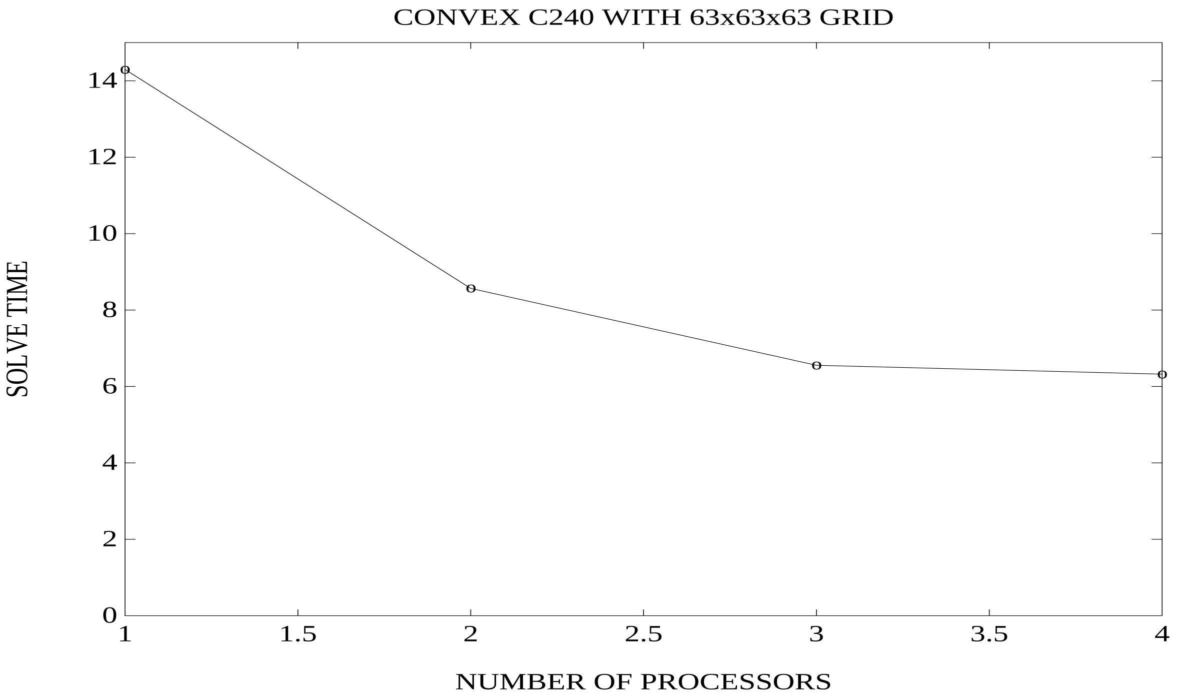


Figure 8.1: Execution times over 4 processors on the Convex C240.

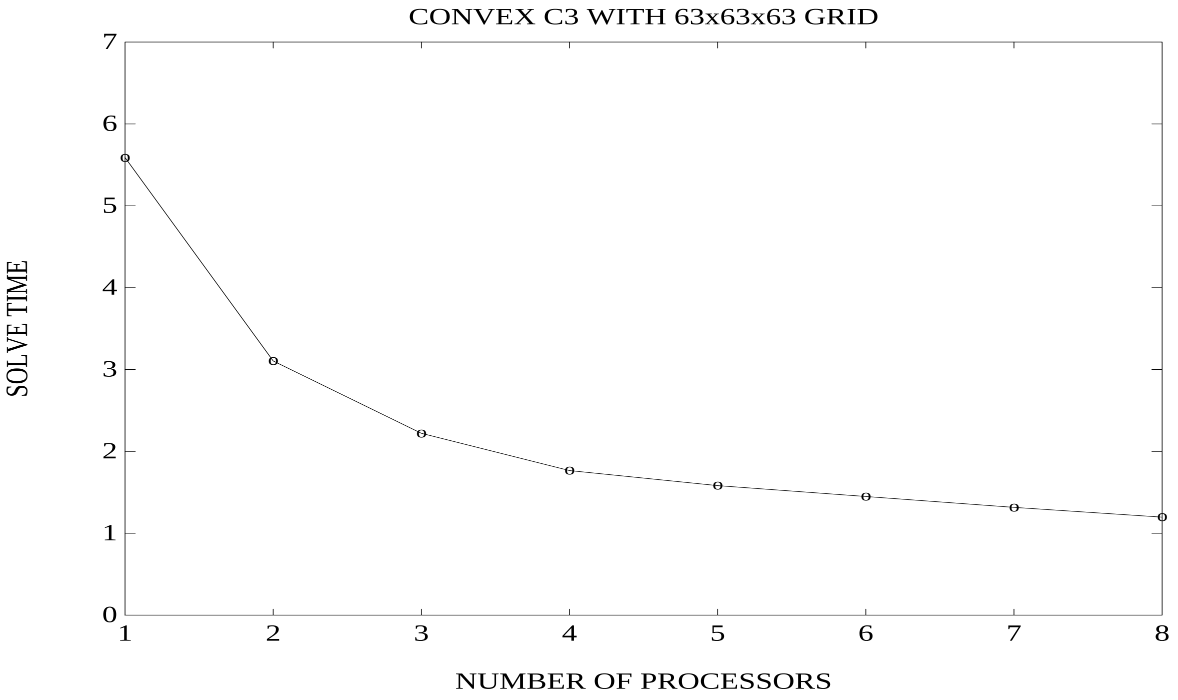


Figure 8.2: Execution times over 8 processors on the Convex C3.

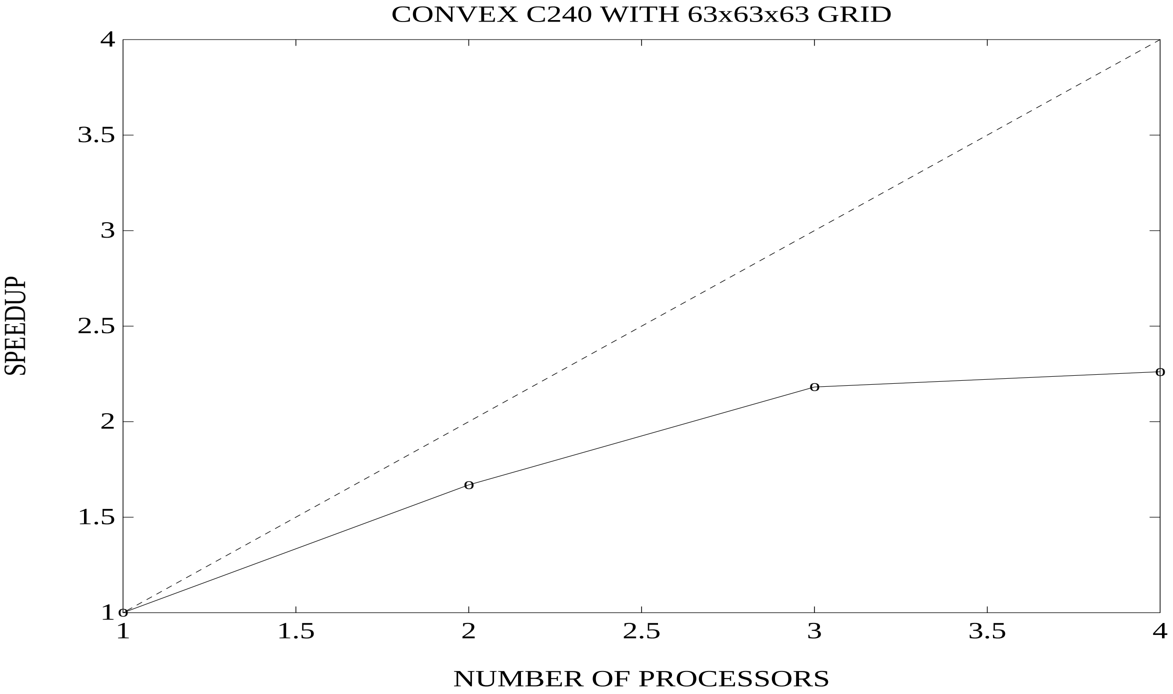


Figure 8.3: Speedup over 4 processors on the Convex C240.

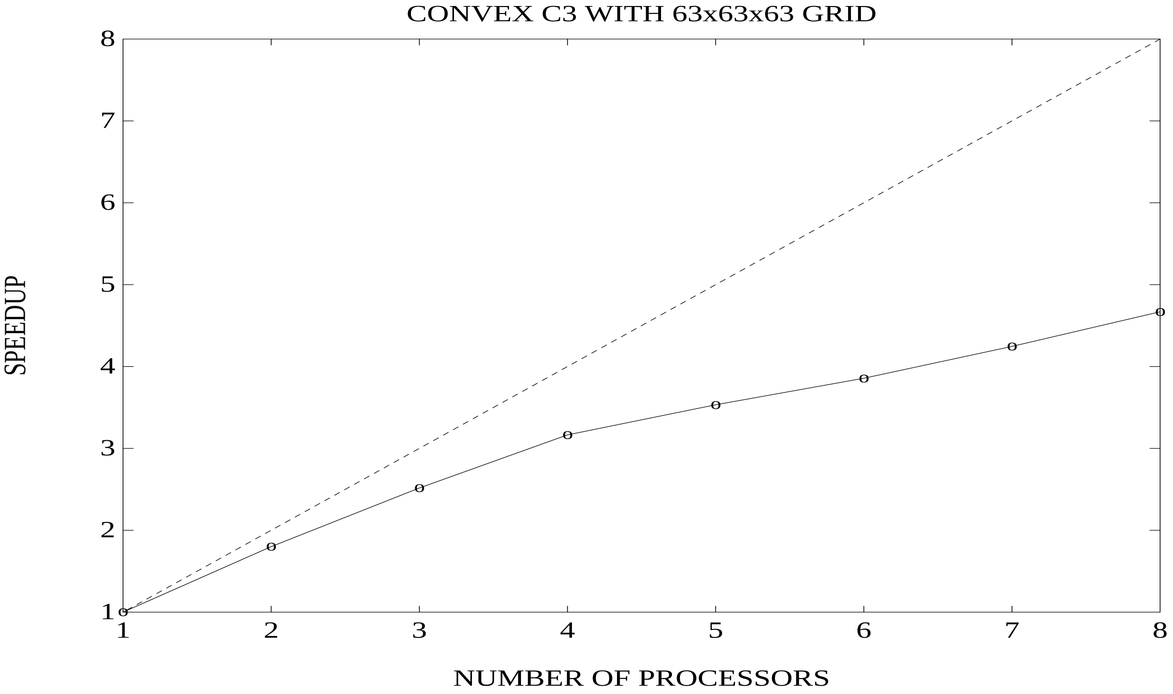


Figure 8.4: Speedup over 8 processors on the Convex C3.

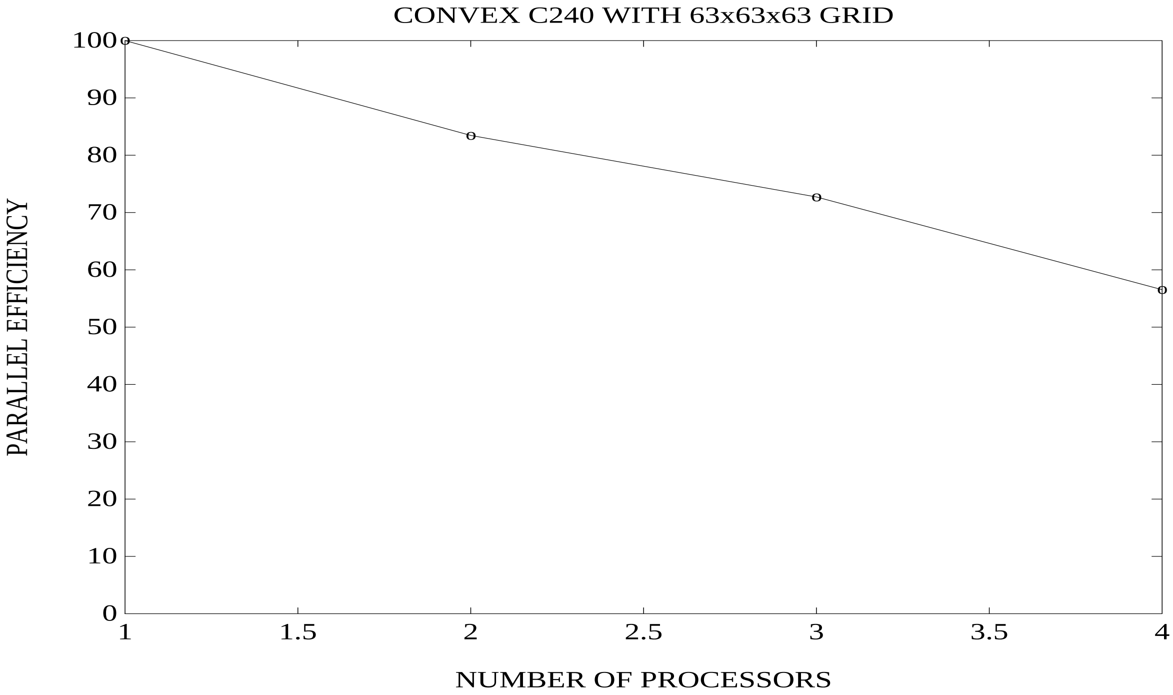


Figure 8.5: Parallel efficiency over 4 processors on the Convex C240.

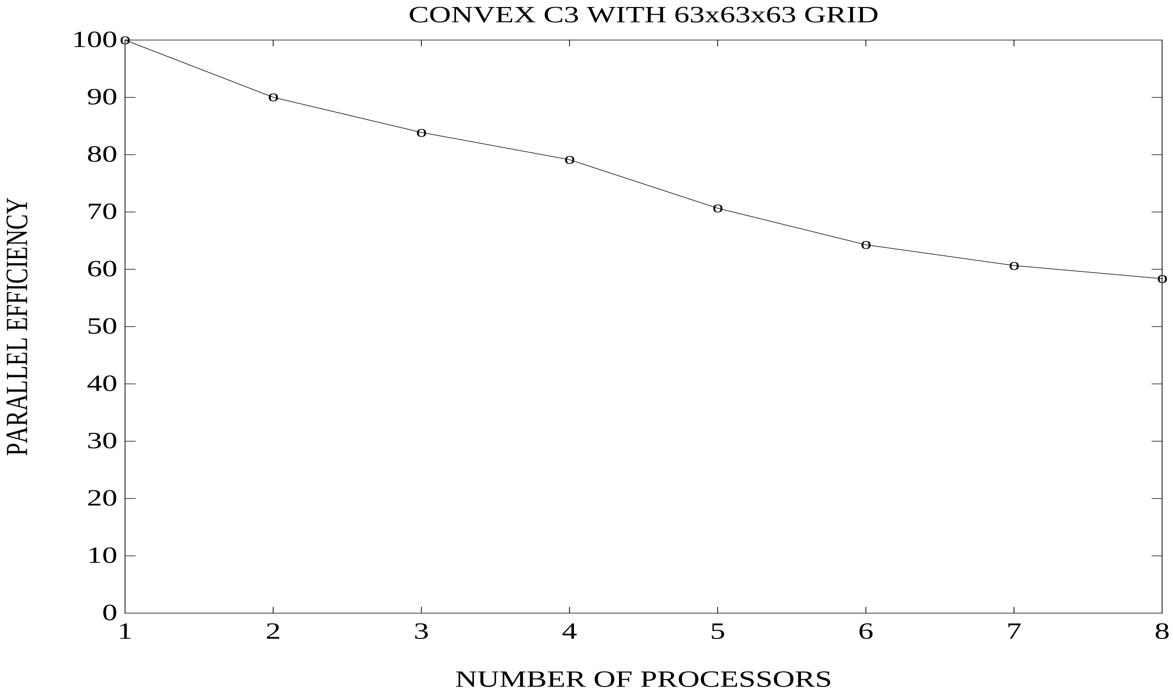


Figure 8.6: Parallel efficiency over 8 processors on the Convex C3.

Table 8.1: Performance summary of the multilevel solver on several architectures.

Machine	Megaflops
Cray Y-MP (4 processors)	641.0
Cray Y-MP (1 processor)	224.0
Convex C3 (8 processors)	237.0
Convex C3 (1 processor)	57.4
Convex C240 (4 processors)	39.5
Convex C240 (1 processor)	18.9
IBM RS6000	12.7
Sun SPARC 1	0.6

8.2 Performance summary on a selection of computers

Table 8.1 summarizes the performance characteristics of the software on a number of sequential, vector, and parallel supercomputers and workstations. Figure 8.7 shows the megaflop rate behavior on the Cray Y-MP, as the number of processors in increased from one to four.

We wish to remark that the Cray Y-MP figures in this chapter are for the vanilla multigrid method MV, without Galerkin expressions, coefficient averaging, or the other techniques employed in methods MH and MG. However, we expect similar behavior for MH and MG on the Cray Y-MP. All other timings and megaflop rates appearing in this chapter for the Convex computers and the miscellaneous workstations were for the method MH.

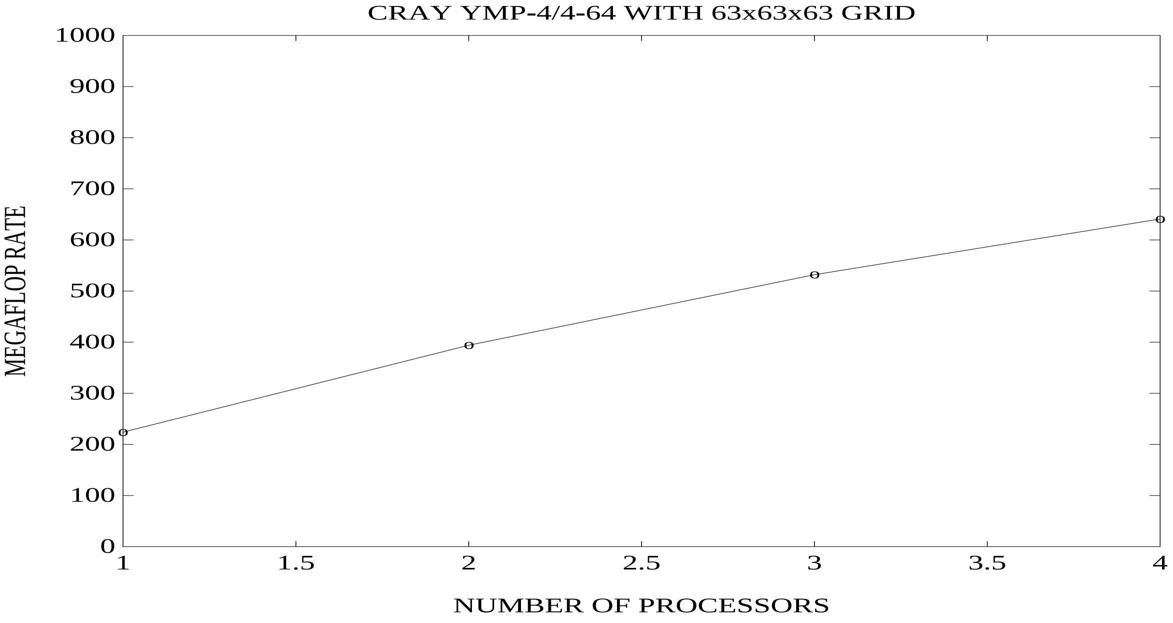


Figure 8.7: Megaflop rates over 4 processors on the Cray Y-MP.

A. Symbolic Stencil Calculus

In Chapter 3 we discussed in detail how to algebraically enforce the variational conditions:

$$A_{k-1} = I_k^{k-1} A_k I_{k-1}^k, \qquad I_k^{k-1} = (I_{k-1}^k)^T, \tag{A.1}$$

in the case of box or finite element discretizations on logically non-uniform Cartesian meshes. Our approach was to represent the system operator A_k and the prolongation operator I_{k-1}^k as *stencils*, and to compute the Galerkin coarse grid operator A_{k-1} as a stencil, the entries of which were combinations of the fine operator stencil entries and the prolongation stencil entries. We described in detail the symbolic *stencil calculus* in the one-dimensional case for computing these general matrix-matrix products; our discussion was motivated by the work of R. Falgout [63], who used a form of stencil calculus for related two-dimensional problems. A complete description as pertains to two-dimensional stencil calculations can be found in the thesis [63].

The two- and three-dimensional cases are simple generalizations of the one-dimensional case, although the expressions quickly become unmanageable. By implementing the stencil calculus in a symbolic manipulation environment such as MAPLE or MATHEMATICA, the complex expressions can be obtained without error. In this Appendix, we will present the expressions obtained for general system operator stencils and prolongation operator stencils in the one-, two-, and three-dimensional cases. We also present the expressions for operator-based prolongation in all three cases.

Note that, using these expressions, the Galerkin conditions can be enforced for a variety of discretization stencils and prolongation stencils on logically non-uniform Cartesian meshes. Our calculations in two and three dimensions are for nine- and twenty-seven-point stencils, which would correspond to bilinear and trilinear finite element discretizations on rectangles and three-dimensional boxes. These stencils include as special cases most other stencils used (see our discussions below for more detail). If a different discretization stencil or prolongation stencil is desired, the Galerkin expressions can be obtained from the expressions printed below by simply setting the corresponding fine operator stencil entries or prolongation operator stencil entries to zero.

A.1 Galerkin operator expressions

In this section, we collect together the expressions for the Galerkin coarse matrix stencils for one-, two-, and three-dimensional problems. These expressions do not appear to be available in the literature (it seems that only the expressions for the two-dimensional case have appeared in print, in the special case of symmetric matrices and a particular prolongation operator, in an Appendix to the paper [19]). The three-dimensional expressions do not appear to have been published at all, apparently due to their complexity and how many pages the expressions require. Since the Galerkin approach seems to be the only robust multilevel approach for certain degenerate problems, and since logically non-uniform Cartesian box-method and finite element method meshes are widely used for both two- and three-dimensional problems, we are including these expressions here.

A.1.1 One dimension

As detailed in Chapter 3 for the one-dimensional case, the fine grid operator stencil, the restriction stencil, and the prolongation stencil are given by:

$$A_h = \left[\begin{array}{ccc} -W_i & C_i & -E_i \end{array} \right]_h^h, \qquad I_h^H = \left[\begin{array}{ccc} PW_i & PC_i & PE_i \end{array} \right]_{h(H)}^H,$$

$$I_H^h = [\ PE_{i-1} \quad 0 \quad PW_{i+1}\ ]_{H(h)}^h \vee [\ PC_i\ ]_{H(H)}^h.$$

As computed in Chapter 3, the Galerkin coarse grid operator stencil is:

$$A_H = \left[\begin{array}{ccc} -W_i^H & C_i^H & -E_i^H \end{array} \right]_H^H,$$

where

$$\begin{aligned} C_i^H &= (PC_i)^2 C_i + (PW_i)^2 C_{i-1} + (PE_i)^2 C_{i+1} - PW_i PC_i E_i - PE_i PC_i W_{i+1} \\ &\quad - PW_i W_i - PE_i E_i, \\ -E_i^H &= PE_i PW_{i+2} C_{i+1} - PC_i PW_{i+2} E_i - PC_{i+2} PE_i E_{i+1}, \\ -W_i^H &= PW_i PE_{i-2} C_{i-1} - PC_i PE_{i-2} W_i - PC_{i-2} PW_i W_{i-1}. \end{aligned}$$

Remark A.1. In the case that the original matrix A_h is symmetric, or $W_i = E_{i-1}$, then since the Galerkin coarse space matrix $A_H = (I_H^h)^T A_h I_H^h$ is also clearly symmetric, only the center entry C_i^H and either W_i^H or E_i^H need be computed and stored. In this case, the expressions for the Galerkin coarse grid matrix stencil entries can be expressed completely in terms of C_i and one of W_i or E_i.

Recall that a box-method, finite difference, or finite element (piecewise linear basis functions) discretization of Poisson's equation on a uniform one-dimensional mesh will yield the following stencil representation of the system matrix:

$$A_h = \left[\begin{array}{ccc} -1 & 2 & -1 \end{array} \right]_h^h, \tag{A.2}$$

where the meshwidth h has been divided out of the matrix. Linear interpolation and the resulting restriction in this case are represented by:

$$I_H^h = \left[\begin{array}{ccc} \frac{1}{2} & 0 & \frac{1}{2} \end{array} \right]_{H(h)}^h \vee [\ 1\ ]_{H(H)}^h, \qquad I_h^H = \left[\begin{array}{ccc} \frac{1}{2} & 1 & \frac{1}{2} \end{array} \right]_{h(H)}^H.$$

As commented earlier, nested finite element discretizations automatically satisfy the Galerkin conditions, if the prolongation operator corresponds to the natural inclusion operation in the finite element space; linear interpolation of the grid function representation of a finite element function corresponds to the inclusion for piecewise linear basis functions. Therefore, with the above choice of stencils, the stencil calculus should reproduce (A.2) as the Galerkin matrix stencil on the coarse mesh:

$$A_H = \left[\begin{array}{ccc} -1 & 2 & -1 \end{array} \right]_H^H.$$

This is easily verified using the above expressions for the stencil entries.

A.1.2 Two dimensions

The two-dimensional stencil calculus proceeds exactly as the one-dimensional case, although grid functions are now two-dimensional. For example, the unit vector e_H has the *grid function* representation on a logically non-uniform Cartesian mesh as follows:

$$e_H = \begin{array}{ccccc} & \vdots & \vdots & \vdots & \\ \cdots & 0 & 0 & 0 & \cdots \\ \cdots & 0 & 1 & 0 & \cdots \\ \cdots & 0 & 0 & 0 & \cdots \\ & \vdots & \vdots & \vdots & \end{array},$$

where we take the horizontal direction as the "x-direction", and the vertical direction as the "y-direction". Stencils will operator on these two-dimensional grid functions. The prolongation operator must now handle four special cases with the standard nested non-uniform Cartesian mesh:

(1) Fine points coincident with coarse mesh points.
(2) Fine points lying on a coarse mesh x-line but not of Type (1).
(3) Fine points lying on a coarse mesh y-line but not of Type (1).
(4) Fine points points not on a coarse mesh point or line.

By using subscripts $H(i)$ to indicate how the prolongation operator acts on the mesh point Type (i), we can represent the prolongation operator for handling these special cases using a notion similar to the one-dimensional case.

Now, recall that a box-method, finite difference method, or finite element method (either linear basis functions over triangles or bilinear basis functions over rectangles) discretization of a second order elliptic partial differential equation over a non-uniform Cartesian in two-dimensions yields either a five-, seven-, or nine-point stencil, all of which can be considered to be special cases of a general nine-point stencil.

This general discretized differential operator, along with a general prolongation operator for a two-dimensional non-uniform Cartesian mesh and its corresponding restriction operator, can be represented in two-dimensional stencil form as:

$$A_h = \left[\begin{array}{ccc} -NW_{ij} & -N_{ij} & -NE_{ij} \\ -W_{ij} & C_{ij} & -E_{ij} \\ -SW_{ij} & -S_{ij} & -SE_{ij} \end{array} \right]_h^h, \quad I_h^H = \left[\begin{array}{ccc} PNW_{ij} & PN_{ij} & PNE_{ij} \\ PW_{ij} & PC_{ij} & PE_{ij} \\ PSW_{ij} & PS_{ij} & PSE_{ij} \end{array} \right]_{h(H)}^H,$$

$$I_H^h = \left[\begin{array}{ccc} PSE_{i-1,j+1} & 0 & PSW_{i+1,j+1} \\ 0 & 0 & 0 \\ PNE_{i-1,j-1} & 0 & PNW_{i+1,j-1} \end{array} \right]_{H(4)}^h \vee \left[\begin{array}{c} PS_{i,j+1} \\ 0 \\ PN_{i,j-1} \end{array} \right]_{H(3)}^h$$

$$\vee \left[\begin{array}{ccc} PE_{i-1,j} & 0 & PW_{i+1,j} \end{array} \right]_{H(2)}^h \vee \left[\begin{array}{c} PC_{ij} \end{array} \right]_{H(1)}^h.$$

It is easily verified by applying the prolongation stencil I_H^h above to the unit grid function e_H that the restriction operator above satisfies $I_h^H = (I_H^h)^T$. With these stencils, the resulting Galerkin coarse grid operator stencil has the form:

$$A_H = \left[\begin{array}{ccc} -NW_{ij}^H & -N_{ij}^H & -NE_{ij}^H \\ -W_{ij}^H & C_{ij}^H & -E_{ij}^H \\ -SW_{ij}^H & -S_{ij}^H & -SE_{ij}^H \end{array} \right]_H^H.$$

Using our MAPLE implementation of the two-dimensional stencil calculus, we can compute the expressions for the individual stencil components, which are as follows:

$$\begin{array}{lll}
C_{ij}^H & = +PSW_{ij} & (C_{i-1,j-1}PSW_{ij} - N_{i-1,j-1}PW_{ij} \\
 & & -E_{i-1,j-1}PS_{ij} - NE_{i-1,j-1}PC_{ij}) \\
 & +PW_{ij} & (-S_{i-1,j}PSW_{ij} + C_{i-1,j}PW_{ij} \\
 & & -N_{i-1,j}PNW_{ij} - SE_{i-1,j}PS_{ij} \\
 & & -E_{i-1,j}PC_{ij} - NE_{i-1,j}PN_{ij}) \\
 & +PNW_{ij} & (-S_{i-1,j+1}PW_{ij} + C_{i-1,j+1}PNW_{ij} \\
 & & -SE_{i-1,j+1}PC_{ij} - E_{i-1,j+1}PN_{ij}) \\
 & +PS_{ij} & (-W_{i,j-1}PSW_{ij} - NW_{i,j-1}PW_{ij} \\
 & & +C_{i,j-1}PS_{ij} - N_{i,j-1}PC_{ij} \\
 & & -E_{i,j-1}PSE_{ij} - NE_{i,j-1}PE_{ij}) \\
 & +PC_{ij} & (-SW_{ij}PSW_{ij} - W_{ij}PW_{ij} \\
 & & -NW_{ij}PNW_{ij} - S_{ij}PS_{ij} \\
 & & +C_{ij}PC_{ij} - N_{ij}PN_{ij} \\
 & & -SE_{ij}PSE_{ij} - E_{ij}PE_{ij} \\
 & & -NE_{ij}PNE_{ij}) \\
 & +PN_{ij} & (-SW_{i,j+1}PW_{ij} - W_{i,j+1}PNW_{ij} \\
 & & -S_{i,j+1}PC_{ij} + C_{i,j+1}PN_{ij} \\
 & & -SE_{i,j+1}PE_{ij} - E_{i,j+1}PNE_{ij})
\end{array}$$

$$
\begin{array}{lll}
 & +PSE_{ij} & (-W_{i+1,j-1}PS_{ij} - NW_{i+1,j-1}PC_{ij} \\
 & & +C_{i+1,j-1}PSE_{ij} - N_{i+1,j-1}PE_{ij}) \\
 & +PE_{ij} & (-SW_{i+1,j}PS_{ij} - W_{i+1,j}PC_{ij} \\
 & & -NW_{i+1,j}PN_{ij} - S_{i+1,j}PSE_{ij} \\
 & & +C_{i+1,j}PE_{ij} - N_{i+1,j}PNE_{ij}) \\
 & +PNE_{ij} & (-SW_{i+1,j+1}PC_{ij} - W_{i+1,j+1}PN_{ij} \\
 & & -S_{i+1,j+1}PE_{ij} + C_{i+1,j+1}PNE_{ij})
\end{array}
$$

$$
\begin{array}{lll}
-N_{ij}^{H} & = +PW_{ij} & (-N_{i-1,j}PSW_{i,j+2} - NE_{i-1,j}PS_{i,j+2}) \\
 & +PNW_{ij} & (C_{i-1,j+1}PSW_{i,j+2} - N_{i-1,j+1}PW_{i,j+2} \\
 & & -E_{i-1,j+1}PS_{i,j+2} - NE_{i-1,j+1}PC_{i,j+2}) \\
 & +PC_{ij} & (-NW_{ij}PSW_{i,j+2} - N_{ij}PS_{i,j+2} \\
 & & -NE_{ij}PSE_{i,j+2}) \\
 & +PN_{ij} & (-W_{i,j+1}PSW_{i,j+2} - NW_{i,j+1}PW_{i,j+2} \\
 & & +C_{i,j+1}PS_{i,j+2} - N_{i,j+1}PC_{i,j+2} \\
 & & -E_{i,j+1}PSE_{i,j+2} - NE_{i,j+1}PE_{i,j+2}) \\
 & +PE_{ij} & (-NW_{i+1,j}PS_{i,j+2} - N_{i+1,j}PSE_{i,j+2}) \\
 & +PNE_{ij} & (-W_{i+1,j+1}PS_{i,j+2} - NW_{i+1,j+1}PC_{i,j+2} \\
 & & +C_{i+1,j+1}PSE_{i,j+2} - N_{i+1,j+1}PE_{i,j+2})
\end{array}
$$

$$
\begin{array}{lll}
-S_{ij}^{H} & = +PSW_{ij} & (-S_{i-1,j-1}PW_{i,j-2} + C_{i-1,j-1}PNW_{i,j-2} \\
 & & -SE_{i-1,j-1}PC_{i,j-2} - E_{i-1,j-1}PN_{i,j-2}) \\
 & +PW_{ij} & (-S_{i-1,j}PNW_{i,j-2} - SE_{i-1,j}PN_{i,j-2}) \\
 & +PS_{ij} & (-SW_{i,j-1}PW_{i,j-2} - W_{i,j-1}PNW_{i,j-2} \\
 & & -S_{i,j-1}PC_{i,j-2} + C_{i,j-1}PN_{i,j-2} \\
 & & -SE_{i,j-1}PE_{i,j-2} - E_{i,j-1}PNE_{i,j-2}) \\
 & +PC_{ij} & (-SW_{ij}PNW_{i,j-2} - S_{ij}PN_{i,j-2} \\
 & & -SE_{ij}PNE_{i,j-2}) \\
 & +PSE_{ij} & (-SW_{i+1,j-1}PC_{i,j-2} - W_{i+1,j-1}PN_{i,j-2} \\
 & & -S_{i+1,j-1}PE_{i,j-2} + C_{i+1,j-1}PNE_{i,j-2}) \\
 & +PE_{ij} & (-SW_{i+1,j}PN_{i,j-2} - S_{i+1,j}PNE_{i,j-2})
\end{array}
$$

$$
\begin{array}{lll}
-E_{ij}^{H} & = +PS_{ij} & (-E_{i,j-1}PSW_{i+2,j} - NE_{i,j-1}PW_{i+2,j}) \\
 & +PC_{ij} & (-SE_{ij}PSW_{i+2,j} - E_{ij}PW_{i+2,j} \\
 & & -NE_{ij}PNW_{i+2,j}) \\
 & +PN_{ij} & (-SE_{i,j+1}PW_{i+2,j} - E_{i,j+1}PNW_{i+2,j}) \\
 & +PSE_{ij} & (C_{i+1,j-1}PSW_{i+2,j} - N_{i+1,j-1}PW_{i+2,j} \\
 & & -E_{i+1,j-1}PS_{i+2,j} - NE_{i+1,j-1}PC_{i+2,j}) \\
 & +PE_{ij} & (-S_{i+1,j}PSW_{i+2,j} + C_{i+1,j}PW_{i+2,j} \\
 & & -N_{i+1,j}PNW_{i+2,j} - SE_{i+1,j}PS_{i+2,j} \\
 & & -E_{i+1,j}PC_{i+2,j} - NE_{i+1,j}PN_{i+2,j}) \\
 & +PNE_{ij} & (-S_{i+1,j+1}PW_{i+2,j} + C_{i+1,j+1}PNW_{i+2,j} \\
 & & -SE_{i+1,j+1}PC_{i+2,j} - E_{i+1,j+1}PN_{i+2,j})
\end{array}
$$

$$
\begin{array}{lll}
-W_{ij}^{H} & = +PSW_{ij} & (-W_{i-1,j-1}PS_{i-2,j} - NW_{i-1,j-1}PC_{i-2,j} \\
 & & +C_{i-1,j-1}PSE_{i-2,j} - N_{i-1,j-1}PE_{i-2,j}) \\
 & +PW_{ij} & (-SW_{i-1,j}PS_{i-2,j} - W_{i-1,j}PC_{i-2,j} \\
 & & -NW_{i-1,j}PN_{i-2,j} - S_{i-1,j}PSE_{i-2,j} \\
 & & +C_{i-1,j}PE_{i-2,j} - N_{i-1,j}PNE_{i-2,j}) \\
 & +PNW_{ij} & (-SW_{i-1,j+1}PC_{i-2,j} - W_{i-1,j+1}PN_{i-2,j} \\
 & & -S_{i-1,j+1}PE_{i-2,j} + C_{i-1,j+1}PNE_{i-2,j}) \\
 & +PS_{ij} & (-W_{i,j-1}PSE_{i-2,j} - NW_{i,j-1}PE_{i-2,j}) \\
 & +PC_{ij} & (-SW_{ij}PSE_{i-2,j} - W_{ij}PE_{i-2,j} \\
 & & -NW_{ij}PNE_{i-2,j}) \\
 & +PN_{ij} & (-SW_{i,j+1}PE_{i-2,j} - W_{i,j+1}PNE_{i-2,j})
\end{array}
$$

$$
\begin{array}{lll}
-NE_{ij}^{H} & = -PC_{ij} & NE_{ij}PSW_{i+2,j+2} \\
 & +PN_{ij} & (-E_{i,j+1}PSW_{i+2,j+2} - NE_{i,j+1}PW_{i+2,j+2}) \\
 & +PE_{ij} & (-N_{i+1,j}PSW_{i+2,j+2} - NE_{i+1,j}PS_{i+2,j+2}) \\
 & +PNE_{ij} & (C_{i+1,j+1}PSW_{i+2,j+2} - N_{i+1,j+1}PW_{i+2,j+2} \\
 & & -E_{i+1,j+1}PS_{i+2,j+2} - NE_{i+1,j+1}PC_{i+2,j+2})
\end{array}
$$

$$\begin{array}{rll}
-NW_{ij}^{H} & = +PW_{ij} & (-NW_{i-1,j}PS_{i-2,j+2} - N_{i-1,j}PSE_{i-2,j+2}) \\
 & +PNW_{ij} & (-W_{i-1,j+1}PS_{i-2,j+2} - NW_{i-1,j+1}PC_{i-2,j+2} \\
 & & +C_{i-1,j+1}PSE_{i-2,j+2} - N_{i-1,j+1}PE_{i-2,j+2}) \\
 & -PC_{ij} & NW_{ij}PSE_{i-2,j+2} \\
 & +PN_{ij} & (-W_{i,j+1}PSE_{i-2,j+2} - NW_{i,j+1}PE_{i-2,j+2}) \\[1ex]
-SE_{ij}^{H} & = +PS_{ij} & (-SE_{i,j-1}PW_{i+2,j-2} - E_{i,j-1}PNW_{i+2,j-2}) \\
 & -PC_{ij} & SE_{i,j}PNW_{i+2,j-2} \\
 & +PSE_{ij} & (-S_{i+1,j-1}PW_{i+2,j-2} + C_{i+1,j-1}PNW_{i+2,j-2} \\
 & & -SE_{i+1,j-1}PC_{i+2,j-2} - E_{i+1,j-1}PN_{i+2,j-2}) \\
 & +PE_{ij} & (-S_{i+1,j}PNW_{i+2,j-2} - SE_{i+1,j}PN_{i+2,j-2}) \\[1ex]
-SW_{ij}^{H} & = +PW_{ij} & (-NW_{i-1,j}PS_{i-2,j+2} - N_{i-1,j}PSE_{i-2,j+2}) \\
 & +PNW_{ij} & (-W_{i-1,j+1}PS_{i-2,j+2} - NW_{i-1,j+1}PC_{i-2,j+2} \\
 & & +C_{i-1,j+1}PSE_{i-2,j+2} - N_{i-1,j+1}PE_{i-2,j+2}) \\
 & -PC_{ij} & NW_{ij}PSE_{i-2,j+2} \\
 & +PN_{ij} & (-W_{i,j+1}PSE_{i-2,j+2} - NW_{i,j+1}PE_{i-2,j+2})
\end{array}$$

Remark A.2. In the case that the original matrix A_h is symmetric, then the Galerkin coarse space matrix $A_H = (I_H^h)^T A_h I_H^h$ is also clearly symmetric. In this case, only the center entry and four of the remaining eight stencil entries need be computed and stored. This is a result of the following stencil symmetries, representing the symmetry of the corresponding matrices, which hold for both the original fine grid matrix stencil and the Galerkin coarse grid matrix stencil:

$$S_{ij} = N_{i,j-1}, \quad W_{ij} = E_{i-1,j}, \quad SE_{ij} = NW_{i+1,j-1}, \quad SW_{ij} = NE_{i-1,j-1}.$$

For example, these stencil symmetries can be used to eliminate the entries S_{ij}, W_{ij}, SE_{ij}, and SW_{ij} from the above expressions for the Galerkin matrix stencil entries, and only the resulting stencil component expressions for C_{ij}^H, N_{ij}^H, E_{ij}^H, NW_{ij}^H, NE_{ij}^H need be computed and stored.

Recall the well-known fact that a box-method, finite difference method, or finite element method (piecewise linear on triangles) discretization of Poisson's equation on a uniform two-dimensional Cartesian mesh placed on the unit square will yield the following stencil representation of the system matrix:

$$A_h = \left[\begin{array}{rrr} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{array} \right]_h^h, \tag{A.3}$$

where the meshwidth h has been divided out of the matrix. If the triangles are formed from the cubes which make up the uniform Cartesian mesh by connecting the lower-left vertex to the upper-right, then linear interpolation and the corresponding restriction on this uniform Cartesian mesh of triangles are represented by the stencils:

$$I_H^h = \left[\begin{array}{ccc} 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 \end{array} \right]_{H(4)}^h \vee \left[\begin{array}{c} \frac{1}{2} \\ 0 \\ \frac{1}{2} \end{array} \right]_{H(3)}^h \vee \left[\begin{array}{ccc} \frac{1}{2} & 0 & \frac{1}{2} \end{array} \right]_{H(2)}^h \vee \left[\, 1 \, \right]_{H(1)}^h, \quad I_h^H = \left[\begin{array}{ccc} 0 & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & 1 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & 0 \end{array} \right]_{h(H)}^H.$$

As commented earlier, nested finite element discretizations automatically satisfy the Galerkin conditions, if the prolongation operator corresponds to the natural inclusion operation in the finite element space; linear interpolation of the grid function representation of a finite element function corresponds to the inclusion for linear basis functions on triangles. Therefore, with the above choice of stencils, the stencil calculus should reproduce (A.3) as the resulting coarse grid Galerkin matrix stencil. This is easily verified (although now somewhat tedious in two dimensions) using the above expressions for the stencil components.

Consider now a finite element discretization of Poisson's equation on a uniform two-dimensional Cartesian mesh placed on the unit square, employing piecewise bilinear basis functions on rectangles. As is well-known, this discretization will yield the following stencil representation of the system matrix:

$$A_h = \frac{1}{3} \left[\begin{array}{rrr} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{array} \right]_h^h, \tag{A.4}$$

where the meshwidth h has been divided out of the matrix. Bilinear interpolation and the corresponding restriction in this case are represented by:

$$I_H^h = \begin{bmatrix} \frac{1}{4} & 0 & \frac{1}{4} \\ 0 & 0 & 0 \\ \frac{1}{4} & 0 & \frac{1}{4} \end{bmatrix}_{H(4)}^h \vee \begin{bmatrix} \frac{1}{2} \\ 0 \\ \frac{1}{2} \end{bmatrix}_{H(3)}^h \vee \begin{bmatrix} \frac{1}{2} & 0 & \frac{1}{2} \end{bmatrix}_{H(2)}^h \vee \begin{bmatrix} 1 \end{bmatrix}_{H(1)}^h, \quad I_h^H = \begin{bmatrix} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{2} & 1 & \frac{1}{2} \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \end{bmatrix}_{h(H)}^H .$$

Since the bilinear prolongation operator corresponds to the natural inclusion operation in the finite element space constructed from bilinear basis functions, with the above choice of stencils, the stencil calculus should reproduce (A.4) as the resulting coarse mesh Galerkin stencil; this is easily verified.

A.1.3 Three dimensions

The three-dimensional stencil calculus proceeds exactly as in the one- and two-dimensional cases, although grid functions are now three-dimensional. For example, the unit vector e_H has the *grid function* representation:

$$e_H = \begin{array}{ccccc} & & \text{(down}-\text{plane)} & & \\ & \vdots & \vdots & \vdots & \\ \cdots & 0 & 0 & 0 & \cdots \\ \cdots & 0 & 1 & 0 & \cdots \\ \cdots & 0 & 0 & 0 & \cdots \\ & \vdots & \vdots & \vdots & \end{array} \times \begin{array}{ccccc} & & \text{(level}-\text{plane)} & & \\ & \vdots & \vdots & \vdots & \\ \cdots & 0 & 0 & 0 & \cdots \\ \cdots & 0 & 1 & 0 & \cdots \\ \cdots & 0 & 0 & 0 & \cdots \\ & \vdots & \vdots & \vdots & \end{array} \times \begin{array}{ccccc} & & \text{(up}-\text{plane)} & & \\ & \vdots & \vdots & \vdots & \\ \cdots & 0 & 0 & 0 & \cdots \\ \cdots & 0 & 1 & 0 & \cdots \\ \cdots & 0 & 0 & 0 & \cdots \\ & \vdots & \vdots & \vdots & \end{array} .$$

Within each plane, the horizontal direction is taken as the "x-direction", and the vertical direction as the "y-direction"; the different planes then represent the "z-direction". Stencils will operate on these three-dimensional grid functions. The prolongation operator must now handle eight special cases with the standard nested non-uniform Cartesian mesh:

(1) Fine points coincident with coarse mesh points.
(2) Fine points lying on a coarse mesh x-line but not of Type (1).
(3) Fine points lying on a coarse mesh y-line but not of Types (1)–(2).
(4) Fine points lying on a coarse mesh z-line but not of Types (1)–(3).
(5) Fine points lying on a coarse mesh xy-plane but not of Types (1)–(4).
(6) Fine points lying on a coarse mesh yz-plane but not of Types (1)–(5).
(7) Fine points lying on a coarse mesh xz-plane but not of Types (1)–(6).
(8) Fine points points not on a coarse mesh point, line, or plane.

By using subscripts $H(i)$ to indicate how the prolongation operator acts on the mesh point Type (i), we can represent the prolongation operator for handling these special cases using a notion similar to the one- and two-dimensional cases.

Now, recall that a box-method, finite difference method, or finite element method (trilinear basis functions over three-dimensional boxes) discretization of a second order elliptic partial differential equation over a non-uniform Cartesian product mesh in three-dimensions yields either a seven- or twenty-seven-point stencil, both of which can be considered to be special cases of a general twenty-seven-point stencil.

This general discretized differential operator, along with a general prolongation operator for a three-dimensional non-uniform Cartesian mesh and its corresponding restriction operator, can be represented in three-dimensional stencil form as follows:

$$A_h = \begin{bmatrix} -dNW_{ij} & -dN_{ij} & -dNE_{ij} \\ -dW_{ij} & -dC_{ij} & -dE_{ij} \\ -dSW_{ij} & -dS_{ij} & -dSE_{ij} \end{bmatrix} \times \begin{bmatrix} -oNW_{ij} & -oN_{ij} & -oNE_{ij} \\ -oW_{ij} & oC_{ij} & -oE_{ij} \\ -oSW_{ij} & -oS_{ij} & -oSE_{ij} \end{bmatrix}$$

$$\times \begin{bmatrix} -uNW_{ij} & -uN_{ij} & -uNE_{ij} \\ -uW_{ij} & -uC_{ij} & -uE_{ij} \\ -uSW_{ij} & -uS_{ij} & -uSE_{ij} \end{bmatrix}_h^h ,$$

$$I_h^H = \left[\begin{array}{ccc} dPNW_{ij} & dPN_{ij} & dPNE_{ij} \\ dPW_{ij} & dPC_{ij} & dPE_{ij} \\ dPSW_{ij} & dPS_{ij} & dPSE_{ij} \end{array}\right] \times \left[\begin{array}{ccc} oPNW_{ij} & oPN_{ij} & oPNE_{ij} \\ oPW_{ij} & oPC_{ij} & oPE_{ij} \\ oPSW_{ij} & oPS_{ij} & oPSE_{ij} \end{array}\right]$$

$$\times \left[\begin{array}{ccc} uPNW_{ij} & uPN_{ij} & uPNE_{ij} \\ uPW_{ij} & uPC_{ij} & uPE_{ij} \\ uPSW_{ij} & uPS_{ij} & uPSE_{ij} \end{array}\right]_{h(H)}^{H},$$

$$I_H^h = \left[\begin{array}{ccc} uPSE_{i-1,j+1,k-1} & 0 & uPSW_{i+1,j+1,k-1} \\ 0 & 0 & 0 \\ uPNE_{i-1,j-1,k-1} & 0 & uPNW_{i+1,j-1,k-1} \end{array}\right] \times \left[\begin{array}{ccc} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{array}\right]$$

$$\times \left[\begin{array}{ccc} dPSE_{i-1,j+1,k+1} & 0 & dPSW_{i+1,j+1,k+1} \\ 0 & 0 & 0 \\ dPNE_{i-1,j-1,k+1} & 0 & dPNW_{i+1,j-1,k+1} \end{array}\right]_{H(8)}^{h} \vee \left[\begin{array}{ccc} dPN_{i,j-1,k+1} & 0 & dPS_{i,j+1,k+1} \\ 0 & 0 & 0 \\ uPN_{i,j-1,k-1} & 0 & uPS_{i,j+1,k-1} \end{array}\right]_{H(7)}^{h}$$

$$\vee \left[\begin{array}{ccc} dPE_{i-1,j,k+1} & 0 & dPW_{i+1,j,k+1} \\ 0 & 0 & 0 \\ uPE_{i-1,j,k-1} & 0 & uPW_{i+1,j,k-1} \end{array}\right]_{H(6)}^{h} \vee \left[\begin{array}{ccc} oPSE_{i-1,j+1,k} & 0 & oPSW_{i+1,j+1,k} \\ 0 & 0 & 0 \\ oPNE_{i-1,j-1,k} & 0 & oPNW_{i+1,j-1,k} \end{array}\right]_{H(5)}^{h}$$

$$\vee \left[\begin{array}{c} dPC_{i,j,k+1} \\ 0 \\ uPC_{i,j,k-1} \end{array}\right]_{H(4)}^{h} \vee \left[\begin{array}{c} oPS_{i,j+1,k} \\ 0 \\ oPN_{i,j-1,k} \end{array}\right]_{H(3)}^{h} \vee \left[\begin{array}{c} oPE_{i-1,j,k} \\ 0 \\ oPW_{i+1,j,k} \end{array}\right]_{H(2)}^{h} \vee \left[\ oPC_{ij}\ \right]_{H(1)}^{h}.$$

The multiple "crossed" stencils are interpreted as the first stencil operating on the "down" plane, the middle stencil operating on the "level" plane, and the third stencil operating on the "up" plane. Note that restriction operator above satisfies $I_h^H = (I_H^h)^T$; this is verified by applying the prolongation operator stencil to the unit grid function, which produces a grid function corresponding to the restriction operator stencil.

In the prolongation operator definition, although there is no simple way to represent which of the three coordinate directions the one-dimensional stencils act, or which two of the three coordinate directions the two-dimensional stencils act, this should be clear from the indices of the stencil components (which is one reason we chose to represent the prolongation operator stencil using this index scheme, in addition to the fact that the corresponding restriction operator stencil has the very simple form above).

With these stencils, the Galerkin coarse grid operator stencil has the form:

$$A_H = \left[\begin{array}{ccc} -dNW_{ij}^H & -dN_{ij}^H & -dNE_{ij}^H \\ -dW_{ij}^H & -dC_{ij}^H & -dE_{ij}^H \\ -dSW_{ij}^H & -dS_{ij}^H & -dSE_{ij}^H \end{array}\right] \times \left[\begin{array}{ccc} -oNW_{ij}^H & -oN_{ij}^H & -oNE_{ij}^H \\ -oW_{ij}^H & oC_{ij}^H & -oE_{ij}^H \\ -oSW_{ij}^H & -oS_{ij}^H & -oSE_{ij}^H \end{array}\right]$$

$$\times \left[\begin{array}{ccc} -uNW_{ij}^H & -uN_{ij}^H & -uNE_{ij}^H \\ -uW_{ij}^H & -uC_{ij}^H & -uE_{ij}^H \\ -uSW_{ij}^H & -uS_{ij}^H & -uSE_{ij}^H \end{array}\right]_{H}^{H}.$$

Before we present the expressions for the Galerkin matrix stencil components, we wish to make some simplifying assumptions to keep the length of this Appendix reasonable. In the case that the original matrix A_h is symmetric, then the Galerkin matrix $A_H = (I_H^h)^T A_h I_H^h$ is also clearly symmetric. In this case, only the center entry and thirteen of the remaining twenty-six stencil entries need be computed and stored. This is a result of the following stencil symmetries, representing the symmetry of the corresponding matrices, which hold for both the original fine grid matrix stencil and the Galerkin coarse grid matrix stencil:

$$S_{ijk} = N_{i,j-1,k}, \qquad W_{ijk} = E_{i-1,j,k}, \qquad SE_{ijk} = NW_{i+1,j-1,k}, \qquad SW_{ijk} = NE_{i-1,j-1,k},$$

$$dC_{ijk} = uC_{i,j,k-1}, \qquad dW_{ijk} = uE_{i-1,j,k-1}, \qquad dE_{ijk} = uW_{i+1,j,k-1},$$

$$dN_{ijk} = uS_{i,j+1,k-1}, \qquad dNW_{ijk} = uSE_{i-1,j+1,k-1}, \qquad dNE_{ijk} = uSW_{i+1,j+1,k-1},$$

$$dS_{ijk} = uN_{i,j-1,k-1}, \qquad dSW_{ijk} = uNE_{i-1,j-1,k-1}, \qquad dSE_{ijk} = uNW_{i+1,j-1,k-1}.$$

We now employ these stencil symmetries to eliminate the entries on the left of each equality, and only the resulting stencil component expressions for C_{ij}^H and the Galerkin coarse matrix equivalents of the thirteen components on the right of each equality above need be computed and stored. Using our MAPLE implementation of the three-dimensional stencil calculus, we can compute the expressions for the center component and the remaining thirteen required components for the symmetric case. To keep the appendix a reasonable length, we will present only four representative components:

$$oC_{ijk}^H, \qquad oE_{ijk}^H, \qquad oNE_{ijk}^H, \qquad uNE_{ijk}^H.$$

The remaining ten components have similar forms to the above four; the components oN_{ijk}^H and uC_{ijk}^H are of the same form as oE_{ijk}^H, whereas oNW_{ijk}^H, uE_{ijk}^H, uW_{ijk}^H, uN_{ijk}^H, and uS_{ijk}^H have the same form as oNE_{ijk}^H, and the remaining components uNW_{ijk}^H, uSE_{ijk}^H, and uSW_{ijk}^H have the same form as uNE_{ijk}^H.

$$
\begin{array}{lll}
oC_{ijk}^H & = +oPN_{ijk} & (-uNE_{i-1,j,k-1}dPW_{ijk} \\
& & -oNE_{i-1,j,k}oPW_{ijk} - uSW_{i,j+1,k}uPW_{ijk} \\
& & -uE_{i-1,j+1,k-1}dPNW_{ijk} - oE_{i-1,j+1,k}oPNW_{ijk} \\
& & -uW_{i,j+1,k}uPNW_{ijk} - uN_{i,j,k-1}dPC_{ijk} \\
& & -oN_{i,j,k}oPC_{ijk} - uS_{i,j+1,k}uPC_{ijk} \\
& & -uC_{i,j+1,k-1}dPN_{ijk} + oC_{i,j+1,k}oPN_{ijk} \\
& & -uC_{i,j+1,k}uPN_{ijk} - uNW_{i+1,j,k-1}dPE_{ijk} \\
& & -oNW_{i+1,j,k}oPE_{ijk} - uSE_{i,j+1,k}uPE_{ijk} \\
& & -uW_{i+1,j+1,k-1}dPNE_{ijk} - oE_{i,j+1,k}oPNE_{ijk} \\
& & -uE_{i,j+1,k}uPNE_{ijk}) \\
& +dPN_{ijk} & (-oNE_{i-1,j,k-1}dPW_{ijk} - uSW_{i,j+1,k-1}oPW_{ijk} \\
& & -oE_{i-1,j+1,k-1}dPNW_{ijk} - uW_{i,j+1,k-1}oPNW_{ijk} \\
& & -oN_{i,j,k-1}dPC_{ijk} - uS_{i,j+1,k-1}oPC_{ijk} \\
& & +oC_{i,j+1,k-1}dPN_{ijk} - uC_{i,j+1,k-1}oPN_{ijk} \\
& & -oNW_{i+1,j,k-1}dPE_{ijk} - uSE_{i,j+1,k-1}oPE_{ijk} \\
& & -oE_{i,j+1,k-1}dPNE_{ijk} - uE_{i,j+1,k-1}oPNE_{ijk}) \\
& +dPC_{ijk} & (-oNE_{i-1,j-1,k-1}dPSW_{ijk} - uSW_{i,j,k-1}oPSW_{ijk} \\
& & -oE_{i-1,j,k-1}dPW_{ijk} - uW_{i,j,k-1}oPW_{ijk} \\
& & -oNW_{i,j,k-1}dPNW_{ijk} - uNW_{i,j,k-1}oPNW_{ijk} \\
& & -oN_{i,j-1,k-1}dPS_{ijk} - uS_{i,j,k-1}oPS_{ijk} \\
& & +oC_{i,j,k-1}dPC_{ijk} - uC_{i,j,k-1}oPC_{ijk} \\
& & -oN_{i,j,k-1}dPN_{ijk} - uN_{i,j,k-1}oPN_{ijk} \\
& & -oNW_{i+1,j-1,k-1}dPSE_{ijk} - uSE_{i,j,k-1}oPSE_{ijk} \\
& & -oE_{i,j,k-1}dPE_{ijk} - uE_{i,j,k-1}oPE_{ijk} \\
& & -oNE_{i,j,k-1}dPNE_{ijk} - uNE_{i,j,k-1}oPNE_{ijk}) \\
& +uPC_{ijk} & (-uNE_{i-1,j-1,k}oPSW_{ijk} - oNE_{i-1,j-1,k+1}uPSW_{ijk} \\
& & -uE_{i-1,j,k}oPW_{ijk} - oE_{i-1,j,k+1}uPW_{ijk} \\
& & -uSE_{i-1,j+1,k}oPNW_{ijk} - oNW_{i,j,k+1}uPNW_{ijk} \\
& & -uN_{i,j-1,k}oPS_{ijk} - oN_{i,j-1,k+1}uPS_{ijk} \\
& & -uC_{i,j,k}oPC_{ijk} + oC_{i,j,k+1}uPC_{ijk} \\
& & -uS_{i,j+1,k}oPN_{ijk} - oN_{i,j,k+1}uPN_{ijk} \\
& & -uNW_{i+1,j-1,k}oPSE_{ijk} - oNW_{i+1,j-1,k+1}uPSE_{ijk} \\
& & -uW_{i+1,j,k}oPE_{ijk} - oE_{i,j,k+1}uPE_{ijk} \\
& & -uSW_{i+1,j+1,k}oPNE_{ijk} - oNE_{i,j,k+1}uPNE_{ijk}) \\
& +oPC_{ijk} & (-uW_{i+1,j,k-1}dPE_{ijk} - oE_{i-1,j,k}oPW_{ijk} \\
& & -uSE_{i-1,j+1,k-1}dPNW_{ijk} - uNE_{i-1,j-1,k-1}dPSW_{ijk} \\
& & -uN_{i,j-1,k-1}dPS_{ijk} - oNE_{i-1,j-1,k}oPSW_{ijk} \\
& & -uE_{i-1,j,k-1}dPW_{ijk} - oNW_{i+1,j-1,k}oPSE_{ijk} \\
& & -uC_{i,j,k-1}dPC_{ijk} - uNW_{i+1,j-1,k-1}dPSE_{ijk} \\
& & -uSW_{i+1,j+1,k-1}dPNE_{ijk} - uS_{i,j+1,k-1}dPN_{ijk} \\
& & -oN_{i,j-1,k}oPS_{ijk} - uNE_{i,j,k}uPNE_{ijk} \\
& & -oNE_{i,j,k}oPNE_{ijk} - uE_{i,j,k}uPE_{ijk} \\
& & -uSE_{i,j,k}uPSE_{ijk} - oN_{i,j,k}oPN_{ijk} \\
& & -oE_{i,j,k}oPE_{ijk} - uS_{i,j,k}uPS_{ijk} \\
& & +oC_{i,j,k}oPC_{ijk} - uSW_{i,j,k}uPSW_{ijk} \\
& & -uN_{i,j,k}uPN_{ijk} - uC_{i,j,k}uPC_{ijk} \\
& & -uW_{i,j,k}uPW_{ijk} - oNW_{i,j,k}oPNW_{ijk} \\
& & -uNW_{i,j,k}uPNW_{ijk}) \\
& +uPS_{ijk} & (-uE_{i-1,j-1,k}oPSW_{ijk} - oE_{i-1,j-1,k+1}uPSW_{ijk}
\end{array}
$$

$$
\begin{aligned}
& -uSE_{i-1,j,k}oPW_{ijk} - oNW_{i,j-1,k+1}uPW_{ijk} \\
& -uC_{i,j-1,k}oPS_{ijk} + oC_{i,j-1,k+1}uPS_{ijk} \\
& -uS_{i,j,k}oPC_{ijk} - oN_{i,j-1,k+1}uPC_{ijk} \\
& -uW_{i+1,j-1,k}oPSE_{ijk} - oE_{i,j-1,k+1}uPSE_{ijk} \\
& -uSW_{i+1,j,k}oPE_{ijk} - oNE_{i,j-1,k+1}uPE_{ijk}) \\
+oPS_{ijk} \quad & (-uE_{i-1,j-1,k-1}dPSW_{ijk} - oE_{i-1,j-1,k}oPSW_{ijk} \\
& -uW_{i,j-1,k}uPSW_{ijk} - uSE_{i-1,j,k-1}dPW_{ijk} \\
& -oNW_{i,j-1,k}oPW_{ijk} - uNW_{i,j-1,k}uPW_{ijk} \\
& -uC_{i,j-1,k-1}dPS_{ijk} + oC_{i,j-1,k}oPS_{ijk} \\
& -uC_{i,j-1,k}uPS_{ijk} - uS_{i,j,k-1}dPC_{ijk} \\
& -oN_{i,j-1,k}oPC_{ijk} - uN_{i,j-1,k}uPC_{ijk} \\
& -uW_{i+1,j-1,k-1}dPSE_{ijk} - oE_{i,j-1,k}oPSE_{ijk} \\
& -uE_{i,j-1,k}uPSE_{ijk} - uSW_{i+1,j,k-1}dPE_{ijk} \\
& -oNE_{i,j-1,k}oPE_{ijk} - uNE_{i,j-1,k}uPE_{ijk}) \\
+dPS_{ijk} \quad & (-oE_{i-1,j-1,k-1}dPSW_{ijk} - uW_{i,j-1,k-1}oPSW_{ijk} \\
& -oNW_{i,j-1,k-1}dPW_{ijk} - uNW_{i,j-1,k-1}oPW_{ijk} \\
& +oC_{i,j-1,k-1}dPS_{ijk} - uC_{i,j-1,k-1}oPS_{ijk} \\
& -oN_{i,j-1,k-1}dPC_{ijk} - uN_{i,j-1,k-1}oPC_{ijk} \\
& -oE_{i,j-1,k-1}dPSE_{ijk} - uE_{i,j-1,k-1}oPSE_{ijk} \\
& -oNE_{i,j-1,k-1}dPE_{ijk} - uNE_{i,j-1,k-1}oPE_{ijk}) \\
+uPNW_{ijk} \quad & (-uN_{i-1,j,k}oPW_{ijk} - oN_{i-1,j,k+1}uPW_{ijk} \\
& -uC_{i-1,j+1,k}oPNW_{ijk} + oC_{i-1,j+1,k+1}uPNW_{ijk} \\
& -uNW_{i,j,k}oPC_{ijk} - oNW_{i,j,k+1}uPC_{ijk} \\
& -uW_{i,j+1,k}oPN_{ijk} - oE_{i-1,j+1,k+1}uPN_{ijk}) \\
+oPNW_{ijk} \quad & (-uN_{i-1,j,k-1}dPW_{ijk} - oN_{i-1,j,k}oPW_{ijk} \\
& -uS_{i-1,j+1,k}uPW_{ijk} - uC_{i-1,j+1,k-1}dPNW_{ijk} \\
& +oC_{i-1,j+1,k}oPNW_{ijk} - uC_{i-1,j+1,k}uPNW_{ijk} \\
& -uNW_{i,j,k-1}dPC_{ijk} - oNW_{i,j,k}oPC_{ijk} \\
& -uSE_{i-1,j+1,k}uPC_{ijk} - uW_{i,j+1,k-1}dPN_{ijk} \\
& -oE_{i-1,j+1,k}oPN_{ijk} - uE_{i-1,j+1,k}uPN_{ijk}) \\
+uPW_{ijk} \quad & (-uN_{i-1,j-1,k}oPSW_{ijk} - oN_{i-1,j-1,k+1}uPSW_{ijk} \\
& -uC_{i-1,j,k}oPW_{ijk} + oC_{i-1,j,k+1}uPW_{ijk} \\
& -uS_{i-1,j+1,k}oPNW_{ijk} - oN_{i-1,j,k+1}uPNW_{ijk} \\
& -uNW_{i,j-1,k}oPS_{ijk} - oNW_{i,j-1,k+1}uPS_{ijk} \\
& -uW_{i,j,k}oPC_{ijk} - oE_{i-1,j,k+1}uPC_{ijk} \\
& -uSW_{i,j+1,k}oPN_{ijk} - oNE_{i-1,j,k+1}uPN_{ijk}) \\
+dPNW_{ijk} \quad & (-oN_{i-1,j,k-1}dPW_{ijk} - uS_{i-1,j+1,k-1}oPW_{ijk} \\
& +oC_{i-1,j+1,k-1}dPNW_{ijk} - uC_{i-1,j+1,k-1}oPNW_{ijk} \\
& -oNW_{i,j,k-1}dPC_{ijk} - uSE_{i-1,j+1,k-1}oPC_{ijk} \\
& -oE_{i-1,j+1,k-1}dPN_{ijk} - uE_{i-1,j+1,k-1}oPN_{ijk}) \\
+oPW_{ijk} \quad & (-uN_{i-1,j-1,k-1}dPSW_{ijk} - oN_{i-1,j-1,k}oPSW_{ijk} \\
& -uS_{i-1,j,k}uPSW_{ijk} - uC_{i-1,j,k-1}dPW_{ijk} \\
& +oC_{i-1,j,k}oPW_{ijk} - uC_{i-1,j,k}uPW_{ijk} \\
& -uS_{i-1,j+1,k-1}dPNW_{ijk} - oN_{i-1,j,k}oPNW_{ijk} \\
& -uN_{i-1,j,k}uPNW_{ijk} - uNW_{i,j-1,k-1}dPS_{ijk} \\
& -oNW_{i,j-1,k}oPS_{ijk} - uSE_{i-1,j,k}uPS_{ijk} \\
& -uW_{i,j,k-1}dPC_{ijk} - oE_{i-1,j,k}oPC_{ijk} \\
& -uE_{i-1,j,k}uPC_{ijk} - uSW_{i,j+1,k-1}dPN_{ijk} \\
& -oNE_{i-1,j,k}oPN_{ijk} - uNE_{i-1,j,k}uPN_{ijk}) \\
+uPSW_{ijk} \quad & (-uC_{i-1,j-1,k}oPSW_{ijk} + oC_{i-1,j-1,k+1}uPSW_{ijk} \\
& -uS_{i-1,j,k}oPW_{ijk} - oN_{i-1,j-1,k+1}uPW_{ijk} \\
& -uW_{i,j-1,k}oPS_{ijk} - oE_{i-1,j-1,k+1}uPS_{ijk} \\
& -uSW_{i,j,k}oPC_{ijk} - oNE_{i-1,j-1,k+1}uPC_{ijk}) \\
+oPSW_{ijk} \quad & (-uC_{i-1,j-1,k-1}dPSW_{ijk} + oC_{i-1,j-1,k}oPSW_{ijk} \\
& -uC_{i-1,j-1,k}uPSW_{ijk} - uS_{i-1,j,k-1}dPW_{ijk} \\
& -oN_{i-1,j-1,k}oPW_{ijk} - uN_{i-1,j-1,k}uPW_{ijk} \\
& -uW_{i,j-1,k-1}dPS_{ijk} - oE_{i-1,j-1,k}oPS_{ijk} \\
& -uE_{i-1,j-1,k}uPS_{ijk} - uSW_{i,j,k-1}dPC_{ijk} \\
& -oNE_{i-1,j-1,k}oPC_{ijk} - uNE_{i-1,j-1,k}uPC_{ijk}) \\
+dPW_{ijk} \quad & (-oN_{i-1,j-1,k-1}dPSW_{ijk} - uS_{i-1,j,k-1}oPSW_{ijk} \\
& +oC_{i-1,j,k-1}dPW_{ijk} - uC_{i-1,j,k-1}oPW_{ijk} \\
& -oN_{i-1,j,k-1}dPNW_{ijk} - uN_{i-1,j,k-1}oPNW_{ijk} \\
& -oNW_{i,j-1,k-1}dPS_{ijk} - uSE_{i-1,j,k-1}oPS_{ijk}
\end{aligned}
$$

$$
\begin{array}{ll}
 & -oE_{i-1,j,k-1}dPC_{ijk} - uE_{i-1,j,k-1}oPC_{ijk} \\
 & -oNE_{i-1,j,k-1}dPN_{ijk} - uNE_{i-1,j,k-1}oPN_{ijk}) \\
+uPNE_{ijk} & (-uNE_{i,j,k}oPC_{ijk} - oNE_{i,j,k+1}uPC_{ijk} \\
 & -uE_{i,j+1,k}oPN_{ijk} - oE_{i,j+1,k+1}uPN_{ijk} \\
 & -uN_{i+1,j,k}oPE_{ijk} - oN_{i+1,j,k+1}uPE_{ijk} \\
 & -uC_{i+1,j+1,k}oPNE_{ijk} + oC_{i+1,j+1,k+1}uPNE_{ijk}) \\
+uPE_{ijk} & (-uNE_{i,j-1,k}oPS_{ijk} - oNE_{i,j-1,k+1}uPS_{ijk} \\
 & -uE_{i,j,k}oPC_{ijk} - oE_{i,j,k+1}uPC_{ijk} \\
 & -uSE_{i,j+1,k}oPN_{ijk} - oNW_{i+1,j,k+1}uPN_{ijk} \\
 & -uN_{i+1,j-1,k}oPSE_{ijk} - oN_{i+1,j-1,k+1}uPSE_{ijk} \\
 & -uC_{i+1,j,k}oPE_{ijk} + oC_{i+1,j,k+1}uPE_{ijk} \\
 & -uS_{i+1,j+1,k}oPNE_{ijk} - oN_{i+1,j,k+1}uPNE_{ijk}) \\
+dPNE_{ijk} & (-oNE_{i,j,k-1}dPC_{ijk} - uSW_{i+1,j+1,k-1}oPC_{ijk} \\
 & -oE_{i,j+1,k-1}dPN_{ijk} - uW_{i+1,j+1,k-1}oPN_{ijk} \\
 & -oN_{i+1,j,k-1}dPE_{ijk} - uS_{i+1,j+1,k-1}oPE_{ijk} \\
 & +oC_{i+1,j+1,k-1}dPNE_{ijk} - uC_{i+1,j+1,k-1}oPNE_{ijk}) \\
+oPNE_{ijk} & (-uNE_{i,j,k-1}dPC_{ijk} - oNE_{i,j,k}oPC_{ijk} \\
 & -uSW_{i+1,j+1,k}uPC_{ijk} - uE_{i,j+1,k-1}dPN_{ijk} \\
 & -oE_{i,j+1,k}oPN_{ijk} - uW_{i+1,j+1,k}uPN_{ijk} \\
 & -uN_{i+1,j,k-1}dPE_{ijk} - oN_{i+1,j,k}oPE_{ijk} \\
 & -uS_{i+1,j+1,k}uPE_{ijk} - uC_{i+1,j+1,k-1}dPNE_{ijk} \\
 & +oC_{i+1,j+1,k}oPNE_{ijk} - uC_{i+1,j+1,k}uPNE_{ijk}) \\
+oPSE_{ijk} & (-uE_{i,j-1,k-1}dPS_{ijk} - oE_{i,j-1,k}oPS_{ijk} \\
 & -uW_{i+1,j-1,k}uPS_{ijk} - uSE_{i,j,k-1}dPC_{ijk} \\
 & -oNW_{i+1,j-1,k}oPC_{ijk} - uNW_{i+1,j-1,k}uPC_{ijk} \\
 & -uC_{i+1,j-1,k-1}dPSE_{ijk} + oC_{i+1,j-1,k}oPSE_{ijk} \\
 & -uC_{i+1,j-1,k}uPSE_{ijk} - uS_{i+1,j,k-1}dPE_{ijk} \\
 & -oN_{i+1,j-1,k}oPE_{ijk} - uN_{i+1,j-1,k}uPE_{ijk}) \\
+dPSE_{ijk} & (-oE_{i,j-1,k-1}dPS_{ijk} - uW_{i+1,j-1,k-1}oPS_{ijk} \\
 & -oNW_{i+1,j-1,k-1}dPC_{ijk} - uNW_{i+1,j-1,k-1}oPC_{ijk} \\
 & +oC_{i+1,j-1,k-1}dPSE_{ijk} - uC_{i+1,j-1,k-1}oPSE_{ijk} \\
 & -oN_{i+1,j-1,k-1}dPE_{ijk} - uN_{i+1,j-1,k-1}oPE_{ijk}) \\
+uPSE_{ijk} & (-uE_{i,j-1,k}oPS_{ijk} - oE_{i,j-1,k+1}uPS_{ijk} \\
 & -uSE_{i,j,k}oPC_{ijk} - oNW_{i+1,j-1,k+1}uPC_{ijk} \\
 & -uC_{i+1,j-1,k}oPSE_{ijk} + oC_{i+1,j-1,k+1}uPSE_{ijk} \\
 & -uS_{i+1,j,k}oPE_{ijk} - oN_{i+1,j-1,k+1}uPE_{ijk}) \\
+oPE_{ijk} & (-uNE_{i,j-1,k-1}dPS_{ijk} - oNE_{i,j-1,k}oPS_{ijk} \\
 & -uSW_{i+1,j,k}uPS_{ijk} - uE_{i,j,k-1}dPC_{ijk} \\
 & -oE_{i,j,k}oPC_{ijk} - uW_{i+1,j,k}uPC_{ijk} \\
 & -uSE_{i,j+1,k-1}dPN_{ijk} - oNW_{i+1,j,k}oPN_{ijk} \\
 & -uNW_{i+1,j,k}uPN_{ijk} - uN_{i+1,j-1,k-1}dPSE_{ijk} \\
 & -oN_{i+1,j-1,k}oPSE_{ijk} - uS_{i+1,j,k}uPSE_{ijk} \\
 & -uC_{i+1,j,k-1}dPE_{ijk} + oC_{i+1,j,k}oPE_{ijk} \\
 & -uC_{i+1,j,k}uPE_{ijk} - uS_{i+1,j+1,k-1}dPNE_{ijk} \\
 & -oN_{i+1,j,k}oPNE_{ijk} - uN_{i+1,j,k}uPNE_{ijk}) \\
+dPE_{ijk} & (-oNE_{i,j-1,k-1}dPS_{ijk} - uSW_{i+1,j,k-1}oPS_{ijk} \\
 & -oE_{i,j,k-1}dPC_{ijk} - uW_{i+1,j,k-1}oPC_{ijk} \\
 & -oNW_{i+1,j,k-1}dPN_{ijk} - uNW_{i+1,j,k-1}oPN_{ijk} \\
 & -oN_{i+1,j-1,k-1}dPSE_{ijk} - uS_{i+1,j,k-1}oPSE_{ijk} \\
 & +oC_{i+1,j,k-1}dPE_{ijk} - uC_{i+1,j,k-1}oPE_{ijk} \\
 & -oN_{i+1,j,k-1}dPNE_{ijk} - uN_{i+1,j,k-1}oPNE_{ijk}) \\
+uPN_{ijk} & (-uNE_{i-1,j,k}oPW_{ijk} - oNE_{i-1,j,k+1}uPW_{ijk} \\
 & -uE_{i-1,j+1,k}oPNW_{ijk} - oE_{i-1,j+1,k+1}uPNW_{ijk} \\
 & -uN_{i,j,k}oPC_{ijk} - oN_{i,j,k+1}uPC_{ijk} \\
 & -uC_{i,j+1,k}oPN_{ijk} + oC_{i,j+1,k+1}uPN_{ijk} \\
 & -uNW_{i+1,j,k}oPE_{ijk} - oNW_{i+1,j,k+1}uPE_{ijk} \\
 & -uW_{i+1,j+1,k}oPNE_{ijk} - oE_{i,j+1,k+1}uPNE_{ijk}) \\
+dPSW_{ijk} & (oC_{i-1,j-1,k-1}dPSW_{ijk} - uC_{i-1,j-1,k-1}oPSW_{ijk} \\
 & -oN_{i-1,j-1,k-1}dPW_{ijk} - uN_{i-1,j-1,k-1}oPW_{ijk} \\
 & -oE_{i-1,j-1,k-1}dPS_{ijk} - uE_{i-1,j-1,k-1}oPS_{ijk} \\
 & -oNE_{i-1,j-1,k-1}dPC_{ijk} - uNE_{i-1,j-1,k-1}oPC_{ijk})
\end{array}
$$

$$
-oE^{H}_{ijk} \quad = +dPS_{ijk} \quad (-oE_{i,j-1,k-1}dPSW_{i+2,j,k} - uE_{i,j-1,k-1}oPSW_{i+2,j,k}
$$

$$
\begin{aligned}
& -oNE_{i,j-1,k-1}dPW_{i+2,j,k} - uNE_{i,j-1,k-1}oPW_{i+2,j,k}) \\
+oPS_{ijk} \quad & (-uW_{i+1,j-1,k-1}dPSW_{i+2,j,k} - oE_{i,j-1,k}oPSW_{i+2,j,k} \\
& -uE_{i,j-1,k}uPSW_{i+2,j,k} - uSW_{i+1,j,k-1}dPW_{i+2,j,k} \\
& -oNE_{i,j-1,k}oPW_{i+2,j,k} - uNE_{i,j-1,k}uPW_{i+2,j,k}) \\
+uPS_{ijk} \quad & (-uW_{i+1,j-1,k}oPSW_{i+2,j,k} - oE_{i,j-1,k+1}uPSW_{i+2,j,k} \\
& -uSW_{i+1,j,k}oPW_{i+2,j,k} - oNE_{i,j-1,k+1}uPW_{i+2,j,k}) \\
+dPC_{ijk} \quad & (-oNW_{i+1,j-1,k-1}dPSW_{i+2,j,k} - uSE_{i,j,k-1}oPSW_{i+2,j,k} \\
& -oE_{i,j,k-1}dPW_{i+2,j,k} - uE_{i,j,k-1}oPW_{i+2,j,k} \\
& -oNE_{i,j,k-1}dPNW_{i+2,j,k} - uNE_{i,j,k-1}oPNW_{i+2,j,k}) \\
+oPC_{ijk} \quad & (-uNW_{i+1,j-1,k-1}dPSW_{i+2,j,k} - oNW_{i+1,j-1,k}oPSW_{i+2,j,k} \\
& -uSE_{i,j,k}uPSW_{i+2,j,k} - uW_{i+1,j,k-1}dPW_{i+2,j,k} \\
& -oE_{i,j,k}oPW_{i+2,j,k} - uE_{i,j,k}uPW_{i+2,j,k} \\
& -uSW_{i+1,j+1,k-1}dPNW_{i+2,j,k} - oNE_{i,j,k}oPNW_{i+2,j,k} \\
& -uNE_{i,j,k}uPNW_{i+2,j,k}) \\
+uPC_{ijk} \quad & (-uNW_{i+1,j-1,k}oPSW_{i+2,j,k} - oNW_{i+1,j-1,k+1}uPSW_{i+2,j,k} \\
& -uW_{i+1,j,k}oPW_{i+2,j,k} - oE_{i,j,k+1}uPW_{i+2,j,k} \\
& -uSW_{i+1,j+1,k}oPNW_{i+2,j,k} - oNE_{i,j,k+1}uPNW_{i+2,j,k}) \\
+dPN_{ijk} \quad & (-oNW_{i+1,j,k-1}dPW_{i+2,j,k} - uSE_{i,j+1,k-1}oPW_{i+2,j,k} \\
& -oE_{i,j+1,k-1}dPNW_{i+2,j,k} - uE_{i,j+1,k-1}oPNW_{i+2,j,k}) \\
+oPN_{ijk} \quad & (-uNW_{i+1,j,k-1}dPW_{i+2,j,k} - oNW_{i+1,j,k}oPW_{i+2,j,k} \\
& -uSE_{i,j+1,k}uPW_{i+2,j,k} - uW_{i+1,j+1,k-1}dPNW_{i+2,j,k} \\
& -oE_{i,j+1,k}oPNW_{i+2,j,k} - uE_{i,j+1,k}uPNW_{i+2,j,k}) \\
+uPN_{ijk} \quad & (-uNW_{i+1,j,k}oPW_{i+2,j,k} - oNW_{i+1,j,k+1}uPW_{i+2,j,k} \\
& -uW_{i+1,j+1,k}oPNW_{i+2,j,k} - oE_{i,j+1,k+1}uPNW_{i+2,j,k}) \\
+dPSE_{ijk} \quad & (oC_{i+1,j-1,k-1}dPSW_{i+2,j,k} - uC_{i+1,j-1,k-1}oPSW_{i+2,j,k} \\
& -oN_{i+1,j-1,k-1}dPW_{i+2,j,k} - uN_{i+1,j-1,k-1}oPW_{i+2,j,k} \\
& -oE_{i+1,j-1,k-1}dPS_{i+2,j,k} - uE_{i+1,j-1,k-1}oPS_{i+2,j,k} \\
& -oNE_{i+1,j-1,k-1}dPC_{i+2,j,k} - uNE_{i+1,j-1,k-1}oPC_{i+2,j,k}) \\
+oPSE_{ijk} \quad & (-uC_{i+1,j-1,k-1}dPSW_{i+2,j,k} + oC_{i+1,j-1,k}oPSW_{i+2,j,k} \\
& -uC_{i+1,j-1,k}uPSW_{i+2,j,k} - uS_{i+1,j,k-1}dPW_{i+2,j,k} \\
& -oN_{i+1,j-1,k}oPW_{i+2,j,k} - uN_{i+1,j-1,k}uPW_{i+2,j,k} \\
& -uW_{i+2,j-1,k-1}dPS_{i+2,j,k} - oE_{i+1,j-1,k}oPS_{i+2,j,k} \\
& -uE_{i+1,j-1,k}uPS_{i+2,j,k} - uSW_{i+2,j,k-1}dPC_{i+2,j,k} \\
& -oNE_{i+1,j-1,k}oPC_{i+2,j,k} - uNE_{i+1,j-1,k}uPC_{i+2,j,k}) \\
+uPSE_{ijk} \quad & (-uC_{i+1,j-1,k}oPSW_{i+2,j,k} + oC_{i+1,j-1,k+1}uPSW_{i+2,j,k} \\
& -uS_{i+1,j,k}oPW_{i+2,j,k} - oN_{i+1,j-1,k+1}uPW_{i+2,j,k} \\
& -uW_{i+2,j-1,k}oPS_{i+2,j,k} - oE_{i+1,j-1,k+1}uPS_{i+2,j,k} \\
& -uSW_{i+2,j,k}oPC_{i+2,j,k} - oNE_{i+1,j-1,k+1}uPC_{i+2,j,k}) \\
+dPE_{ijk} \quad & (-oN_{i+1,j-1,k-1}dPSW_{i+2,j,k} - uS_{i+1,j,k-1}oPSW_{i+2,j,k} \\
& +oC_{i+1,j,k-1}dPW_{i+2,j,k} - uC_{i+1,j,k-1}oPW_{i+2,j,k} \\
& -oN_{i+1,j,k-1}dPNW_{i+2,j,k} - uN_{i+1,j,k-1}oPNW_{i+2,j,k} \\
& -oNW_{i+2,j-1,k-1}dPS_{i+2,j,k} - uSE_{i+1,j,k-1}oPS_{i+2,j,k} \\
& -oE_{i+1,j,k-1}dPC_{i+2,j,k} - uE_{i+1,j,k-1}oPC_{i+2,j,k} \\
& -oNE_{i+1,j,k-1}dPN_{i+2,j,k} - uNE_{i+1,j,k-1}oPN_{i+2,j,k}) \\
+oPE_{ijk} \quad & (-uN_{i+1,j-1,k-1}dPSW_{i+2,j,k} - oN_{i+1,j-1,k}oPSW_{i+2,j,k} \\
& -uS_{i+1,j,k}uPSW_{i+2,j,k} - uC_{i+1,j,k-1}dPW_{i+2,j,k} \\
& +oC_{i+1,j,k}oPW_{i+2,j,k} - uC_{i+1,j,k}uPW_{i+2,j,k} \\
& -uS_{i+1,j+1,k-1}dPNW_{i+2,j,k} - oN_{i+1,j,k}oPNW_{i+2,j,k} \\
& -uN_{i+1,j,k}uPNW_{i+2,j,k} - uNW_{i+2,j-1,k-1}dPS_{i+2,j,k} \\
& -oNW_{i+2,j-1,k}oPS_{i+2,j,k} - uSE_{i+1,j,k}uPS_{i+2,j,k} \\
& -uW_{i+2,j,k-1}dPC_{i+2,j,k} - oE_{i+1,j,k}oPC_{i+2,j,k} \\
& -uE_{i+1,j,k}uPC_{i+2,j,k} - uSW_{i+2,j+1,k-1}dPN_{i+2,j,k} \\
& -oNE_{i+1,j,k}oPN_{i+2,j,k} - uNE_{i+1,j,k}uPN_{i+2,j,k}) \\
+uPE_{ijk} \quad & (-uN_{i+1,j-1,k}oPSW_{i+2,j,k} - oN_{i+1,j-1,k+1}uPSW_{i+2,j,k} \\
& -uC_{i+1,j,k}oPW_{i+2,j,k} + oC_{i+1,j,k+1}uPW_{i+2,j,k} \\
& -uS_{i+1,j+1,k}oPNW_{i+2,j,k} - oN_{i+1,j,k+1}uPNW_{i+2,j,k} \\
& -uNW_{i+2,j-1,k}oPS_{i+2,j,k} - oNW_{i+2,j-1,k+1}uPS_{i+2,j,k} \\
& -uW_{i+2,j,k}oPC_{i+2,j,k} - oE_{i+1,j,k+1}uPC_{i+2,j,k} \\
& -uSW_{i+2,j+1,k}oPN_{i+2,j,k} - oNE_{i+1,j,k+1}uPN_{i+2,j,k}) \\
+dPNE_{ijk} \quad & (-oN_{i+1,j,k-1}dPW_{i+2,j,k} - uS_{i+1,j+1,k-1}oPW_{i+2,j,k} \\
& +oC_{i+1,j+1,k-1}dPNW_{i+2,j,k} - uC_{i+1,j+1,k-1}oPNW_{i+2,j,k} \\
& -oNW_{i+2,j,k-1}dPC_{i+2,j,k} - uSE_{i+1,j+1,k-1}oPC_{i+2,j,k} \\
& -oE_{i+1,j+1,k-1}dPN_{i+2,j,k} - uE_{i+1,j+1,k-1}oPN_{i+2,j,k})
\end{aligned}
$$

$$
\begin{aligned}
+oPNE_{ijk} \quad & (-uN_{i+1,j,k-1}\,dPW_{i+2,j,k} - oN_{i+1,j,k}\,oPW_{i+2,j,k} \\
& -uS_{i+1,j+1,k}\,uPW_{i+2,j,k} - uC_{i+1,j+1,k-1}\,dPNW_{i+2,j,k} \\
& +oC_{i+1,j+1,k}\,oPNW_{i+2,j,k} - uC_{i+1,j+1,k}\,uPNW_{i+2,j,k} \\
& -uNW_{i+2,j,k-1}\,dPC_{i+2,j,k} - oNW_{i+2,j,k}\,oPC_{i+2,j,k} \\
& -uSE_{i+1,j+1,k}\,uPC_{i+2,j,k} - uW_{i+2,j+1,k-1}\,dPN_{i+2,j,k} \\
& -oE_{i+1,j+1,k}\,oPN_{i+2,j,k} - uE_{i+1,j+1,k}\,uPN_{i+2,j,k}) \\
+uPNE_{ijk} \quad & (-uN_{i+1,j,k}\,oPW_{i+2,j,k} - oN_{i+1,j,k+1}\,uPW_{i+2,j,k} \\
& -uC_{i+1,j+1,k}\,oPNW_{i+2,j,k} + oC_{i+1,j+1,k+1}\,uPNW_{i+2,j,k} \\
& -uNW_{i+2,j,k}\,oPC_{i+2,j,k} - oNW_{i+2,j,k+1}\,uPC_{i+2,j,k} \\
& -uW_{i+2,j+1,k}\,oPN_{i+2,j,k} - oE_{i+1,j+1,k+1}\,uPN_{i+2,j,k})
\end{aligned}
$$

$$
\begin{aligned}
-oNE^{H}_{ijk} \quad = +dPC_{ijk} \quad & (-oNE_{i,j,k-1}\,dPSW_{i+2,j+2,k} - uNE_{i,j,k-1}\,oPSW_{i+2,j+2,k}) \\
+oPC_{ijk} \quad & (-uSW_{i+1,j+1,k-1}\,dPSW_{i+2,j+2,k} - oNE_{i,j,k}\,oPSW_{i+2,j+2,k} \\
& -uNE_{i,j,k}\,uPSW_{i+2,j+2,k}) \\
+uPC_{ijk} \quad & (-uSW_{i+1,j+1,k}\,oPSW_{i+2,j+2,k} - oNE_{i,j,k+1}\,uPSW_{i+2,j+2,k}) \\
+dPN_{ijk} \quad & (-oE_{i,j+1,k-1}\,dPSW_{i+2,j+2,k} - uE_{i,j+1,k-1}\,oPSW_{i+2,j+2,k} \\
& -oNE_{i,j+1,k-1}\,dPW_{i+2,j+2,k} - uNE_{i,j+1,k-1}\,oPW_{i+2,j+2,k}) \\
+oPN_{ijk} \quad & (-uW_{i+1,j+1,k-1}\,dPSW_{i+2,j+2,k} - oE_{i,j+1,k}\,oPSW_{i+2,j+2,k} \\
& -uE_{i,j+1,k}\,uPSW_{i+2,j+2,k} - uSW_{i+1,j+2,k-1}\,dPW_{i+2,j+2,k} \\
& -oNE_{i,j+1,k}\,oPW_{i+2,j+2,k} - uNE_{i,j+1,k}\,uPW_{i+2,j+2,k}) \\
+uPN_{ijk} \quad & (-uW_{i+1,j+1,k}\,oPSW_{i+2,j+2,k} - oE_{i,j+1,k+1}\,uPSW_{i+2,j+2,k} \\
& -uSW_{i+1,j+2,k}\,oPW_{i+2,j+2,k} - oNE_{i,j+1,k+1}\,uPW_{i+2,j+2,k}) \\
+dPE_{ijk} \quad & (-oN_{i+1,j,k-1}\,dPSW_{i+2,j+2,k} - uN_{i+1,j,k-1}\,oPSW_{i+2,j+2,k} \\
& -oNE_{i+1,j,k-1}\,dPS_{i+2,j+2,k} - uNE_{i+1,j,k-1}\,oPS_{i+2,j+2,k}) \\
+oPE_{ijk} \quad & (-uS_{i+1,j+1,k-1}\,dPSW_{i+2,j+2,k} - oN_{i+1,j,k}\,oPSW_{i+2,j+2,k} \\
& -uN_{i+1,j,k}\,uPSW_{i+2,j+2,k} - uSW_{i+2,j+1,k-1}\,dPS_{i+2,j+2,k} \\
& -oNE_{i+1,j,k}\,oPS_{i+2,j+2,k} - uNE_{i+1,j,k}\,uPS_{i+2,j+2,k}) \\
+uPE_{ijk} \quad & (-uS_{i+1,j+1,k}\,oPSW_{i+2,j+2,k} - oN_{i+1,j,k+1}\,uPSW_{i+2,j+2,k} \\
& -uSW_{i+2,j+1,k}\,oPS_{i+2,j+2,k} - oNE_{i+1,j,k+1}\,uPS_{i+2,j+2,k}) \\
+dPNE_{ijk} \quad & (oC_{i+1,j+1,k-1}\,dPSW_{i+2,j+2,k} - uC_{i+1,j+1,k-1}\,oPSW_{i+2,j+2,k} \\
& -oN_{i+1,j+1,k-1}\,dPW_{i+2,j+2,k} - uN_{i+1,j+1,k-1}\,oPW_{i+2,j+2,k} \\
& -oE_{i+1,j+1,k-1}\,dPS_{i+2,j+2,k} - uE_{i+1,j+1,k-1}\,oPS_{i+2,j+2,k} \\
& -oNE_{i+1,j+1,k-1}\,dPC_{i+2,j+2,k} - uNE_{i+1,j+1,k-1}\,oPC_{i+2,j+2,k}) \\
+oPNE_{ijk} \quad & (-uC_{i+1,j+1,k-1}\,dPSW_{i+2,j+2,k} + oC_{i+1,j+1,k}\,oPSW_{i+2,j+2,k} \\
& -uC_{i+1,j+1,k}\,uPSW_{i+2,j+2,k} - uS_{i+1,j+2,k-1}\,dPW_{i+2,j+2,k} \\
& -oN_{i+1,j+1,k}\,oPW_{i+2,j+2,k} - uN_{i+1,j+1,k}\,uPW_{i+2,j+2,k} \\
& -uW_{i+2,j+1,k-1}\,dPS_{i+2,j+2,k} - oE_{i+1,j+1,k}\,oPS_{i+2,j+2,k} \\
& -uE_{i+1,j+1,k}\,uPS_{i+2,j+2,k} - uSW_{i+2,j+2,k-1}\,dPC_{i+2,j+2,k} \\
& -oNE_{i+1,j+1,k}\,oPC_{i+2,j+2,k} - uNE_{i+1,j+1,k}\,uPC_{i+2,j+2,k}) \\
+uPNE_{ijk} \quad & (-uC_{i+1,j+1,k}\,oPSW_{i+2,j+2,k} + oC_{i+1,j+1,k+1}\,uPSW_{i+2,j+2,k} \\
& -uS_{i+1,j+2,k}\,oPW_{i+2,j+2,k} - oN_{i+1,j+1,k+1}\,uPW_{i+2,j+2,k} \\
& -uW_{i+2,j+1,k}\,oPS_{i+2,j+2,k} - oE_{i+1,j+1,k+1}\,uPS_{i+2,j+2,k} \\
& -uSW_{i+2,j+2,k}\,oPC_{i+2,j+2,k} - oNE_{i+1,j+1,k+1}\,uPC_{i+2,j+2,k})
\end{aligned}
$$

$$
\begin{aligned}
-uNE^{H}_{ijk} \quad = -oPC_{ijk} \quad & uNE_{i,j,k}\,dPSW_{i+2,j+2,k+2} \\
+uPC_{ijk} \quad & (-oNE_{i,j,k+1}\,dPSW_{i+2,j+2,k+2} - uNE_{i,j,k+1}\,oPSW_{i+2,j+2,k+2}) \\
+oPN_{ijk} \quad & (-uE_{i,j+1,k}\,dPSW_{i+2,j+2,k+2} - uNE_{i,j+1,k}\,dPW_{i+2,j+2,k+2}) \\
+uPN_{ijk} \quad & (-oE_{i,j+1,k+1}\,dPSW_{i+2,j+2,k+2} - uE_{i,j+1,k+1}\,oPSW_{i+2,j+2,k+2} \\
& -oNE_{i,j+1,k+1}\,dPW_{i+2,j+2,k+2} - uNE_{i,j+1,k+1}\,oPW_{i+2,j+2,k+2}) \\
+oPE_{ijk} \quad & (-uN_{i+1,j,k}\,dPSW_{i+2,j+2,k+2} - uNE_{i+1,j,k}\,dPS_{i+2,j+2,k+2}) \\
+uPE_{ijk} \quad & (-oN_{i+1,j,k+1}\,dPSW_{i+2,j+2,k+2} - uN_{i+1,j,k+1}\,oPSW_{i+2,j+2,k+2} \\
& -oNE_{i+1,j,k+1}\,dPS_{i+2,j+2,k+2} - uNE_{i+1,j,k+1}\,oPS_{i+2,j+2,k+2}) \\
+oPNE_{ijk} \quad & (-uC_{i+1,j+1,k}\,dPSW_{i+2,j+2,k+2} - uN_{i+1,j+1,k}\,dPW_{i+2,j+2,k+2} \\
& -uE_{i+1,j+1,k}\,dPS_{i+2,j+2,k+2} - uNE_{i+1,j+1,k}\,dPC_{i+2,j+2,k+2}) \\
+uPNE_{ijk} \quad & (oC_{i+1,j+1,k+1}\,dPSW_{i+2,j+2,k+2} - uC_{i+1,j+1,k+1}\,oPSW_{i+2,j+2,k+2} \\
& -oN_{i+1,j+1,k+1}\,dPW_{i+2,j+2,k+2} - uN_{i+1,j+1,k+1}\,oPW_{i+2,j+2,k+2} \\
& -oE_{i+1,j+1,k+1}\,dPS_{i+2,j+2,k+2} - uE_{i+1,j+1,k+1}\,oPS_{i+2,j+2,k+2} \\
& -oNE_{i+1,j+1,k+1}\,dPC_{i+2,j+2,k+2} - uNE_{i+1,j+1,k+1}\,oPC_{i+2,j+2,k+2})
\end{aligned}
$$

Remark A.3. Consider a finite element discretization of Poisson's equation on the unit cube, employing a uniform three-dimensional Cartesian mesh. We use piecewise linear basis functions on tetrahedral elements which subdivide each cube in the uniform Cartesian mesh. The four master-element basis functions on the master-element $\tilde{e}$ having the three vertices $\{(1,0,0),(0,1,0),(0,0,1)\}$ have the form:

$$\left\{\begin{array}{l}\tilde{\phi}_1(x,y,z)=x,\\ \tilde{\phi}_2(x,y,z)=y,\\ \tilde{\phi}_3(x,y,z)=z,\\ \tilde{\phi}_4(x,y,z)=1-x-y-z.\end{array}\right\}$$

For Poisson's equation, we can evaluate the integrals required for the master-element stiffness matrix analytically, yielding:

$$\left[\int_0^1\int_0^{1-x}\int_0^{1-x-y}\nabla\tilde{\phi}_i\cdot\nabla\tilde{\phi}_j\ dzdydx\right]=\frac{1}{6}\begin{bmatrix}3&-1&-1&-1\\-1&1&0&0\\-1&0&1&0\\-1&0&0&1\end{bmatrix}.$$

There are exactly ten ways to break a cube into tetrahedra such that the vertices of the tetrahedra coincide with the vertices of the cube (cf. [156] for analysis and references to similar geometric results for tetrahedra). Choosing to subdivide the unit cube into five or six tetrahedra, the unit cube element stiffness matrix can be constructed from the individual tetrahedral master-element stiffness matrices. It is not difficult to show after some bookkeeping that the resulting stencil representations of the system matrix, depending on whether five or six tetrahedra are used to divide the cube, are as follows:

$$A_h=\begin{bmatrix}0&0&0\\-1&-1&0\\0&0&0\end{bmatrix}\times\begin{bmatrix}0&-1&-1\\0&8&0\\-1&-1&0\end{bmatrix}\times\begin{bmatrix}0&0&0\\0&-1&-1\\0&0&0\end{bmatrix}_h^h, \tag{A.5}$$

$$A_h=\begin{bmatrix}0&1&0\\-1&-4&0\\1&0&0\end{bmatrix}\times\begin{bmatrix}0&-4&-1\\-2&20&-2\\-1&-4&0\end{bmatrix}\times\begin{bmatrix}0&0&1\\0&-4&-1\\0&1&0\end{bmatrix}_h^h, \tag{A.6}$$

where the meshwidth h has been divided out of the matrices. More generally, note that for a variable coefficient operator rather than the Laplace operator, the nonzero structure of the stencil produced by a tetrahedral refinement of a non-uniform Cartesian mesh will be:

$$A_h=\begin{bmatrix}0&-dN_{ij}&0\\-dW_{ij}&-dC_{ij}&0\\-dSW_{ij}&0&0\end{bmatrix}\times\begin{bmatrix}0&-oN_{ij}&-oNE_{ij}\\-oW_{ij}&oC_{ij}&-oE_{ij}\\-oSW_{ij}&-oS_{ij}&0\end{bmatrix}\times\begin{bmatrix}0&0&-uNE_{ij}\\0&-uC_{ij}&-uE_{ij}\\0&-uS_{ij}&0\end{bmatrix}_h^h.$$

The additional zero elements in the first case occur due to the symmetries present for the Laplace operator on a uniform Cartesian mesh. This is also true in two dimensions; the usually seven-point stencil produced by a triangular refinement of a non-uniform Cartesian mesh reduces to the familiar five-point stencil in the case of the Laplace operator.

Linear interpolation of a grid function on the uniform Cartesian mesh, and the corresponding restriction operator, are represented by:

$$I_h^H=\begin{bmatrix}0&\frac{1}{2}&0\\\frac{1}{2}&\frac{1}{2}&0\\\frac{1}{2}&0&0\end{bmatrix}\times\begin{bmatrix}0&\frac{1}{2}&\frac{1}{2}\\\frac{1}{2}&1&\frac{1}{2}\\\frac{1}{2}&\frac{1}{2}&0\end{bmatrix}\times\begin{bmatrix}0&0&\frac{1}{2}\\0&\frac{1}{2}&\frac{1}{2}\\0&\frac{1}{2}&0\end{bmatrix}_{h(H)}^H,$$

$$I_H^h=\begin{bmatrix}0&0&0\\0&0&0\\\frac{1}{2}&0&0\end{bmatrix}\times\begin{bmatrix}0&0&0\\0&0&0\\0&0&0\end{bmatrix}\times\begin{bmatrix}0&0&\frac{1}{2}\\0&0&0\\0&0&0\end{bmatrix}_{H(8)}^h$$

$$\vee \begin{bmatrix} 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 \end{bmatrix}^{h}_{H(7,6,5)} \quad \vee \begin{bmatrix} \frac{1}{2} \\ 0 \\ \frac{1}{2} \end{bmatrix}^{h}_{H(4,3,2)} \quad \vee \left[\ 1\ \right]^{h}_{H(1)}.$$

Since the linear prolongation operator corresponds to the natural inclusion operation in the finite element space constructed from linear basis functions, then with the above choice of stencils for the Poisson equation case, the stencil calculus should reproduce both (A.5) and (A.6) as the resulting coarse mesh Galerkin stencils. This is difficult to show by hand, but is easily verified in MAPLE.

Consider now a finite element discretization of Poisson's equation on the unit cube, employing a uniform three-dimensional Cartesian mesh and piecewise trilinear basis functions on the resulting cube elements. The eight master-element basis functions on the master-element $\tilde{e} = [-1,1] \times [-1,1] \times [-1,1]$ have the form:

$$\left\{ \begin{array}{ll} \tilde{\phi}_1(x,y,z) = \frac{(1-x)(1-y)(1-z)}{8}, & \tilde{\phi}_5(x,y,z) = \frac{(1-x)(1-y)(1+z)}{8}, \\ \tilde{\phi}_2(x,y,z) = \frac{(1+x)(1-y)(1-z)}{8}, & \tilde{\phi}_6(x,y,z) = \frac{(1+x)(1-y)(1+z)}{8}, \\ \tilde{\phi}_3(x,y,z) = \frac{(1+x)(1+y)(1-z)}{8}, & \tilde{\phi}_7(x,y,z) = \frac{(1+x)(1+y)(1+z)}{8}, \\ \tilde{\phi}_4(x,y,z) = \frac{(1-x)(1+y)(1-z)}{8}, & \tilde{\phi}_8(x,y,z) = \frac{(1-x)(1+y)(1+z)}{8}. \end{array} \right\}$$

For Poisson's equation, we can evaluate the integrals required for the master-element stiffness matrix analytically, yielding:

$$\left[\int_{-1}^{1} \int_{-1}^{1} \int_{-1}^{1} \nabla \tilde{\phi}_i \cdot \nabla \tilde{\phi}_j \ dxdydz \right] = \frac{1}{6} \begin{bmatrix} 4 & 0 & -1 & 0 & 0 & -1 & -1 & -1 \\ 0 & 4 & 0 & -1 & -1 & 0 & -1 & -1 \\ -1 & 0 & 4 & 0 & -1 & -1 & 0 & -1 \\ 0 & -1 & 0 & 4 & -1 & -1 & -1 & 0 \\ 0 & -1 & -1 & -1 & 4 & 0 & -1 & 0 \\ -1 & 0 & -1 & -1 & 0 & 4 & 0 & -1 \\ -1 & -1 & 0 & -1 & -1 & 0 & 4 & 0 \\ -1 & -1 & -1 & 0 & 0 & -1 & 0 & 4 \end{bmatrix}.$$

It then follows after a little bookkeeping that the resulting stencil representation of the system matrix is as follows:

$$A_h = \frac{1}{6} \begin{bmatrix} -1 & -2 & -1 \\ -2 & 0 & -2 \\ -1 & -2 & -1 \end{bmatrix} \times \frac{1}{6} \begin{bmatrix} -2 & 0 & -2 \\ 0 & 32 & 0 \\ -2 & 0 & -2 \end{bmatrix} \times \frac{1}{6} \begin{bmatrix} -1 & -2 & -1 \\ -2 & 0 & -2 \\ -1 & -2 & -1 \end{bmatrix}^{h}_{h}, \qquad \text{(A.7)}$$

where the meshwidth h has been divided out of the matrix. More generally for a variable coefficient operator, all twenty-seven stencil components will be nonzero; the zeros occurring above are due to symmetries which are present for the Laplace operator on a uniform Cartesian mesh. Trilinear interpolation of a grid function on the uniform Cartesian mesh, and the corresponding restriction operator, are represented by:

$$I_h^H = \begin{bmatrix} \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \end{bmatrix} \times \begin{bmatrix} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{2} & 1 & \frac{1}{2} \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \end{bmatrix} \times \begin{bmatrix} \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \end{bmatrix}^{H}_{h(H)},$$

$$I_H^h = \begin{bmatrix} \frac{1}{8} & 0 & \frac{1}{8} \\ 0 & 0 & 0 \\ \frac{1}{8} & 0 & \frac{1}{8} \end{bmatrix} \times \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} \frac{1}{8} & 0 & \frac{1}{8} \\ 0 & 0 & 0 \\ \frac{1}{8} & 0 & \frac{1}{8} \end{bmatrix}^{h}_{H(8)}$$

$$\vee \begin{bmatrix} \frac{1}{4} & 0 & \frac{1}{4} \\ 0 & 0 & 0 \\ \frac{1}{4} & 0 & \frac{1}{4} \end{bmatrix}^{h}_{H(7,6,5)} \quad \vee \begin{bmatrix} \frac{1}{2} \\ 0 \\ \frac{1}{2} \end{bmatrix}^{h}_{H(4,3,2)} \quad \vee \left[\ 1\ \right]^{h}_{H(1)}.$$

Since the trilinear prolongation operator corresponds to the natural inclusion operation in the finite element space constructed from trilinear basis functions, then with the above choice of stencils, the stencil calculus should reproduce (A.7) as the resulting coarse mesh Galerkin stencil; this is easily verifiable in MAPLE, but is a tedious calculation by hand.

Finally, consider a box-method discretization of Poisson's equation on the unit cube, employing a uniform three-dimensional Cartesian mesh, which as shown in Chapter 2 yields the following seven-point stencil representation of the system matrix:

$$A_h = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 0 & -1 & 0 \\ -1 & 6 & -1 \\ 0 & -1 & 0 \end{bmatrix} \times \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix}_h^h . \tag{A.8}$$

If linear prolongation is employed, then it is not difficult to show using the MAPLE implementation of the stencil calculus that the Galerkin coarse mesh stencils converge rapidly to the stencil (A.6). Similarly, if trilinear prolongation is used, then the Galerkin coarse mesh stencils converge rapidly to the stencil (A.7). Unfortunately, in the linear case the stencil expands to fifteen-point already on the second finest mesh, but it remains fifteen-point on coarser meshes. In the bilinear case, the stencil expands to twenty-seven-point on the second finest mesh, but remains twenty-seven-point on all coarser meshes.

When a general second order elliptic operator is considered, and a box-method discretization is employed on a fine non-uniform Cartesian mesh yielding a seven point stencil, then the stencils spread in exactly this same fashion on the second finest mesh. The fixed nonzero structure of the second finest mesh stencil is preserved on all coarser meshes, so the stencil spreading occurs only from the finest to the second finest mesh in the general case as well.

A.2 Operator-based prolongations by stencil compression

We outlined in 3 an approach for producing prolongation operators which attempt to conserve flux at box boundaries in a box-method-based multilevel scheme. There is an analogous interpretation completely in terms of the discretized stencil, which we also outlined. These approaches were first developed in [2] for two-dimensional problems, and extensive experiments for three-dimensional problems have appeared in [51]. Related algebraically-based prolongation operators are constructed and discussed in [166]. In this section, we will give the stencils for one-, two-, and three-dimensional operator-based prolongation operators based on stencil compression, as described in Chapter 3.

A.2.1 One dimension

In Chapter 3, we developed the one-dimensional operator-based prolongation operator motivated by physical considerations, and outlined an equivalent approach based completely on the discrete differential operator. In the second approach, the prolongation operator stencil components were shown to have the form:

$$PC_i = 1, \qquad PE_i = \frac{W_{i+1}}{C_{i+1}}, \qquad PW_i = \frac{E_{i-1}}{C_{i-1}}.$$

A.2.2 Two dimensions

As outlined in Chapter 3, the higher-dimensional operator-based prolongations are obtained by stencil compression. The prolongation operator stencil components in the two-dimensional case are as follows:

$$\begin{aligned}
PC_{ij} &= 1 \\
PN_{ij} &= (SW_{i,j+1} + S_{i,j+1} + SE_{i,j+1})/(C_{i,j+1} - W_{i,j+1} - E_{i,j+1}) \\
PS_{ij} &= (NW_{i,j-1} + N_{i,j-1} + NE_{i,j-1})/(C_{i,j-1} - W_{i,j-1} - E_{i,j-1}) \\
PE_{ij} &= (NW_{i+1,j} + W_{i+1,j} + SW_{i+1,j})/(C_{i+1,j} - S_{i+1,j} - N_{i+1,j}) \\
PW_{ij} &= (NE_{i-1,j} + E_{i-1,j} + SE_{i-1,j})/(C_{i-1,j} - S_{i-1,j} - N_{i-1,j}) \\
PNE_{ij} &= (SW_{i+1,j+1} + S_{i+1,j+1} * PE_{ij} + W_{i+1,j+1} * PN_{ij})/(C_{i+1,j+1}) \\
PNW_{ij} &= (SE_{i-1,j+1} + S_{i-1,j+1} * PW_{ij} + E_{i-1,j+1} * PN_{ij})/(C_{i-1,j+1}) \\
PSE_{ij} &= (NW_{i+1,j-1} + N_{i+1,j-1} * PE_{ij} + W_{i+1,j-1} * PS_{ij})/(C_{i+1,j-1}) \\
PSW_{ij} &= (NE_{i-1,j-1} + N_{i-1,j-1} * PW_{ij} + E_{i-1,j-1} * PS_{ij})/(C_{i-1,j-1})
\end{aligned}$$

A.2.3 Three dimensions

The three-dimensional operator-based prolongations are also obtained by stencil compression. The prolongation operator stencil components in the three-dimensional case are as follows. Note that for simplicity in the three-dimensional case, we have assumed symmetry, as in the presentation of the three-dimensional Galerkin coarse grid matrix stencil components.

$$oPC_{ijk} = 1$$

$$\begin{aligned} oPN_{ijk} &= (uNE_{i-1,j,k-1} + uN_{i,j,k-1} + uNW_{i+1,j,k-1} + oNE_{i-1,j,k} + oN_{ijk} + oNW_{i+1,j,k} \\ &\quad + uSW_{i,j+1,k} + uS_{i,j+1,k} + uSE_{i,j+1,k})/(oC_{i,j+1,k} - oE_{i-1,j+1,k} - oE_{i,j+1,k} \\ &\quad - uC_{i,j+1,k-1} - uE_{i-1,j+1,k-1} - uW_{i+1,j+1,k-1} - uC_{i,j+1,k} - uW_{i,j+1,k} - uE_{i,j+1,k}) \end{aligned}$$

$$\begin{aligned} oPS_{ijk} &= (uSE_{i-1,j,k-1} + uS_{i,j,k-1} + uSW_{i+1,j,k-1} + oNW_{i,j-1,k} + oN_{i,j-1,k} + oNE_{i,j-1,k} \\ &\quad + uNW_{i,j-1,k} + uN_{i,j-1,k} + uNE_{i,j-1,k})/(oC_{i,j-1,k} - oE_{i-1,j-1,k} - oE_{i,j-1,k} \\ &\quad - uC_{i,j-1,k-1} - uE_{i-1,j-1,k-1} - uW_{i+1,j-1,k-1} - uC_{i,j-1,k} - uW_{i,j-1,k} - uE_{i,j-1,k}) \end{aligned}$$

$$\begin{aligned} oPE_{ijk} &= (uSE_{i,j+1,k-1} + oNW_{i+1,j,k} + uNW_{i+1,j,k} + uE_{i,j,k-1} + oE_{ijk} + uW_{i+1,j,k} \\ &\quad + uNE_{i,j-1,k-1} + oNE_{i,j-1,k} + uSW_{i+1,j,k})/(oC_{i+1,j,k} - uC_{i+1,j,k-1} - uC_{i+1,j,k} \\ &\quad - oN_{i+1,j,k} - uS_{i+1,j+1,k-1} - uN_{i+1,j,k} - oN_{i+1,j-1,k} - uN_{i+1,j-1,k-1} - uS_{i+1,j,k}) \end{aligned}$$

$$\begin{aligned} oPW_{ijk} &= (uSW_{i,j+1,k-1} + oNE_{i-1,j,k} + uNE_{i-1,j,k} + uW_{i,j,k-1} + oE_{i-1,j,k} + uE_{i-1,j,k} \\ &\quad + uNW_{i,j-1,k-1} + oNW_{i,j-1,k} + uSE_{i-1,j,k})/(oC_{i-1,j,k} - uC_{i-1,j,k-1} - uC_{i-1,j,k} \\ &\quad - oN_{i-1,j,k} - uS_{i-1,j+1,k-1} - uN_{i-1,j,k} - oN_{i-1,j-1,k} - uN_{i-1,j-1,k-1} - uS_{i-1,j,k}) \end{aligned}$$

$$\begin{aligned} oPNE_{ijk} &= (uNE_{i,j,k-1} + oNE_{ijk} + uSW_{i+1,j+1,k} \\ &\quad + (uN_{i+1,j,k-1} + oN_{i+1,j,k} + uS_{i+1,j+1,k})oPE_{ijk} \\ &\quad + (uE_{i,j+1,k-1} + oE_{i,j+1,k} + uW_{i+1,j+1,k})oPN_{ijk}) \\ &\quad /(oC_{i+1,j+1,k} - uC_{i+1,j+1,k-1} - uC_{i+1,j+1,k}) \end{aligned}$$

$$\begin{aligned} oPNW_{ijk} &= (uNW_{i,j,k-1} + oNW_{ijk} + uSE_{i-1,j+1,k} \\ &\quad + (uN_{i-1,j,k-1} + oN_{i-1,j,k} + uS_{i-1,j+1,k})oPW_{ijk} \\ &\quad + (uW_{i,j+1,k-1} + oE_{i-1,j+1,k} + uE_{i-1,j+1,k})oPN_{ijk}) \\ &\quad /(oC_{i-1,j+1,k} - uC_{i-1,j+1,k-1} - uC_{i-1,j+1,k}) \end{aligned}$$

$$\begin{aligned} oPSE_{ijk} &= (uSE_{i,j,k-1} + oNW_{i+1,j-1,k} + uNW_{i+1,j-1,k} \\ &\quad + (uS_{i+1,j,k-1} + oN_{i+1,j-1,k} + uN_{i+1,j-1,k})oPE_{ijk} \\ &\quad + (uE_{i,j-1,k-1} + oE_{i,j-1,k} + uW_{i+1,j-1,k})oPS_{ijk}) \\ &\quad /(oC_{i+1,j-1,k} - uC_{i+1,j-1,k-1} - uC_{i+1,j-1,k}) \end{aligned}$$

$$\begin{aligned} oPSW_{ijk} &= (uSW_{i,j,k-1} + oNE_{i-1,j-1,k} + uNE_{i-1,j-1,k} \\ &\quad + (uS_{i-1,j,k-1} + oN_{i-1,j-1,k} + uN_{i-1,j-1,k})oPW_{ijk} \\ &\quad + (uW_{i,j-1,k-1} + oE_{i-1,j-1,k} + uE_{i-1,j-1,k})oPS_{ijk}) \\ &\quad /(oC_{i-1,j-1,k} - uC_{i-1,j-1,k-1} - uC_{i-1,j-1,k}) \end{aligned}$$

$$\begin{aligned} dPC_{ijk} &= (uNW_{i,j,k-1} + uW_{i,j,k-1} + uSW_{i,j,k-1} + uN_{i,j,k-1} + uC_{i,j,k-1} + uS_{i,j,k-1} \\ &\quad + uNE_{i,j,k-1} + uE_{i,j,k-1} + uSE_{i,j,k-1})/(oC_{i,j,k-1} - oN_{i,j,k-1} - oN_{i,j-1,k-1} \\ &\quad - oNW_{i,j,k-1} - oE_{i-1,j,k-1} - oNE_{i-1,j-1,k-1} \\ &\quad - oNE_{i,j,k-1} - oE_{i,j,k-1} - oNW_{i+1,j-1,k-1}) \end{aligned}$$

$$\begin{aligned} dPN_{ijk} &= (uSW_{i,j+1,k-1} + uS_{i,j+1,k-1} + uSE_{i,j+1,k-1} \\ &\quad + (oNE_{i-1,j,k-1} + oN_{i,j,k-1} + oNW_{i+1,j,k-1})dPC_{ijk} \\ &\quad + (uW_{i,j+1,k-1} + uC_{i,j+1,k-1} + uE_{i,j+1,k-1})oPN_{ijk}) \\ &\quad /(oC_{i,j+1,k-1} - oE_{i-1,j+1,k-1} - oE_{i,j+1,k-1}) \end{aligned}$$

$$\begin{aligned} dPS_{ijk} &= (uNW_{i,j-1,k-1} + uN_{i,j-1,k-1} + uNE_{i,j-1,k-1} \\ &\quad + (oNW_{i,j-1,k-1} + oN_{i,j-1,k-1} + oNE_{i,j-1,k-1})dPC_{ijk} \\ &\quad + (uW_{i,j-1,k-1} + uC_{i,j-1,k-1} + uE_{i,j-1,k-1})oPS_{ijk}) \\ &\quad /(oC_{i,j-1,k-1} - oE_{i-1,j-1,k-1} - oE_{i,j-1,k-1}) \end{aligned}$$

$$\begin{aligned} dPE_{ijk} &= (uNW_{i+1,j,k-1} + uW_{i+1,j,k-1} + uSW_{i+1,j,k-1} \\ &\quad + (uN_{i+1,j,k-1} + uC_{i+1,j,k-1} + uS_{i+1,j,k-1})oPE_{ijk} \\ &\quad + (oNW_{i+1,j,k-1} + oE_{i,j,k-1} + oNE_{i,j-1,k-1})dPC_{ijk}) \\ &\quad /(oC_{i+1,j,k-1} - oN_{i+1,j,k-1} - oN_{i+1,j-1,k-1}) \end{aligned}$$

$$\begin{aligned}
dPW_{ijk} &= (uNE_{i-1,j,k-1} + uE_{i-1,j,k-1} + uSE_{i-1,j,k-1}\\
&\quad +(uN_{i-1,j,k-1} + uC_{i-1,j,k-1} + uS_{i-1,j,k-1})oPW_{ijk}\\
&\quad +(oNE_{i-1,j,k-1} + oE_{i-1,j,k-1} + oNW_{i,j-1,k-1})dPC_{ijk})\\
&\quad /(oC_{i-1,j,k-1} - oN_{i-1,j,k-1} - oN_{i-1,j-1,k-1})
\end{aligned}$$

$$\begin{aligned}
dPNE_{ijk} &= (uSW_{i+1,j+1,k-1} + uW_{i+1,j+1,k-1}oPN_{ijk} + uS_{i+1,j+1,k-1}oPE_{ijk}\\
&\quad +uC_{i+1,j+1,k-1}oPNE_{ijk} + oNE_{i,j,k-1}dPC_{ijk} + oE_{i,j+1,k-1}dPN_{ijk}\\
&\quad +oN_{i+1,j,k-1}dPE_{ijk})/oC_{i+1,j+1,k-1}
\end{aligned}$$

$$\begin{aligned}
dPNW_{ijk} &= (uSE_{i-1,j+1,k-1} + uE_{i-1,j+1,k-1}oPN_{ijk} + uS_{i-1,j+1,k-1}oPW_{ijk}\\
&\quad +uC_{i-1,j+1,k-1}oPNW_{ijk} + oNW_{i,j,k-1}dPC_{ijk} + oE_{i-1,j+1,k-1}dPN_{ijk}\\
&\quad +oN_{i-1,j,k-1}dPW_{ijk})/oC_{i-1,j+1,k-1}
\end{aligned}$$

$$\begin{aligned}
dPSE_{ijk} &= (uNW_{i+1,j-1,k-1} + uW_{i+1,j-1,k-1}oPS_{ijk} + uN_{i+1,j-1,k-1}oPE_{ijk}\\
&\quad +uC_{i+1,j-1,k-1}oPSE_{ijk} + oNW_{i+1,j-1,k-1}dPC_{ijk} + oE_{i,j-1,k-1}dPS_{ijk}\\
&\quad +oN_{i+1,j-1,k-1}dPE_{ijk})/oC_{i+1,j-1,k-1}
\end{aligned}$$

$$\begin{aligned}
dPSW_{ijk} &= (uNE_{i-1,j-1,k-1} + uE_{i-1,j-1,k-1}oPS_{ijk} + uN_{i-1,j-1,k-1}oPW_{ijk}\\
&\quad +uC_{i-1,j-1,k-1}oPSW_{ijk} + oNE_{i-1,j-1,k-1}dPC_{ijk} + oE_{i-1,j-1,k-1}dPS_{ijk}\\
&\quad +oN_{i-1,j-1,k-1}dPW_{ijk})/oC_{i-1,j-1,k-1}
\end{aligned}$$

$$\begin{aligned}
uPC_{ijk} &= (uSE_{i-1,j+1,k} + uE_{i-1,j,k} + uNE_{i-1,j-1,k} + uS_{i,j+1,k} + uC_{ijk} + uN_{i,j-1,k}\\
&\quad +uSW_{i+1,j+1,k} + uW_{i+1,j,k} + uNW_{i+1,j-1,k})/(oC_{i,j,k+1} - oN_{i,j,k+1} - oN_{i,j-1,k+1}\\
&\quad -oNW_{i,j,k+1} - oE_{i-1,j,k+1} - oNE_{i-1,j-1,k+1}\\
&\quad -oNE_{i,j,k+1} - oE_{i,j,k+1} - oNW_{i+1,j-1,k+1})
\end{aligned}$$

$$\begin{aligned}
uPN_{ijk} &= (uNE_{i-1,j,k} + uN_{ijk} + uNW_{i+1,j,k}\\
&\quad +(oNE_{i-1,j,k+1} + oN_{i,j,k+1} + oNW_{i+1,j,k+1})uPC_{ijk}\\
&\quad +(uE_{i-1,j+1,k} + uC_{i,j+1,k} + uW_{i+1,j+1,k})oPN_{ijk})\\
&\quad /(oC_{i,j+1,k+1} - oE_{i-1,j+1,k+1} - oE_{i,j+1,k+1})
\end{aligned}$$

$$\begin{aligned}
uPS_{ijk} &= (uSE_{i-1,j,k} + uS_{ijk} + uSW_{i+1,j,k}\\
&\quad +(oNW_{i,j-1,k+1} + oN_{i,j-1,k+1} + oNE_{i,j-1,k+1})uPC_{ijk}\\
&\quad +(uE_{i-1,j-1,k} + uC_{i,j-1,k} + uW_{i+1,j-1,k})oPS_{ijk})\\
&\quad /(oC_{i,j-1,k+1} - oE_{i-1,j-1,k+1} - oE_{i,j-1,k+1})
\end{aligned}$$

$$\begin{aligned}
uPE_{ijk} &= (uSE_{i,j+1,k} + uS_{i+1,j+1,k} + uNE_{i,j-1,k}\\
&\quad +(uS_{i+1,j+1,k} + uC_{i+1,j,k} + uN_{i+1,j-1,k})oPE_{ijk}\\
&\quad +(oNW_{i+1,j,k+1} + oE_{i,j,k+1} + oNE_{i,j-1,k+1})uPC_{ijk})\\
&\quad /(oC_{i+1,j,k+1} - oN_{i+1,j,k+1} - oN_{i+1,j-1,k+1})
\end{aligned}$$

$$\begin{aligned}
uPW_{ijk} &= (uSW_{i,j+1,k} + uW_{ijk} + uNW_{i,j-1,k}\\
&\quad +(uS_{i-1,j+1,k} + uC_{i-1,j,k} + uN_{i-1,j-1,k})oPW_{ijk}\\
&\quad +(oNE_{i-1,j,k+1} + oE_{i-1,j,k+1} + oNW_{i,j-1,k+1})uPC_{ijk})\\
&\quad /(oC_{i-1,j,k+1} - oN_{i-1,j,k+1} - oN_{i-1,j-1,k+1})
\end{aligned}$$

$$\begin{aligned}
uPNE_{ijk} &= (uNE_{ijk} + uE_{i,j+1,k}oPN_{ijk} + uN_{i+1,j,k}oPE_{ijk}\\
&\quad +uC_{i+1,j+1,k}oPNE_{ijk} + oNE_{i,j,k+1}uPC_{ijk} + oE_{i,j+1,k+1}uPN_{ijk}\\
&\quad +oN_{i+1,j,k+1}uPE_{ijk})/oC_{i+1,j+1,k+1}
\end{aligned}$$

$$\begin{aligned}
uPNW_{ijk} &= (uNW_{ijk} + uW_{i,j+1,k}oPN_{ijk} + uN_{i-1,j,k}oPW_{ijk}\\
&\quad +uC_{i-1,j+1,k}oPNW_{ijk} + oNW_{i,j,k+1}uPC_{ijk} + oE_{i-1,j+1,k+1}uPN_{ijk}\\
&\quad +oN_{i-1,j,k+1}uPW_{ijk})/oC_{i-1,j+1,k+1}
\end{aligned}$$

$$\begin{aligned}
uPSE_{ijk} &= (uSE_{ijk} + uE_{i,j-1,k}oPS_{ijk} + uS_{i+1,j,k}oPE_{ijk}\\
&\quad +uC_{i+1,j-1,k}oPSE_{ijk} + oNW_{i+1,j-1,k+1}uPC_{ijk} + oE_{i,j-1,k+1}uPS_{ijk}\\
&\quad +oN_{i+1,j-1,k+1}uPE_{ijk})/oC_{i+1,j-1,k+1}
\end{aligned}$$

$$\begin{aligned}
uPSW_{ijk} &= (uSW_{ijk} + uW_{i,j-1,k}oPS_{ijk} + uS_{i-1,j,k}oPW_{ijk}\\
&\quad +uC_{i-1,j-1,k}oPSW_{ijk} + oNE_{i-1,j-1,k+1}uPC_{ijk} + oE_{i-1,j-1,k+1}uPS_{ijk}\\
&\quad +oN_{i-1,j-1,k+1}uPW_{ijk})/oC_{i-1,j-1,k+1}
\end{aligned}$$

A.3 Stencil calculus in MAPLE and MATHEMATICA

The MAPLE and MATHEMATICA routines implementing the stencil calculus in two and three dimensions, which were used to produce the expressions in this Appendix and which can be used to produce FORTRAN or C expressions directly, can be obtained from the author.

B. Information about the Software

In the course of this work, a package of three-dimensional linear and nonlinear partial differential equation solvers has been developed. At the core of the package are the linear multilevel methods discussed in the earlier chapters. The package is applicable to three-dimensional nonlinear Poisson-like scalar equations, allowing for nonlinearities that depend on the scalar unknown (but not on derivatives); nonlinearities of this type occur in the Poisson-Boltzmann equation as well as in other applications, such as semiconductor modeling. The class of problems for which the package is applicable can be extended with suitable modifications to the software (the methods need not be modified). In this Appendix, we summarize the essential features and ideas in the package, as well as how the package is used.

B.1 The FORTRAN-language multilevel numerical software

The software is restricted to *logically* non-uniform Cartesian three-dimensional meshes due to the multilevel solver employed, and the box-method discretization routine we provide requires a *physically* non-uniform Cartesian mesh as well. This discretization routine allows arbitrary Dirichlet or Neumann boundary conditions at any boundary point on a three-dimensional rectangular box, allowing for specification of arbitrary contact placements and geometries, as may be necessary in semiconductor modeling. Although we provide a box-method discretization with the software, the multilevel method can form the coarse level equations algebraically, the only requirement on the fine level discretization being that the equations be representable as stencils, and that the nonlinear term be diagonal. This is always true for either box or finite element (with mass lumping) discretization on a logically non-uniform Cartesian mesh.

To use the package as provided, the user defines the problem coefficients, the mesh points and domain geometry, as well as the boundary conditions and types, in a very simple and well-defined way. All necessary workspace is passed in by the user; the software will indicate exactly how much workspace is required for a particular problem configuration. The nonlinearity must be defined in a subroutine supplied by the user which is to be called by the software, and a second subroutine must also be supplied which provides the derivative of the nonlinear term for use in forming the Jacobian required by the Newton solver. Using the solver involves a single subroutine call with a simple and understandable argument list.

The linear multilevel method at the core of the software, as described earlier, is designed specifically for discontinuous coefficients as occur in material interface problems. Techniques employed include the coefficient averaging methods or the Galerkin methods, combined with operator-induced prolongation procedures which attempt to enforce flux conservation at box boundaries when a grid function is interpolated from a coarse to a fine mesh. The averaging methods are usually sufficient for the Poisson-Boltzmann problem, but particularly complex molecular surfaces seem to require the Galerkin approach for robustness; this is computationally more costly and requires more memory than the averaging approach (seven-diagonal matrices produced by the box-method on the fine mesh expand to twenty-seven-diagonal matrices on coarse meshes), but it always converges rapidly. Galerkin coefficient expressions are extremely complex in three dimensions, and were obtained with the help of the MAPLE and MATHEMATICA symbolic manipulation packages.

The user is given many choices in configuring the multilevel method, including choice of smoother, number of levels employed, iteration strategy, choice of prolongation operator, coarse problem formulation, as well as iterative (CG) or direct (banded LINPACK) solve on the coarsest mesh. A conjugate gradient method is also employed to accelerate the multilevel methods, and while this approach is mathematically guaranteed to improve the convergence rate of the multilevel method with a corresponding increase in computational labor, it is also the case that in practice the CPU time to solve the problem is usually reduced using the acceleration (i.e., the acceleration more than pays for itself).

Nonlinear problems are handled with the methods described earlier: either a nonlinear multilevel method, or a more robust (essentially globally convergent) inexact-Newton solver coupled with the fast linear multilevel method. Configuration options include a choice of several stopping criteria, degree of inexactness in the approximate Newton solves, and damping strategy, in addition to all possible configuration options for the multilevel Jacobian system solver. A nonlinear operator-based prolongation procedure has been developed for the nonlinear case as well. Also included in the software are several implementations of more traditional nonlinear methods, including nonlinear SOR (SOR-outer iteration with one-dimensional Newton-solves) and nonlinear CG (the Flether-Reeves nonlinear extension to the linear Hestenes-Steifel algorithm).

An interface routine is provided for using the software specifically as a nonlinear Poisson-Boltzmann equation solver. To handle the severe numerical problems occurring with nonlinearities of exponential-type present in the Poisson-Boltzmann equation, we developed argument-capping functions which avoid nonvectorizable statements. Calls to the standard intrinsic functions are replaced by these modified functions, and overflows are successfully avoided during early transient iterations without loosing the execution efficiency of the intrinsic functions.

The entire package is written in standard FORTRAN 77. The package installs itself in single or double precision on the specified machine, inserting the appropriate compiler directives automatically, and runs without modification on most current systems, including the Cray C90, Cray Y-MP, Convex C3, Convex C240, Convex C220, IBM RS/6000, Sun 4, Sun 3, and most other systems with a standard FORTRAN 77 compiler.

Due to the various choices made during the development of this package, the software executes at very high rates on a number of modern computers; see for example Chapter 8 for benchmarks.

B.2 The C-language X11 user interface

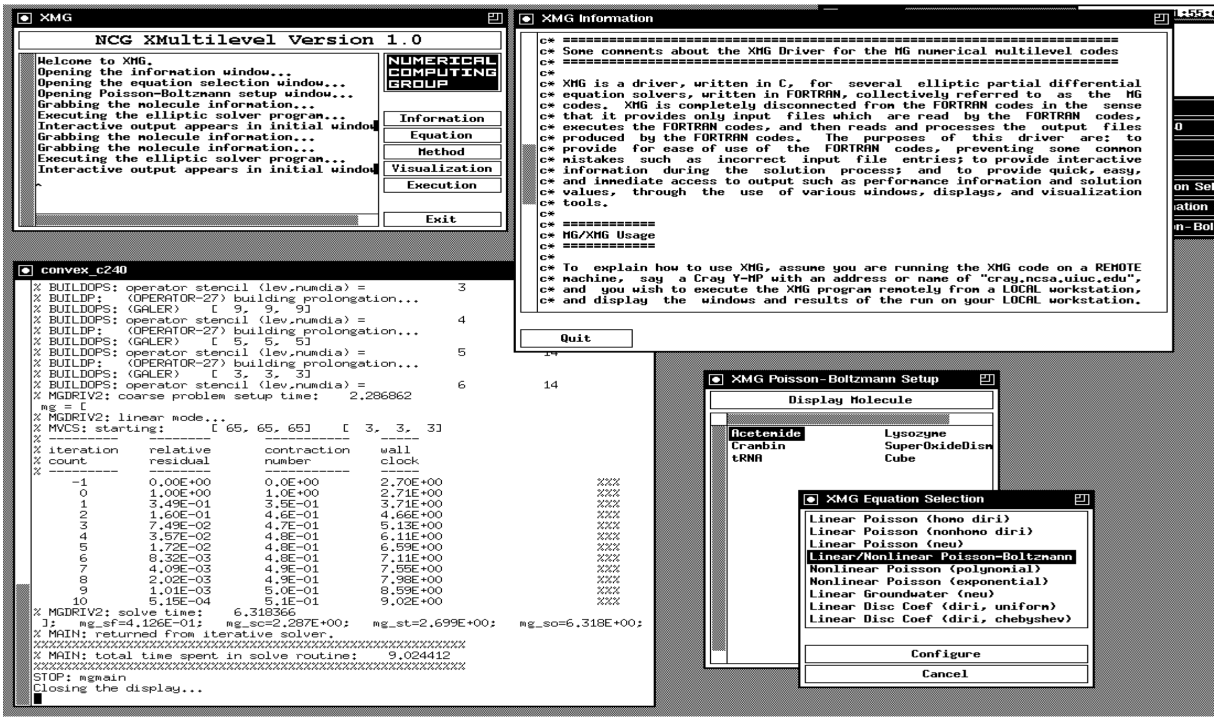
An XWindow interface has been written (in C) for the FORTRAN numerical software. The interface is completely disconnected from the numerical software in the sense that it provides only input files which are read by the numerical software, executes the numerical software, and then reads and processes the output files produced by the numerical software. The purposes of this driver are: to provide for ease of use of the numerical software, preventing some common mistakes such as incorrect input file entries; to provide interactive information during the solution process; and to provide quick, easy, and immediate access to output such as performance information and solution values, through the use of various windows, displays, and visualization tools.

The C-language interface runs as is on any machine that has a standard C compiler and the standard X11 distribution Release 4 or newer. The interface uses only the Athena Widget set (along with Xlib and the Xt intrinsics), so that the interface is also quite portable. We have tested it on the machines mentioned above in the description of the numerical software, and the interface runs without modification in each case.

A sample display produced by the interface is given in Figure B.1.

B.3 Obtaining the software

The software package, including the numerical software as well as the XWindow interface software, can be obtained from the author.



```
Welcome to XMG.
Opening the information window...
Opening the equation selection window...
Opening Poisson-Boltzmann setup window...
Grabbing the molecule information...
Executing the elliptic solver program...
Interactive output appears in initial window
Grabbing the molecule information...
Grabbing the molecule information...
Executing the elliptic solver program...
Interactive output appears in initial window
^
```

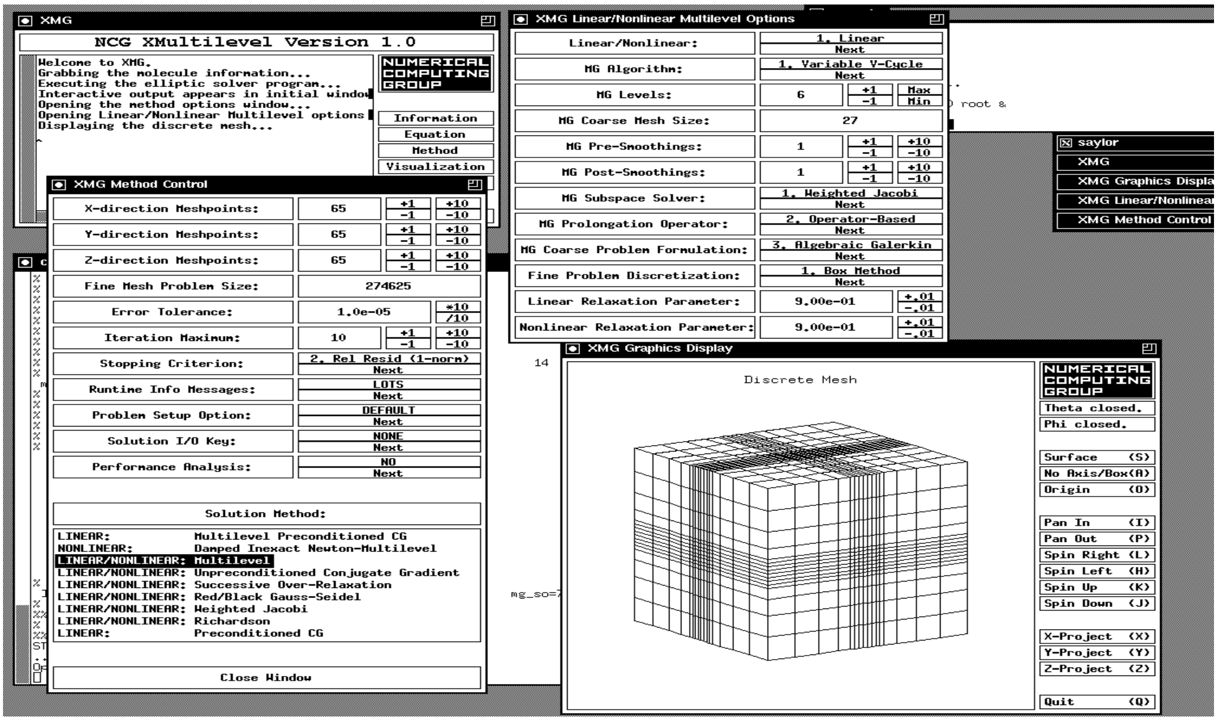
```
c* ========================================================================
c* Some comments about the XMG Driver for the MG numerical multilevel codes
c* ========================================================================
c*
c* XMG is a driver, written in C,  for  several  elliptic partial differential
c* equation solvers, written in FORTRAN, collectively referred to  as  the  MG
c* codes.  XMG is completely disconnected from the FORTRAN codes in the  sense
c* that it provides only input  files which  are read  by the  FORTRAN  codes,
c* executes the FORTRAN codes, and then reads and processes the  output  files
c* produced  by the FORTRAN codes.  The  purposes  of  this  driver  are:  to
c* provide  for ease of use of  the  FORTRAN  codes,  preventing  some  common
c* mistakes  such  as  incorrect  input  file  entries; to provide interactive
c* information  during  the  solution  process;  and  to  provide quick, easy,
c* and immediate access to output such as performance information and solution
c* values,  through  the  use  of various windows, displays, and visualization
c* tools.
c*
c* ============
c* MG/XMG Usage
c* ============
c*
c* To  explain how to use XMG, assume you are running the XMG code on a REMOTE
c* machine,  say  a Cray Y-MP with an address or name of "cray.ncsa.uiuc.edu",
c* and  you wish to execute the XMG program remotely from a LOCAL workstation,
c* and display  the  windows and results of the run on your LOCAL workstation.
```

convex_c240

```
% BUILDOPS: operator stencil (lev,numdia) =          3
% BUILDP:   (OPERATOR-27) building prolongation...
% BUILDOPS: (GALER)    [  9,  9,  9]
% BUILDOPS: operator stencil (lev,numdia) =          4
% BUILDP:   (OPERATOR-27) building prolongation...
% BUILDOPS: (GALER)    [  5,  5,  5]
% BUILDOPS: operator stencil (lev,numdia) =          5          14
% BUILDP:   (OPERATOR-27) building prolongation...
% BUILDOPS: (GALER)    [  3,  3,  3]
% BUILDOPS: operator stencil (lev,numdia) =          6          14
% MGDRIV2: coarse problem setup time:    2.286862
 mg = [
% MGDRIV2: linear mode...
% MVCS: starting:      [ 65, 65, 65]    [  3,  3,  3]
% ---------    --------       -----------    -----
% iteration    relative       contraction    wall
% count        residual       number         clock
% ---------    --------       -----------    -----
     -1        0.00E+00       0.0E+00        2.70E+00             %%%
      0        1.00E+00       1.0E+00        2.71E+00             %%%
      1        3.49E-01       3.5E-01        3.71E+00             %%%
      2        1.60E-01       4.6E-01        4.66E+00             %%%
      3        7.49E-02       4.7E-01        5.13E+00             %%%
      4        3.57E-02       4.8E-01        6.11E+00             %%%
      5        1.72E-02       4.8E-01        6.59E+00             %%%
      6        8.32E-03       4.8E-01        7.11E+00             %%%
      7        4.09E-03       4.9E-01        7.55E+00             %%%
      8        2.02E-03       4.9E-01        7.98E+00             %%%
      9        1.01E-03       5.0E-01        8.59E+00             %%%
     10        5.15E-04       5.1E-01        9.02E+00             %%%
% MGDRIV2: solve time:    6.318366
 ];   mg_sf=4.126E-01;   mg_sc=2.287E+00;   mg_st=2.699E+00;   mg_so=6.318E+00;
% MAIN: returned from iterative solver.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% MAIN: total time spent in solve routine:    9.024412
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
STOP: mgmain
Closing the display...
```

XMG Equation Selection:

- Linear Poisson (homo diri)
- Linear Poisson (nonhomo diri)
- Linear Poisson (neu)
- Linear/Nonlinear Poisson-Boltzmann
- Nonlinear Poisson (polynomial)
- Nonlinear Poisson (exponential)
- Linear Groundwater (neu)
- Linear Disc Coef (diri, uniform)
- Linear Disc Coef (diri, chebyshev)



```
Welcome to XMG.
Grabbing the molecule information...
Executing the elliptic solver program...
Interactive output appears in initial window
Opening the method options window...
Opening Linear/Nonlinear Multilevel options
Displaying the discrete mesh...
^
```

XMG Method Control:

X-direction Meshpoints:	65
Y-direction Meshpoints:	65
Z-direction Meshpoints:	65
Fine Mesh Problem Size:	274625
Error Tolerance:	1.0e-05
Iteration Maximum:	10
Stopping Criterion:	2. Rel Resid (1-norm)
Runtime Info Messages:	LOTS
Problem Setup Option:	DEFAULT
Solution I/O Key:	NONE
Performance Analysis:	NO

Solution Method:

```
LINEAR:             Multilevel Preconditioned CG
NONLINEAR:          Damped Inexact Newton-Multilevel
LINEAR/NONLINEAR:   Multilevel
LINEAR/NONLINEAR:   Unpreconditioned Conjugate Gradient
LINEAR/NONLINEAR:   Successive Over-Relaxation
LINEAR/NONLINEAR:   Red/Black Gauss-Seidel
LINEAR/NONLINEAR:   Weighted Jacobi
LINEAR/NONLINEAR:   Richardson
LINEAR:             Preconditioned CG
```

XMG Linear/Nonlinear Multilevel Options:

Linear/Nonlinear:	1. Linear
MG Algorithm:	1. Variable V-Cycle
MG Levels:	6
MG Coarse Mesh Size:	27
MG Pre-Smoothings:	1
MG Post-Smoothings:	1
MG Subspace Solver:	1. Weighted Jacobi
MG Prolongation Operator:	2. Operator-Based
MG Coarse Problem Formulation:	3. Algebraic Galerkin
Fine Problem Discretization:	1. Box Method
Linear Relaxation Parameter:	9.00e-01
Nonlinear Relaxation Parameter:	9.00e-01

Figure B.1: Sample displays produced by the XMG software.

Bibliography

[1] R. A. Adams, *Sobolev Spaces*, Academic Press, San Diego, CA, 1978.

[2] R. E. Alcouffe, A. Brandt, J. E. Dendy, Jr., and J. W. Painter, *The multi-grid method for the diffusion equation with strongly discontinuous coefficients*, SIAM J. Sci. Statist. Comput., 2 (1981), pp. 430–454.

[3] S. A. Allison, J. J. Sines, and A. Wierzbicki, *Solutions of the full Poisson-Boltzmann equation with application to diffusion-controlled reactions*, J. Phys. Chem., 93 (1989), pp. 5819–5823.

[4] V. I. Arnold, *Mathematical Methods of Classical Mechanics*, Springer-Verlag, New York, NY, 1989.

[5] S. Ashby, M. Holst, T. Manteuffel, and P. Saylor, *The role of the inner product in stopping criteria for conjugate gradient iterations*, Tech. Rep. UCRL-JC-112586, Lawrence Livermore National Laboratory, 1992.

[6] S. Ashby, M. Holst, T. Manteuffel, and P. Saylor, *CgCode: A software package for solving linear systems with conjugate gradient methods (version 1.0)*, tech. rep., Lawrence Livermore National Laboratory, 1993.

[7] S. F. Ashby, T. A. Manteuffel, and P. E. Saylor, *A taxonomy for conjugate gradient methods*, Tech. Rep. UCRL-98508, Lawrence Livermore National Laboratory, March 1988. To appear in SIAM J. Numer. Anal.

[8] G. P. Astrakhantsev, *An iterative method of solving elliptic net problems*, USSR Comput. Math. and Math. Phys., 11 (1971), pp. 439–448.

[9] O. Axelsson and V. Barker, *Finite Element Solution of Boundary Value Problems*, Academic Press, Orlando, FL, 1984.

[10] I. Babuśka and A. K. Aziz, *Survey lectures on the mathematical foundations of the finite element method*, in The Mathematical Foundations of the Finite Element Method with Applications to Partial Differential Equations, A. K. Aziz, ed., Academic Press, New York, NY, 1972, pp. 5–359.

[11] N. S. Bakhvalov, *On the convergence of a relaxation method with natural constraints on the elliptic operator*, USSR Comput. Math. and Math. Phys., 6 (1966), pp. 861–885.

[12] R. E. Bank and C. C. Douglas, *Sharp estimates for multigrid rates of convergence with general smoothing and acceleration*, SIAM J. Numer. Anal., 22 (1985), pp. 617–633.

[13] R. E. Bank and T. F. Dupont, *Analysis of a two-level scheme for solving finite element equations*, Tech. Rep. CNA–159, Center for Numerical Analysis, University of Texas at Austin, 1980.

[14] R. E. Bank and T. F. Dupont, *An optimal order process for solving finite element equations*, Math. Comp., 36 (1981), pp. 35–51.

[15] R. E. Bank and D. J. Rose, *Parameter selection for Newton-like methods applicable to nonlinear partial differential equations*, SIAM J. Numer. Anal., 17 (1980), pp. 806–822.

[16] R. E. Bank and D. J. Rose, *Global approximate Newton methods*, Numer. Math., 37 (1981), pp. 279–295.

[17] R. E. Bank and D. J. Rose, *Analysis of a multilevel iterative method for nonlinear finite element equations*, Math. Comp., 39 (1982), pp. 453–465.

[18] R. E. Bank and D. J. Rose, *Some error estimates for the box method*, SIAM J. Numer. Anal., 24 (1987), pp. 777–787.

[19] A. Behie and P. Forsyth, Jr., *Comparison of fast iterative methods for symmetric systems*, IMA Journal of Numerical Analysis, 3 (1983), pp. 41–63.

[20] A. Behie and P. Forsyth, Jr., *Multigrid solution of the pressure equation in reservoir simulation*, Soc. Petr. Engr. J., 23 (1983), pp. 623–632.

[21] F. Bodine, M. Holst, and T. Kerkhoven, *The three-dimensional depletion approximation computed with multigrid*, in Proceedings of the International Workshop on Computational Electronics, Leeds, UK, North Holland, 1993.

[22] P. E. Björstad and J. Mandel, *On the spectra of sums of orthogonal projections with applications to parallel computing*, BIT, 31 (1991), pp. 76–88.

[23] D. Braess and W. Hackbusch, *A new convergence proof for the multigrid method including the V-cycle*, SIAM J. Numer. Anal., 20 (1983), pp. 967–975.

[24] J. H. Bramble and J. E. Pasciak, *New convergence estimates for multigrid algorithms*, Math. Comp., 49 (1987), pp. 311–329.

[25] J. H. Bramble and J. E. Pasciak, *The analysis of smoothers for multigrid algorithms*, Math. Comp., 58 (1992), pp. 467–488.

[26] J. H. Bramble, J. E. Pasciak, J. Wang, and J. Xu, *Convergence estimates for multigrid algorithms without regularity assumptions*, Math. Comp., 57 (1991), pp. 23–45.

[27] J. H. Bramble, J. E. Pasciak, J. Wang, and J. Xu, *Convergence estimates for product iterative methods with applications to domain decomposition and multigrid*, Math. Comp., 57 (1991), pp. 1–21.

[28] A. Brandt, *Multi-level adaptive technique (MLAT) for fast numerical solution to boundary value problems*, in Proceedings of the Third International Conference on Numerical Methods in Fluid Mechanics, Paris, July 1972; Lecture Notes in Physics, Number 18, H. Cabannes and R. Temam, eds., Berlin, 1972, Springer-Verlag, pp. 82–89.

[29] A. Brandt, *Multi-level adaptive solutions to boundary-value problems*, Math. Comp., 31 (1977), pp. 333–390.

[30] A. Brandt, *Algebraic multigrid theory: The symmetric case*, Appl. Math. Comp., 19 (1986), pp. 23–56.

[31] S. L. Brenner and R. E. Roberts, *A variational solution of the Poisson-Boltzmann equation for a spherical colloidal particle*, J. Phys. Chem., 77 (1973), pp. 2367–2370.

[32] J. M. Briggs and J. A. McCammon, *Computation unravels mysteries of molecular biophysics*, Computers in Physics, 6 (1990), pp. 238–243.

[33] F. E. Browder, *Existence theory for boundary value problems for quasi-linear elliptic systems with strongly nonlinear lower order terms*, in Proceedings of the American Mathematical Society Symposium on Partial Differential Equations, 1971.

[34] X.-C. Cai and O. B. Widlund, *Multiplicative Schwarz algorithms for some nonsymmetric and indefinite problems*, Tech. Rep. 595, Courant Institute of Mathematical Science, New York University, New York, NY, 1992.

[35] D. L. Chapman, Phil. Mag., 25 (1913), p. 475.

[36] P. G. Ciarlet, *The Finite Element Method for Elliptic Problems*, North-Holland, New York, NY, 1978.

[37] L. Collatz, *Functional Analysis and Numerical Mathematics*, Academic Press, New York, NY, 1966.

[38] P. Concus, G. H. Golub, and D. P. O'Leary, *A generalized conjugate gradient method for the numerical solution of elliptic partial differential equations*, in Sparse Matrix Computations, J. R. Bunch and D. J. Rose, eds., Academic Press, New York, NY, 1976, pp. 309–332.

[39] N. Davidson, *Statistical Mechanics*, McGraw-Hill Book Company, New York, NY, 1962.

[40] H. T. Davis, *Introduction to Nonlinear Differential and Integral Equations*, Dover Publications, Inc., New York, NY, 1960.

[41] M. E. Davis, J. D. Madura, B. A. Luty, and J. A. McCammon, *Electrostatics and diffusion of molecules in solution: Simulations with the University of Houston Brownian dynamics program*, Comp. Phys. Comm., 62 (1990), pp. 187–197.

[42] M. E. Davis and J. A. McCammon, *Solving the finite difference linearized Poisson-Boltzmann equation: A comparison of relaxation and conjugate gradient methods*, J. Comput. Chem., 10 (1989), pp. 386–391.

[43] M. E. Davis and J. A. McCammon, *Electrostatics in biomolecular structure and dynamics*, Chem. Rev., 90 (1990), pp. 509–521.

[44] L. Debnath and P. Mikusinński, *Introduction to Hilbert Spaces with Applications*, Academic Press, New York, NY, 1990.

[45] P. Debye and E. Hückel, Physik. Z., 24 (1923), p. 185.

[46] N. Decker, J. Mandel, and S. Parter, *On the role of regularity in multigrid methods*, in Multigrid Methods: Theory, Application, and Supercomputing, S. McCormick, ed., Marcel Dekker, Inc., 1988, pp. 140–177.

[47] R. S. Dembo, S. C. Eisenstat, and T. Steihaug, *Inexact Newton methods*, SIAM J. Numer. Anal., 19 (1982), pp. 400–408.

[48] J. E. Dendy, Jr., *Black box multigrid*, J. Comput. Phys., 48 (1982), pp. 366–386.

[49] J. E. Dendy, Jr., *Black box multigrid for nonsymmetric problems*, Appl. Math. Comp., 13 (1983), pp. 261–283.

[50] J. E. Dendy, Jr., *Black box multigrid for systems*, Appl. Math. Comp., 19 (1986), pp. 57–74.

[51] J. E. Dendy, Jr., *Two multigrid methods for three-dimensional problems with discontinuous and anisotropic coefficients*, SIAM J. Sci. Statist. Comput., 8 (1987), pp. 673–685.

[52] J. E. Dendy, Jr., *Black box multigrid for periodic and singular problems*, Appl. Math. Comp., 25 (1988), pp. 1–10.

[53] J. E. Dendy, Jr. and J. M. Hyman, *Multi-grid and ICCG for problems with interfaces*, in Elliptic Problem Solvers, M. Schultz, ed., New York, NY, 1981, Academic Press.

[54] J. E. Dennis, Jr. and J. J. Moré, *A characterization of superlinear convergence and its application to quasi-Newton methods*, Math. Comp., 28 (1974), pp. 549–560.

[55] J. E. Dennis, Jr. and J. J. Moré, *Quasi-Newton methods, motivation and theory*, Siam Review, 19 (1977), pp. 46–89.

[56] J. E. Dennis, Jr. and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1983.

[57] C. C. Douglas, *Multi-grid algorithms for elliptic boundary-value problems*, Tech. Rep. 223, Dept. of Computer Science, Yale University, 1982.

[58] M. Dryja and O. B. Widlund, *Towards a unified theory of domain decomposition algorithms for elliptic problems*, in Third International Symposium on Domain Decomposition Methods for Partial Differential Equations, T. F. Chan, R. Glowinski, J. Périaux, and O. B. Widlund, eds., Philadelphia, PA, 1989, SIAM, pp. 3–21.

[59] M. Dryja and O. B. Widlund, *Domain decomposition algorithms with small overlap*, Tech. Rep. 606, Courant Institute of Mathematical Science, New York University, New York, NY, 1992.

[60] D. E. Edmunds, V. B. Moscatelli, and J. R. L. Webb, *Strongly nonlinear elliptic operators in unbounded domains*, Publ. Math. Bordeaux, 4 (1974), pp. 6–32.

[61] S. C. Eisenstat and H. F. Walker, *Globally convergent inexact Newton methods*, tech. rep., Dept. of Mathematics and Statistics, Utah State University, 1992.

[62] I. Ekland and R. Temam, *Convex Analysis and Variational Problems*, North-Holland, New York, NY, 1976.

[63] R. D. Falgout, *Algebraic-Geometric Multigrid Methods for Poisson-Type Equations*, PhD thesis, Dept. of Applied Mathematics, University of Virginia, May 1991.

[64] R. P. Fedorenko, *A relaxation method for solving elliptic difference equations*, USSR Comput. Math. and Math. Phys., 1 (1961), pp. 1092–1096.

[65] R. P. Fedorenko, *The speed of convergence of one iterative process*, USSR Comput. Math. and Math. Phys., 4 (1964), pp. 227–235.

[66] R. Fletcher and C. Reeves, *Function minimization by conjugate gradients*, Comput. J., 7 (1964), pp. 149–154.

[67] B. Friedman, *Principles and Techniques of Applied Mathematics*, Dover Publications, New York, NY, 1956.

[68] S. Fucik and A. Kufner, *Nonlinear Differential Equations*, Elsevier Scientific Publishing Company, New York, NY, 1980.

[69] H. Fujita, *On the nonlinear equations $\Delta u + e^u = 0$ and $\partial v/\partial t = \Delta v + e^v$*, Bull. Amer. Math. Soc., 75 (1969), pp. 132–135.

[70] D. Gilbarg and N. S. Trudinger, *Elliptic Partial Differential Equations of Second Order*, Springer-Verlag, New York, NY, 1977.

[71] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization*, Academic Press, New York, NY, 1981.

[72] M. K. Gilson, K. A. Sharp, and B. H. Honig, *Calculating the electrostatic potential of molecules in solution: Method and error assessment*, J. Comput. Chem., 9 (1988), pp. 327–335.

[73] R. Glowinski, J. L. Lions, and R. Trémolières, *Numerical Analysis of Variational Inequalities*, North-Holland, New York, NY, 1981.

[74] C. I. Goldstein, *Multigrid analysis of finite element methods with numerical integration*, Math. Comp., 56 (1991), pp. 409–436.

[75] L. Greengard, *The Rapid Evaluation of Potential Fields in Particle Systems*, PhD thesis, Department of Computer Science, Yale University, April, 1987. Also available as Research Report YALEU/DCS/RR-533.

[76] P. Grisvard, *Elliptic Problems in Nonsmooth Domains*, Pitman Publishing, Marshfield, MA, 1985.

[77] G. Guoy, J. Phys., 9 (1910), p. 457.

[78] W. Hackbusch, *Ein Iteratives Verfahren zur Schnellen Auflösung Elliptischer Randwert–Problem*, Tech. Rep. 76–12, Universität zu Köln, 1976.

[79] W. Hackbusch, *A fast iterative method solving Poisson's equation in a general region*, in Proceedings of the Conference on the Numerical Treatment of Differential Equations, Oberwolfach, July 1976; Lecture Notes in Mathematics, Number 631, R. Bulirsch, R. D. Griegorieff, and J. Schröder, eds., Berlin, 1978, Springer-Verlag, pp. 51–62.

[80] W. Hackbusch, *On the fast solutions of nonlinear elliptic equations*, Numer. Math., 32 (1979), pp. 83–95.

[81] W. Hackbusch, *Survey of convergence proofs for multi-grid iterations*, in Conference Proceedings, Bonn, Oct. 1979; Special Topics in Applied Mathematics, J. Frehse, D. Pallaschke, and U. Trottenberg, eds., Amsterdam, 1979, North-Holland, pp. 151–164.

[82] W. Hackbusch, *Convergence of multi-grid iterations applied to difference equations*, Math. Comp., 34 (1980), pp. 425–440.

[83] W. Hackbusch, *On the convergence of multi-grid iterations*, Beiträge Numer. Math., 9 (1981), pp. 213–239.

[84] W. Hackbusch, *Multi-grid convergence theory*, in Multigrid Methods: Proceedings of Köln-Porz Conference on Multigrid Methods, Lecture notes in Mathematics 960, W. Hackbusch and U. Trottenberg, eds., Berlin, Germany, 1982, Springer-Verlag.

[85] W. Hackbusch, *Multi-grid Methods and Applications*, Springer-Verlag, Berlin, Germany, 1985.

[86] W. Hackbusch, *Iterative Solution of Large Sparse Systems of Equations*, Springer-Verlag, Berlin, Germany, 1994.

[87] W. Hackbusch and A. Reusken, *On global multigrid convergence for nonlinear problems*, in Robust Multigrid Methods, W. Hackbusch, ed., Braunschweig, 1988, Vieweg, pp. 105–113.

[88] W. Hackbusch and A. Reusken, *Analysis of a damped nonlinear multilevel method*, Numer. Math., 55 (1989), pp. 225–246.

[89] P. R. Halmos, *Introduction to Hilbert Space*, Chelsea Publishing Company, New York, NY, 1957.

[90] P. R. Halmos, *Finite-Dimensional Vector Spaces*, Springer-Verlag, Berlin, Germany, 1958.

[91] H. S. Harned and B. B. Owen, *The Physical Chemistry of Electrolytic Solutions*, Reinhold Publishing Company, New York, NY, 1958.

[92] P. Hess, *A strongly nonlinear elliptic boundary value problem*, J. Math. Anal. and Appl., 43 (1973), pp. 241–249.

[93] M. R. Hestenes and E. Stiefel, *Methods of conjugate gradients for solving linear systems*, J. Research of NBS, 49 (1952), pp. 409–435.

[94] M. Holst, *The Poisson-Boltzmann Equation: Analysis and Multilevel Numerical Solution.* Unpublished report, 1994 (updated and extended form of the Ph.D. thesis [97]).

[95] M. Holst, *CgCode: Software for solving linear systems with conjugate gradient methods*, Master's thesis, Numerical Computing Group, Department of Computer Science, University of Illinois at Urbana-Champaign, May 1990.

[96] M. Holst, *Notes on the KIVA-II software and chemically reactive fluid mechanics*, Tech. Rep. UCRL-ID-112019, Lawrence Livermore National Laboratory, 1992.

[97] M. Holst, *Multilevel Methods for the Poisson-Boltzmann Equation*, PhD thesis, Numerical Computing Group, Science, University of Illinois at Urbana-Champaign, 1993. Also published as Tech. Rep. UIUCDCS-R-03-1821.

[98] M. Holst, *An Algebraic Schwarz Theory*, Tech. Rep. CRPC-94-12, Applied Mathematics and CRPC, California Institute of Technology, 1994.

[99] M. Holst, *A robust and efficient numerical method for nonlinear protein modeling equations*, Tech. Rep. CRPC-94-9, Applied Mathematics and CRPC, California Institute of Technology, 1994.

[100] M. Holst, R. Kozack, F. Saied, and S. Subramaniam, *Treatment of electrostatic effects in proteins: Multigrid-based Newton iterative method for solution of the full nonlinear Poisson-Boltzmann equation*, Tech. Rep. UIUC-BI-MB-93-01, The Beckman Institute for Advanced Science and Technology, 1993.

[101] M. Holst, R. Kozack, F. Saied, and S. Subramaniam, *Protein electrostatics: Rapid multigrid-based Newton algorithm for solution of the full nonlinear Poisson-Boltzmann equation*, J. Biomol. Struct. Dyn., 11 (1994), pp. 1437–1445.

[102] M. Holst, R. Kozack, F. Saied, and S. Subramaniam, *Treatment of electrostatic effects in proteins: Multigrid-based-Newton iterative method for solution of the full nonlinear Poisson-Boltzmann equation*, Proteins: Structure, Function, and Genetics, 18 (1994), pp. 231–245.

[103] M. Holst and F. Saied, *Vector multigrid: An accuracy and performance study*, Tech. Rep. UIUCDCS-R-90-1636, Numerical Computing Group, Department of Computer Science, University of Illinois at Urbana-Champaign, 1990.

[104] M. Holst and F. Saied, *Multigrid methods for computational ocean acoustics on vector and parallel computers*, in Proceedings of the Third IMACS Symposium on Computational Acoustics, New York, NY, North Holland, 1991.

[105] M. Holst and F. Saied, *Parallel performance of some multigrid solvers for three-dimensional parabolic equations*, Tech. Rep. UIUCDCS-R-91-1697, Numerical Computing Group, Department of Computer Science, University of Illinois at Urbana-Champaign, 1991.

[106] M. Holst and F. Saied, *Multigrid solution of the Poisson-Boltzmann equation*, Tech. Rep. UIUCDCS-R-92-1744, Numerical Computing Group, Department of Computer Science, University of Illinois at Urbana-Champaign, 1992.

[107] M. Holst and F. Saied, *Multigrid solution of the Poisson-Boltzmann equation*, J. Comput. Chem., 14 (1993), pp. 105–113.

[108] M. Holst and F. Saied, *A short note comparing multigrid and domain decomposition for protein modeling equations*, Tech. Rep. CRPC-94-10, Applied Mathematics and CRPC, California Institute of Technology, 1994.

[109] M. Holst and F. Saied, *Multigrid and domain decomposition methods for electrostatics problems*, in Domain Decomposition Methods in Science and Engineering (Proceedings of the Seventh International Conference on Domain Decomposition, October 27-30, 1993, The Pennsylvania State University), D. E. Keyes and J. Xu, eds., American Mathematical Society, Providence, 1995.

[110] M. Holst and F. Saied, *Numerical solution of the nonlinear Poisson-Boltzmann equation: Developing more robust and efficient methods*, J. Comput. Chem., 16 (1995), pp. 337–364.

[111] E. Isaacson and H. B. Keller, *Analysis of Numerical Methods*, John Wiley & Sons, Inc., New York, NY, 1966.

[112] B. Jayaram, K. A. Sharp, and B. Honig, *The electrostatic potential of B-DNA*, Biopolymers, 28 (1989), pp. 975–993.

[113] J. W. Jerome, *Approximation of Nonlinear Evolution Systems*, Academic Press, New York, NY, 1983.

[114] J. W. Jerome, *Consistency of semiconductor modeling: An existence/stability analysis for the stationary van roosbroeck system*, SIAM J. Appl. Math., 45 (1985), pp. 565–590.

[115] F. Jones, *Lebesgue Integration on Euclidean Space*, Jones and Bartlett Publishers, Inc., Boston, MA, 1993.

[116] A. H. Juffer, E. F. F. Botta, B. A. M. van Keulen, A. van der Ploeg, and H. J. C. Berendsen, *The electric potential of a macromolecule in a solvent: A fundamental approach*, J. Comput. Phys., 97 (1991), pp. 144–171.

[117] L. V. Kantorovich and G. P. Akilov, *Functional Analysis*, Pergamon Press, New York, NY, 1982.

[118] L. V. Kantorovich and V. I. Krylov, *Approximate Methods of Higher Analysis*, P. Noordhoff, Ltd, Groningen, The Netherlands, 1958.

[119] J. P. Keener, *Principles of Applied Mathematics*, Addison-Wesley Publishing Company, Redwood City, CA, 1988.

[120] H. B. Keller, *Numerical Methods for Two-Point Boundary-Value Problems*, Dover Publications, New York, NY, 1992.

[121] O. D. Kellog, *Foundations of Potential Theory*, Dover Publications, New York, NY, 1953.

[122] T. Kerkhoven, *Piecewise linear Petrov-Galerkin analysis of the box-method*, SIAM J. Numer. Anal., (1997). (To appear).

[123] T. Kerkhoven and J. W. Jerome, *L_∞ stability of finite element approximations of elliptic gradient equations*, Numer. Math., 57 (1990), pp. 561–575.

[124] S. Kesavan, *Topics in Functional Analysis and Applications*, John Wiley & Sons, Inc., New York, NY, 1989.

[125] A. N. Kolmogorov and S. V. Fomin, *Introductory Real Analysis*, Dover Publications, New York, NY, 1970.

[126] M. Kovˇara and J. Mandel, *A multigrid method for three-dimsional elasticity and algebraic convergence estimates*, Appl. Math. Comp., 23 (1987), pp. 121–135.

[127] R. Kozack, *Private communication*, 1992.

[128] R. Kress, *Linear Integral Equations*, Springer-Verlag, Berlin, Germany, 1989.

[129] E. Kreyszig, *Introductory Functional Analysis with Applications*, John Wiley & Sons, Inc., New York, NY, 1990.

[130] A. Kufner, O. John, and S. Fucik, *Function Spaces*, Noordhoff International Publishing, Leyden, The Netherlands, 1977.

[131] C. Lanczos, *The Variational Principles of Mechanics*, Dover Publications, Inc., New York, NY, 1949.

[132] P. D. Lax, *Theory of Functions of a Real Variable*, New York University, New York, NY, 1959.

[133] P. L. Lions, *On the Schwarz Alternating Method. I*, in First International Symposium on Domain Decomposition Methods for Partial Differential Equations, R. Glowinski, G. H. Golub, G. A. Meurant, and J. Périaux, eds., Philadelphia, PA, 1988, SIAM, pp. 1–42.

[134] C. Liu, Z. Liu, and S. McCormick, *An efficient multigrid scheme for elliptic equations with discontinuous coefficients*, Tech. Rep. (In preparation), Computational Mathematics Group, University of Colorado at Denver, 1991.

[135] B. A. Luty, M. E. Davis, and J. A. McCammon, *Solving the finite-difference non-linear Poisson-Boltzmann equation*, J. Comput. Chem., 13 (1992), pp. 1114–1118.

[136] J.-F. Maitre and F. Musy, *Multigrid methods: Convergence theory in a variational framework*, SIAM J. Numer. Anal., 21 (1984), pp. 657–671.

[137] J.-F. Maitre and F. Musy, *Multigrid methods for symmetric variational problems: A general theory and convergence estimates for usual smoothers*, Appl. Math. Comp., 21 (1987), pp. 21–43.

[138] J. Mandel, *On some two-level iterative methods*, in Defect Correction Methods, K. Böhmer and H. J. Stetter, eds., Springer Verlag, 1984, pp. 75–88.

[139] J. Mandel, *Multigrid convergence for nonsymmetric, indefinite variational problems and one smoothing step*, in Proceedings of the Second Copper Mountain Conference on Multigrid Methods, S. McCormick, ed., Marcel Dekker, 1986, pp. 201–216.

[140] J. Mandel, *On multigrid and iterative aggregation methods for nonsymmetric problems*, in Multigrid Methods II: Proceedings of the Second European Conference on Multigrid Methods held at Cologne, W. Hackbusch and U. Trottenberg, eds., Berlin, Germany, 1987, Springer-Verlag, pp. 219–231.

[141] J. Mandel, *Some recent advances in multigrid methods*, Advances in Electronics and Electron Physics, 82 (1991), pp. 327–377.

[142] J. Mandel, S. McCormick, and R. Bank, *Variational multigrid theory*, in Multigrid Methods, S. McCormick, ed., SIAM, 1987, pp. 131–177.

[143] S. F. McCormick, *An algebraic interpretation of multigrid methods*, SIAM J. Numer. Anal., 19 (1982), pp. 548–560.

[144] S. F. McCormick, *Multigrid methods for variational problems: Further results*, SIAM J. Numer. Anal., 21 (1984), pp. 255–263.

[145] S. F. McCormick, *Multigrid methods for variational problems: General theory for the V-cycle*, SIAM J. Numer. Anal., 22 (1985), pp. 634–643.

[146] S. F. McCormick and J. W. Ruge, *Multigrid methods for variational problems*, SIAM J. Numer. Anal., 19 (1982), pp. 924–929.

[147] S. F. McCormick and J. W. Ruge, *Unigrid for multigrid simulation*, Math. Comp., 41 (1983), pp. 43–62.

[148] D. A. McQuarrie, *Statistical Mechanics*, Harper and Row, New York, NY, 1973.

[149] S. G. Nash, *Truncated-Newton Methods*, PhD thesis, Deptartment of Computer Science, Stanford University, 1982.

[150] J. Nečas, *Méthodes Directes en Théorie des Équations Elliptiques*, Academia, Prague, Czechoslovakia, 1967.

[151] A. Nicholls, *Private communication*, 1993.

[152] A. Nicholls and B. Honig, *A rapid finite difference algorithm, utilizing successive over-relaxation to solve the Poisson-Boltzmann equation*, J. Comput. Chem., 12 (1991), pp. 435–445.

[153] R. A. Nicolaides, *On the l^2 convergence of an algorithm for solving finite element equations*, Math. Comp., 31 (1977), pp. 892–906.

[154] C. Niedermeier and K. Schulten, *Molecular dynamics simulations in heterogeneous dielectrica and Debye-Hückel media – application to the protein bovine pancreatic trypsin inhibitor*, tech. rep., Department of Physics and Beckman Institute, University of Illinois at Urbana-Champaign, 1990.

[155] D. I. Okunbor, *Canonical Integration Methods for Hamiltonian Dynamical Systems*, PhD thesis, Department of Computer Science, University of Illinois at Urbana-Champaign, 1992. Available as Technical Report UIUCDCS-R-92-1785.

[156] E. G. Ong, *Uniform refinement of a tetrahedron*, Tech. Rep. CAM 91-01, Department of Mathematics, UCLA, 1991.

[157] J. M. Ortega, *Numerical Analysis: A Second Course*, Academic Press, New York, NY, 1972.

[158] J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York, NY, 1970.

[159] P. Oswald, *Stable subspace splittings for Sobolev spaces and their applications*, Tech. Rep. MATH-93-7, Institut für Angewandte Mathematik, Friedrich-Schiller-Universitatät Jena, D-07740 Jena, FRG, September 1993.

[160] A. A. Rashin and J. Malinsky, *New method for the computation of ionic distribution around rod-like polyelectrolytes with helical distribution of charges. I. General approach and a nonlinearized Poisson-Boltzmann equation*, J. Comput. Chem., 12 (1991), pp. 981–993.

[161] K. Rektorys, *Variational Methods in Mathematics, Science and Engineering*, D. Reidel Publishing Company, Boston, MA, 1977.

[162] M. Renardy and R. C. Rogers, *An Introduction to Partial Differential Equations*, Springer-Verlag, New York, NY, 1993.

[163] A. Reusken, *Convergence of the multigrid full approximation scheme for a class of elliptic mildly nonlinear boundary value problems*, Numer. Math., 52 (1988), pp. 251–277.

[164] A. Reusken, *Convergence of the multilevel full approximation scheme including the V-cycle*, Numer. Math., 53 (1988), pp. 663–686.

[165] U. Rüde, *Mathematical and Computational Techniques for Multilevel Adaptive Methods*, vol. 13 of SIAM Frontiers Series, SIAM, Philadelphia, PA, 1993.

[166] J. W. Ruge and K. Stüben, *Algebraic multigrid*, in Multigrid Methods, S. McCormick, ed., SIAM, 1987, pp. 73–130.

[167] M. Schechter, *Modern Methods in Partial Differential Equations*, McGraw-Hill, New York, NY, 1977.

[168] R. B. Setlow and E. C. Pollard, *Molecular Biophysics*, Addison-Wesley Publishing Company, Reading, MA, 1962.

[169] K. A. Sharp, *Incorporating solvent and ion screening into molecular dynamics using the finite-difference Poisson-Boltzmann method*, J. Comput. Chem., 12 (1991), pp. 454–468.

[170] K. A. Sharp and B. Honig, *Calculating total electrostatic energies with the nonlinear Poisson-Boltzmann equation*, J. Phys. Chem., 94 (1990), pp. 7684–7692.

[171] K. A. Sharp and B. Honig, *Electrostatic interactions in macromolecules: Theory and applications*, Annu. Rev. Biophys. Biophys. Chem., 19 (1990), pp. 301–332.

[172] R. E. Showalter, *Hilbert Space Methods for Partial Differential Equations*, Pitman Publishing, Marshfield, MA, 1979.

[173] R. V. Southwell, *Relaxation Methods in Theoretical Physics*, Clarendon Press, Oxford, 1946.

[174] G. Strang and G. Fix, *An Analysis of the Finite Element Method*, Prentice-Hall, Englewood Cliffs, NJ, 1973.

[175] K. Stüben and U. Trottenberg, *Multigrid methods: Fundamental algorithms, model problem analysis and applications*, in Multigrid Methods: Proceedings of Köln-Porz Conference on Multigrid Methods, Lecture notes in Mathematics 960, W. Hackbusch and U. Trottenberg, eds., Berlin, Germany, 1982, Springer-Verlag.

[176] C. Tanford, *Physical Chemistry of Macromolecules*, John Wiley & Sons, New York, NY, 1961.

[177] A. C. Ting, H. H. Chen, and Y. C. Lee, *Exact solutions of a nonlinear boundary value problem: The vortices of the two-dimensional sinh-Poisson equation*, Physica D, 26 (1987), pp. 37–66.

[178] H. A. van der Vorst, *High performance preconditioning*, SIAM J. Sci. Statist. Comput., 10 (1989), pp. 1174–1185.

[179] R. S. Varga, *Matrix Iterative Analysis*, Prentice-Hall, Englewood Cliffs, NJ, 1962.

[180] E. L. Wachspress, *Iterative Solution of Elliptic Systems and Applications to the Neutron Diffusion Equations of Reactor Physics*, Prentice-Hall, Englewood Cliffs, NJ, 1966.

[181] J. Wang, *Convergence analysis without regularity assumptions for multigrid algorithms based on SOR smoothing*, SIAM J. Numer. Anal., 29 (1992), pp. 987–1001.

[182] P. Wesseling, *A convergence proof for a multiple grid method*, in Numerical Analysis, Proceedings from Dundee 1979, Lecture Notes in Mathematics 733, G. A. Watson, ed., Berlin, Germany, 1980, Springer-Verlag.

[183] O. B. Widlund, *Optimal iterative refinement methods*, in Third International Symposium on Domain Decomposition Methods for Partial Differential Equations, T. F. Chan, R. Glowinski, J. Périaux, and O. B. Widlund, eds., Philadelphia, PA, 1989, SIAM, pp. 114–125.

[184] J. Xu, *Theory of Multilevel Methods*, PhD thesis, Department of Mathematics, Penn State University, University Park, PA, July 1989. Technical Report AM 48.

[185] J. Xu, *Iterative methods by space decomposition and subspace correction*, SIAM Review, 34 (1992), pp. 581–613.

[186] B. J. Yoon and A. M. Lenhoff, *A boundary element method for molecular electrostatics with electrolyte effects*, J. Comput. Chem., 11 (1990), pp. 1080–1086.

[187] D. M. Young, *Iterative Solution of Large Linear Systems*, Academic Press, New York, NY, 1971.

[188] H. Yserentant, *On the convergence of multi-level methods for strongly nonuniform families of grids and any number of smoothing steps per level*, Computing, 30 (1983), pp. 305–313.

[189] H. Yserentant, *Old and new convergence proofs for multigrid methods*, Acta Numerica, (1993), pp. 285–326.

[190] D. Zwillinger, *Handbook of Differential Equations*, Academic Press, San Diego, CA, 1992.

List of Figures

List of Tables

Vita

Michael Jay Holst was born on the 4th of January, 1964 in St. Paul, Minnesota. Shortly thereafter, he moved with his parents Dale and Shirley, and three brothers Greg, Brian, and Jon, to the rugged land of Colorado, where he spent the remainder of his youth. He attended high school in Berthoud, Colorado during the brief periods of time not spent throwing the discus and working odd jobs, until his graduation from high school in 1982.

After spending a semester at nearly every major university west of the Mississippi River, taking courses mainly in philosophy, mathematics, and computer science, he graduated in 1987 from Colorado State University with a degree in mathematics and computer science. During his undergraduate studies, Michael had several "internships" to prepare him for a possible future life in academics, which included working as a cook, dishwasher, waiter, construction worker, rose picker, painter, and grave digger.

Deciding against a promising career as a coffee importer, Michael moved to the University of Illinois in 1987 to attend graduate school in numerical analysis, which seemed to be the ideal compromise to satisfy his strong interests in both mathematics and computer science. After struggling through exams and the difficult task of finding a topic which was interesting, manageable, and something that his advisor would buy, he settled on numerical solution of nonlinear partial differential equations, focusing on a particularly interesting application from biophysics. Several busy years later, which included travels through India, Thailand, and Japan to keep his continuing interests in philosophy at bay, he completed his thesis.

Michael currently resides in Pasadena, California, pursuing postdoctoral research on nonlinear partial differential equations in the Applied Mathematics Department/CRPC at the California Institute of Technology.

Quote from an anonymous triathaloner (i.e., graduate student):

> *I'm of the opinion, that if you are in the middle of doing something difficult, and you keep doing it long enough, eventually it will end.*