

Rochester Institute of Technology

RIT Digital Institutional Repository

Theses

5-17-2013

Multigrid Solution of the Debye-Hückel Equation

Benjamin Quanah Parker

Follow this and additional works at: <https://repository.rit.edu/theses>

Recommended Citation

Parker, Benjamin Quanah, "Multigrid Solution of the Debye-Hückel Equation" (2013). Thesis. Rochester Institute of Technology. Accessed from

This Thesis is brought to you for free and open access by the RIT Libraries. For more information, please contact repository@rit.edu.

Multigrid Solution of the Debye-Hückel Equation
by
Benjamin Quanah Parker

A Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of Master of Science

School of Mathematical Sciences
College of Science

Thesis Committee:

David S. Ross
Chris W. Wahle
George M. Thurston

Rochester Institute of Technology
Rochester, NY
May 17, 2013

Abstract

Multigrid algorithms are fast solvers for elliptic partial differential equations. In this thesis, we apply multigrid methods to the model of protein charge-regulation of Hollenbeck, et al. The model of protein charge-regulation requires computing work of charging matrices for two low dielectric spheres in a salt solution, which in turn requires many solutions of the linearized Poisson-Boltzmann equation, or the Debye-Hückel equation. We use multigrid methods to reduce the run time of computing solutions to the Debye-Hückel equation, and compare the results of some simple and more complicated examples. Using the work of Brandt and others, we also construct an interpolation scheme that takes the potentially complicated behavior of the coefficient into account.

Introduction

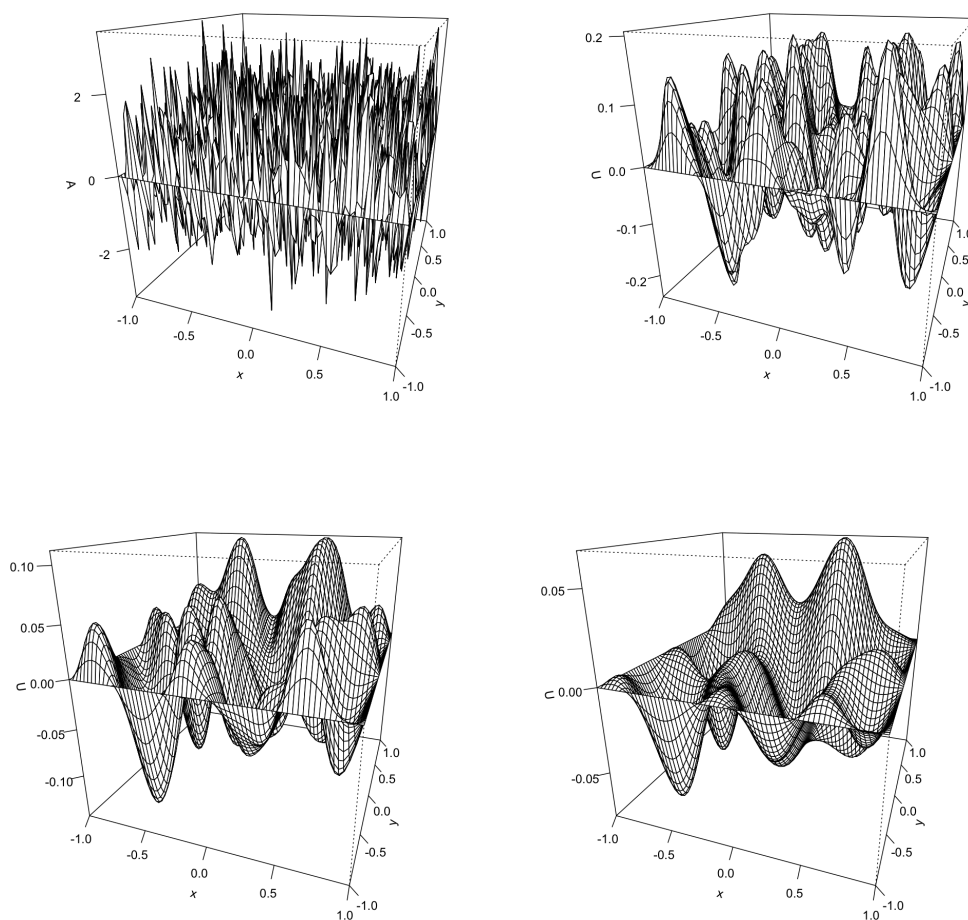
Introduction to Multigrid

Multigrid methods are a set of algorithms used to solve boundary value problems which have had notable success in creating fast and accurate numerical solutions for elliptic partial differential equations. Multigrid methods are comprised of different working pieces, or components. These components consist of prolongation, or interpolation to finer grids, restriction of numerical approximations to coarser grids, and smoothing of errors. The creation and success of multigrid methods stems from research conducted by Brandt, Hackbush, and others [8]. Multigrid Methods are ideal for solving linear, elliptic partial differential equations. These methods have also been extended to other classes of problems, including hyperbolic and parabolic partial differential equations [36].

When solving linear elliptic partial differential equations, one obtains greater accuracy by solving the given problem on a large set of points. Brandt notes that as the problem becomes larger in size, the resources used are often wasted because the size of the discrete problem has a notable influence on the accuracy of the numerical solution [8]. Multigrid methods are a means of taking these large problems and solving related, smaller problems which provide precise numerical solutions at a reduced cost. That is, one solves smaller problems on coarser grids that will provide approximations to the same problems on finer grids; these methods save time and memory resources while providing precise numerical solutions to partial dif-

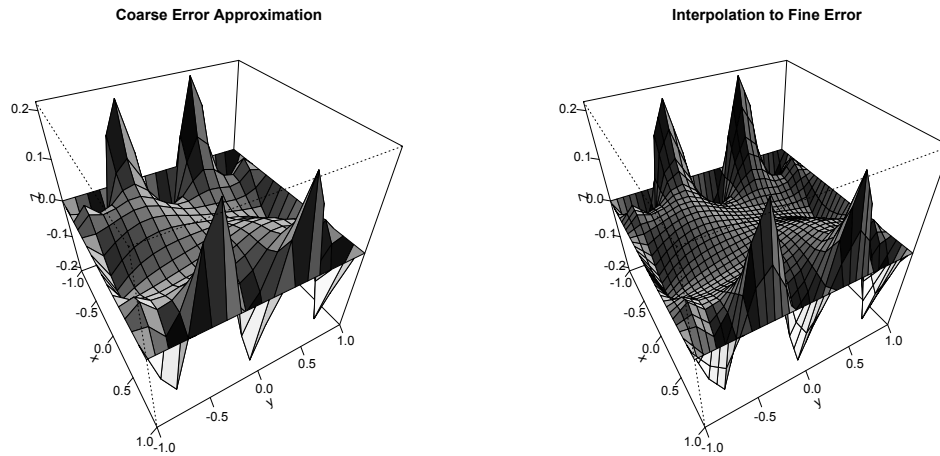
ferential equations.

The basic components of multigrid methods are that of smoothing, restriction, and interpolation. As Brandt [8] notes, the error can be represented as a Fourier Series, on which the smoother acts to dampen highly-oscillatory, or high frequency errors; one can employ the Jacobi or Gauss-Seidel method to use as a smoother. Since the convergence of these methods as stand-alone solvers depends on the size of the problem, convergence begins to slow down once there are no more high-frequency errors to dampen. The remaining errors are of low-frequency, and for the given size of problem, do not oscillate rapidly relative to the scale of the grid:

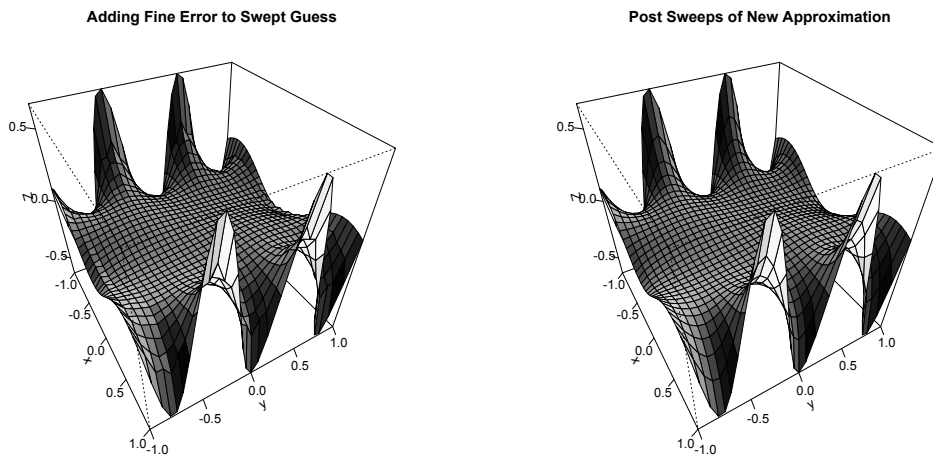


At this point, one switches to a coarser problem and solves a related problem for the error,

or correction. This approximation of the correction is then interpolated to the fine grid.



The interpolated correction is added to the approximation on the fine grid. One then performs additional sweeps of the smoother to the new approximation on the fine grid.



The same components of multigrid are then applied to the error, at which point the error is added back to the original numerical approximation and smoothing is applied again. One typically applies this process repeatedly until the desired accuracy is obtained. Variants of this process include the V-Cycle, W-Cycle, and the Full-Multigrid Cycle which we introduce in Chapter 3.

The Debye-Hückel Equation

For this thesis, we are interested in finding fast numerical solutions to the Debye-Hückel equation below given by Hollenbeck et al. [13]:

$$-\nabla \cdot (\epsilon \nabla \varphi) + \kappa^2 \varphi = \rho.$$

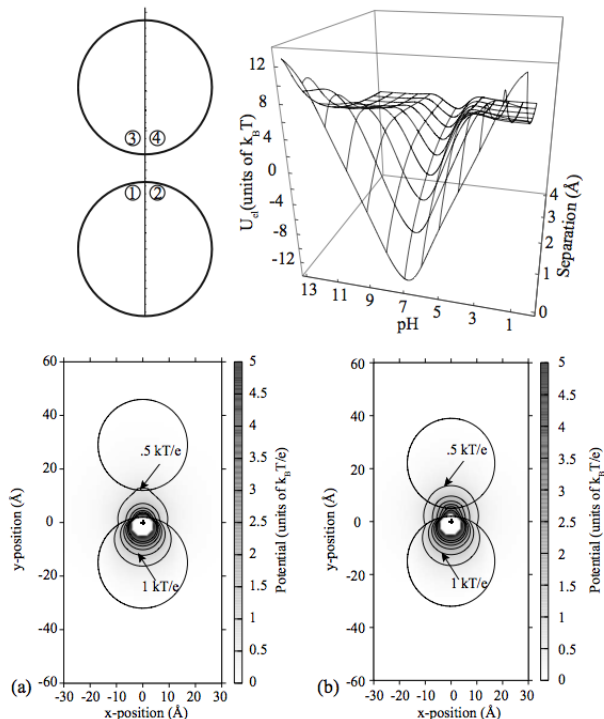
This equation is a linearized Poisson-Boltzmann equation. The linearization corresponds to the Debye-Hückel Theory of dilute electrolyte solutions [22]. In the above equation, ϵ is the dielectric coefficient, κ is the Debye screening parameter, and ρ is the charge distribution. The function ϵ , the dielectric coefficient, is a measure of the amount of charge which a medium, otherwise known as a dielectric, can store charge. Similarly, κ^2 represents the ionic strength of the salt solution in which the proteins are immersed, and varies in space depending upon the composition of the solution the proteins inhabit (citation here!).

The work-of-charging matrices [13] to be computed for the model of protein charge regulation requires many solutions of the Debye-Hückel equation because of the numerous configurations of charged sites on the proteins. Developing a fast solver for this equation is therefore important to further progress in research. For the model developed by Hollenbeck, et al. [13], the proteins are represented by two low-dielectric spheres immersed in a salt solution.

The phenomenon of charge regulation influences the attraction between protein molecules [13]. Professor Thurston and his co-workers at RIT are particularly interested in its influence on the alpha-, beta-, and gamma-crystallin proteins in the lens of the human eye, and its role in cataract disease. The charged state of a protein varies with time in an ionic solution, as random thermal agitation and electrical attraction combine to attract protons to chargeable sites on the protein and to dislodge them. The statistics of this random process are studied

through work-of-charging matrices, matrices that represent the influences of chargeable sites on one another via electrical potentials. To compute such work-of-charging matrices one must solve a Debye-Huckel equation for each chargeable site on the protein configuration of interest. The RIT group is studying configurations with 50-100 chargeable sites, so the computation of a single work-of-charging matrix requires the solution of that number of three-dimensional Debye-Huckel problems. Speeding up these calculations is important to the progress of this research.

In the problems in which the RIT group is interested, the proteins are represented as low-dielectric spheres in a high-dielectric ionic solution.



In this model of protein charge-regulation, both the dielectric coefficient ϵ and the Debye screening parameter κ are discontinuous, piecewise constant functions of position. We will construct and tailor the standard multigrid method components to accurately represent the numerical solutions of the electric potential φ .

Literature Review of Multigrid: Prior and Current Research

The multigrid methods gained widespread attention, in part from an early paper by Brandt [8]. This family of methods combines corrections of numerical solutions to boundary value problems with smoothing methods such as Gauss-Seidel to obtain fast and precise numerical methods for solving elliptic partial differential equations. Brandt introduces a number of multigrid algorithms used to solve linear and non-linear elliptic partial differential equations. In addition, Brandt explains how one can derive smoothing and convergence factors that give estimates for the efficiency and speed of these methods [8].

One may use other methods such as iterant recombination to improve the accuracy of numerical solutions obtained using multigrid. [36]. Another approach involves the use of Krylov Subspace methods such as the Conjugate Gradient method. Pflaum, for example, introduces the use of the Conjugate Gradient method into multigrid methods. Pflaum's results showed that inserting a V-Cycle after every tenth iteration gave favorable errors when dealing with non-smooth and jumping coefficients [38].

Others have looked at modifying specific components of the multigrid algorithm with notable success. The methods that have been improved focus on the prolongation and restriction operators, while others have focused on iterant recombination and residual scaling [48–50]. The latter of these methods accelerates the convergence of the multigrid methods when slow convergence is detected. As is noted by Zhang [46], smoothing the error components can also come at a certain cost, particularly since SOR may not distinguish between low- and high-frequency error components; while some may be reduced, others will not be affected depending on the parameter ω chosen for the relaxation methods. As a result, Zhang derives two relaxation parameters which reduce both low and high frequencies.

In addition to accelerating convergence, other methods have been adapted to deal with discontinuous coefficients. Zeeuw [45], for example, creates a black box multi-grid solve for linear elliptic partial differential equations, while Alcouffe, Brandt, Dendy, and Painter [2] uses finite volume type methods and a modified interpolation operator to solve elliptic partial differential equations on rectangular domains. While observing favorable results and convergence rates in each case, the results assume that the interface at which the discontinuity occurs aligns or is directly on the gridlines of the grid on which these problems are solved. Others such as Coco [11] have developed methods that deal with discontinuous coefficients where the discontinuities occur at arbitrary interfaces in the computational domain. The results of such methods are second order accurate in the infinity norm for one and two-dimensional problems, as well as convergence rates close to those of standard multigrid methods.

Chapter 1

Motivation and Objectives

The motivating factors for conducting this research are the speed and precision to which linear elliptic partial differential equations can be solved using multigrid methods. Many have applied multigrid methods to different classes of problems, including the Navier-Stokes Equations [24, 51], the Steady-State Euler equations [7, 12], convection-diffusion equations [14], and anisotropic partial differential equations [35]. We focus on fine-tuning certain components of the multigrid method to increase the convergence rate of the overall algorithm. In addition, this research seeks to understand the interplay of the number of grids, grid step sizes, the relaxation schemes used, and how an optimal solver can be created when taking these factors into consideration.

To explore these questions, we look to modify specific components of the multigrid algorithm which provide the greatest precision while minimizing the run time and computational work of the algorithm. Since the coefficients ε and κ may have steep gradients in certain parts of the domain, we create interpolation and restriction operators which take the behavior of the coefficients into account. We explore this concept further in the third chapter.

To begin, one must first understand the interworking components of the multigrid algo-

rithms. We look at the V, W, and Full-Multigrid cycles as examples for solving linear partial differential equations. We direct our focus on devising new methods tailored to specific components of the multigrid algorithms, including prolongation (interpolation), restriction, and relaxation methods. In general, one can extend a two-grid analysis of such methods to algorithms that use multiple grids [36].

For linear elliptic partial differential equations with constant coefficients, the discretization of the partial differential equation

$$-\Delta u = f, x \in \Omega$$

with suitable (Dirichlet) boundary conditions is given by a matrix equation

$$Au = f, \quad (1)$$

where A is a symmetric, positive-definite matrix, u is an array of unknown values of the continuous solution u in the domain Ω , and f is the array of the values of the right-hand-side function f at the corresponding grid points in the domain Ω .

To solve the discretized partial differential equation, we need to define the space over which we are providing numerical solutions. We let Ω_k , $k = 1, 2, \dots, m$ be a hierarchy of grids which we use to solve a given partial differential equation. That is, for a domain Ω and boundary $\partial\Omega$, we let

$$\Omega_k = \{(x, y) \in \Omega \cup \partial\Omega : x = ih_x^{(k)}, y = jh_y^{(k)}\},$$

where $h_x^{(k)}$ and $h_y^{(k)}$ are the grid step sizes in the x- and y-directions on the k th grid, and $i, j \in \mathbb{Z}$. The solution to the discrete problem (1) is defined on the finest grid Ω_k .

As previously mentioned, the methods we are using are restricted to the V, W, and Full-Multigrid Methods for the problems in which we are interested. We provide examples for the model problem, as well as problems whose coefficients vary in two and three space dimensions. We also derive a new interpolation operator in chapter 3. This scheme is compared to bilinear interpolation for use in the model problem given in chapter 3 and the linearized Poisson Boltzman Equation, otherwise known as the Debye-Hückel equation given by

$$\nabla \cdot (\epsilon \nabla \phi) = \kappa^2 \phi - \rho,$$

where ϵ is the dielectric coefficient, ϕ is the electric potential, κ is the Debye screening parameter, and ρ is the charge distribution [13].

We will also compare existing methods that have fine-tuned current solvers for elliptic partial differential equations, and extend them in two ways: The first involves making additions to the multigrid components that will accelerate the rate of convergence of the multigrid methods, including the V, W, and FMG cycles. We will also test the performance of this algorithm in a three-dimensional setting by using the model elliptic PDE and the Debye-Hückel equation.

Chapter 2

Multigrid Basics

2.1 Introduction

The methods we will use to analyze error and convergence of the new scheme include Local Mode (Fourier) Analysis and the spectral properties of the proposed multigrid operator [8,36]. This will give insight into how the operator affects the numerical approximations and error over the course of the scheme, while the spectral radius of the operator will provide an estimate of the factor by which error is reduced after each cycle.

2.2 The Model Problem

As a start to analysis, many have used the model Poisson Equation with homogeneous boundary conditions, given by

$$-\Delta u = f, x \in \Omega$$

$$u = 0, x \in \partial\Omega,$$

where we let $\Omega = (0, 1)^2$, the unit square. For two space dimensions, this reduces to

$$-\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) = f.$$

Using separation of variables on the corresponding equation $-\Delta u = \lambda u$, one can show that the eigenfunctions of $-\Delta u$ are given by

$$u_{m,n}(x, y) = \sin(m\pi x) \sin(n\pi y),$$

where m and n are positive integers. To move to the discrete case, we need to solve the above partial differential equation on a grid of discrete points that coincides with the original domain. Define

$$\Omega_k = \{(x, y) \in \Omega : x = ih_x, y = jh_y\}$$

where $i, j \in \mathbb{Z}$. We also define the boundary of the discrete domain

$$\partial\Omega_k = \{(x, y) \in \partial\Omega : x = ih_x, y = jh_y\}.$$

In practice, however, one can take Ω_k to be the set of all such discrete points that lie in the interior of the original domain and on the boundary.

In addition to the discrete domain Ω_k , otherwise known as the fine grid, multigrid methods solve analogous problems for the error, or corrections on coarser grids. For a given fine grid, the step sizes on the next coarsest grid are doubled. The step sizes h_x and h_y are taken to

be $2h_x$ and $2h_y$ on the coarse grid, which Brandt refers to as the standard coarsening [8]. We then define the next coarsest grid as

$$\Omega_{k-1} := \{(x, y) \in \Omega \cup \partial\Omega : x = 2ih_x, y = 2jh_y\},$$

where again i and j are taken to be integers ranging over appropriate values.

When moving to the discrete equation, we define the following operator and grid functions:

Definition 3.2.1 For the discrete problem, we define L_h (L_k) as the discrete laplace operator on the k th grid with a grid step size h .

Throughout this paper, we use the above two notations interchangeably. We also take f_h (f_k) to be the right-hand-side grid function for the discrete problem on the k th grid, and u_h (u_k) be the grid for which we seek a solution on the k th grid.

For the corresponding discrete equation

$$L_h u_h = f_h,$$

we obtain the similar eigenfunctions as above [36], which we show further in Chapter 4 of this thesis. In this case, the eigenfunctions are defined over the set of points

$$\Omega_h = \{(x, y) \in \mathbb{R}^2 | x = ih_x, y = jh_y, i, j \in \mathbb{Z}\}.$$

The use of local Fourier analysis will show how these eigenfunctions are affected by the Coarse Grid Correction method and the use of various smoothers. It is known, for example, that the eigenfunctions of the Jacobi smoothers are the same as those for the Laplace operator, making the analysis simple even though the Jacobi method has a larger smoothing factor than Gauss-Seidel and SOR [36]. One can perform similar analysis for Gauss-Seidel and SOR smoothers, though they do not share the same properties with the discrete operator that the Jacobi smoother does [18].

2.3 Prolongation and Restriction

The prolongation (interpolation) operator, often denoted I_k^{k+1} or I_{2h}^h , transfers a grid function u_{k-1} (u_{2h}) defined on a coarse grid Ω_{k-1} to the next finest grid Ω_k . In two dimensions, we can interpolate a given function to the next grid by bilinear interpolation [36]:

$$u_h(x, y) = \begin{cases} u_{2h}(x, y) & \text{if } (x, y) \in \Omega_{2h} \\ \frac{1}{2}(u_{2h}(x-h, y) + u_{2h}(x+h, y)) & \text{if } (x, y) \in \Omega_{2h} \setminus X_{2h} \\ \frac{1}{2}(u_{2h}(x, y-h) + u_{2h}(x, y+h)) & \text{if } (x, y) \in \Omega_{2h} \setminus Y_{2h} \\ \frac{1}{4}(u_{2h}(x-h, y-h) + u_{2h}(x-h, y+h) + \\ u_{2h}(x+h, y-h) + u_{2h}(x+h, y+h)) & \text{if } (x, y) \in \Omega_h \setminus \Omega_{2h} \end{cases}$$

Here, we denote the set of coarse x - and y - points as X_{2h} and Y_{2h} , respectively.

For the restriction operator, we give three rules below. Assume that (x, y) is a point in $\Omega_h \cap \Omega_{2h}$:

$$u_{2h}(x, y) = u_h(x, y) \quad (1)$$

$$u_{2h}(x, y) = \frac{1}{8}(4u_h(x, y) + u_h(x - h, y) + u_h(x + h, y) + u_h(x, y - h) + u_h(x, y + h)) \quad (2)$$

$$u_{2h}(x, y) = \frac{1}{16}(4u_h(x, y) + 2u_h(x - h, y) + 2u_h(x + h, y) + 2u_h(x, y - h) + 2u_h(x, y + h) + u_h(x - h, y - h) + u_h(x - h, y + h) + u_h(x + h, y - h) + u_h(x + h, y + h)) \quad (3)$$

Rules (1), (2), and (3) are called injection, half-weighting, and full-weighting, respectively [36]. For full-weighting, one can show that this restriction operator is the transpose of the interpolation operator up to a constant, i.e.

$$I_{2h}^h = c(I_h^{2h}), \quad c \neq 0.$$

This comes from the definition of the stencils for full-weighting and interpolation. On a given grid Ω_{k-1} , the stencil for interpolating a given point to the fine grid is given by

$$\frac{1}{4} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \quad (3.3.3)$$

while for full-weighting restriction we have

$$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}. \quad (3.3.4)$$

We can interpret the two stencils as follows: The center value in the first stencil corresponds to the new fine value we are initializing on the fine grid, as are the corresponding values on the same row and column as the center value. Distributing the factor of $1/4$ inside of the stencil gives

$$\begin{bmatrix} 1/4 & 1/2 & 1/4 \\ 1/2 & 1 & 1/2 \\ 1/4 & 1/2 & 1/4 \end{bmatrix}$$

Thus, for the center value we set it as the arithmetic average of the surrounding coarse points at the corners. The other values with the weight $1/2$ correspond to taking the fine value to be the average of the two adjacent coarse points at the corners of the stencil.

For the full-weighting stencil, the values inside the stencil correspond to the weights assigned to the surrounding fine points. The point in the center of the stencil is the fine point that corresponds to the coarse point whose value we are setting.

As an example in two dimensions, suppose we are solving a problem with zero Dirichlet boundary conditions on the unit square $\Omega = (0, 1)^2$. We order the points in the domain in a lexicographical order, so that the interior grid points at which we find our solution are ordered from smaller to larger x - and y -values. For this example, we assume that the coarse grid Ω_{k-1} contains nine points in the interior of the domain.

$$I_{k-1}^k = \begin{bmatrix} 1/4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/4 & 1/4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/4 & 1/4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/2 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/4 & 0 & 0 & 1/4 & 0 & 0 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0 \\ 1/4 & 1/4 & 0 & 1/4 & 1/4 & 0 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 0 & 1/2 & 0 & 0 & 0 & 0 \\ 0 & 1/4 & 1/4 & 0 & 1/4 & 1/4 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 0 & 0 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 1/4 & 0 & 0 & 1/4 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1/2 & 1/2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1/4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1/4 & 1/4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/4 & 1/4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/4 \end{bmatrix}$$

For the restriction operator I_k^{k-1} , we take the transpose of the above matrix and multiply by a factor of $1/4$, which corresponds to the stencil given in (3.3.4). Similar stencils and matrices can be derived for the corresponding problem in three space dimensions.

2.4 Operator-Based Interpolation

For smooth, continuous coefficients, bilinear interpolation is the favored choice in multigrid methods. For problems such as the Debye-Hückel equation

$$-\nabla \cdot (\epsilon \nabla \varphi) + \kappa^2 \varphi = \rho$$

presented by Hollenbeck et al. [13], one must exercise more caution in tailoring multigrid methods to handle potentially steep gradients in the coefficients ϵ and κ . Further challenges may arise when the coefficients (in particular, ϵ) are discontinuous. To amend this problem, we shall use the results of (cite Brandt source that uses the volume differencing scheme) and [36] to create an interpolation operator which is second order and satisfies the jump conditions of an elliptic partial differential equation with a jump condition at an interface in the interior of the domain.

We begin by looking at interpolating the course grid approximation to the next finest grid. It should be noted that although [2] uses a volume-type discretization, we can still obtain a feasible interpolation scheme using a similar approach with finite differences.

For the value of the numerical solution on the fine grid, we seek to satisfy the following condition given in [36]:

$$L_h I_{2h}^h \varphi_{2h} = 0.$$

That is, we assume that the function satisfying the original partial differential equation is approximately linearly on a small portion of the domain. Not that in the case of a constant coefficient ϵ , the above formulations yield bilinear interpolation. In three and higher dimen-

sions, the approach we take below is similar. For now, let $\varphi_h := I_{2h}^h \varphi_{2h}$.

We begin by creating an interpolation scheme for the fine grid value φ_{ij} that lies on a coarse y -line and fine x -line. Thus, the above condition on the interpolation scheme reduces to interpolation in the x -direction while y is held constant:

$$\begin{aligned} L_h I_{2h}^h \varphi_{2h} = 0 &\implies \\ \frac{-\epsilon_{i+1/2,j} \varphi_{i+1,j} + (\epsilon_{i+1/2,j} + \epsilon_{i-1/2,j}) \varphi_{ij} - \epsilon_{i-1/2,j} \varphi_{i-1,j}}{h^2} = 0 &\implies \\ \varphi_{ij} = \frac{\epsilon_{i-1/2,j} \varphi_{i-1,j} + \epsilon_{i+1/2,j} \varphi_{i+1,j}}{\epsilon_{i-1/2,j} + \epsilon_{i+1/2,j}} \end{aligned}$$

We can derive a similar approximation for the value of φ_{ij} which lies on a fine y -line and coarse x -line. As before, this reduces to a discrete analogue of the derivative with respect to just y because we are treating x as constant.

$$\begin{aligned} L_h I_{2h}^h \varphi_{2h} = 0 &\implies \\ \frac{-\epsilon_{i,j+1/2} \varphi_{i,j+1} + (\epsilon_{i,j+1/2} + \epsilon_{i,j-1/2}) \varphi_{ij} - \epsilon_{i,j-1/2} \varphi_{i,j-1}}{h^2} = 0 &\implies \\ \varphi_{ij} = \frac{\epsilon_{i,j-1/2} \varphi_{i,j-1} + \epsilon_{i,j+1/2} \varphi_{i,j+1}}{\epsilon_{i,j-1/2} + \epsilon_{i,j+1/2}} \end{aligned}$$

Finally, we find an approximation to the value of φ_{ij} which does not lie on the coarse grid. That is, the point (x_i, y_j) does not lie on an x - or y -gridline from the coarse grid. We then have

$$\begin{aligned} L_h \varphi_h = \\ - \left(\frac{\epsilon_{i+1/2,j} \varphi_{i+1,j} - (\epsilon_{i+1/2,j} + \epsilon_{i-1/2,j}) \varphi_{ij} + \epsilon_{i-1/2,j} \varphi_{i-1,j}}{h^2} \right) \end{aligned}$$

$$-\left(\frac{\epsilon_{i,j+1/2}\varphi_{i,j+1} - (\epsilon_{i,j+1/2} + \epsilon_{i,j-1/2})\varphi_{ij} + \epsilon_{i,j-1/2}\varphi_{i,j-1}}{h^2}\right) + \kappa_{ij}^2\varphi_{ij} = 0.$$

We then multiply the above equation through by h^2 and collect the terms with φ_{ij} on one side to obtain

$$\begin{aligned} &(\epsilon_{i-1/2,j} + \epsilon_{i+1/2,j} + \epsilon_{i,j-1/2} + \epsilon_{i,j+1/2} + (h\kappa_{ij})^2)\varphi_{ij} = \\ &\epsilon_{i-1/2,j}\varphi_{i-1,j} + \epsilon_{i+1/2,j}\varphi_{i+1,j} + \epsilon_{i,j-1/2}\varphi_{i,j-1} + \epsilon_{i,j+1/2}\varphi_{i,j+1}, \end{aligned}$$

so

$$\varphi_{ij} = \frac{\epsilon_{i-1/2,j}\varphi_{i-1,j} + \epsilon_{i+1/2,j}\varphi_{i+1,j} + \epsilon_{i,j-1/2}\varphi_{i,j-1} + \epsilon_{i,j+1/2}\varphi_{i,j+1}}{\epsilon_{i-1/2,j} + \epsilon_{i+1/2,j} + \epsilon_{i,j-1/2} + \epsilon_{i,j+1/2} + (h\kappa_{ij})^2}.$$

In the above formula, note that the values $\varphi_{i-1,j}$, $\varphi_{i+1,j}$, $\varphi_{i,j-1}$, and $\varphi_{i,j+1}$ are fine-grid values obtained from the interpolation on coarse lines given above.

2.5 Work Units

Another aspect of the multigrid convergence analysis comes from Brandt's use of the multigrid work unit WU , which he defines as the total work done on the finest grid over the course of one relaxation sweep [8]. Work units can be measured as the number of floating point operations on a given grid; if the finest grid has N points, and there are w floating point operations per grid point, then one work unit is approximately $w \times N$ floating point operations. Additionally, one may also take into consideration the work it takes to interpolate and restrict numerical solutions to finer and coarser grids. Brandt notes that a majority of the computational work takes place with relaxation sweeps; the work invested in restriction and prolongation is considered negligible since it comprises around twenty percent of the

total work. Restriction and prolongation, however, are important in other regards since they affect the representation of the error frequencies on the grids.

To give an estimate of the total computational work of restriction and interpolation relative to the total work in a multigrid V-Cycle (see next section), we suppose that the number of points in the x - and y -directions is N , so we seek a solution on a grid of N^2 points. We then look at multigrid in terms of four components: One application of the smoother before moving to a coarse grid and one application after returning to a fine grid, computing the residual, restriction, and interpolation.

For the smoother, we employ a Gauss-Seidel smoother for the standard five-point laplacian given by

$$u_{ij} = \frac{h^2 f_{ij} + u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1}}{4}.$$

On any given grid, we perform five additions per grid point when computing a new value of the approximation to $u(x, y)$. Then one work unit (WU) is approximately $5N^2$ additions. Since the smoother is applied once before moving to a coarse grid and once after interpolating to a fine grid, an estimate on the total number of additions over one multigrid V-cycle is

$$\begin{aligned} 2 \times \left(5N^2 + \frac{5N^2}{4} + \frac{5N^2}{4^2} + \dots + \frac{5N^2}{4^{k-1}} \right) = \\ 10N^2 \left(1 + \frac{1}{4} + \frac{1}{4^2} + \dots + \frac{1}{4^{k-1}} \right) \leq \\ 10N^2 \frac{1}{1 - \frac{1}{4}} = \frac{40N^2}{3} \end{aligned}$$

additions.

After smoothing, we compute residual as follows:

$$r_{ij} = f_{ij} + \frac{u_{i+1,j} - 2u_{ij} + u_{i-1,j}}{h^2} + \frac{u_{i,j+1} - 2u_{ij} + u_{i,j-1}}{h^2}.$$

This incurs six additions per grid point on the given grid, yielding $6N^2$ additions on the fine grid. Since we compute the residual on all grids but the coarsest, the estimated number of additions is given by

$$\begin{aligned} 6N^2 + \frac{6N^2}{4} + \frac{6N^2}{4^2} + \dots + \frac{6N^2}{4^{k-1}} &= \\ 6N^2 \left(1 + \frac{1}{4} + \frac{1}{4^2} + \dots + \frac{1}{4^{k-1}} \right) &\leq \\ 6N^2 \frac{1}{1 - \frac{1}{4}} &= 8N^2 \end{aligned}$$

For restriction, we compute the values at the coarse grid points as weighted sums of the surrounding fine grid points. For each coarse grid point, the new value takes eight additions to compute. Restricting a fine grid residual to a coarse grid then takes approximately

$$\frac{N}{2} \times \frac{N}{2} \times 8 = 2N^2$$

additions. Interpolation from the coarse grid Ω_{k-1} to the fine k th grid takes approximately

$$2\frac{N}{2} \times \frac{N}{2} + 2\frac{N}{2} \times \frac{N}{2} + 4\frac{N}{2} \times \frac{N}{2} = 2N^2$$

additions.

Now considering the work done over all grids, and assuming we can neglect the work done on the coarsest grid, an upper bound of the total work performed by interpolation is

$$\begin{aligned} 2N^2 + \frac{2N^2}{4} + \frac{2N^2}{4^2} + \dots + \frac{2N^2}{4^{k-1}} &= \\ 2N^2 \left(1 + \frac{1}{4} + \frac{1}{4^2} + \dots + \frac{1}{4^{k-1}} \right) &\leq \\ 2N^2 \times \frac{1}{1 - \frac{1}{4}} &= \frac{8N^2}{3} \end{aligned}$$

additions. We obtain a similar estimate for restriction, since on the next coarsest grid, we do about one quarter of the work we did on the previous fine grid in two dimensions. Thus, the estimated number of additions for restriction over all grids is

$$2N^2 \left(1 + \frac{1}{4} + \frac{1}{4^2} + \dots + \frac{1}{4^{k-1}} \right) \leq \frac{8N^2}{3}.$$

Thus, the approximate percentage of work spent on interpolation and restriction combined is

$$\frac{\frac{8}{3} + \frac{8}{3}}{\frac{40}{3} + \frac{8}{3} + \frac{8}{3} + 8} = \frac{\frac{16}{3}}{\frac{80}{3}} = \frac{1}{5}$$

or twenty percent.

Now consider the following: Define the ratio of step sizes $\hat{\rho} = \frac{h_k}{h_{k-1}}$ between two successive grids. If we have problem in d dimensions and a mesh size ratio of $\hat{\rho} = \frac{1}{2}$, then the work

done on the grid G^{M-j} is given by $\hat{\rho}^{dj}$ work units [8]. From this analysis, we can derive the multigrid convergence factor $\hat{\mu}$, the rate by which all errors are reduced per one work unit of relaxation, assuming we ignore the computational work of restriction and interpolation [8].

Suppose that during the course of the scheme, we perform s relaxation sweeps on each level. Recall that the relaxation factor is given by $\bar{\mu}$, so that after s sweeps, the high-frequency errors have been reduced by a factor of $\bar{\mu}^s$. We can then provide an upper bound of the total computational work (excluding restriction and prolongation) over each grid, given by

$$s + s\hat{\rho}^d + s\hat{\rho}^{2d} + \dots + s\hat{\rho}^{(M-1)d} < \frac{s}{1 - \hat{\rho}^d},$$

for which we then define the multigrid convergence factor as [8]

$$\hat{\mu} = \bar{\mu}^{1 - \hat{\rho}^d}.$$

This convergence factor represents the rate of error reduction per unit of work.

2.6 The Coarse Grid Correction Scheme

One can derive multigrid methods based on the coarse grid correction scheme [36], which we give below.

The two-grid correction and relaxation cycle forms the basis for methods which recurse on multiple grids, including the V-Cycle, W-Cycle, and Full-Multigrid-Cycle. For non-linear problems, one may use the Full-Approximation-Storage Method outlined by Brandt [8].

Algorithm 1 Coarse Grid Correction

- (1) Provide an initial guess u_h^0 on the finest grid Ω_h
 - (2) Relax u_h^0 v_1 times on the grid Ω_h
 - (3) Compute the residual $r_h = f_h - L_h u_h$
 - (4) Restrict the residual to the next coarsest grid: $r_{2h} = I_h^{2h} r_h$
 - (5) Solve the error equation $A_{2h} e_{2h} = r_{2h}$
 - (6) Interpolate the error to the next finest grid: $e_h = I_{2h}^h e_{2h}$
 - (7) Add the error to the initial swept-over guess: $u_h^0 = u_h^0 + e_h$
 - (8) Perform v_2 additional relaxation sweeps on u_h^0
-

2.7 The Multilevel Operator: Definition and Derivation

For the definition of the multigrid (multilevel) operator, we use the following theorem from [36]. In the theorem below, the operator I_k is the identity operator on the k th grid.

Theorem 1. *The multigrid iteration operator M_l is given by the following recursion:*

$$M_0 = 0,$$

$$M_k = S_k^{v_2} (I_k - I_{k-1}^k (I_{k-1} - M_{k-1}^\gamma) (L_{k-1})^{-1} I_k^{k-1} L_k) S_k^{v_1}, \quad (k = 1, \dots, l).$$

Proof. By definition, $M_0 u_k^0 = 0$, and for $k = 1$, we obtain the two-level operator

$$M_1 = S_1^{v_2} (I_1 - I_0^1 (L_0)^{-1} I_1^0 L_1) S_1^{v_1} = S_1^{v_2} (I_1 - I_0^1 (I_0 - M_0^\gamma) (L_0)^{-1} I_1^0 L_1) S_1^{v_1}.$$

Now assume the recursive definition holds for $k \geq 1$. To show this holds true for the next level $k + 1$, suppose we wish to solve the fine-grid problem on the grid Ω_{k+1} :

$$L_{k+1} u_{k+1} = f_{k+1}.$$

We step through the multigrid process by applying the smoother on grid $k+1$ v_1 times, and compute the residual on the fine grid:

$$r_{k+1} = f_{k+1} - L_{k+1} S_{k+1}^{v_1} u_{k+1}^0.$$

We then restrict this residual to the coarser grid by setting $r_k = I_{k+1}^k r_{k+1}$, and approximately solve the corresponding equation

$$L_k e_k = r_k \quad (1)$$

for the correction e_k . To do so, we devise an iterative scheme using the multilevel operator M_k . From (1), we have that

$$e_k = L_k^{-1} r_k.$$

The correction e_k also satisfies

$$(I_k - M_k) e_k = (I_k - M_k) L_k^{-1} r_k \quad (2)$$

since M_k is nonsingular and a convergent operator, i.e. $\|M_k\| < 1$. In addition, we also assume M_k has no zero eigenvalues. Then (2) can be written as

$$e_k = M_k e_k + (I_k - M_k) L_k^{-1} r_k \quad (3),$$

and so we define the following (multilevel) iterative scheme:

$$e_k^{n+1} = M_k e_k^n + (I_k - M_k) L_k^{-1} r_k \quad (4).$$

Applying this iteration γ times to the iterate e_k^0 with $e_k^0 = 0$ yields

$$e_k^\gamma = (I_k - M_k^\gamma) L_k^{-1} r_k.$$

Once this approximation to the correction has been given on the coarse grid, we interpolate this correction to the fine grid and add it to the approximation of the numerical solution, and perform v_2 addition smoothing sweeps:

$$S_{k+1}^{v_2} (S_{k+1}^{v_1} u_{k+1}^0 + I_k^{k+1} e_k^\gamma).$$

Placing these steps together yields

$$S_{k+1}^{v_2} (S_{k+1}^{v_1} u_{k+1}^0 + I_k^{k+1} (I_k - M_k^\gamma) L_k^{-1} I_{k+1}^k (f_{k+1} - L_{k+1} S_{k+1}^{v_1} u_{k+1}^0)).$$

Simplifying the above expression yields

$$\begin{aligned} & S_{k+1}^{v_2} (I_{k+1} - I_k^{k+1} (I_k - M_k^\gamma) L_k^{-1} I_{k+1}^k L_{k+1}) S_{k+1}^{v_1} u_{k+1}^0 + \\ & S_{k+1}^{v_2} I_k^{k+1} (I_k - M_k^\gamma) L_k^{-1} I_{k+1}^k f_{k+1}, \end{aligned}$$

with

$$M_{k+1} = S_{k+1}^{v_2}(I_{k+1} - I_k^{k+1}(I_k - M_k^\gamma)L_k^{-1}I_{k+1}^k L_{k+1})S_{k+1}^{v_1}$$

□

2.8 Common Multigrid Algorithms

Introduction

The following sections present common algorithms used in multigrid methods. Further details of these algorithms may be found in [36].

2.8.1 V-Cycle

The following is the algorithm for the V-Cycle:

Algorithm 2 V-Cycle

- (1) Relax u_k^0 v_1 times on the grid Ω_h
 - (2) Compute the residual $r_k = f_k - L_k u_k$
 - (3) If $k > 1$, call VCYCLE(A_{k-1} , v_{k-1} , r_{k-1}) with $v_{k-1} = 0$ as an initial guess
 - (4) Else if $k = 1$, solve $A_k v_k = r_k$ directly
 - (5) Interpolate the error to the next finest grid: $\tilde{v}_k = I_{k-1}^k v_k$
 - (6) Add the error to the initial swept-over guess: $u_k^0 = u_k^0 + \tilde{v}_k$
 - (7) Perform v_2 additional relaxation sweeps on u_k^0
 - (8) Return the new approximation u_k^0
-

2.8.2 W-Cycle

The following is the algorithm for the V-Cycle:

Algorithm 3 W-Cycle

- (1) Relax u_k^0 v_1 times on the grid Ω_h
 - (2) Compute the residual $r_k = f_k - L_k u_k$
 - (3) For $n = 1, 2$
 - (4) If $k > 1$, call $WCYCLE(A_{k-1}, v_{k-1}, r_{k-1})$ with $v_{k-1} = 0$ as an initial guess
 - (5) Else if $k = 1$, solve $A_k v_k = r_k$ directly
 - (6) Interpolate the error to the next finest grid: $\tilde{v}_k = I_{k-1}^k v_k$
 - (7) Add the error to the initial swept-over guess: $u_k^0 = u_k^0 + \tilde{v}_k$
 - (8) Perform v_2 additional relaxation sweeps on u_k^0
 - (9) End for
 - (10) Return the new approximation u_k^0
-

2.8.3 FMG-Cycle

The following is the algorithm for the Full Multigrid (FMG) Cycle. Here, r denotes the number of applications of the V-Cycle on a given grid. Other cycles may also be used in place of the V-Cycle [36].

Algorithm 4 FMG-Cycle

- (1) Solve $L_1 u_1 = f_1$ exactly on the coarsest grid to obtain u_1^{FMG}
 - (2) Interpolate this guess using FMG-Interpolation to the next grid Ω_2
 - (3) For $k = 2, \dots, M$
 - (4) Call $V - CYCLE^r(A_k, u_k, f_k)$
 - (5) If $k < M$
 - (6) Use FMG Interpolation on u_k to obtain a guess on the next highest grid Ω_{k+1}
 - (7) End If
 - (8) End For
-

2.9 Norm Estimates and Theorems

We give the following error estimate theorems for the multigrid methods for the γ -cycle ($\gamma \geq 2$) and the FMG-cycle. For $\gamma = 1$, the error estimates in the following theorem are not independent of the grid step size(s), so a more thorough treatment is needed [36].

For the following theorem, we first provide definitions of the operators given in the state-

ment. The following corollary from Trottenberg, Oosterlee, and Schüller [36]:

Corollary. For $k = 1, \dots, l - 1$, the equations

$$M_{k+1} = M_{k+1}^k + A_k^{k+1}(M_k)^\gamma A_{k+1}^k$$

hold, where

$$A_k^{k+1} := (S_{k+1})^{v_2} I_k^{k+1} : G(\Omega_k) \rightarrow G(\Omega_{k+1})$$

$$A_{k+1}^k := (L_k)^{-1} I_{k+1}^k L_{k+1} (S_{k+1})^{v_1} : G(\Omega_{k+1}) \rightarrow G(\Omega_k)$$

and

$$M_{k+1}^k = S_{k+1}^{v_2} (I_{k+1} - I_k^{k+1} (L_k)^{-1} I_{k+1}^k L_{k+1}) S_{k+1}^{v_1}.$$

Proof. From **Theorem 1**, we have that

$$M_{k+1} = S_{k+1}^{v_2} (I_{k+1} - I_k^{k+1} (I_k - (M_k)^\gamma) (L_k)^{-1} I_{k+1}^k L_{k+1}) S_{k+1}^{v_1},$$

which may be written as

$$S_{k+1}^{v_2} (I_{k+1} - I_k^{k+1} I_k (L_k)^{-1} I_{k+1}^k L_{k+1} + I_k^{k+1} (M_k)^\gamma (L_k)^{-1} I_{k+1}^k L_{k+1}) S_{k+1}^{v_1}.$$

This simplifies to

$$S_{k+1}^{v_2} (I_{k+1} - I_k^{k+1} (L_k)^{-1} I_{k+1}^k L_{k+1}) S_{k+1}^{v_1} +$$

$$\begin{aligned}
& S_{k+1}^{v_2} I_k^{k+1} (M_k)^\gamma (L_k)^{-1} I_{k+1}^k L_{k+1} S_{k+1}^{v_1} = \\
& S_{k+1}^{v_2} (I_{k+1} - I_k^{k+1} (L_k)^{-1} I_{k+1}^k L_{k+1}) S_{k+1}^{v_1} + A_k^{k+1} (M_k)^\gamma A_{k+1}^k = \\
& M_{k+1}^k + A_k^{k+1} (M_k)^\gamma A_{k+1}^k.
\end{aligned}$$

□

The following theorem from [36] gives estimates for the norm of the multigrid operator for $\gamma \geq 2$.

Theorem 2. *Let the following estimates hold uniformly with respect to k :*

$$\|M_{k+1}^k\| \leq \sigma^*, \quad \|A_k^{k+1}\| \cdot \|A_{k+1}^k\| \leq C.$$

Then we have $\|M_l\| \leq \eta_l$. where η_l is recursively defined as

$$\eta_1 = \sigma^*, \quad \eta_{k+1} = \sigma^* + C\eta_k^\gamma, \quad (k = 1, 2, \dots, l-1).$$

If we additionally assume that

$$4C\sigma^* \leq 1 \text{ and } \gamma = 2,$$

we obtain the following uniform estimate for $M_h = M_l$ ($h = h_l$):

$$\|M_h\| \leq \eta := \frac{1 - \sqrt{1 - 4C\sigma^*}}{2C} \leq 2\sigma^*, \quad l \geq 1.$$

Before proving the above theorem, we require the following Lemmas.

Lemma 1. Let

$$f(\eta) = C\eta^2 + \sigma^*$$

and let

$$g(\eta) = m\eta + \sigma^*$$

with $m = \frac{\eta^* - \sigma^*}{\eta^*}$, $\eta^* = \frac{1 - \sqrt{1 - 4C\sigma^*}}{2C}$, and $4C\sigma^* \leq 1$. The sequences $\{\eta_k\}_{k=1}^\infty$ and $\{\mu_k\}_{k=1}^\infty$ with

$$\eta_{k+1} = f(\eta_k) = C\eta_k^2 + \sigma^*,$$

$$\mu_{k+1} = g(\mu_k) = m\mu_k + \sigma^*,$$

and

$$\eta_1 = \mu_1 = \sigma^*$$

are monotonic and bounded. Hence, both sequences converge [40].

Proof. To complete the proof, we first show that μ_k is monotonic and bounded for all positive integers k . The fact that η_k is bounded follows from a further proof that $\eta_k \leq \mu_k$ for all integers $k \geq 1$.

We prove μ_k is bounded by induction on k . Indeed, $\mu_1 < \mu_2$ since $\sigma^* < \sigma^* + m\sigma^*$. Now suppose this holds for some integer $k \geq 2$. By the inductive step we have $\mu_{k-1} < \mu_k$. Since g is strictly increasing, it follows that $g(\mu_{k-1}) < g(\mu_k)$ which implies $\mu_k < \mu_{k+1}$.

To show that $\{\mu_k\}_{k=1}^\infty$ is bounded, observe that the linear recurrence relation formed by

the sequence $\{\mu_k\}_{k=1}^\infty$ has a closed form solution when we trace down to the base case of $\mu_1 = \sigma^*$:

$$\begin{aligned}\mu_k &= m\mu_{k-1} + \sigma^* = m^2\mu_{k-2} + m\sigma^* + \sigma^* = \dots = \\ m^{k-1}\sigma^* + m^{k-2}\sigma^* + \dots + m\sigma^* + \sigma^* &= \sigma^* \frac{1 - m^k}{1 - m} = \sigma^* \frac{1 - m^k}{1 - \frac{\eta^* - \sigma^*}{\eta^*}} = \\ \sigma^* \frac{1 - m^k}{\frac{\sigma^*}{\eta^*}} &= \eta^*(1 - m^k).\end{aligned}$$

This expression approaches η^* from below as $k \rightarrow \infty$. Now we show that $\{\eta_k\}_{k=1}^\infty$ is monotonic by induction on k . Observe that $\eta_1 < \eta_2$ since $\sigma^* < \sigma^* + C(\sigma^*)^2$. Now assume that $\eta_{k-1} < \eta_k$ holds for some integer $k \geq 2$. Since f is also strictly increasing on $[\sigma^*, \eta^*]$, we have that

$$f(\eta_{k-1}) < f(\eta_k) \implies \eta_k < \eta_{k+1}.$$

Finally, we prove the claim that $\eta_k \leq \mu_k$ for all integers $k \geq 1$. As a base case, we have that $\eta_1 \leq \mu_1$ by construction, so assume that this holds for some integer $k \geq 1$. The induction step gives $\eta_k \leq \mu_k$, so to prove the claim we use the fact that $g \geq f$ for all $\eta \in [\sigma^*, \eta^*]$:

$$\eta_{k+1} = f(\eta_k) \leq f(\mu_k) \leq g(\mu_k) = \mu_{k+1}.$$

Thus, $\{\eta_k\}_{k=1}^\infty$ and $\{\mu_k\}_{k=1}^\infty$ converge. □

The next lemma gives a proof of convergence of the sequence $\{\eta_k\}_{k=1}^\infty$ in the case that $4C\sigma^* < 1$.

Lemma 2. Let $\sigma^* = \frac{p}{4C}$, with $0 < p < 1$. Then the sequence $\{\eta_k\}_{k=1}^\infty$ converges to η^* .

Proof. It suffices to show that

$$|\eta^* - \eta_k| \rightarrow 0 \quad (2.1)$$

as $k \rightarrow \infty$. Since η^* is a fixed point of the map $f(\eta) = C\eta^2 + \sigma^*$, (2.1) can be written as

$$\begin{aligned} & |C(\eta^*)^2 + \sigma^* - C\eta_k^2 - \sigma^*| = \\ & C|(\eta^*)^2 - \eta_k^2| = C|\eta^* + \eta_k||\eta^* - \eta_k| = \\ & C|\eta^* + \eta_k||C(\eta^*)^2 - C(\eta_{k-1})^2| = \\ & C^2|\eta^* + \eta_k||\eta^* + \eta_{k-1}||\eta^* - \eta_{k-1}| = \dots = C^k|\eta^* + \eta_k||\eta^* + \eta_{k-1}| \dots |\eta^* + \eta_2||\eta^* + \eta_1||\eta^* - \eta_1| = \\ & C^k \left(\prod_{j=1}^k |\eta^* + \eta_j| \right) (\eta^* - \sigma^*). \end{aligned}$$

From *Lemma 1*, we have $\eta_j \leq \mu_k$ and $\mu_k \rightarrow \eta^*$ as $k \rightarrow \infty$, so

$$\begin{aligned} |\eta^* - \eta_{k+1}| & \leq C^k \left(\prod_{j=1}^k |\eta^* + \eta_j| \right) (\eta^* - \sigma^*) = \\ C^k \left(\prod_{j=1}^k \frac{1 - \sqrt{1 - 4C\sigma^*}}{C} \right) (\eta^* - \sigma^*) & = C^k \frac{(1 - \sqrt{1 - 4C\frac{p}{4C}})^k}{C^k} (\eta^* - \sigma^*) = \\ & (1 - \sqrt{1 - p})^k (\eta^* - \sigma^*), \end{aligned}$$

which approaches zero as $k \rightarrow \infty$. □

The next lemma gives a short proof for the convergence of $\{\eta_k\}_{k=1}^\infty$ in the case that σ^* , the norm of the two-level operator M_k^{k+1} , is equal to $\frac{1}{4C}$. In this case, we have to be more careful since $f(\eta) = \eta^2 + \sigma^*$ is a **nonexpansive map** on the interval (σ^*, η^*) [1], i.e.

$$|f(\eta) - f(\xi)| \leq K|\eta - \xi| \quad \forall \eta, \xi \in (\sigma^*, \eta^*),$$

where $K = 1$.

Lemma 3. Suppose now that the norm of the two-level operator in **Theorem 2** is $\sigma^* = \frac{1}{4C}$, so now $\eta^* = \frac{1}{2C}$. Then the sequence $\{\eta_k\}_{k=1}^\infty$ converges to the unique fixed point η^* of the map $f(\eta) = C\eta^2 + \sigma^*$

Proof. We begin by showing that the fixed point of the map $f(\eta) = C\eta^2 + \sigma^*$ is unique. Since η^* is a fixed point of f , it follows that

$$\eta^2 = C(\eta^*)^2 + \frac{1}{4C} \implies C(\eta^*)^2 - \eta^* + \frac{1}{4C} = 0,$$

which admits one root given by

$$\eta^* = \frac{1 \pm \sqrt{1 - 4C \frac{1}{4C}}}{2C} = \frac{1}{2C},$$

so η^* is necessarily unique.

To complete the proof, we utilize **Theorem 2.1** from Ahmed [1]. In order to prove that the sequence $\{\eta_k\}_{k=1}^\infty$ converges, we must show that

$$\lim_{k \rightarrow \infty} |\eta_{k+1} - \eta_k| = 0 \quad (1)$$

and

$$\lim_{k \rightarrow \infty} \eta_k \quad (2)$$

exists. The second condition follows from *Lemma 1*, since it was shown that $\{\eta_k\}_{k=1}^{\infty}$ is bounded and monotonic. Thus, the sequence is Cauchy on $[\sigma^*, \eta^*]$ which implies condition (1) [40], so $\eta_k \rightarrow \eta^*$ as $k \rightarrow \infty$.

□

We now proceed with the proof of **Theorem 2**.

Proof. We first observe that

$$\|M_{k+1}\| = \|M_{k+1}^k + A_k^{k+1}(M_k)^\gamma A_{k+1}^k\| \leq$$

$$\|M_{k+1}^k\| + \|A_k^{k+1}\| \cdot \|A_{k+1}^k\| \cdot \|M_k\|^\gamma = \sigma^* + C\eta_k^2 = \eta_{k+1}.$$

Now under the additional assumptions, we must show that as $k \rightarrow \infty$, η_k approaches the limit point η which satisfies

$$\eta = \sigma^* + C\eta^2,$$

a quadratic in η with the root

$$\eta = \frac{1 - \sqrt{1 - 4C\sigma^*}}{2C}.$$

Then by Lemmas 1, 2, and 3, the sequence $\{f(\eta_k)\}_{k=1}^{\infty}$ with $\eta_1 = \sigma^*$ converges to the fixed point

$$\eta^* = \frac{1 - \sqrt{1 - 4C\sigma^*}}{2C}$$

in $[0, \frac{1}{2C}]$. By rationalizing the denominator of η^* , we obtain the estimate from the above theorem:

$$\frac{1 - \sqrt{1 - 4C\sigma^*}}{2C} = \frac{1 - (1 - 4C\sigma^*)}{2C(1 + \sqrt{1 - 4C\sigma^*})} = \frac{2\sigma^*}{1 + \sqrt{1 - 4C\sigma^*}} \leq 2\sigma^*.$$

□

The following theoretical estimates from [36] for Full Multigrid are also given:

Theorem 3. *Let the following assumptions hold:*

(1) *Let the norm of the multigrid iteration operator M_h be uniformly bounded:*

$$\|M_h\| \leq \eta < 1$$

(2) *Let the norm of the FMG interpolation operator Π_{k-1}^k be uniformly bounded:*

$$\|\Pi_{k-1}^k\| \leq P < 1$$

(3) *The discretization error and the FMG interpolation error are assumed to be of order κ and order κ_{FMG} , respectively:*

$$\|u - u_h\| \leq Kh^\kappa, \quad (h = h_k; k = 1, 2, \dots),$$

$$\|u - \Pi_{k-1}^k u\| \leq \bar{K}h^{\kappa_{FMG}}, \quad (h = h_k; k = 1, 2, \dots),$$

In addition, assume that

$$\eta^r < \frac{1}{2^\kappa P}.$$

Then the following estimate holds for any $l \geq 1$:

$$\|u_l - u_l^{FMG}\| \leq \delta h_l^\kappa,$$

where

$$\delta = \eta^r \frac{B}{1 - \eta^r A},$$

with $A = 2^\kappa P$ and any bound B such that $K(1 + A) + \bar{K}h^{\kappa_{FMG} - \kappa} \leq B$.

Proof. By the definition of the Full-Multigrid Method, we have for all $l \geq 1$

$$u_l^{FMG} - u_l = (M_l)^r (u_0^l - u_l),$$

with

$$u_0^l = \Pi_{l-1}^l u_{l-1}^{FMG}.$$

We first make use of the following identity:

$$u_l^0 - u_l = \Pi_{l-1}^l (u_{l-1}^{FMG} - u_{l-1}) + \Pi_{l-1}^l (u_{l-1} - u) + (\Pi_{l-1}^l u - u) + (u - u_l).$$

Then we have that

$$\begin{aligned}
||u_l^{FMG} - u_l|| &= ||(M_l)^r(u_l^0 - u_l)|| \leq \eta^r ||u_l^0 - u_l|| \leq \\
\eta^r (P||u_{l-1}^{FMG} - u_{l-1}|| + P||u_{l-1} - u|| + ||\Pi_{l-1}^l u - u|| + ||u - u_l||) &= \\
\eta^r (P||u_{l-1}^{FMG} - u_{l-1}|| + PKh_l^\kappa + \bar{K}h_l^{\kappa_{FMG}} + Kh_l^\kappa). &
\end{aligned}$$

Let $\delta_l = ||u_l^{FMG} - u_l||/h_l^\kappa, l \geq 1$. After dividing the above inequality through by h_l^κ , the above estimate is written as

$$\delta_l \leq \eta^r (A\delta_{l-1} + AK + \bar{K}h_l^{\kappa_{FMG}-\kappa} + K),$$

where $A = 2^\kappa P$. Now let B be any bound such that $K(1 + A) + \bar{K}h^{\kappa_{FMG}-\kappa} \leq B$, and let $p = \eta^r A$. Then the above inequality may be written as

$$\delta_l \leq p\delta_{l-1} + \tilde{B}$$

where $\tilde{B} = \eta^r B$. Expanding δ_{l-1} and using the fact that

$$\delta_0 = \frac{||u_0^{FMG} - u_0||}{h_0^\kappa} = \frac{0}{h_0^\kappa} = 0,$$

we obtain

$$\delta_l \leq p^l \delta_0 + (1 + p + p^2 + \dots + p^{l-1})\tilde{B} = (1 + p + p^2 + \dots + p^{l-1})\tilde{B}.$$

This implies that

$$\delta_l \leq \frac{\tilde{B}}{1-p} = \eta^r \frac{B}{1-\eta^r A}$$

and so

$$||u_l - u_l^{FMG}|| \leq \delta h_l^\kappa.$$

□

Chapter 3

Local Mode Analysis

3.1 Introduction and Definitions

Local Mode Analysis [8] allows one to conclude the properties of multigrid, providing information about the rate at which the error is reduced and how fast the multigrid algorithm itself will converge. We begin by identifying important concepts needed before proceeding with a case analysis of the model Poisson equation. We provide definitions for the two-dimensional case that can extend to any n-dimensional setting.

Definition 4.1.1. Let $\theta = (\theta_1, \theta_2)$ and $x = (x_1, x_2)$ be vectors in \mathbb{R}^2 , and let $G^h = \{(x, y) \in \mathbb{R}^2 : x = ih_x, y = jh_y, i, j \in \mathbb{Z}\}$ be the grid with step sizes h_x and h_y . Then the grid function associated with the discrete operator L_h is given by

$$\varphi(\theta, x) = e^{i\theta \cdot x}.$$

One can show that the above grid functions $\varphi(\theta, x)$ are in fact the eigenfunctions of L_h .

Definition 4.1.2. Let $\theta = (\theta_1, \theta_2)$ be a vector in \mathbb{R}^2 . Define θ to be a **low-frequency**

component if

$$\theta \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]^2,$$

where

$$|\theta| := \max\{|\theta_1|, |\theta_2|\}.$$

Otherwise, define θ to be a **high frequency component** if

$$\theta \in [-\pi, \pi]^2 \setminus \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]^2.$$

Consequently, define the following sets T^{high} and T^{low} [36] as

$$T^{high} = \left\{ \theta \in \mathbb{R}^2 : \theta \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]^2 \right\}$$

and

$$T^{low} = \left\{ \theta \in \mathbb{R}^2 : \theta \in [-\pi, \pi]^2 \setminus \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]^2 \right\}.$$

These sets represent the high- and low-frequency components, respectively.

3.2 H-Ellipticity of the Discrete Laplace Operator

To formulate a discrete solution to the original problem, we need to assess how well the discrete operator satisfies the properties of the original elliptic operator $L = -\Delta$. We arrive at the following definition from Brandt [36]:

Definition Let L_h be the discrete analogue of the differential operator L . The h -Ellipticity measure of the operator L_h is defined as

$$E_h(L_h) = \frac{\min \{ \tilde{L}_h(\theta) : \theta \in T^{high} \}}{\max \{ \tilde{L}_h(\theta) : -\pi \leq \theta \leq \pi \}},$$

with $0 < c \leq E_h(L_h) \leq 1$.

One can show that the 5-point Laplace operator given by

$$L_h(u) = -\frac{u_{i+1,j} - 2u_{ij} + u_{i-1,j}}{h^2} - \frac{u_{i,j+1} - 2u_{ij} + u_{i,j-1}}{h^2}$$

has a nonzero h -Ellipticity. We observe this is the case by applying the operator to the mode $e^{i\theta \cdot x}$:

$$L_h e^{i\theta \cdot x} = \frac{1}{h^2} (4e^{i\theta \cdot x} - e^{i\theta_1} e^{i\theta \cdot x} - e^{-i\theta_1} e^{i\theta \cdot x} - e^{i\theta_2} e^{i\theta \cdot x} - e^{-i\theta_2} e^{i\theta \cdot x}) =$$

$$L_h e^{i\theta \cdot x} = \tilde{L}_h(\theta) e^{i\theta \cdot x},$$

where

$$\tilde{L}_h(\theta) = \frac{1}{h^2} (4 - e^{i\theta_1} - e^{-i\theta_1} - e^{i\theta_2} - e^{-i\theta_2}).$$

Then we have that $E_h(L_h) = 0.25$, which occurs when $\theta = (0, \frac{\pi}{2})$ [36].

3.3 Local Mode Analysis

The tools that we will employ for error analysis are those upon which multigrid is based: Local Fourier Analysis and the operator properties of the multigrid methods, expressed as operator equations [8]. The Local Fourier analysis allows one to show what happens to certain frequencies and components of the error, and the effects of the method can be seen by how the method itself acts on the eigenfunctions (derived in section 3.1) of the discrete differential operator. Local Fourier analysis serves as a useful framework from which one can view the effects of smoothing operators and the entire multigrid scheme on the low- and high-frequency components of the error [8, 18, 36].

Suppose we are attempting to solve a linear elliptic partial differential equation given by

$$a \frac{\partial^2 u}{\partial x^2} + c \frac{\partial^2 u}{\partial y^2} = f$$

with some prescribed boundary conditions, and where a and c are constant coefficients. Furthermore, suppose we are discretizing by finite differences using a second order approximation to the second partial derivatives of u , and that our step sizes in the x - and y -directions are the same. On the k th grid, we have that the exact grid function u^k satisfies.

$$a \frac{u_{m+1,n}^k - 2u_{mn}^k + u_{m-1,n}^k}{h_k^2} + c \frac{u_{m,n+1}^k - 2u_{mn}^k + u_{m,n-1}^k}{h_k^2} = f_{mn}^k \quad (1)$$

Where f_{mn}^k is the right-hand-side function on the k th grid evaluated at the point (x_m, y_n) and h_k is the step size on grid k . When using Gauss-Seidel, we update the values u_{mn} , $u_{m-1,n}$, and $u_{m,n-1}$ of an approximate solution u with new values \bar{u}_{mn} , $\bar{u}_{m-1,n}$, and $\bar{u}_{m,n-1}$ to obtain

the following:

$$a \frac{u_{m+1,n} - 2\bar{u}_{mn} + \bar{u}_{m-1,n}}{h_k^2} + c \frac{u_{m,n+1} - 2\bar{u}_{mn} + \bar{u}_{m,n-1}}{h_k^2} = f_{mn}^k \quad (2)$$

This new approximation \bar{u} may not immediately satisfy the first finite difference equation above, and additional applications, or sweeps, of Gauss-Seidel may be necessary to improve it. Nonetheless, we introduce an important quantity, the smoothing factor, given by

$$\mu = \frac{\|\bar{v}\|}{\|v\|}, \quad \bar{v} = u^k - \bar{u}, \quad v = u^k - u,$$

where v and \bar{v} are the error functions before and after one application of Gauss-Seidel (or any other smoother). In order to determine the smoothing factor in practice, we let $\theta = (\theta_1, \theta_2)$ be the Fourier components of the error and write v and \bar{v} as

$$v = A_\theta e^{i(m\theta_1 + n\theta_2)}$$

and

$$\bar{v} = \bar{A}_\theta e^{i(m\theta_1 + n\theta_2)}.$$

Thus, subtracting (2) from (1) and multiplying through by h_k yields

$$\begin{aligned} & a \left((u_{m+1,n}^k - u_{m+1,n}) - 2(u_{mn}^k - \bar{u}_{mn}) - (u_{m-1,n}^k - \bar{u}_{m-1,n}) \right) + \\ & c \left((u_{m,n+1}^k - u_{m,n+1}) - 2(u_{mn}^k - \bar{u}_{mn}) - (u_{m,n-1}^k - \bar{u}_{m,n-1}) \right) = \end{aligned}$$

$$a(v_{m+1,n} - 2\bar{v}_{mn} - \bar{v}_{m-1,n}) + c(v_{m,n+1} - 2\bar{v}_{mn} - \bar{v}_{m,n-1}) = 0,$$

or in terms of their Fourier representations,

$$\begin{aligned} & a \left(A_\theta e^{i((m+1)\theta_1 + n\theta_2)} - 2\bar{A}_\theta e^{i(m\theta_1 + n\theta_2)} + \bar{A}_\theta e^{i((m-1)\theta_1 + n\theta_2)} \right) + \\ & c \left(A_\theta e^{i(m\theta_1 + (n+1)\theta_2)} - 2\bar{A}_\theta e^{i(m\theta_1 + n\theta_2)} + \bar{A}_\theta e^{i(m\theta_1 + (n-1)\theta_2)} \right) = 0. \end{aligned}$$

By dividing through by the common factor $e^{i(m\theta_1 + n\theta_2)}$ and collecting like terms with A_θ and \bar{A}_θ , we obtain

$$a \left(A_\theta e^{i\theta_1} - 2\bar{A}_\theta + \bar{A}_\theta e^{-i\theta_1} \right) + c \left(A_\theta e^{i\theta_2} - 2\bar{A}_\theta + \bar{A}_\theta e^{-i\theta_2} \right) = 0$$

or

$$A_\theta (ae^{i\theta_1} + ce^{i\theta_2}) + \bar{A}_\theta (ae^{-i\theta_1} + ce^{-i\theta_2} - 2a - 2c) = 0.$$

Thus, the convergence factor μ can be expressed as

$$\mu(\theta) = \frac{\|\bar{A}_\theta\|}{\|A_\theta\|} = \frac{|ae^{i\theta_1} + ce^{i\theta_2}|}{|2a + 2c - ae^{-i\theta_1} - ce^{-i\theta_2}|}.$$

Since we seek to find the worst factor by which error is reduced for high-frequency components, we seek to find the θ which maximizes μ subject to the constraint that

$$\hat{\rho}\pi \leq |\theta| \leq \pi,$$

where $|\theta| = \max\{|\theta_1|, |\theta_2|\}$ and $\hat{\rho}$ is the ratio of the current and next coarsest grid steps, i.e. $\hat{\rho} = \frac{h_k}{h_{k-1}}$. This allows one to find the worst factor $\bar{\mu}$ by which high frequency error components are reduced at each step. For a Poisson equation with $a = c$, we have $\bar{\mu} = 0.5$ [8].

The smoothers perform well initially, but convergence slows down after the first two or three sweeps. This corresponds to a smoothing of the low-frequency errors, but the purpose of multigrid is not to eliminate the low-frequency error components on the given grid. When we move to coarser grids, the lower frequency components become high frequencies on those grids. This allows those same frequencies to be reduced while even lower frequencies are approximated more precisely before we add them back to the numerical approximation of the solution. This analysis allows one to pick or create a suitable smoother that accounts for the behavior of the partial differential equation [8].

3.3.1 Spectral Properties of the Multigrid Operators

Another component of the analysis involves computing the largest eigenvalue of the multigrid operator to see at what rate the numerical approximation converges to the solution. From the section on the Coarse Grid Correction cycle, we can write the algorithm in terms of matrices and vectors as

$$\begin{aligned} u_h^1 &= S^{v_2}(u_h^0 + I_{2h}^h L_{2h}^{-1} I_h^{2h} (f_h - L_h S^{v_1} u_h^0)) = \\ &S^{v_2}(u_h^0 + M(f_h - L_h S^{v_1} u_h^0)) = \\ &S^{v_2}(I - M S^{v_1} L_h) u_h^0 + S^{v_2} M f, \end{aligned}$$

where $M = I_{2h}^h L_{2h}^{-1} I_h^{2h}$, I_h^{2h} is the restriction operator which transfers the approximation u_h^0 from the fine grid to the coarse grid, and I_{2h}^h is the interpolation operator which transfers

the approximation u_{2h}^0 from the coarse grid to the fine grid.

This component of the analysis looks at the spectral radius of the multigrid operator given by

$$M_h^{2h} = S^{v_2}(I - MS^{v_1}L_h),$$

which yields the factor by which the error is reduced at each step. Note also that the coarse grid correction scheme without smoothing yields the operator

$$I - ML_h,$$

which is known to have eigenvalues larger than 1 [36]. The use of smoothing operators, however, works in conjunction with coarse grid correction to provide the overall multigrid operator with a spectral radius that is less than 1, independent of the step size of the finest grid.

Chapter 4

Application of Multigrid to the Debye-Hückel Equation

Recall that we seek to solve the Debye-Hückel. For the solutions of this equation, we use the operator-dependent interpolation described in Chapter 3. In addition to testing numerical solutions of this equation with the dielectric coefficient ε represented as a jump function at the interface of the dielectric spheres, we employ the transition function given below from [13].

$$\varepsilon = \varepsilon_m - (\varepsilon_i - \varepsilon_m) \frac{e^{(|r|-R)/S}}{1 + e^{(|r|-R)/S}}$$

In this definition of ε , $|r|$ is the euclidean distance from the center of the sphere, R is the radius of the sphere, and S is the characteristic length over which the change in the dielectric coefficient takes place [13]. We take S to be 0.2 Å in this case.

In the following section, we use local Fourier Analysis from Chapter 4 to justify the use of the operator-dependent interpolation method.

4.1 Local Fourier Analysis of the Debye-Hückel Equation

We begin by taking the same approach as in [36] by writing the differential operator

$$L(u) = -\nabla \cdot (\varepsilon \nabla u) + \kappa^2 u$$

as follows:

$$L_h^- w^- + L_h^+ w^+,$$

where L_h^- is the portion of discrete operator stencil that acts on a previously updated approximation (or correction) w^- and w^+ is an approximation upon which an operation has yet to take place. The stencil for the Debye-Hückel equation used in two dimensions is given by

$$L_h = \frac{1}{h^2} \begin{bmatrix} & & -\epsilon_{i,j+1/2} & & \\ & -\epsilon_{i-1/2,j} & (\epsilon_{i-1/2,j} + \epsilon_{i+1/2,j} + \epsilon_{i,j-1/2} + \epsilon_{i,j+1/2}) & -\epsilon_{i+1/2,j} & \\ & & -\epsilon_{i,j-1/2} & & \end{bmatrix} + \begin{bmatrix} 0 & & \\ 0 & \kappa_{ij}^2 & 0 \\ 0 & & 0 \end{bmatrix}.$$

We split this discrete operator into its constituent parts L_h^- and L_h^+ as follows:

$$\begin{aligned}
L_h^- &= \frac{1}{h^2} \begin{bmatrix} & & 0 & \\ -\epsilon_{i-1/2,j} & (\epsilon_{i-1/2,j} + \epsilon_{i+1/2,j} + \epsilon_{i,j-1/2} + \epsilon_{i,j+1/2}) & & 0 \\ & & -\epsilon_{i,j-1/2} & \\ & & & \end{bmatrix} + \\
&\quad \kappa_{ij}^2 \begin{bmatrix} & 0 & \\ 0 & 1 & 0 \\ & 0 & \end{bmatrix} . \\
L_h^+ &= \frac{1}{h^2} \begin{bmatrix} & -\epsilon_{i,j+1/2} & \\ 0 & 0 & -\epsilon_{i+1/2,j} \\ & 0 & \end{bmatrix} + \kappa_{ij}^2 \begin{bmatrix} & 0 & \\ 0 & 0 & 0 \\ & 0 & \end{bmatrix} = \\
&\quad \frac{1}{h^2} \begin{bmatrix} & -\epsilon_{i,j+1/2} & \\ 0 & 0 & -\epsilon_{i+1/2,j} \\ & 0 & \end{bmatrix} .
\end{aligned}$$

When acting on the formal eigenfunction $e^{i\theta \cdot x}$, we have that

$$\begin{aligned}
L_h^- e^{i\theta \cdot x} &= \left[\frac{1}{h^2} \left((\epsilon_{i-1/2,j} + \epsilon_{i+1/2,j} + \epsilon_{i,j-1/2} + \epsilon_{i,j+1/2}) - \right. \right. \\
&\quad \left. \left. \epsilon_{i-1/2,j} e^{-i\theta_1} - \epsilon_{i,j-1/2} e^{-i\theta_2} \right) + \kappa_{ij}^2 \right] e^{i\theta \cdot x} .
\end{aligned}$$

Similarly, we have that

$$L_h^+ e^{i\theta \cdot x} = \frac{1}{h^2} \left[\epsilon_{i+1/2,j} e^{i\theta_1} + \epsilon_{i,j+1/2} e^{i\theta_2} \right] e^{i\theta \cdot x} .$$

Denote by $\tilde{L}_h^-(\theta)$ and $\tilde{L}_h^+(\theta)$ the formal eigenvalues computed above for L_h^- and L_h^+ , respec-

tively. Then the smoothing factor is given by

$$\mu(\theta) = \frac{|L_h^+(\theta)|}{|L_h^-(\theta)|},$$

for which we need to find the supremum of μ over all high-frequency components θ :

$$\begin{aligned} \mu(\theta) &= \frac{\frac{1}{h^2} |\epsilon_{i+1/2,j} e^{i\theta_1} + \epsilon_{i,j+1/2} e^{i\theta_2}|}{\left| \frac{1}{h^2} ((\epsilon_{i-1/2,j} + \epsilon_{i+1/2,j} + \epsilon_{i,j-1/2} + \epsilon_{i,j+1/2}) - \epsilon_{i-1/2,j} e^{-i\theta_1} - \epsilon_{i,j-1/2} e^{-i\theta_2}) + \kappa_{ij}^2 \right|} \\ &= \frac{|\epsilon_{i+1/2,j} e^{i\theta_1} + \epsilon_{i,j+1/2} e^{i\theta_2}|}{|\epsilon_{i-1/2,j} + \epsilon_{i+1/2,j} + \epsilon_{i,j-1/2} + \epsilon_{i,j+1/2} - \epsilon_{i-1/2,j} e^{-i\theta_1} - \epsilon_{i,j-1/2} e^{-i\theta_2} + h^2 \kappa_{ij}^2|} \end{aligned}$$

In the above expression $\epsilon_{i\pm 1/2,j}$ and $\epsilon_{i,j\pm 1/2}$ is the dielectric coefficient ϵ evaluated at the half-index points $(x_{i\pm 1/2}, y_j)$ and $(x_i, y_{j\pm 1/2})$.

Brandt in [8] states that one may ignore changes in the coefficient by locally treating coefficients as constant. Interfaces with strong jumps and discontinuities, however, must be taken into account because smoothing rates can deteriorate in these regions. If, for example, $\epsilon_{i+1/2,j} \gg \epsilon_{i-1/2,j}$ and $\epsilon_{i,j\pm 1/2}$ are of the same order as $\epsilon_{i-1/2,j}$ then the smoothing factor approaches 1 quickly in this region. It should be noted, however, that $\kappa \geq 0$ can reduce the smoothing factor, especially when ϵ is smooth.

For our purposes, however, the dielectric coefficient is relatively smooth when one is sufficiently far from the interface. Thus, one could use the normal differencing stencil and bilinear interpolation. Near the interface, however, we choose to use the operator-dependent interpolation defined in Chapter 3, with numerical results given in the following chapter.

Chapter 5

Numerical Results

Introduction

The following sections give numerical results for the application of the multigrid V-Cycle to numerous test problems and the Debye-Hückel equation. The error tables given for each problem measure the norm of the residual $r = f - Au$ after each iteration. We first give examples in two dimensions, followed by results for the Debye-Hückel equation in three dimensions.

5.1 2D Results

As an example of the utility of multigrid V-Cycle, we show the results of running the multigrid V-Cycle on the unit square $\Omega = [0, 1]^2$ for the following problem solved by Brandt [8]:

$$-\Delta u = \sin(3(x + y)), (x, y) \in \Omega$$

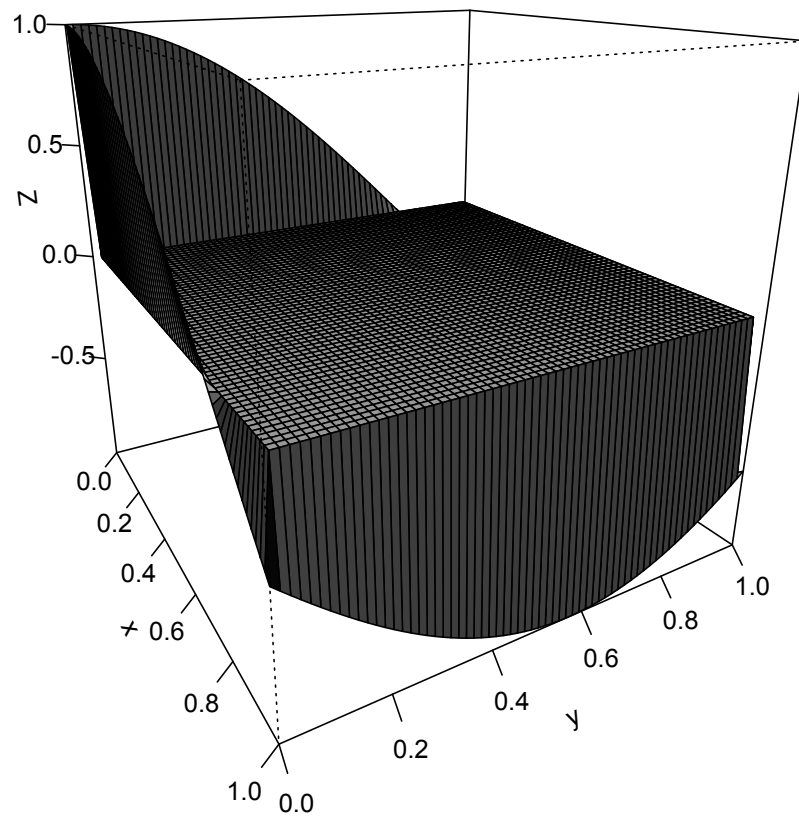
$$u = \cos(2(x + y)), (x, y) \in \partial\Omega.$$

Graphs of the solution, as well as error and residual norms, are given below. For this par-

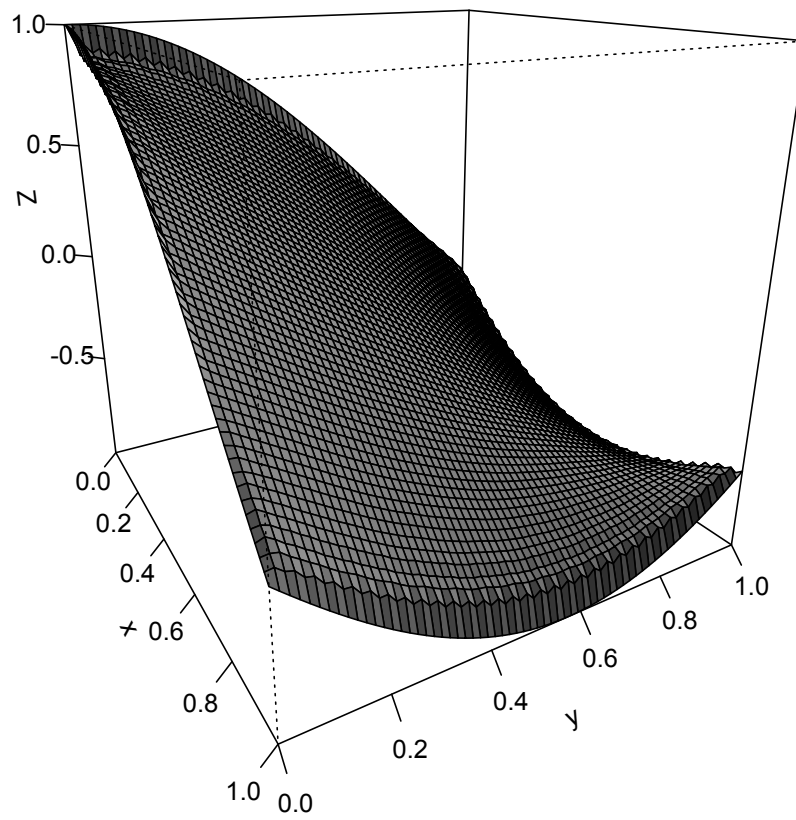
ticular program, a total of six grids were used with an initial step size of $h = \frac{1}{2}$. The table below show the norms of the residuals in three norms for each run of the V-Cycle. To reduce the residual norm to 1.52E-06 in the L^2 -Norm, we had to expend approximately 26 WU.

Cycle	Grid Level	L^1 -Norm	L^2 -Norm	L^∞ -Norm	WU
-	6	7.15E+05	4.94E+04	8.19E+03	0.00E+00
1	6	8.53E+04	5.90E+03	6.89E+02	2.68E+00
2	6	6.44E+03	4.23E+02	4.95E+01	5.25E+00
3	6	5.65E+02	3.18E+01	3.60E+00	7.88E+00
4	6	5.13E+01	2.56E+00	2.75E-01	1.05E+01
5	6	4.76E+00	2.17E-01	2.17E-02	1.31E+01
6	6	4.49E-01	1.91E-02	2.09E-03	1.58E+01
7	6	4.32E-02	1.74E-03	2.07E-04	1.84E+01
8	6	4.23E-03	1.62E-04	2.05E-05	2.10E+01
9	6	4.23E-04	1.55E-05	2.03E-06	2.36E+01
10	6	4.28E-05	1.52E-06	2.02E-07	2.63E+01

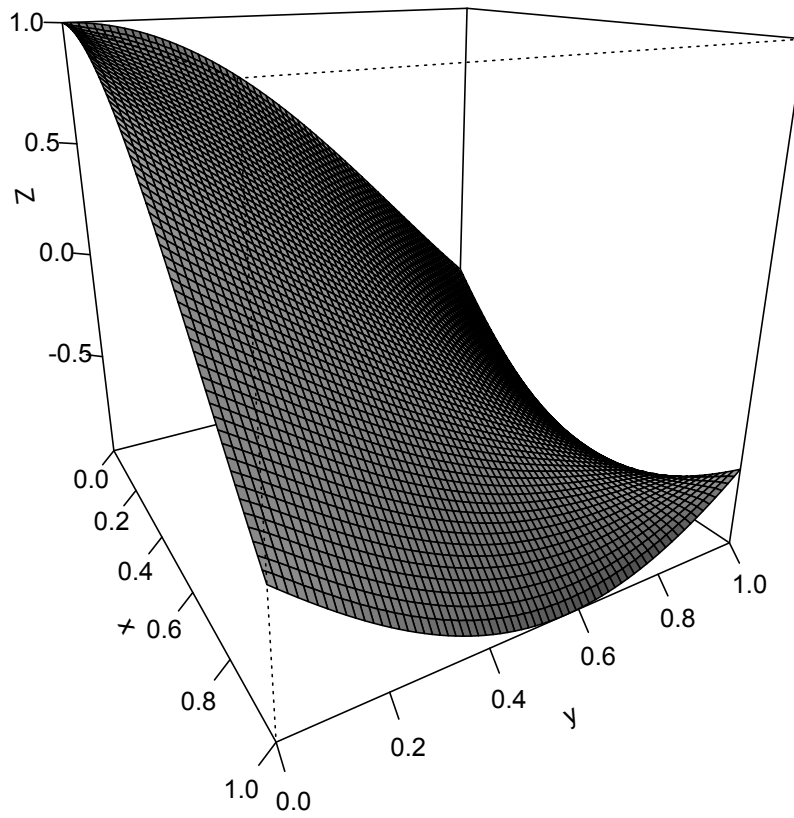
Fortran Results: Iteration 0



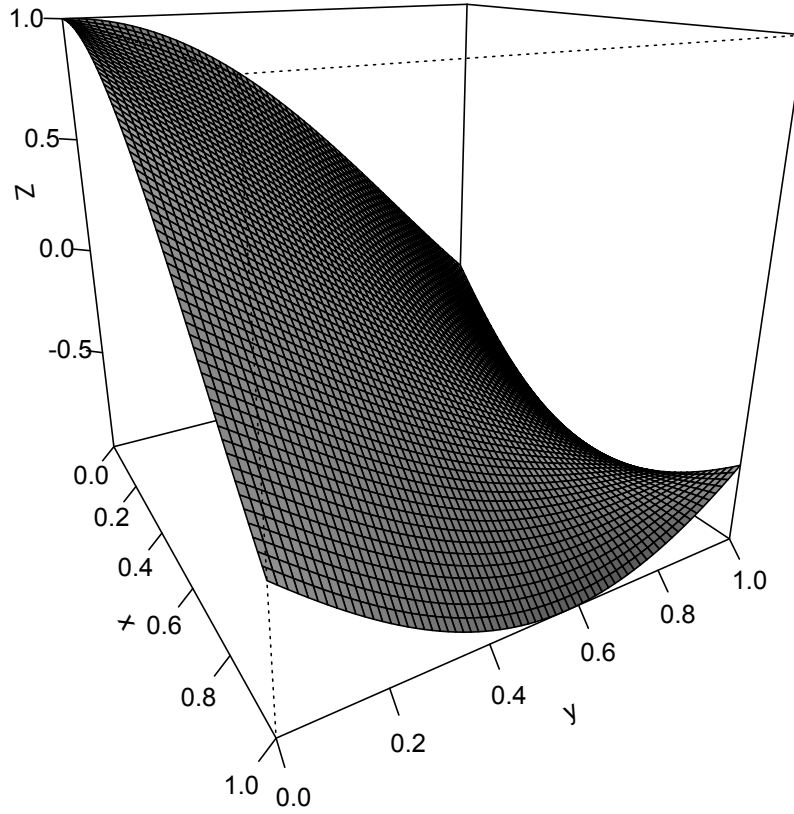
Fortran Results: Iteration 1



Fortran Results: Iteration 2



Fortran Results: Iteration 3



When using six grids, running a total of ten V-Cycles took less than one second. Below we give another table of residual norms and resulting plots when using the W-Cycle. Note that the residual norms are reduced by approximately two orders of magnitude per cycle. A total of approximately 28 WU were expended when computing the solution using five W-Cycles.

Cycle	Grid Level	L^1 -Norm	L^2 -Norm	L^∞ -Norm	WU
-	6	7.15E+05	4.94E+04	8.19E+03	0.00E+00
1	6	1.89E+04	1.17E+03	1.27E+02	5.54E+00
2	6	3.08E+02	1.73E+01	1.83E+00	1.11E+01
3	6	4.92E+00	2.64E-01	2.75E-02	1.66E+01
4	6	7.83E-02	4.08E-03	4.20E-04	2.22E+01
5	6	1.24E-03	6.33E-05	6.47E-06	2.77E+01

5.1.1 Model Problem 1

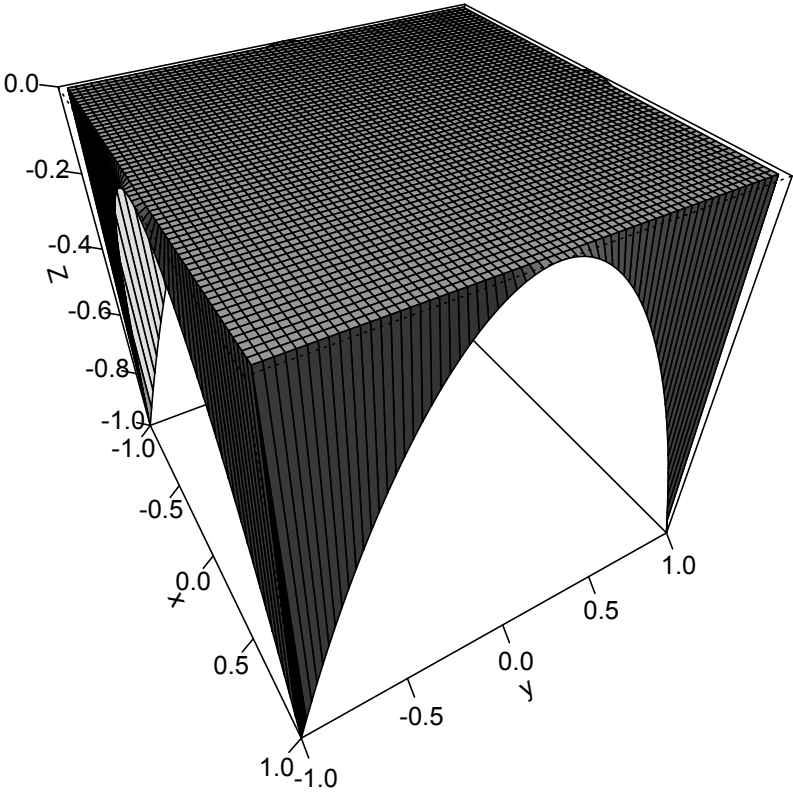
The following results are given for the first model problem

$$\begin{cases} -\frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial y^2} = 4, x \in \Omega \\ u = 1 - x^2 - y^2, x \in \partial\Omega \end{cases},$$

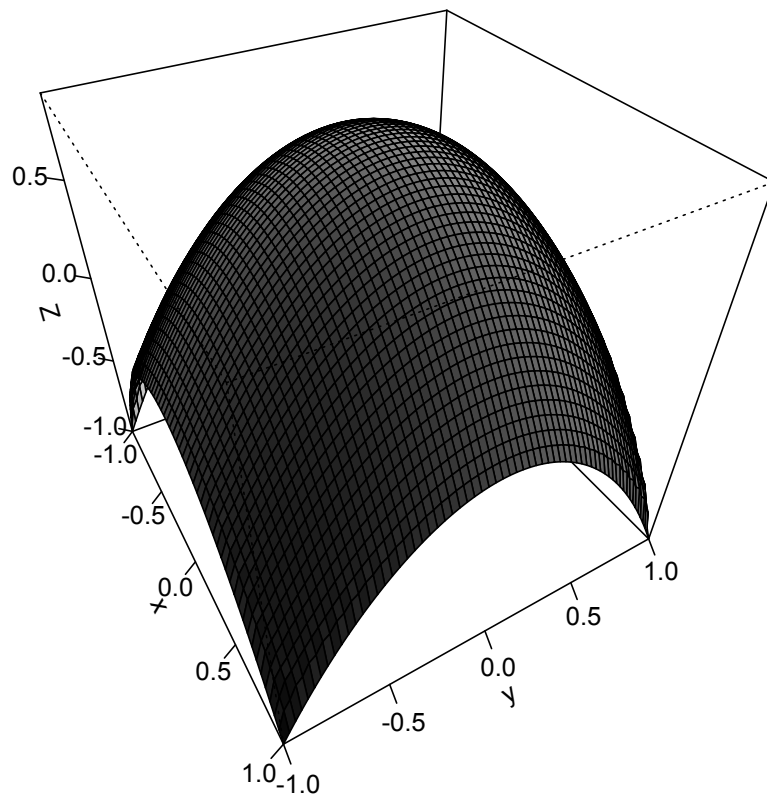
where $\Omega = (-1, 1)^2$. The results are presented for a V-Cycle on six grids with three initial gridpoints in each direction.

Cycle	Grid Level	L^1 -Norm	L^2 -Norm	L^∞ -Norm	WU
-	6	9.73E+04	7.51E+03	1.92E+03	0.00E+00
1	6	1.04E+04	7.95E+02	1.30E+02	2.63E+00
2	6	8.02E+02	5.49E+01	8.35E+00	5.25E+00
3	6	7.21E+01	4.11E+00	5.89E-01	7.88E+00
4	6	6.85E+00	3.37E-01	4.45E-02	1.05E+01
5	6	6.74E-01	2.95E-02	3.88E-03	1.31E+01
6	6	6.86E-02	2.72E-03	3.80E-04	1.58E+01
7	6	7.15E-03	2.60E-04	3.65E-05	1.84E+01
8	6	7.61E-04	2.57E-05	3.49E-06	2.10E+01
9	6	8.25E-05	2.61E-06	3.35E-07	2.36E+01
10	6	9.09E-06	2.71E-07	3.21E-08	2.63E+01

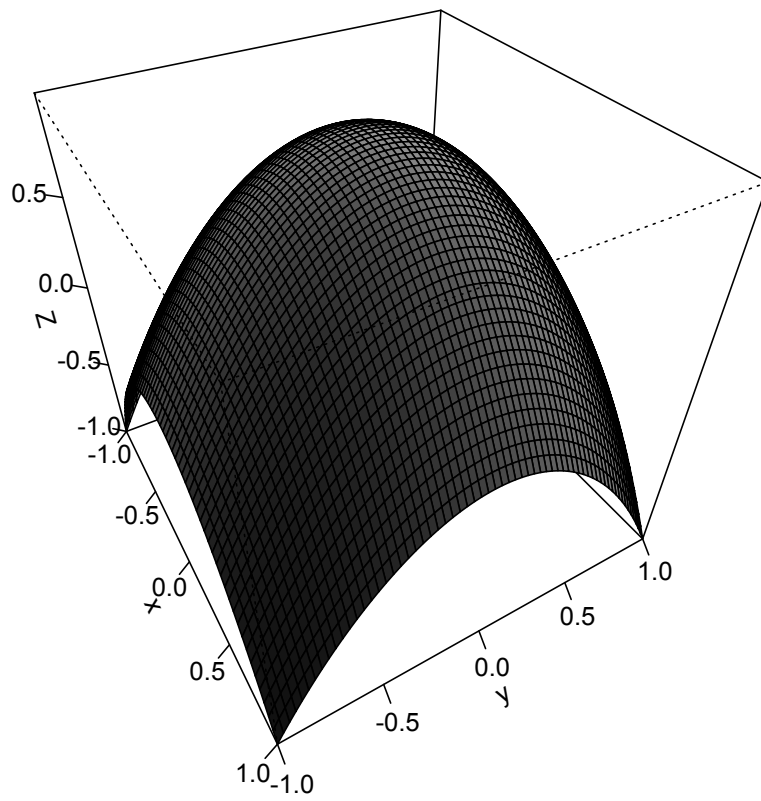
Fortran Results: Iteration 0



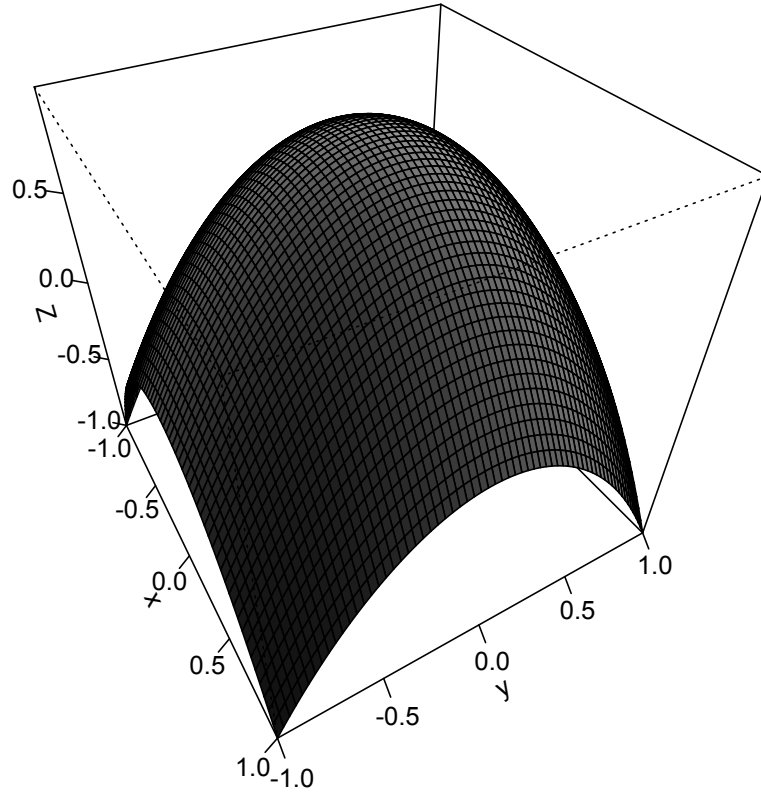
Fortran Results: Iteration 1



Fortran Results: Iteration 2



Fortran Results: Iteration 3



5.1.2 Model Problem 2

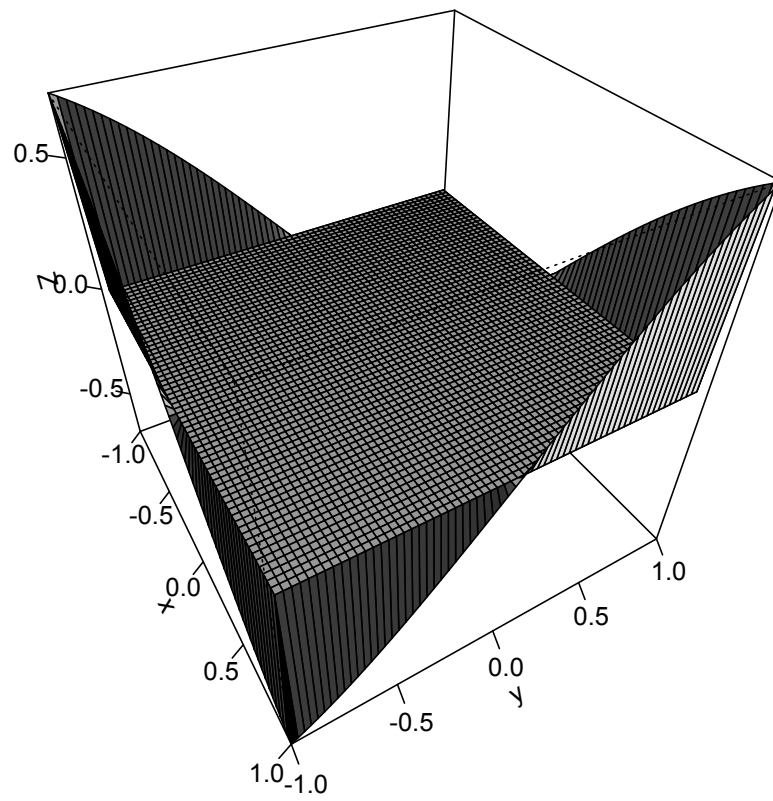
The following results are given for the second model problem

$$\begin{cases} -\frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial y^2} = 2 \sin(x) \cos(x), x \in \Omega \\ u = \sin(x) \cos(x), x \in \partial\Omega \end{cases},$$

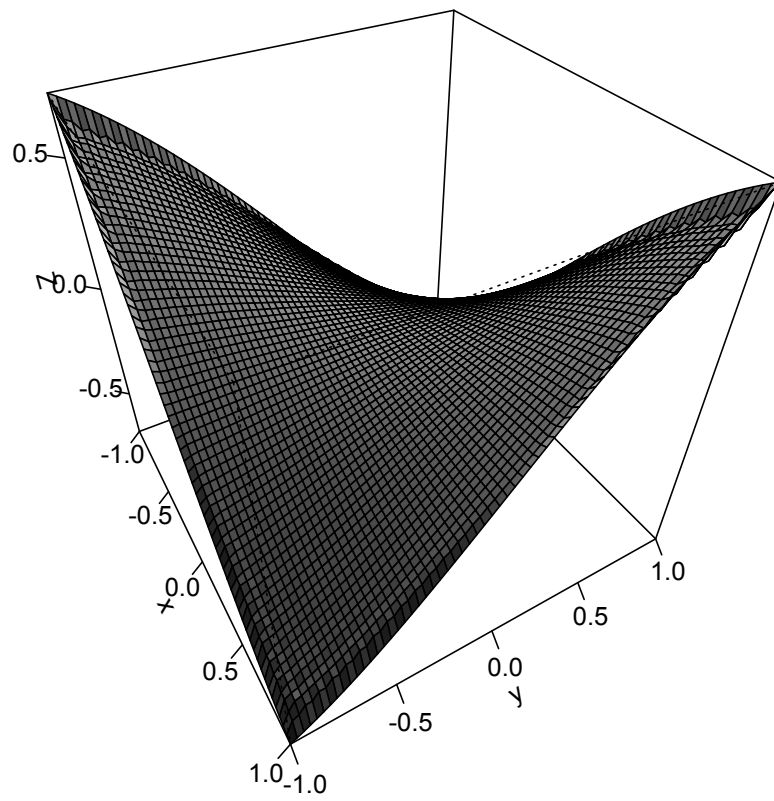
where $\Omega = (-1, 1)^2$. The results are presented for a V-Cycle on six grids with three initial gridpoints in each direction.

Cycle	Grid Level	L^1 -Norm	L^2 -Norm	L^∞ -Norm	WU
-	6	1.00E+05	7.35E+03	1.42E+03	0.00E+00
1	6	1.15E+04	8.31E+02	1.07E+02	2.68E+00
2	6	8.66E+02	5.87E+01	7.25E+00	5.25E+00
3	6	7.66E+01	4.42E+00	5.09E-01	7.88E+00
4	6	7.17E+00	3.63E-01	3.87E-02	1.05E+01
5	6	6.94E-01	3.18E-02	3.41E-03	1.31E+01
6	6	6.97E-02	2.93E-03	3.42E-04	1.58E+01
7	6	7.19E-03	2.82E-04	3.38E-05	1.84E+01
8	6	7.61E-04	2.83E-05	3.34E-06	2.10E+01
9	6	8.24E-05	2.93E-06	3.31E-07	2.36E+01
10	6	9.09E-06	3.13E-07	3.31E-08	2.63E+01

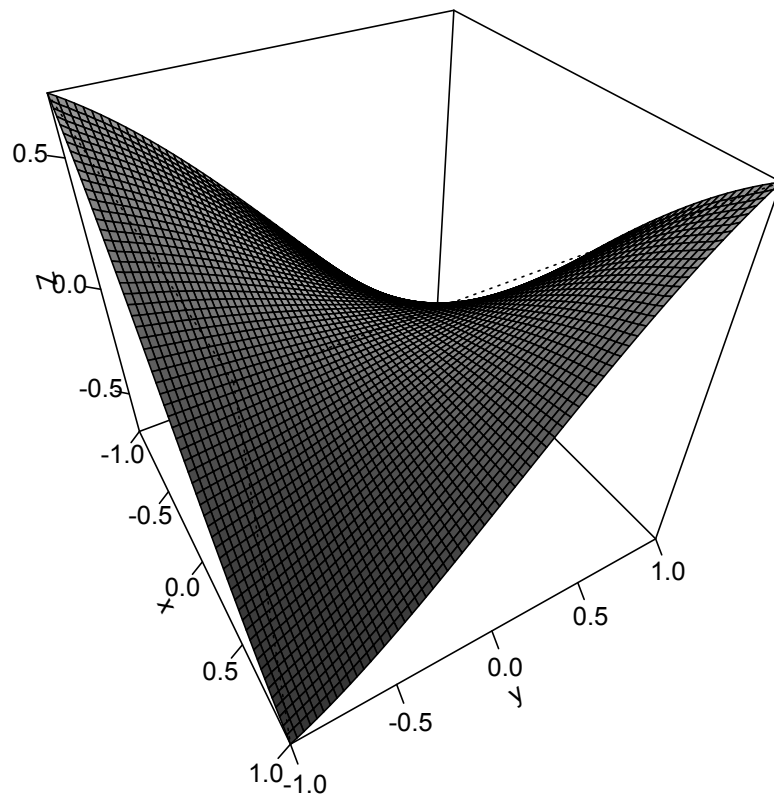
Fortran Results: Iteration 0



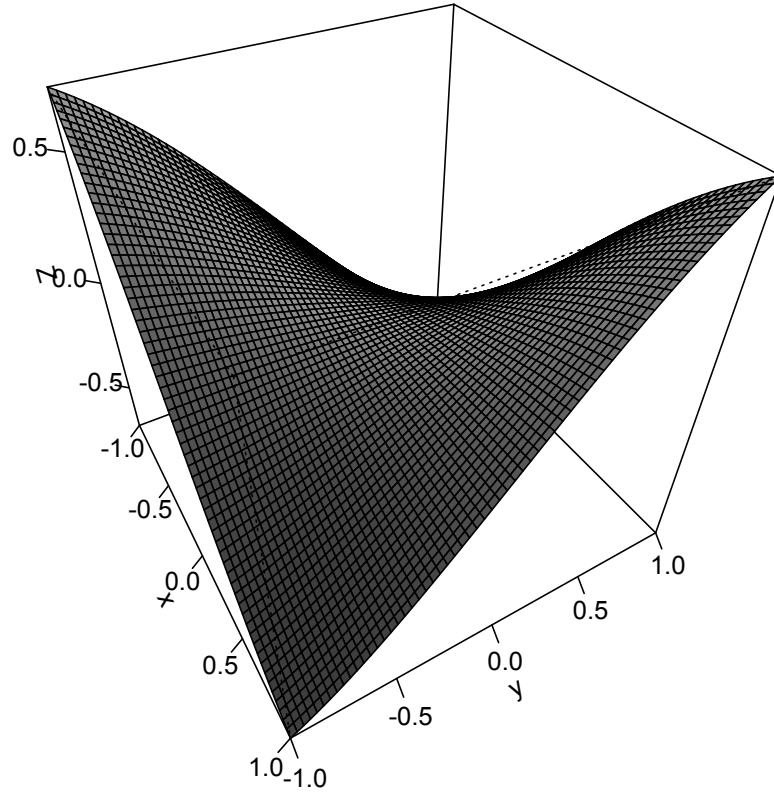
Fortran Results: Iteration 1



Fortran Results: Iteration 2



Fortran Results: Iteration 3



5.1.3 Model Problem 3

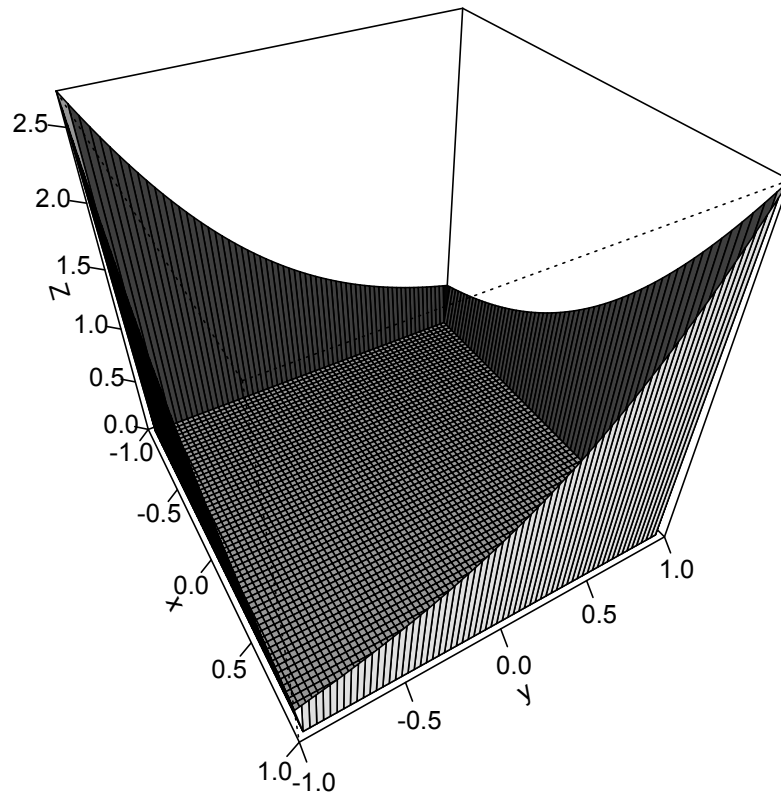
The following results are given for the third model problem

$$\begin{cases} -\frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial y^2} = -e^{xy}(x^2 + y^2), x \in \Omega \\ u = e^{xy}, x \in \partial\Omega \end{cases},$$

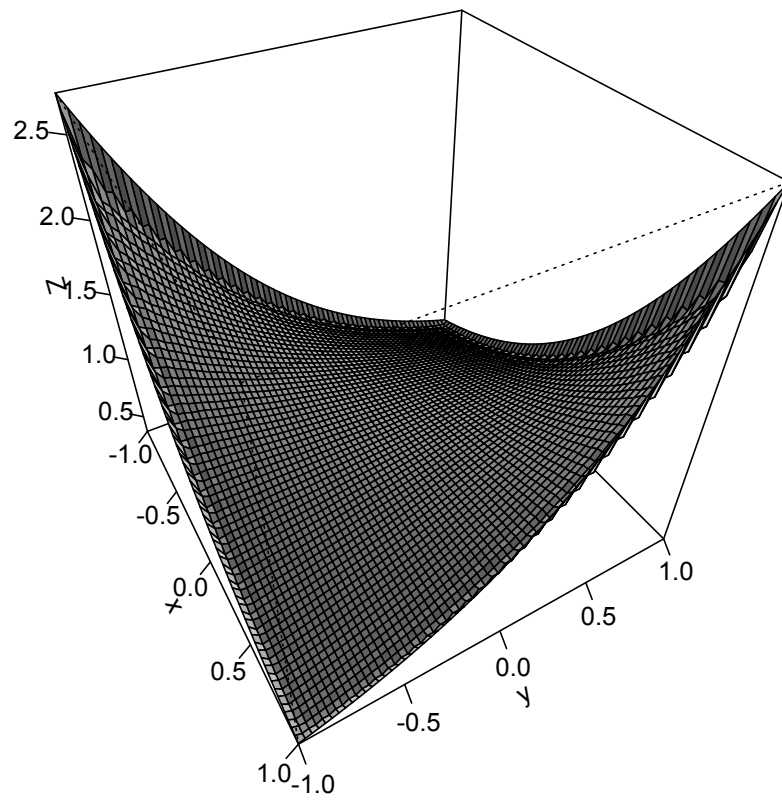
where $\Omega = (-1, 1)^2$. The results are presented for a V-Cycle on six grids with three initial gridpoints in each direction.

Cycle	Grid Level	L^1 -Norm	L^2 -Norm	L^∞ -Norm	WU
-	6	3.04E+05	2.24E+04	5.39E+03	0.00E+00
1	6	3.57E+04	2.58E+03	3.94E+02	2.68E+00
2	6	2.68E+03	1.83E+02	2.68E+01	5.25E+00
3	6	2.35E+02	1.38E+01	1.88E+00	7.88E+00
4	6	2.18E+01	1.12E+00	1.47E-01	1.05E+01
5	6	1.99E+00	9.61E-02	1.25E-02	1.31E+01
6	6	1.90E-01	8.62E-03	1.26E-03	1.58E+01
7	6	1.86E-02	8.02E-04	1.24E-04	1.84E+01
8	6	1.87E-03	7.71E-05	1.22E-05	2.10E+01
9	6	1.91E-04	7.65E-06	1.20E-06	2.36E+01
10	6	1.20E-05	7.82E-07	1.19E-07	2.63E+01

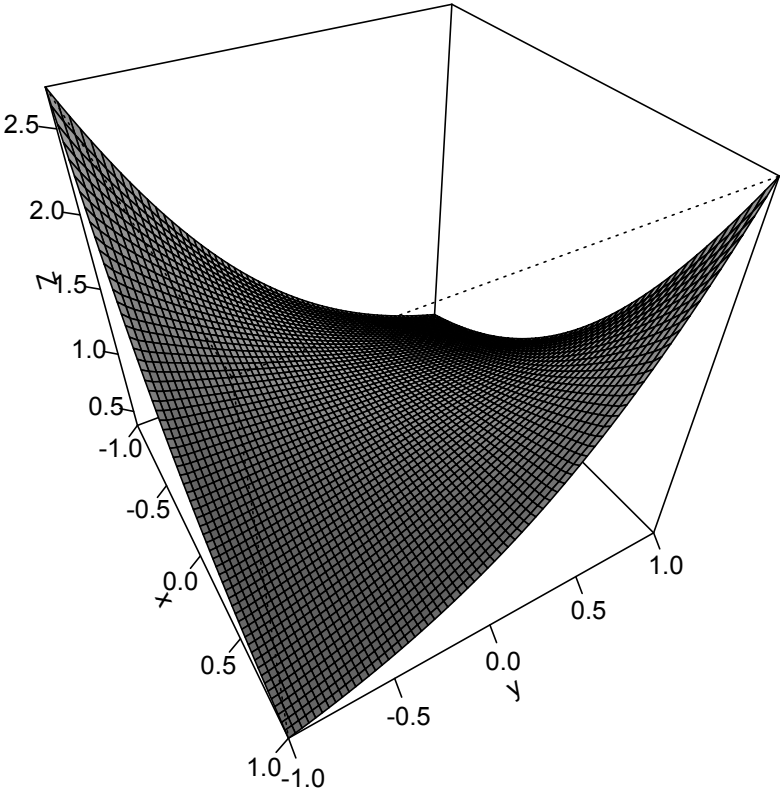
Fortran Results: Iteration 0



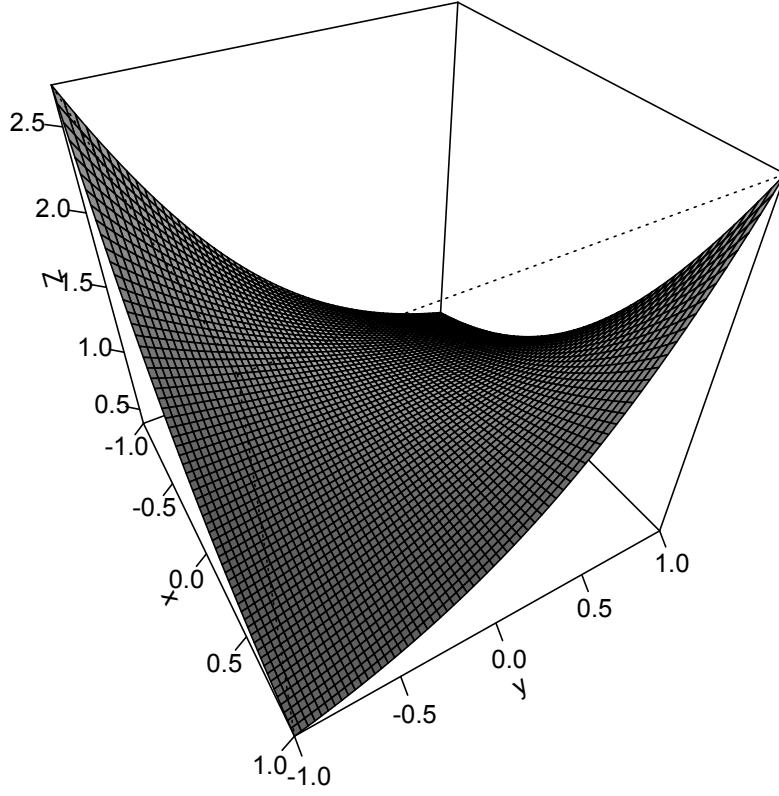
Fortran Results: Iteration 1



Fortran Results: Iteration 2



Fortran Results: Iteration 3



5.1.4 Variable Coefficient Problem 1

The following results are given for the first variable coefficient problem

$$\begin{cases} -\nabla \cdot (\epsilon \nabla u) = -12 - 8x - 8y, x \in \Omega \\ u = (x + y)^2, x \in \partial\Omega \end{cases},$$

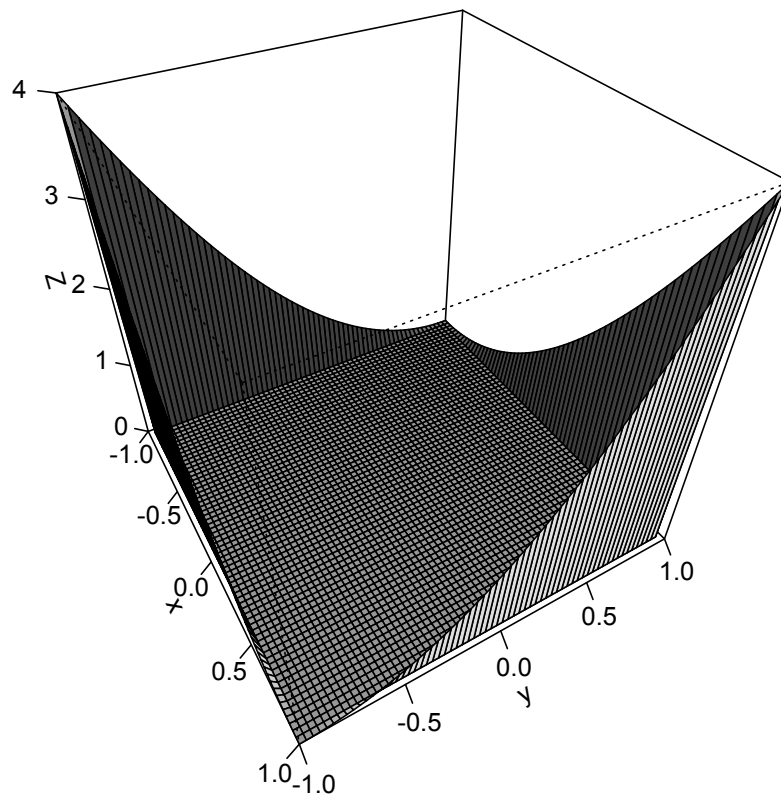
where $\Omega = (-1, 1)^2$ and $\epsilon = 3 + x + y$. The results are presented for a V-Cycle on six grids with three initial gridpoints in each direction. The first table gives results from ten V-cycles using Red-Black Gauss-Seidel, bi-linear interpolation, and full-weighting.

Cycle	Grid Level	L^1 -Norm	L^2 -Norm	L^∞ -Norm	WU
-	6	1.07E+06	1.02E+05	3.93E+04	0.00E+00
1	6	1.22E+05	1.15E+04	2.76E+03	2.63E+00
2	6	9.23E+03	7.98E+02	1.82E+02	5.25E+00
3	6	8.20E+02	5.91E+01	1.25E+01	7.88E+00
4	6	7.72E+01	4.79E+00	9.25E-01	1.05E+01
5	6	7.63E+00	4.18E-01	7.77E-02	1.31E+01
6	6	7.93E-01	3.93E-02	7.54E-03	1.58E+01
7	6	8.64E-02	3.96E-03	7.15E-04	1.84E+01
8	6	9.84E-03	4.28E-04	6.74E-05	2.10E+01
9	6	1.16E-03	4.90E-05	6.68E-06	2.36E+01
10	6	1.41E-04	5.88E-06	8.03E-07	2.63E+01

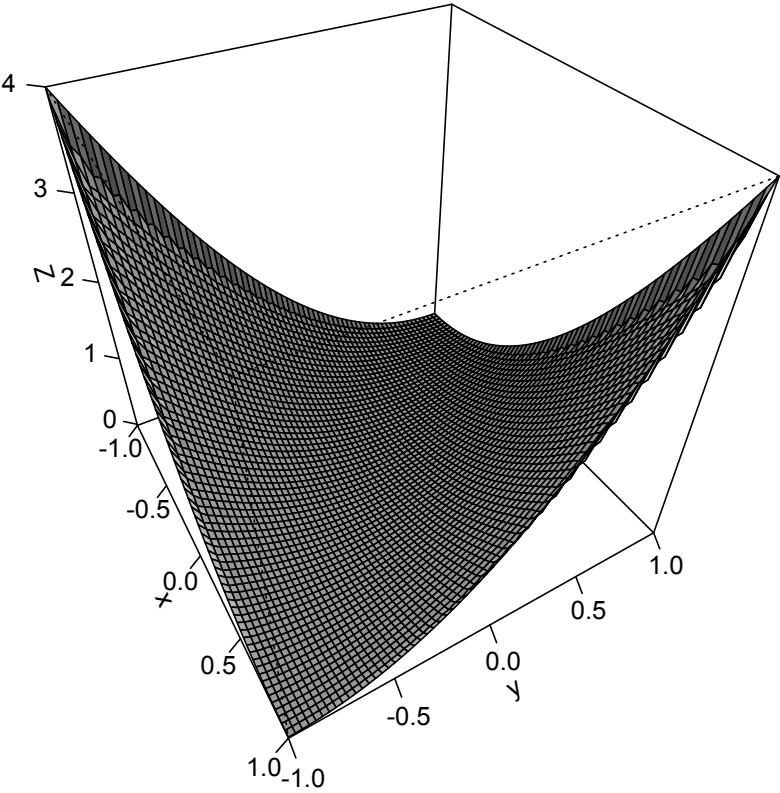
The next table gives the results of ten iterations of the same problem, but with the operator interpolation defined in Chapter 5.

Cycle	Grid Level	L^1 -Norm	L^2 -Norm	L^∞ -Norm	WU
-	6	1.07E+06	1.02E+05	3.93E+04	0.00E+00
1	6	1.22E+05	1.16E+04	2.79E+03	2.63E+00
2	6	9.22E+03	8.13E+02	1.87E+02	5.25E+00
3	6	8.14E+02	6.07E+01	1.32E+01	7.88E+00
4	6	7.53E+01	4.91E+00	1.01E+00	1.05E+01
5	6	7.21E+00	4.21E-01	8.52E-02	1.31E+01
6	6	7.15E-01	3.79E-02	8.47E-03	1.58E+01
7	6	7.34E-02	3.55E-03	8.27E-04	1.84E+01
8	6	7.76E-03	3.45E-04	8.04E-05	2.10E+01
9	6	8.41E-04	3.49E-05	7.85E-06	2.36E+01
10	6	9.34E-05	3.67E-06	7.72E-07	2.63E+01

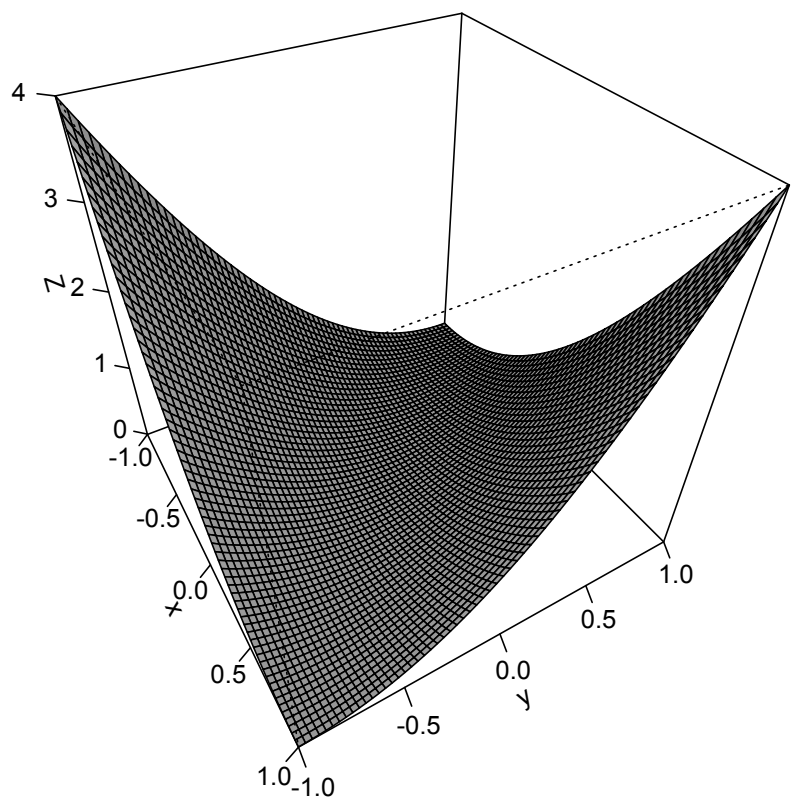
Fortran Results: Iteration 0



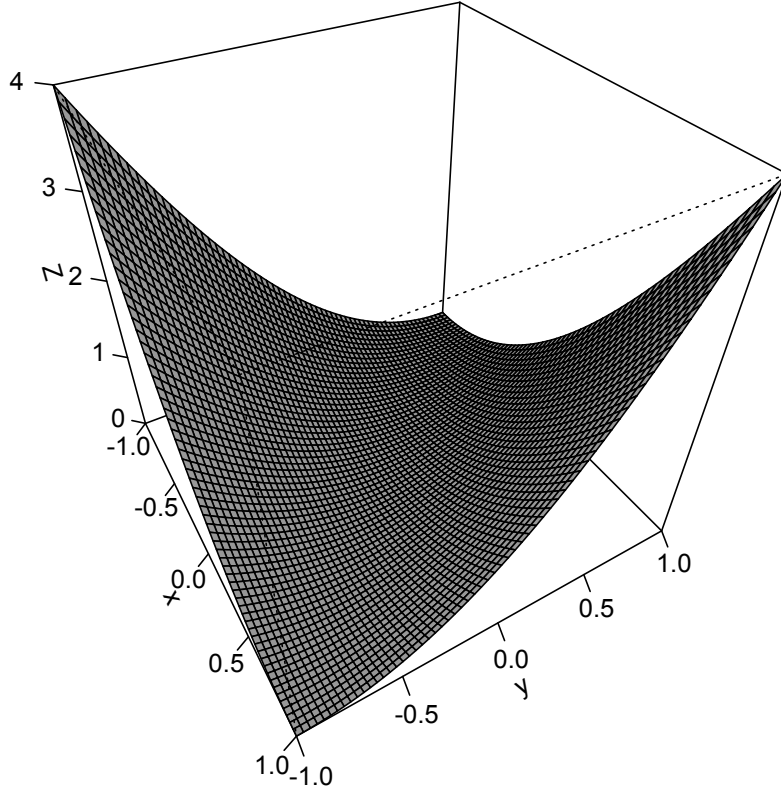
Fortran Results: Iteration 1



Fortran Results: Iteration 2



Fortran Results: Iteration 3



5.1.5 Variable Coefficient Problem 2

The following results are given for the second variable coefficient problem

$$\begin{cases} -\nabla \cdot (\epsilon \nabla u) = -2e^{x+y}(2 + \cos(xy)) + e^{x+y} \sin(xy)(x + y), x \in \Omega \\ u = e^{x+y}, x \in \partial\Omega \end{cases},$$

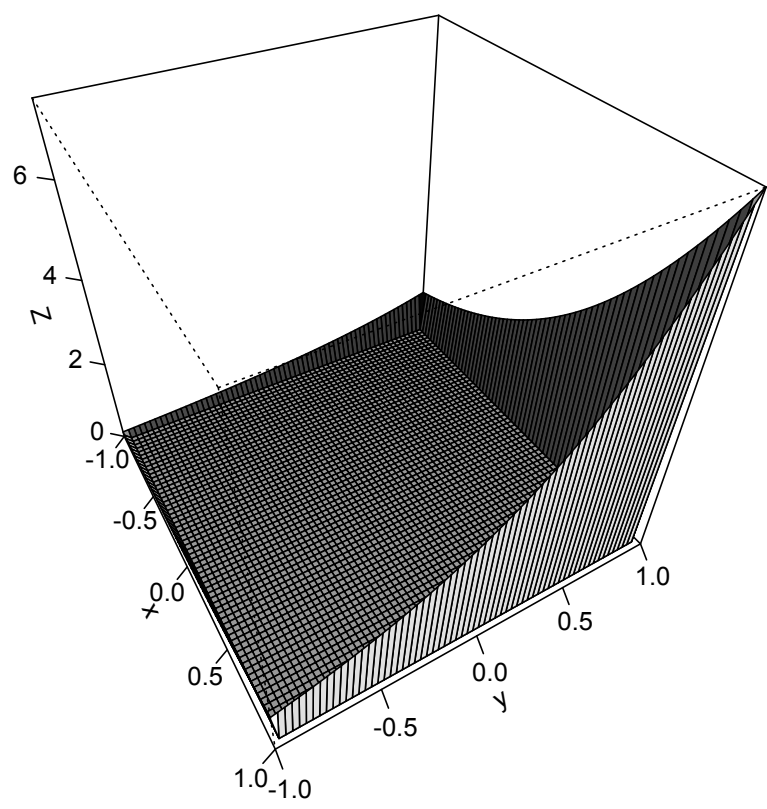
where $\Omega = (-1, 1)^2$ and $\epsilon = 2 + \cos(xy)$. The results are presented for a V-Cycle on six grids with three initial gridpoints in each direction. The first table below gives the results of running ten V-Cycles with bilinear interpolation.

Cycle	Grid Level	L^1 -Norm	L^2 -Norm	L^∞ -Norm	WU
-	6	1.35E+06	1.21E+05	3.78E+04	0.00E+00
1	6	1.58E+05	1.42E+04	2.91E+03	2.63E+00
2	6	1.21E+04	1.03E+03	2.03E+02	5.25E+00
3	6	1.08E+03	7.87E+01	1.47E+01	7.88E+00
4	6	1.01E+02	6.56E+00	1.16E+00	1.05E+01
5	6	9.64E+00	5.81E-01	1.05E-01	1.31E+01
6	6	9.52E-01	5.40E-02	1.08E-02	1.58E+01
7	6	9.68E-02	5.22E-03	1.11E-03	1.84E+01
8	6	1.01E-02	5.21E-04	1.13E-04	2.10E+01
9	6	1.09E-03	5.37E-05	1.17E-05	2.36E+01
10	6	1.19E-04	5.68E-06	1.21E-06	2.63E+01

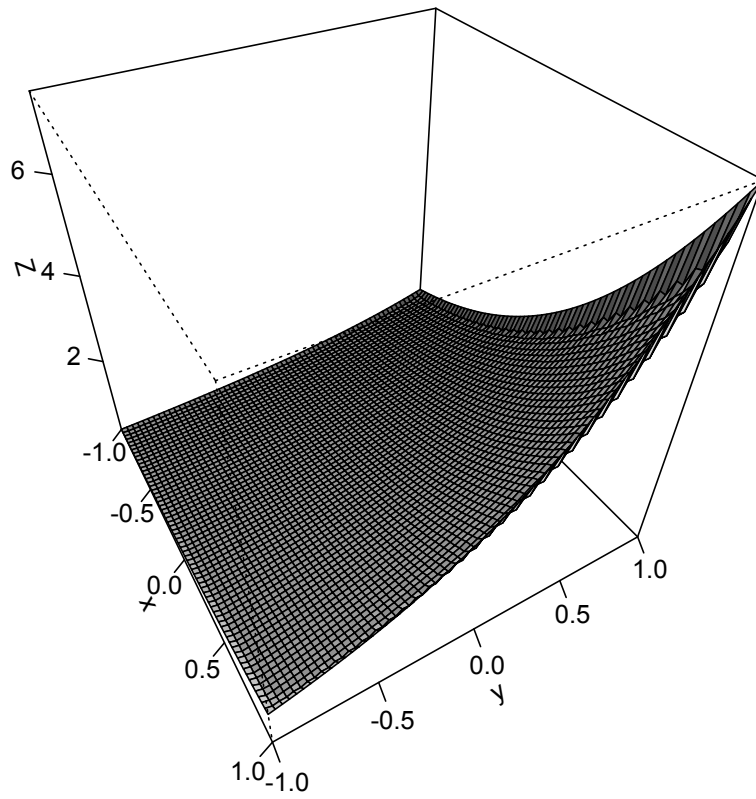
The next table gives the results of 10 iterations of the same problem, but with operator interpolation.

Cycle	Grid Level	L^1 -Norm	L^2 -Norm	L^∞ -Norm	WU
-	6	1.35E+06	1.21E+05	3.78E+04	0.00E+00
1	6	1.57E+05	1.41E+04	2.87E+03	2.63E+00
2	6	1.19E+04	1.01E+03	1.97E+02	5.25E+00
3	6	1.05E+03	7.61E+01	1.39E+01	7.88E+00
4	6	9.63E+01	6.22E+00	1.05E+00	1.05E+01
5	6	9.07E+00	5.39E-01	9.45E-02	1.31E+01
6	6	8.80E-01	4.88E-02	9.52E-03	1.58E+01
7	6	8.78E-02	4.58E-03	9.44E-04	1.84E+01
8	6	9.00E-03	4.45E-04	9.33E-05	2.10E+01
9	6	9.46E-04	4.46E-05	9.27E-06	2.36E+01
10	6	1.02E-04	4.60E-06	9.27E-07	2.63E+01

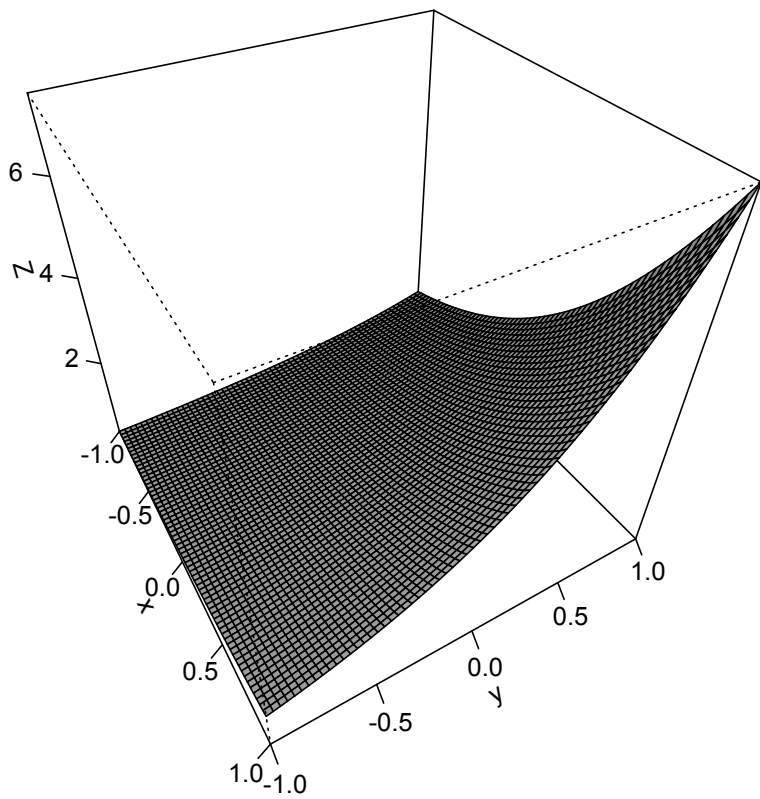
Fortran Results: Iteration 0



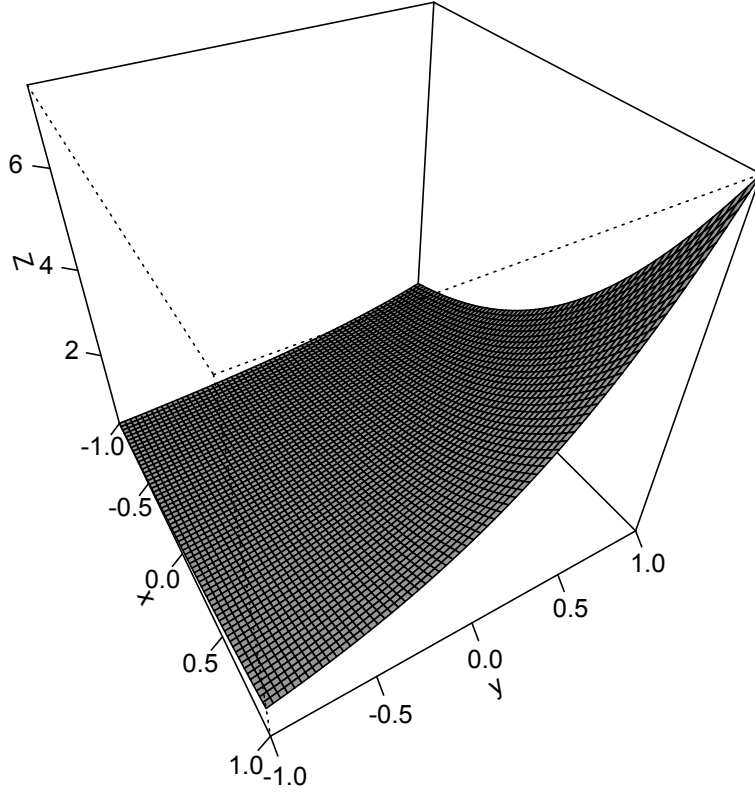
Fortran Results: Iteration 1



Fortran Results: Iteration 2



Fortran Results: Iteration 3



5.1.6 Variable Coefficient Problem 3

The following results are given for the second variable coefficient problem

$$\begin{cases} -\nabla \cdot (\epsilon \nabla u) + \kappa^2 u = -2(x + y) + (3 + \sin(x + y))^2(1 + x + y), x \in \Omega \\ u = 1 + x + y, x \in \partial\Omega \end{cases},$$

where $\Omega = (-1, 1)^2$, $\epsilon = 1 + x^2 + y^2$, and $\kappa = 3 + \sin(x + y)$. The results are presented for a V-Cycle on six grids with three initial gridpoints in each direction. For this problem, we used the operator-dependent interpolation defined in Chapter 5. The first table gives the

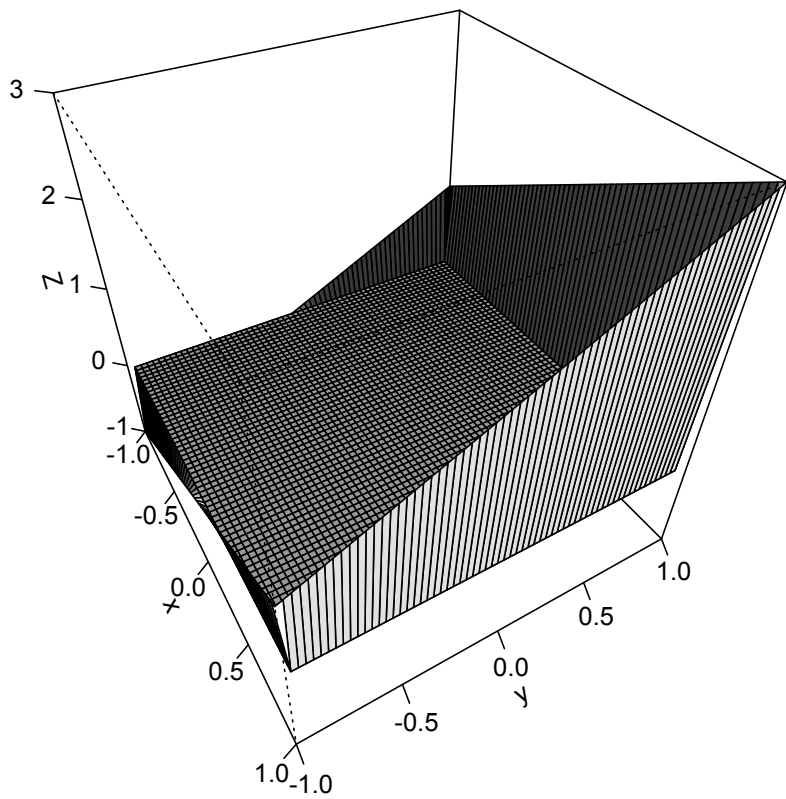
results when using bilinear interpolation, and the second using the operator interpolation.

Cycle	Grid Level	L^1 -Norm	L^2 -Norm	L^∞ -Norm	WU
-	6	7.96E+05	6.02E+04	1.77E+04	0.00E+00
1	6	8.23E+04	6.48E+03	1.23E+03	2.63E+00
2	6	5.49E+03	4.16E+02	7.69E+01	5.25E+00
3	6	4.01E+02	2.74E+01	4.84E+00	7.88E+00
4	6	3.14E+01	1.91E+00	3.31E-01	1.05E+01
5	6	2.50E+00	1.40E-01	2.44E-02	1.31E+01
6	6	2.03E-01	1.07E-02	2.21E-03	1.58E+01
7	6	1.67E-02	8.40E-04	1.95E-04	1.84E+01
8	6	1.41E-03	6.85E-05	1.70E-05	2.10E+01
9	6	1.22E-04	5.75E-06	1.47E-06	2.36E+01
10	6	1.08E-05	4.97E-07	1.28E-07	2.63E+01

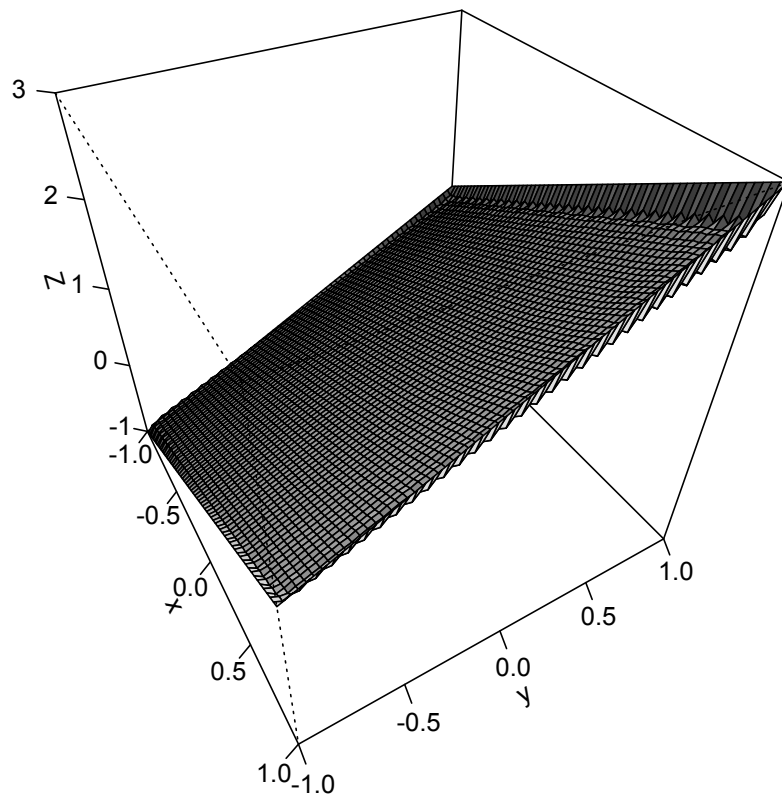
The next table gives the results of 10 iterations of the same problem with operator interpolation.

Cycle	Grid Level	L^1 -Norm	L^2 -Norm	L^∞ -Norm	WU
-	6	7.96E+05	6.02E+04	1.77E+04	0.00E+00
1	6	8.98E+04	6.90E+03	1.29E+03	2.63E+00
2	6	6.95E+03	4.87E+02	8.76E+01	5.25E+00
3	6	6.62E+02	3.69E+01	6.18E+00	7.88E+00
4	6	6.99E+01	3.14E+00	4.83E-01	1.05E+01
5	6	8.08E+00	3.06E-01	4.49E-02	1.31E+01
6	6	1.01E+00	3.46E-02	4.75E-03	1.58E+01
7	6	1.36E-01	4.47E-03	5.03E-04	1.84E+01
8	6	1.90E-02	6.30E-04	5.45E-05	2.10E+01
9	6	2.75E-03	9.28E-05	6.54E-06	2.36E+01
10	6	4.06E-04	1.40E-05	1.00E-06	2.63E+01

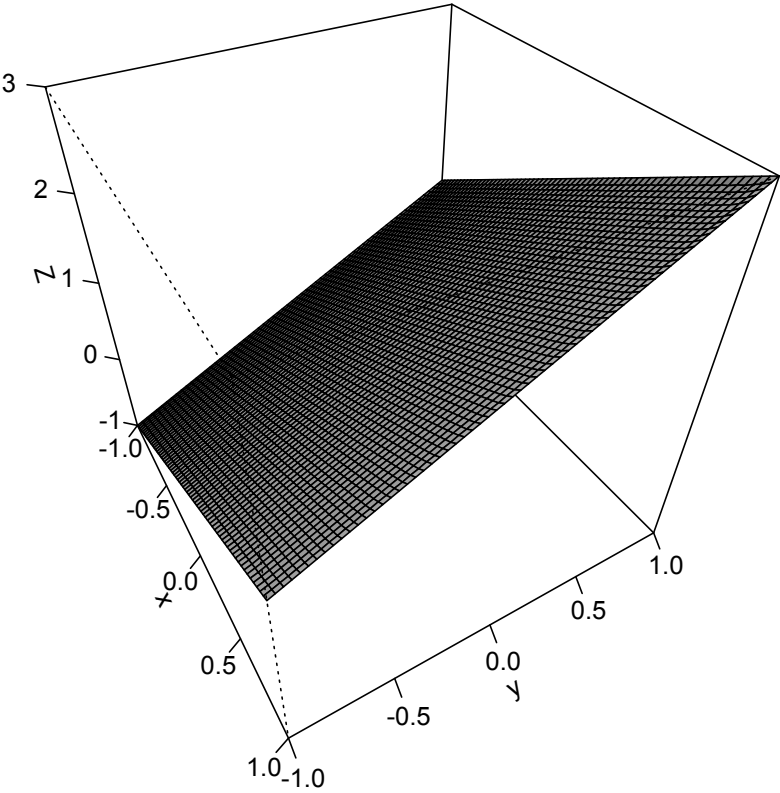
Fortran Results: Iteration 0



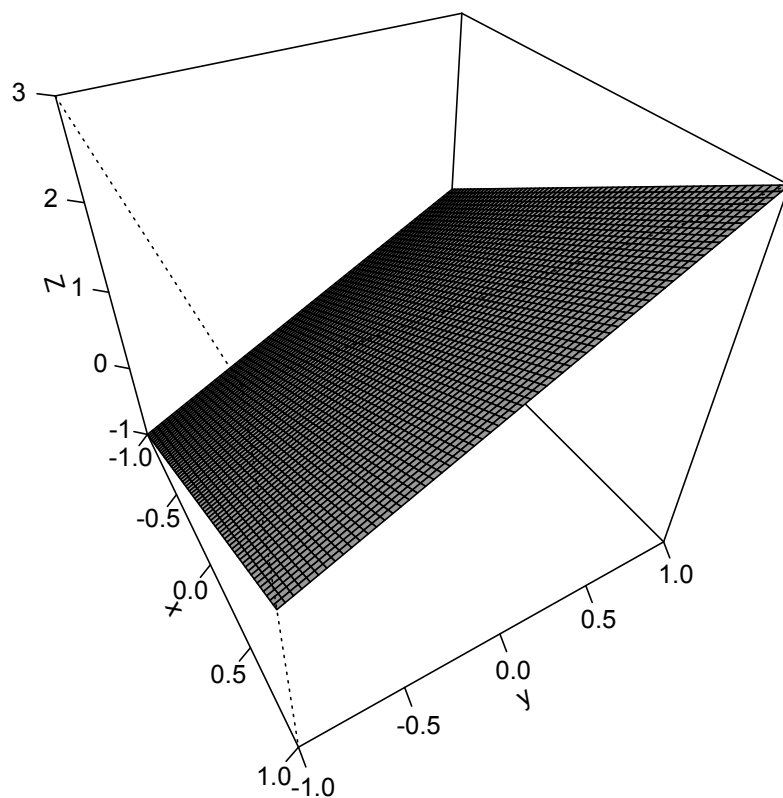
Fortran Results: Iteration 1



Fortran Results: Iteration 2



Fortran Results: Iteration 3



5.2 3D Results

5.2.1 V-Cycle

Below we give two tables with the results of the V-Cycle. The first table gives results for the dirichlet problem $-\Delta u = f$ on the unit cube $\Omega = (0, 1)^3$. We use Dirichlet Boundary conditions by setting $u = 0$ on the boundary $\partial\Omega$, and provide an initial guess of $u_h^0 = 0$ at every point in the discrete domain. We present the results of running twenty cycles over six grids. We also use Red-Black Gauss-Seidel with one pre-smoothing sweep and one

post-smoothing sweep.

Cycle	Grid Level	L^1 -Norm	L^2 -Norm	L^∞ -Norm	WU
-	6	9.78E+07	6.52E+05	1.23E+04	0.00E+00
1	6	2.51E+07	2.13E+05	2.24E+03	2.27E+00
2	6	5.30E+06	4.33E+04	4.42E+02	4.54E+00
3	6	1.16E+06	9.01E+03	9.49E+01	6.81E+00
4	6	2.62E+05	1.91E+03	2.06E+01	9.07E+00
5	6	5.92E+04	4.11E+02	4.51E+00	1.13E+01
6	6	1.35E+04	8.94E+01	9.97E-01	1.36E+01
7	6	3.07E+03	1.96E+01	2.22E-01	1.59E+01
8	6	7.03E+02	4.34E+00	4.97E-02	1.82E+01
9	6	1.61E+02	9.65E-01	1.12E-02	2.04E+01
10	6	3.68E+01	2.16E-01	2.53E-03	2.50E+01
11	6	8.44E+00	4.84E-02	5.74E-04	2.27E+01
12	6	1.93E+00	1.09E-02	1.31E-04	2.72E+01
13	6	4.43E-01	2.46E-03	2.98E-05	2.95E+01
14	6	1.02E-01	5.58E-04	6.82E-06	3.18E+01
15	6	2.33E-02	1.27E-04	1.56E-06	3.40E+01
16	6	5.33E-03	2.87E-05	3.59E-07	3.63E+01
17	6	1.22E-03	6.53E-06	8.24E-08	3.86E+01
18	6	2.79E-04	1.49E-06	1.89E-08	4.08E+01
19	6	6.40E-05	3.38E-07	4.36E-09	4.31E+01
20	6	1.47E-05	7.71E-08	1.00E-09	4.53E+01

The program took approximately 2.1 seconds to run.

5.2.2 W-Cycle

The following are the results of the W-Cycle given over 10 iterations.

Cycle	Grid Level	L^1 -Norm	L^2 -Norm	L^∞ -Norm	WU
-	6	9.78E+07	6.52E+05	1.23E+04	0.00E+00
1	6	1.09E+07	7.77E+04	7.16E+02	3.92E+00
2	6	8.92E+05	5.78E+03	5.18E+01	7.83E+00
3	6	7.45E+04	4.52E+02	3.94E+00	1.18E+01
4	6	6.30E+03	3.63E+01	3.08E-01	1.57E+01
5	6	5.36E+02	2.97E+00	2.45E-02	1.96E+01
6	6	4.57E+01	2.46E-01	1.97E-03	2.35E+01
7	6	3.92E+00	2.05E-02	1.62E-04	2.74E+01
8	6	3.36E-01	1.72E-03	1.33E-05	3.13E+01
9	6	2.88E-02	1.45E-04	1.11E-06	3.52E+01
10	6	2.48E-03	1.22E-05	9.22E-08	3.92E+01

With 10 W-Cycles, this program took approximately 2.1 seconds to run.

5.2.3 Results for the Debye-Hückel equation in 2D

Below we present the results of the multigrid solution to the debye huckel equation in two dimensions. Here, we let the coefficients be defined as

$$\varepsilon(x, y) = \varepsilon_I - (\varepsilon_I - \varepsilon_O) \frac{e^{(|r|-R)/S}}{1 + e^{(|r|-R)/S}}$$

and

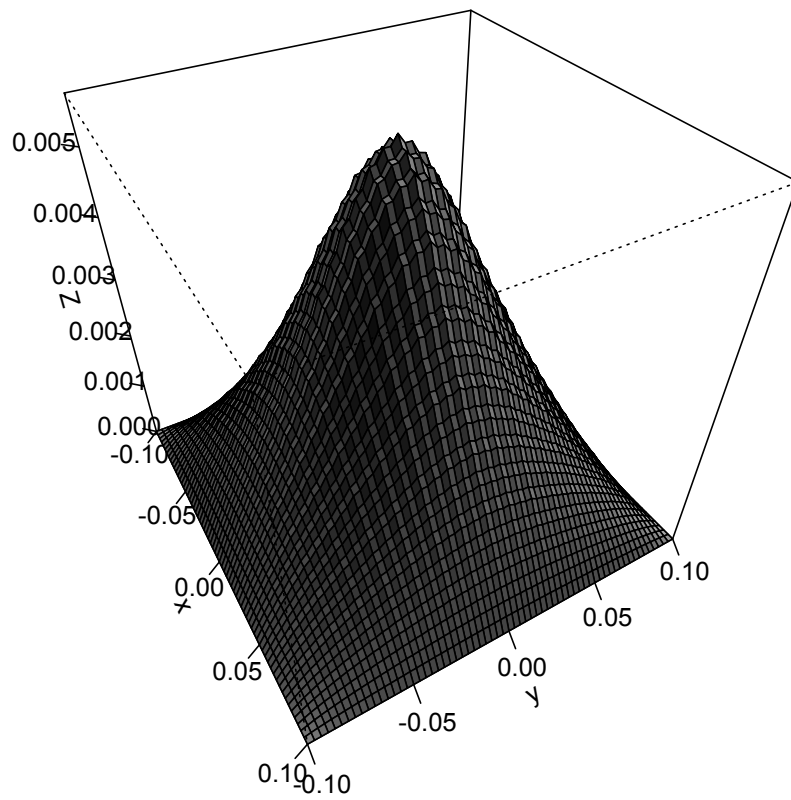
$$\kappa(x, y) = \kappa_I - (\kappa_I - \kappa_O) \frac{e^{(|r|-R)/S}}{1 + e^{(|r|-R)/S}}.$$

For the above functions, R is the radius of the dielectric sphere, S is characteristic length over which the coefficients change, and ε_I , ε_O , κ_I , and κ_O are the values of the dielectric coefficient and the Debye parameter in and outside of the sphere, respectively. Here, we take $R = 0.5$, $S = 0.2$, $\varepsilon_I = 8$, $\varepsilon_O = 80$, $\kappa_I = 0$, and $\kappa_O = 6$.

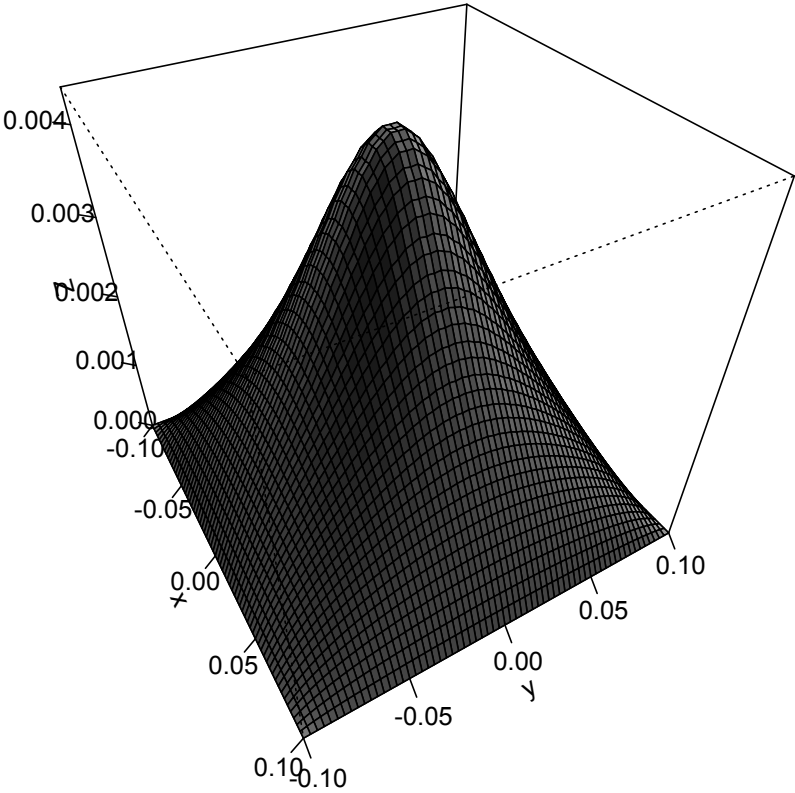
Cycle	Grid Level	L^1 -Norm	L^2 -Norm	L^∞ -Norm	WU
-	6	3.969E+03	6.30E+01	1.00E+00	0.00E+00
1	6	1.428E+04	3.50E+02	1.65E+01	2.63E+00
2	6	2.155E+03	5.78E+01	3.34E+00	5.25E+00
3	6	6.394E+02	1.60E+01	8.26E-01	7.88E+00
4	6	1.187E+02	3.10E+00	1.73E-01	1.05E+01
5	6	3.014E+01	7.63E-01	4.04E-02	1.31E+01
6	6	6.198E+00	1.60E-01	8.76E-03	1.58E+01
7	6	1.465E+00	3.73E-02	2.00E-03	1.84E+01
8	6	3.158E-01	8.12E-03	4.41E-04	2.10E+01
9	6	7.231E-02	1.85E-03	9.95E-05	2.36E+01
10	6	1.591E-02	4.08E-04	2.21E-05	2.63E+01
11	6	3.594E-03	9.19E-05	4.96E-06	2.89E+01
12	6	7.979E-04	2.05E-05	1.11E-06	3.15E+01
13	6	1.792E-04	4.59E-06	2.48E-07	3.42E+01
14	6	3.993E-05	1.02E-06	5.53E-08	3.68E+01
15	6	8.943E-06	2.29E-07	1.24E-08	3.94E+01
16	6	1.997E-06	5.12E-08	2.76E-09	4.20E+01
17	6	4.467E-07	1.14E-08	6.18E-10	4.47E+01
18	6	9.978E-08	2.56E-09	1.38E-10	4.73E+01
19	6	2.233E-08	5.71E-10	3.09E-11	4.99E+01
20	6	4.998E-09	1.28E-10	6.89E-12	5.25E+01

The program took approximately 1.3 seconds to run. The graphical results are given below for the first three iterations.

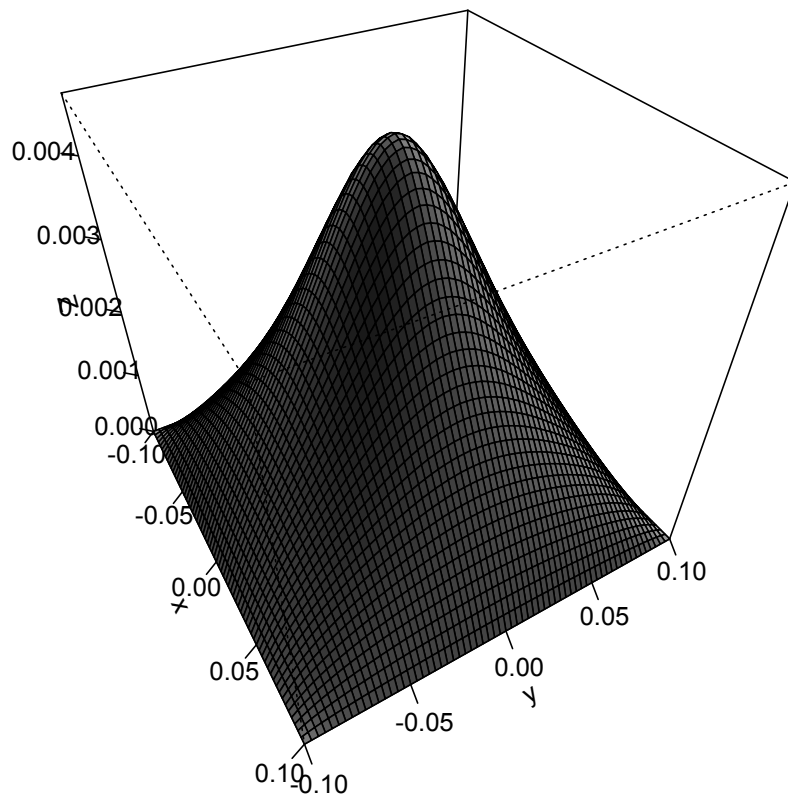
Fortran Results: Iteration 1



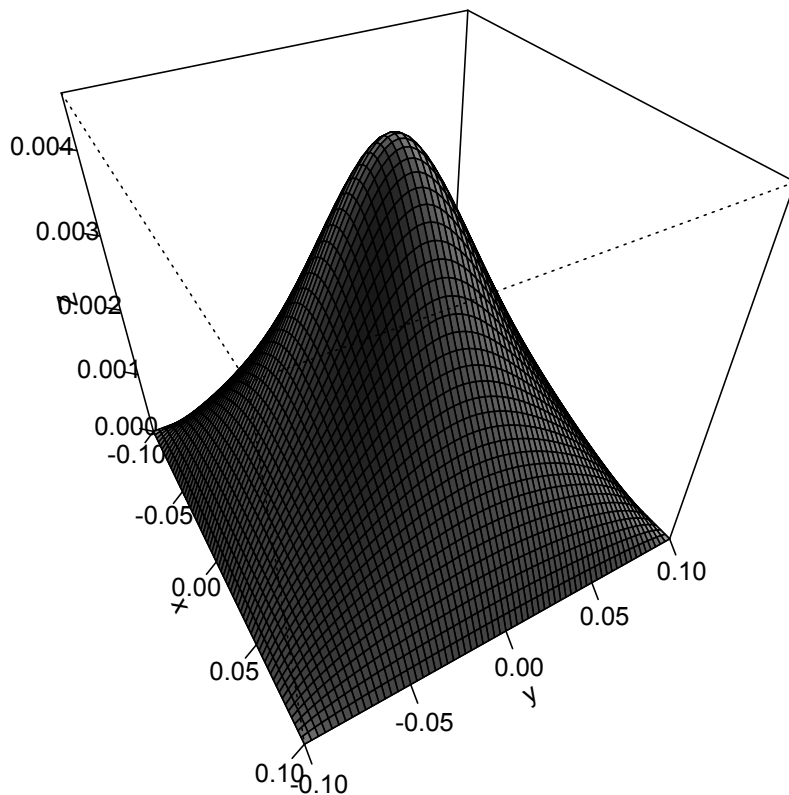
Fortran Results: Iteration 2



Fortran Results: Iteration 3



Fortran Results: Iteration 4



5.2.4 Results for the Debye-Hückel equation in 3D

In three dimensions, we set up the same problem with the coefficients defined analogously in three dimensions as in two dimensions. We take our computational domain to be the unit cube $(-1, 1)^3$, with

$$\epsilon - (\epsilon_I = \epsilon_O) \frac{e^{(r-R)/S}}{1 + e^{(r-R)/S}}$$

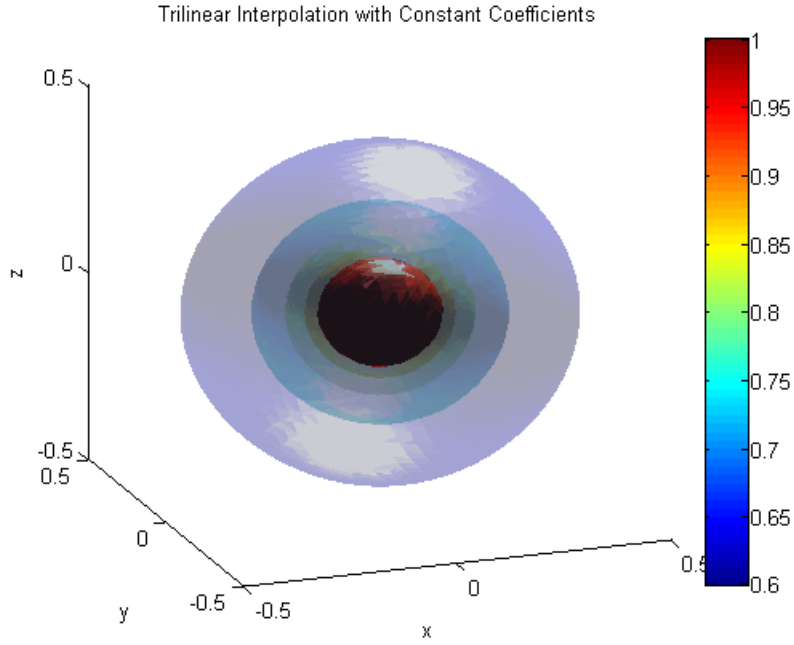
and

$$\kappa - (\kappa_I = \kappa_O) \frac{e^{(r-R)/S}}{1 + e^{(r-R)/S}}.$$

We take the sphere of the low-dielectric sphere to have a radius of $R = 0.5$, and define the characteristic length of change in the coefficients to be $S = 0.008\bar{3}$. The results for the three-dimensional problem using bilinear and operator prolongation are given below, as well as graphical solutions from the use of the operator interpolation.

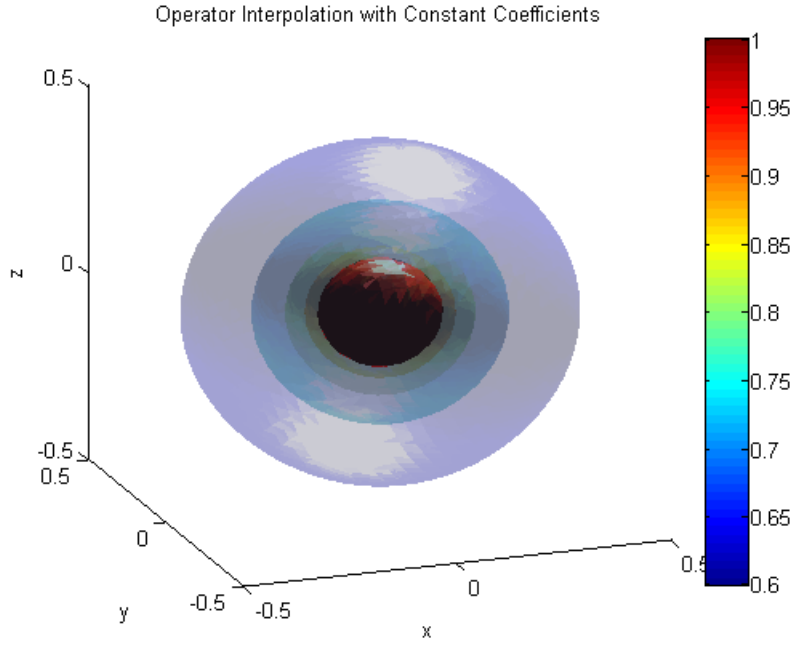
The first table gives the error using trilinear interpolation.

Cycle	Grid Level	L^1 -Norm	L^2 -Norm	L^∞ -Norm	WU
-	6	1.08E+07	7.94E+04	3.28E+04	0.00E+00
1	6	2.61E+06	1.66E+04	6.66E+02	2.27E+00
2	6	2.46E+05	1.13E+03	1.37E+01	4.54E+00
3	6	1.78E+04	6.87E+01	6.98E-01	6.81E+00
4	6	1.53E+03	5.72E+00	6.24E-02	9.07E+00
5	6	1.52E+02	5.10E-01	5.13E-03	1.13E+01
6	6	1.49E+01	4.87E-02	4.94E-04	1.36E+01
7	6	1.55E+00	4.83E-03	5.23E-05	1.59E+01
8	6	1.63E-01	4.95E-04	5.53E-06	1.82E+01
9	6	1.79E-02	5.25E-05	5.79E-07	2.04E+01
10	6	1.98E-03	5.71E-06	6.52E-08	2.27E+01



The second table gives the error using operator interpolation.

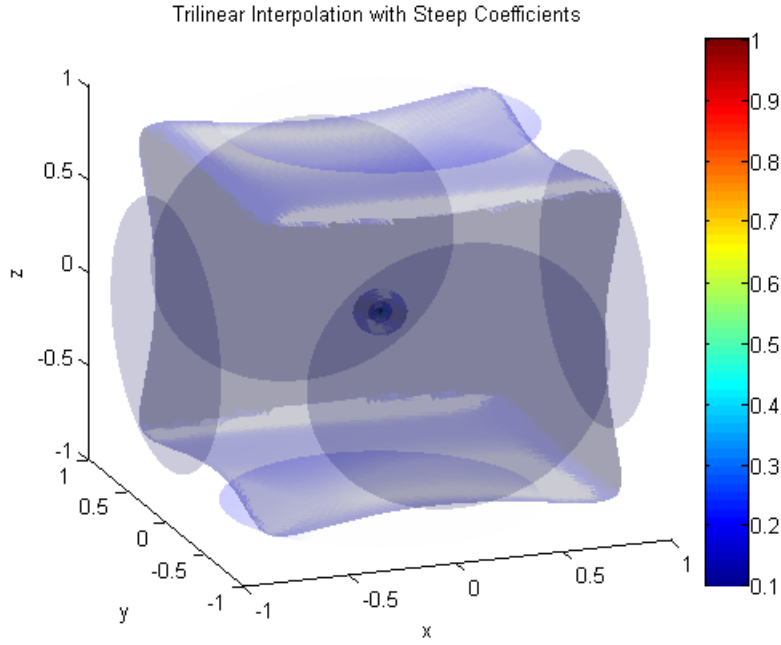
Cycle	Grid Level	L^1 -Norm	L^2 -Norm	L^∞ -Norm	WU
-	6	1.09E+07	7.94E+04	3.28E+04	0.00E+00
1	6	2.61E+06	1.66E+04	6.66E+02	2.27E+00
2	6	2.46E+05	1.13E+03	1.37E+01	4.54E+00
3	6	1.78E+04	6.87E+01	6.98E-01	6.81E+00
4	6	1.53E+03	5.72E+00	6.24E-02	9.07E+00
5	6	1.52E+02	5.10E-01	5.13E-03	1.13E+01
6	6	1.49E+01	4.87E-02	4.94E-04	1.36E+01
7	6	1.55E+00	4.83E-03	5.23E-05	1.59E+01
8	6	1.63E-01	4.95E-04	5.53E-06	1.82E+01
9	6	1.79E-02	5.25E-05	5.79E-07	2.04E+01
10	6	1.98E-03	5.71E-06	6.52E-08	2.27E+01



The second problem lets the coefficients vary in space.

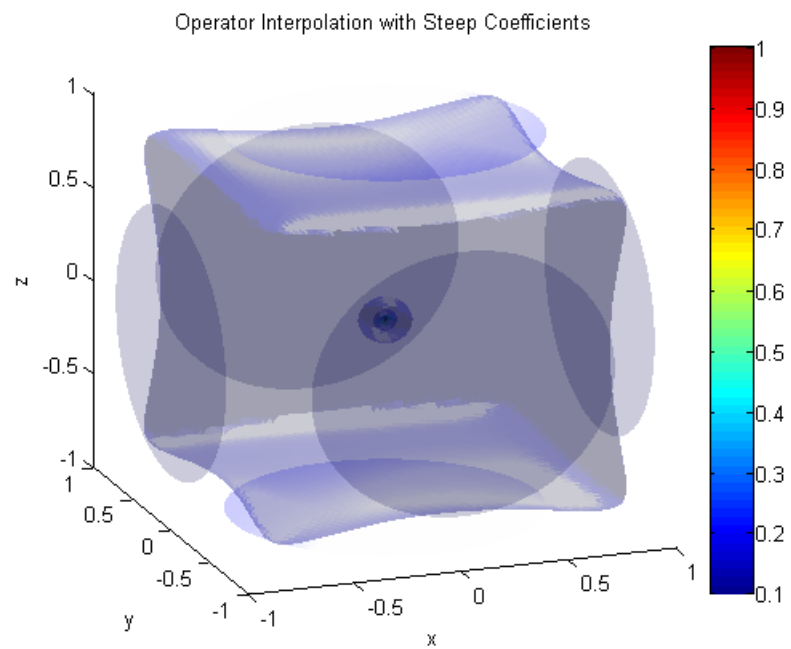
The first table gives the error using trilinear interpolation.

Cycle	Grid Level	L^1 -Norm	L^2 -Norm	L^∞ -Norm	WU
-	6	3.76E+08	2.54E+06	3.28E+04	0.00E+00
1	6	9.21E+07	5.57E+05	7.32E+03	2.27E+00
2	6	8.01E+06	3.43E+04	4.38E+02	4.54E+00
3	6	4.50E+05	1.53E+03	1.61E+01	6.81E+00
4	6	2.82E+04	9.95E+01	1.29E+00	9.07E+00
5	6	2.51E+03	9.52E+00	2.65E-01	1.13E+01
6	6	2.34E+02	1.12E+00	3.66E-02	1.36E+01
7	6	3.13E+01	2.02E-01	6.84E-03	1.59E+01
8	6	4.18E+00	2.77E-02	9.60E-04	1.82E+01
9	6	6.76E-01	5.10E-03	1.74E-04	2.04E+01
10	6	1.01E-01	7.12E-04	2.48E-05	2.27E+01



The second table gives the error using operator interpolation.

Cycle	Grid Level	L^1 -Norm	L^2 -Norm	L^∞ -Norm	WU
-	6	3.76E+08	2.54E+06	3.28E+04	0.00E+00
1	6	9.27E+07	5.76E+05	1.56E+04	2.27E+00
2	6	9.05E+06	4.07E+04	1.22E+03	4.54E+00
3	6	8.29E+05	3.07E+03	1.20E+02	6.81E+00
4	6	9.18E+04	3.16E+02	1.71E+01	9.07E+00
5	6	1.19E+04	3.72E+01	1.48E+00	1.13E+01
6	6	1.59E+03	4.73E+00	9.06E-02	1.36E+01
7	6	2.38E+02	7.10E-01	1.92E-02	1.59E+01
8	6	3.51E+01	1.06E-01	4.27E-03	1.82E+01
9	6	5.38E+00	1.68E-02	4.55E-04	2.04E+01
10	6	8.48E-01	2.74E-03	8.85E-05	2.27E+01



Chapter 6

Conclusion

This thesis covered the main components of multigrid methods, and devised a new interpolation method to handle coefficients with sharp changes in the gradient of the dielectric coefficient and the Debye parameter of the Debye-Hückel equation. The operator interpolation, in comparison to linear interpolation, performed just as well if not slightly better. We observed this behavior for problems in two and three dimensions. For simple model problems, we observed a reduction of the norms of the residuals to precision error within ten iterations. The results for the Debye-Hückel equation were also favorable, with the residual norms reduced by several orders of magnitude over the course of just ten iterations.

Although the interpolation operator devised in this thesis works with coefficients with steep gradients, the next extension of this interpolation method would be adapting it to discontinuous coefficients at an arbitrary interface in the domain. Work has been done to adapt to similar problems, but with assumptions about the location of the discontinuity [2].

Appendix A

Derivation of the Numerical Schemes

A.1 Finite Difference Scheme

The numerical scheme which we use for the solution of the problems in two and three dimensions is given by the $O(h^2)$ center difference approximations to Δu , ∇u , and $\nabla \varepsilon$. Thus, the numerical scheme used in the residual computation and relaxation methods all come from the following approximation to the original differential equation:

$$\begin{aligned} & -\varepsilon_{ijk} \frac{u_{i+1,jk} - 2u_{ijk} + u_{i-1,jk}}{h^2} - \\ & \varepsilon_{ijk} \frac{u_{i,j+1,k} - 2u_{ijk} + u_{i,j-1,k}}{h^2} - \\ & \varepsilon_{ijk} \frac{u_{ij,k+1} - 2u_{ijk} + u_{ij,k-1}}{h^2} - \\ & \left(\frac{\varepsilon_{i+1,jk} - \varepsilon_{i-1,jk}}{2h} \right) \left(\frac{u_{i+1,jk} - u_{i-1,jk}}{2h} \right) - \\ & \left(\frac{\varepsilon_{i,j+1,k} - \varepsilon_{i,j-1,k}}{2h} \right) \left(\frac{u_{i,j+1,k} - u_{i,j-1,k}}{2h} \right) - \\ & \left(\frac{\varepsilon_{ij,k+1} - \varepsilon_{ij,k-1}}{2h} \right) \left(\frac{u_{ij,k+1} - u_{ij,k-1}}{2h} \right) + \kappa_{ijk}^2 u_{ijk} = f_{ijk}. \end{aligned}$$

From here, we can derive the next approximation to the solution u at the point (x_i, y_j, z_k) . Let h_x , h_y , and h_z be the step sizes in the x -, y -, and z -directions. Also, let $\alpha = (\frac{h_x}{h_y})^2$ and $\beta = (\frac{h_x}{h_z})^2$. Then collecting all u_{ijk} terms on one side yields

$$2\varepsilon_{ijk}(1 + \alpha + \beta + (h_x\kappa_{ijk})^2)u_{ijk} = h_x^2 f_{ijk}$$

$$\begin{aligned} & \varepsilon_{ijk}(u_{i+1,jk} + u_{i-1,jk} + \alpha(u_{i,j+1,k} + u_{i,j-1,k}) + \beta(u_{ij,k+1} + u_{ij,k-1})) + \\ & \frac{1}{4}(\varepsilon_{i+1,jk} - \varepsilon_{i-1,jk})(u_{i+1,jk} - u_{i-1,jk}) + \frac{\alpha}{4}(\varepsilon_{i,j+1,k} - \varepsilon_{i,j-1,k})(u_{i,j+1,k} - u_{i,j-1,k}) + \\ & \frac{\beta}{4}(\varepsilon_{ij,k+1} - \varepsilon_{ij,k-1})(u_{ij,k+1} - u_{ij,k-1}). \end{aligned}$$

Note that when $\varepsilon \equiv 1$ for every point in the domain on which we define the problem, we obtain the usual Gauss-Seidel scheme for the problem $-\Delta u = f$.

A.2 Conservative Discretization of the Debye-Hückel Equation

Another discretization one can use with the Debye-Hückel equation

$$-\nabla \cdot (\epsilon \nabla u) + \kappa^2 u = f$$

is given below, and requires no differentiation of the coefficient ϵ . We assume a uniform grid step size h in the x -, y -, and z -directions.

$$-\frac{\epsilon_{i+1/2,jk}(u_{i+1/2,jk})_x - \epsilon_{i-1/2,jk}(u_{i-1/2,jk})_x}{h} -$$

$$\begin{aligned}
& \frac{\epsilon_{i,j+1/2,k}(u_{i,j+1/2,k})_y - \epsilon_{i,j-1/2,k}(u_{i,j-1/2,k})_y}{h} - \\
& \frac{\epsilon_{ij,k+1/2}(u_{ij,k+1/2})_z - \epsilon_{ij,k-1/2}(u_{ij,k-1/2})_z}{h} + \\
& k_{ijk}^2 u_{ijk} = f_{ijk},
\end{aligned}$$

which simplifies to

$$\begin{aligned}
& - \frac{\epsilon_{i+1/2,jk}(u_{i+1,jk} - u_{ijk}) - \epsilon_{i-1/2,jk}(u_{ijk} - u_{i-1,jk})}{h^2} - \\
& \frac{\epsilon_{i,j+1/2,k}(u_{i,j+1,k} - u_{ijk}) - \epsilon_{i,j-1/2,k}(u_{ijk} - u_{i,j-1,k})}{h^2} - \\
& \frac{\epsilon_{ij,k+1/2}(u_{ij,k+1} - u_{ijk}) - \epsilon_{ij,k-1/2}(u_{ijk} - u_{ij,k-1})}{h} + \\
& k_{ijk}^2 u_{ijk} = f_{ijk}
\end{aligned}$$

or

$$\begin{aligned}
& \frac{-\epsilon_{i+1/2,jk}u_{i+1,jk} + (\epsilon_{i+1/2,jk} + \epsilon_{i-1/2,jk})u_{ijk} - \epsilon_{i-1/2,jk}u_{i-1,jk}}{h^2} + \\
& \frac{-\epsilon_{i,j+1/2,k}u_{i,j+1,k} + (\epsilon_{i,j+1/2,k} + \epsilon_{i,j-1/2,k})u_{ijk} - \epsilon_{i,j-1/2,k}u_{i,j-1,k}}{h^2} + \\
& \frac{-\epsilon_{ij,k+1/2}u_{ij,k+1} + (\epsilon_{ij,k+1/2} + \epsilon_{ij,k-1/2})u_{ijk} - \epsilon_{ij,k-1/2}u_{ij,k-1}}{h^2} + \\
& k_{ijk}^2 u_{ijk} = f_{ijk}.
\end{aligned}$$

We make use of the above discretization throughout this paper. Using this scheme in two dimensions just requires the omission of the term corresponding to

$$\frac{\partial}{\partial z} \left(\epsilon \frac{\partial u}{\partial z} \right)$$

and the use of only two indices i and j .

Bibliography

- [1] M.A. Ahmed. A characterization of the convergence of picard iteration to a fixed point for a continuous mapping and an application. *Appl. Math. Comput.*, 169:1298–1304, 2005.
- [2] R.E. Alcouffe, Achi Brandt, J.E. Dendy, and J.W. Painter. The multi-grid method for the diffusion equation with strongly discontinuous coefficients. *SIAM J. Sci. Stat. Comp.*, 2(4):430–454, 1981.
- [3] O. Axelsson and I. Gustafsson. Preconditioning and two-level multigrid methods of arbitrary degree of approximation. *Mathematics of Computation*, 40(161):219–242, 1983.
- [4] Tim Boonen, Jan Van lent, , and Stefan Vandewalle. An algebraic multigrid method for high order time-discretizations of the div-grad and the curl-curl equations. *Applied Numerical Mathematics*, 59(3-4):507–521, 2009.
- [5] D. Braess and W. Hackbusch. A new convergence proof for the multigrid method including the v-cycle. *SIAM*, 20(5):967–975, 1983.
- [6] James H. Bramble, Do Y. Kwak, and Joseph E. Pasciak. Uniform convergence of multi-grid v-cycle iterations for indefinite and nonsymmetric problems. *SIAM*, 31(6):1746–1763, 1994.
- [7] Henrik Branden and Sverker Holmgren. Convergence acceleration for the steady-state euler equations. *Computers & Fluids*, 32(8):1075–1092, 2003.
- [8] Achi Brandt. Multi-level adaptive solutions to boundary-value problems. *Mathematics of Computation*, 31(138):333–390, 1977.
- [9] R. Bulirsch and J. Stoer. *Introduction to Numerical Analysis*. Texts in Applied Mathematics 12. Springer-Verlag, New York, 3rd edition edition, 2002.
- [10] Tony F. Chan and W.L. Wan. Robust multigrid methods for nonsmooth coefficient elliptic linear systems. *Numerical Analysis*, 123(1-2):323–352, 2000.
- [11] Armando Coco. *Finite-Difference Ghost-Cell Multigrid Methods for Elliptic Problems with Mixed Boundary Conditions and Discontinuous Coefficients*. Thesis, Università degli Studi di Catania, 2011.

- [12] Erik Dick. Multigrid acceleration of a flux-difference splitting method for steady euler equations. *Journal of Computational and Applied Mathematics*, 28:173–180, 1989.
- [13] Dawn Hollenbeck et al. Model for evaluating patterned charge-regulation contributions to electrostatic interactions between low-dielectric spheres. *Physical Review E*, 82(3), 2010.
- [14] Murli M. Gupta and Jun Zhang. High accuracy multigrid solution of the 3d convection-diffusion equation. *Applied Mathematics and Computation*, 113(2-3):249–274, 2000.
- [15] Bertil Gustafsson and Per Ltstedt. Fourier analysis of multigrid methods for general systems of pdes. *Mathematics of Computation*, 60(202):473–493, 1993.
- [16] Ale Janka Herv Guillard and Petr Vank. Analysis of an algebraic petrovgalerkin smoothed aggregation multigrid method. *Applied Numerical Mathematics*, 58(12):1861–1874, 2008.
- [17] Michael Holst and Stefan Vandewalle. Schwarz methods: To symmetrize or not to symmetrize. *SIAM Journal on Numerical Analysis*, 34(2):699–722, 1997.
- [18] Wolfgang Joppich and Slobodan Mijalkovic. *Multigrid Methods for Process Simulation*. Springer-Verlag, Wien, 1993.
- [19] K.S. Kang and D.Y. Kwak. Convergence estimates for multigrid algorithms. *Computers & Mathematics with Applications*, 34(9):15–22, 1997.
- [20] János Karátson and István Faragó. Variable preconditioning via quasi-newton methods for nonlinear problems in hilbert space. *SIAM Journal on Numerical Analysis*, 41(4):1242–1262, 2004.
- [21] R. Kettler and P. Wesseling. Aspects of multigrid methods for problems in three dimensions. *Applied Mathematics and Computation*, 19(1-4):159–168, 1986.
- [22] John D. Kirkwood. Theory of solutions of molecules containing widely separated charges with special applications to zwitterions. *The Journal of Chemical Physics*, 2(7):351–361, 1934.
- [23] R. Kulke, Th. Sporkmann, and I. Wolff. Multigrid technique with local grid refinement for solving static field problems [rf circuits]. In *Microwave Symposium Digest, 1998 IEEE MTT-S International*, volume 1, pages 29–32, June 1998.
- [24] N.K. Lambropoulos, D.G. Koubogiannis, , and K.C. Giannakoglou. Acceleration of a navierstokes equation solver for unstructured grids using agglomeration multigrid and parallel processing. *Computer Methods in Applied Mechanics and Engineering*, 193(9-11):781–803, 2004.
- [25] Qunfeng Liu and Jinping Zeng. Convergence analysis of multigrid methods with residual scaling techniques. *Journal of Computational and Applied Mathematics*, 234(10):29322942, 2008.

- [26] Per Ltstedt. Grid independent convergence of the multigrid method for first-order equations. *SIAM*, 29(5):1370–1394, 1992.
- [27] Andrew Lumsdaine and Deyun Wu. Krylov subspace acceleration of waveform relaxation. *SIAM Journal on Numerical Analysis*, 41(1):90–111, 2004.
- [28] Jan Mandel. Multigrid convergence for nonsymmetric, indefinite variational problems and one smoothing step. *Applied Mathematics and Computation*, 19(1-4):201–216, 1986.
- [29] Donald H. Menzel. *Mathematical Physics*. Dover Publications, Inc., New York, 1961.
- [30] Maximilian S. Mesquita and Marcelo J.S. de Lemos. Optimal multigrid solutions of two-dimensional convectionconduction problems. *Applied Mathematics and Computation*, 152(3):725–742, 2004.
- [31] Rafael Muñoz. Theoretical analysis of some spectral multigrid methods. *Computer Methods in Applied Mechanics and Engineering*, 80(1-3):287–294, 1990.
- [32] Naomi H. Naik and John Van Rosendale. The improved robustness of multigrid elliptic solvers based on multiple semicoarsened grids. *SIAM Journal on Numerical Analysis*, 30(1):215–229, 1993.
- [33] So Noguchi, Vasil Spasov, and Hideo Yamashita. Fast 3-d edge element analysis by the geometric multigrid method using an accelerated symmetric gauss-seidel smoother. *IEEE Transactions on Magnetics*, 39(3):1685–1688, 2003.
- [34] Yvan Notay. Convergence analysis of perturbed two-grid and multigrid methods. *SIAM J. Numer. Analysis*, 45(3):1035–1044, 2007.
- [35] F. Oliveira, M.A.V. Pinto, C.H. Marchi, and L.K. Araki. Optimized partial semicoarsening multigrid algorithm for heat diffusion problems and anisotropic grids. *Applied Mathematical Modelling*, 36(10):4665–4676, 2012.
- [36] Cornelius Oosterlee, Anton Schüller, and Ulrich Trottenberg. *Multigrid*. Academic Press, London, 2001.
- [37] James S. Otto. Multigrid convergence for discretizations of singular perturbation problems with grid-aligned flow. *SIAM Journal on Numerical Analysis*, 33(1):399–416, 1996.
- [38] Christoph Pflaum. A multigrid conjugate gradient method. *Applied Numerical Mathematics*, 58(12):1803–1817, 2008.
- [39] J.A. Rabi and M.J.S. de Lemos. Optimization of convergence acceleration in multigrid numerical solutions of conductiveconvective problems. *Applied Mathematics and Computation*, 124(2):215–226, 2001.
- [40] Maxwell Rosenlicht. *Introduction to Analysis*. Dover Publications, Inc., New York, 1968.

- [41] L. Ridgway Scott and Shangyou Zhang. Higher-dimensional nonnested multigrid methods. *Mathematics of Computation*, 58(198):457–466, 1992.
- [42] Yair Shapira. Multigrid methods for 3-d definite and indefinite problems. *Applied Numerical Mathematics*, 26(3):377–398, 1998.
- [43] Ali Tavakoli and Asghar Kerayechian. Improvement of the rate of convergence estimates for multigrid algorithm. *Applied Mathematics and Computation*, 187(2):922–928, 2007.
- [44] Junping Wang. Convergence analysis without regularity assumptions for multigrid algorithms based on sor smoothing. *SIAM*, 29(4):987–1001, 1992.
- [45] P.M. De Zeeuw. Matrix-dependent prolongations and restrictions in a black box multigrid solver. *Journal of Computational and Applied Mathematics*, 33(1):1–27, 1990.
- [46] Jun Zhang. Acceleration of five-point red-black gauss-seidel in multigrid for poisson equation. *Applied Mathematics and Computation*, 80(1):73–93, 1996.
- [47] Jun Zhang. A cost-effective multigrid projection operator. *Journal of Computational and Applied Mathematics*, 76(1-2):325–333, 1996.
- [48] Jun Zhang. Residual scaling techniques in multigrid, ii: Practical applications. *Applied Mathematics and Computation*, 90(2-3):229–252, 1998.
- [49] Jun Zhang. Two-grid analysis of minimal residual smoothing as a multigrid acceleration technique. *Applied Mathematics and Computation*, 96(1):19, 1998.
- [50] Jun Zhang. Acceleration and stabilization properties of minimal residual smoothing technique in multigrid. *Applied Mathematics and Computation*, 100(2-3):151–168, 1999.
- [51] Z. Zhu and C.A.J. Fletcher. A study of sequential solutions for the reduced/complete navier-stokes equations with multigrid acceleration. *Computers & Fluids*, 19(1):43–60, 1991.