University of California
Los Angeles

Wavelet Operators Applied to Multigrid Methods

A dissertation submitted in partial satisfaction of the
requirements for the degree Doctor of Philosophy
in Mathematics

by

Doreen Rona Naomi De Leon

2000

The dissertation of Doreen Rona Naomi De Leon is approved.

James McWilliams

Chris Anderson

Tony Chan

Bjorn Engquist, Committee Chair

University of California, Los Angeles

2000

To my grandfather who was a source of inspiration and knowledge. Also to the rest of my family for their unwavering support.

Table of Contents

List of Figures

List of Tables

Acknowledgments

The algebraic multigrid method used for comparison in Chapter 7 is AMG1R5, written by John Ruge, Klaus Stüben, and Rolf Hempel. This program, in an earlier form, is discussed in [RS87]. I also want to give my thanks to John Ruge, who answered my questions regarding applications of his program. His help is very much appreciated.

I wish to thank my advisor, Bjorn Engquist, whose patience and advice have been so helpful to me in completing my graduate career. His guidance and support have been invaluable. Also, thanks to the other members of my dissertation committee, Tony Chan, Chris Anderson, and James McWilliams. I extend my appreciation as well to the faculty, staff, and students of the math department. For their advice and assistance with some aspects of numerical implementation, particularly involving matrices and vectors, I give an extra thanks to Randolph Cooper and Li-Tien Cheng. I also wish to thank John Westman for his advice and encouragement in my academic endeavors.

VITA

1969	Born, Los Angeles, California, USA.
1991	B.S., Electrical Engineering, UCLA, Los Angeles, California.
1995 – 2000	Teaching Assistant and Research Assistant, UCLA, Los Angeles, California.
1996	M. A., Mathematics, UCLA, Los Angeles, California.

Abstract of the Dissertation

Wavelet Operators Applied to Multigrid Methods

by

Doreen Rona Naomi De Leon
Doctor of Philosophy in Mathematics
University of California, Los Angeles, 2000
Professor Bjorn Engquist, Chair

The multigrid method is very useful in reducing the convergence time for solving systems of algebraic equations obtained from approximating partial differential equations. However, when confronted by certain problems, for example elliptic problems with discontinuous or highly oscillatory coefficients or the advection-dominated advection-diffusion equation, the standard multigrid procedure is less efficient or may break down. Here, we take a new approach to improve the efficiency of these methods for such problems.

The approach taken here is to use wavelets to obtain new coarse grid, interpolation, and restriction operators for the multigrid method. We do this by applying the wavelet transform, formed from the scaling and wavelet functions, to the operators in the original equation. The transform projects onto the coarser scaling and wavelet spaces. Then, by decomposing the resulting operator, we obtain a solution expressed in matrix form, from which we obtain the interpolation, restriction, and coarse grid operators. The coarse grid operator thus formed is a good approximation of the operator formed from homogenization, a technique used to solve elliptic problems with periodic oscillatory coefficients. To make this

method computationally more efficient, an inverse component used in the procedure above is approximated, first by using ILU(0) to obtain the LU-decomposition and then by truncating to allow no fill-in in the inverse.

This method is applied to a variety of problems of interest. Among them are elliptic equations whose coefficients are highly oscillatory or discontinuous. This method proves effective in returning the desired performance of multigrid, i.e., convergence independent of mesh size. Also, we apply this method to the advection-diffusion equation with dominant advection term. Application of the wavelet multigrid method, here using symmetric Gauss-Seidel for the smoother, again yields convergence independent of mesh size. Another problem we look at is an anisotropic second order partial differential equation, for which convergence is slow. A last area investigated is the efficiency of the method with respect to solving the Stokes and incompressible Navier-Stokes equations. For the Stokes problem, we find that although the original method performs well, the method using ILU(0) and truncation fails. For the incompressible Navier-Stokes equations, the original method performs well for zero boundary conditions.

CHAPTER 1

Introduction

The multigrid method is very useful in reducing the convergence time for solving systems of algebraic equations approximating partial differential equations. However, when confronted by certain problems, for example problems with discontinuous or highly oscillatory coefficients as well as advection-dominated problems, the standard multigrid procedure is less efficient or may break down. One problem is that smaller eigenvalues do not necessarily correspond to smooth eigenfunctions. As a result, the coarse grid problem does not approximate the original problem well.

One method to correct for this for elliptic problems with periodic coefficients is by solving a homogenized version of the problem on coarser grids and using the standard interpolation and restriction operators [EL97]. Another approach involves finding local homogenized coarse grid operators and using harmonic interpolation I_{2h}^{h}, with restriction operator $I_{h}^{2h} = (I_{2h}^{h})^{T}$ [EL93, EL96]. This approach is taken because the homogenized operator provides a better approximation of the properties (eigenvalues, eigenfunctions) of the original fine grid operator. Neuss, Jäger, and Wittum [NJW98] also use homogenization in their approach, which uses the finite element method. Here, prolongation is given by the action of the interpolation operator arising from evaluation at the nodes of the fine triangulation on a sum involving the solution to the homogenized problem and the solution to the cell problem. A problem with these homogenization techniques is that they are only applicable to periodic problems and can only be given closed form solutions in certain cases. One alternate approach has been suggested in [ABD81], which employs a modification of the interpolation operators and uses a Galerkin-type formulation for the coarse grid operator. The problem examined here is

$$\begin{aligned} -\nabla \cdot (D(x,y)\nabla U(x,y)) + \sigma(x,y)U(x,y) &= f(x,y), (x,y) \in \Omega, \\ \nu(x,y) \cdot D(x,y)\nabla U(x,y) + \gamma(x,y)U(x,y) &= 0, (x,y) \in \partial\Omega. \end{aligned}$$

More recently, the standard elliptic problem

$$\begin{aligned} -\alpha \triangle u + u &= f, \text{ in } \Omega, \\ u &= g, \text{ on } \partial\Omega, \end{aligned}$$

where $\alpha = const > 0$, has been analyzed using wavelets to modify the multigrid methods. For example, wavelets have been used to modify the hierarchical basis method in [VW95a, VW95b]. The wavelet basis functions are used for the basis functions in the modified hierarchical basis method. This allows the subspaces thus constructed to be orthogonal. Use of a wavelet-based multigrid preconditioner for the conjugate gradient method to solve the system of equations resulting from a wavelet-Galerkin discretization of the boundary value problem with $g = 0$ has been used with good results [GRW96]. Here, the basis functions are the scaling functions (or, low-pass filter functions). Also, wavelets have been used to develop multiple scale solutions to more complex elliptic problems [LC95]. Andreas Rieder [Rie94, RWZ94] takes this one step further by using wavelet decompositions to obtain a multilevel method. His approach uses a choice of the filter operators obtained from wavelets for the restriction and interpolation operators.

With respect to the solution of the advection-diffusion equation with dominant advection term

$$-\nabla \cdot (a(x)\nabla u) + b(x)\nabla u + cu = f(x),\ \ x \in \Omega,$$

where $\|b\| \gg \|a\|$, we see various approaches in the literature. Many multigrid solutions to this problem taken by other researchers involve numbering the unknowns in a certain order. For example, in [BW95], the nodes are numbered so that the unknowns are solved in the direction of flow. Domain decomposition methods, which resemble multigrid methods, are employed in [KL95]. Here, upwinding or stabilized Galerkin approaches are recommended. Arnold Reuskin uses an approximate LU-factorization of the matrix operator (after discretization of the problem) to determine the interpolation, restriction, and coarse grid operators for the multigrid method [Reu95]. His method uses the information from the underlying differential equation in order to get the factorization. P. M. de Zeeuw uses matrix-dependent prolongations (I_{2h}^{h}) and restrictions (I_{h}^{2h}), the Galerkin formulation of the coarse grid operator (given fine grid operator A^h, define the coarse grid operator A^{2h} by $A^{2h} = I_h^{2h} A^h I_{2h}^h$), and as a choice of smoother either the incomplete line LU (ILLU) or incomplete block LU methods [Zee93a, Zee93b]. Another article that employs matrix-dependent interpolations is [GK95]. Here, the coarse grid operator is determined, using the Galerkin approach, to be a Schur complement. Another recent approach involves using upstream discretization combined with downstream relaxation with residual weighting [YVB98].

Other researchers deal more generally with non-elliptic problems or problems with singular perturbations. Irad Yavneh, in [Yav98], discusses various ways of computing the coarse grid, interpolation, and restriction operators. He comes up with requirements for the coarse grid operator (it must approximate not only the first term in a Taylor expansion of the differential operator, but also the second term, i.e., the truncation error), as well as for the interpolation and restriction operators, that should be met for the multigrid scheme to perform well. Boris Diskin [Dis99] uses semi-coarsening and differential approximations applied to the full multigrid method. Jinchao Xu [Xu92b] looks at non-selfadjoint problems or indefinite problems as perturbations of elliptic problems. Here, the original equation is solved exactly on the coarsest grid, and the symmetric part of the operator becomes the operator for the solution on all other grids. Thomas Schlinkmann [Sch98] uses an adaptive full multigrid method, using the wavelet coefficients to determine where grid refinement should occur.

Another area of great interest is the solution of the incompressible Navier-Stokes equations

$$\begin{aligned} -\nabla u + Re(uu_x + vu_y) + p_x &= f^u \text{ in } \Omega \\ -\nabla v + Re(uv_x + vv_y) + p_y &= f^v \text{ in } \Omega \\ u_x + v_y &= 0 \text{ on } \overline{\Omega}. \end{aligned}$$

Much of the research that has been devoted to the numerical solution of said equations involves the use of the FAS (full approximation scheme) in order to solve using multigrid. In the majority of the works, a finite element or finite volume based approach is taken for the discretization of the Navier-Stokes equations, using either staggered grids or non-staggered grids, and, in some cases, different choices of basis functions for the velocity and pressure terms. For example, in [BFP89], a finite volume discretization is used, along with a FAS multigrid scheme, to solve numerically the incompressible Navier-Stokes equations written in conservative form. In [JV95], the Navier-Stokes equations are discretized on triangular grids and the control volume procedure is used (which involves integrating over a control volume to obtain the equations to be discretized). Here, FAS is also used to obtain the coarse grid equations. The solution is obtained by solving the pressure equations (determined by satisfying the continuity equation), then the momentum equations. Under-relaxation is used to prevent instability. R. Webster takes an algebraic multigrid approach to the solution of the incompressible Navier-Stokes equations in [Web94, Web96]. Here, equations are discretized using unstructured finite element meshes, employing different interpolations for pressure and velocity, but elements of equal order. The discrete equations are obtained via enforcing the conservation laws on the control volumes, and under-relaxation is used in the iterative scheme. The later work uses the defect-correction method to solve to second order accuracy, thus eliminating the need for under-relaxation. In [Sch90], Anton Schüller takes a finite difference approach to the solution of the Navier-Stokes equations. This involves a restatement of the problem, which makes the finite difference discretization more efficient and eliminates the need for addition of an artificial stabilizing term. The standard five-point discretization for second order terms and centered differencing for the first order terms are then used.

In [EL], the wavelet transform $\mathcal{W}_j$ which involves both high- and low-pass filter operators (the scaling and wavelet operators, H_j and G_j), is used to derive a new approach, assuming that the matrix on the fine grid is symmetric. Also, some one-dimensional examples are examined. The goal of this work is to examine the results of the above approach in two dimensions, taking into account the implications on the scaling and wavelet operators. The reason we consider this approach is that we see from, say, [DE98] and [AEL97] that the wavelet coarse grid operator provides a good approximation to the homogenized coarse grid operator, but with a reduced number of computations. Also, wavelets can be applied to problems with periodic as well as non-periodic coefficients. Finally, application of wavelet operators to vectors and matrices maintains the properties of the original problem.

The initial procedure followed is much the same as that in [EL], although care is taken to remove the restriction of a symmetric fine grid operator. As a result of removing this restriction, we have been able to apply the wavelet multigrid method to more, and different types of, problems than were examined in [EL]. As a first step, the wavelet transform is applied to both sides of the matrix equation. Given the problem

$$L_j U = F$$

on the fine grid, application of the wavelet transform to both sides of the equation yields

$$(\mathcal{W}_j L_j \mathcal{W}_j^T)\mathcal{W}_j U = \mathcal{W}_j F. \tag{1.1}$$

The above also uses the fact that the wavelet transform is orthogonal. Now, $\mathcal{W}_j$ is defined by

$$\mathcal{W}_j = \begin{pmatrix} H_j \\ G_j \end{pmatrix}.$$

Thus, we can rewrite (1.1) as

$$(\mathcal{W}_j L_j \mathcal{W}_j^T)\begin{pmatrix} U_L \\ U_H \end{pmatrix} = \begin{pmatrix} F_L \\ F_H \end{pmatrix}, \tag{1.2}$$

where L denotes the low-frequency part of the vector and H denotes the high-frequency part. Denote $\tilde{L}_j = \mathcal{W}_j L_j \mathcal{W}_j^T$. Writing $\tilde{L}_j$ in block matrix form, we obtain

$$\tilde{L}_j = \begin{pmatrix} T_j & B_j \\ C_j & D_j \end{pmatrix}.$$

Using this, (1.2) is written as

$$\begin{pmatrix} T_j & B_j \\ C_j & D_j \end{pmatrix}\begin{pmatrix} U_L \\ U_H \end{pmatrix} = \begin{pmatrix} F_L \\ F_H \end{pmatrix}.$$

Solving for U_L gives

$$U_L = (T_j - B_j D_j^{-1} C_j)^{-1}(F_L - B_j D_j^{-1} F_H),$$

where $T_j - B_j D_j^{-1} C_j$ is later seen to be the coarse grid operator. The block UDL decomposition of $\tilde{L}_j$, where U is block upper triangular with unit diagonal, D is block diagonal, and L is block lower triangular with unit diagonal is then obtained. By inverting this and solving for

$$U = \begin{pmatrix} U_L \\ U_H \end{pmatrix},$$

the coarse grid, interpolation, and restriction operators are determined, where the coarse grid operator is the Schur complement of D_j in $\tilde{L}_j$, as seen above. For computation of similar expressions, see [CGM85, CTW97]. One thing to keep in mind is that the blocks T_j, B_j, C_j, and D_j are sparse, due to the properties of the scaling and wavelet operators.

As a vital part of this work, we attempt to make the method more useful in real-world applications. Again, this aspect was not considered in [EL]. First, we observe that the construction of the coarse grid, restriction, and interpolation operators requires inversion of a matrix. Not only does the calculation of the inverse of a matrix require much work, but also inverting a sparse matrix results in a great deal of fill-in, yielding a dense matrix. We attack this problem directly, first calculating the inverse exactly and applying a thresholding strategy to see if any banding can be detected. In practice, however, we want to assume a banded structure. Therefore, we apply ILU(0) to calculate the LU-decomposition of the matrix in a more efficient manner, then compute the inverse via backward and forward substitution. Finally, we take care of the density issue. Though we have efficiently calculated the inverse of the matrix, it is still dense. So, we apply a truncation strategy to obtain a banded, sparser matrix.

In Chapter 2, we discuss some multigrid background. Chapter 3 discusses wavelets, in both one and two dimensions, as background for later chapters. Chapter 4 discusses the application of the wavelet transform to multigrid methods. This chapter is broken up into four parts, an expository part dealing with one-dimensional problems (as explained in [EL]), a discussion of compression results in one dimension (not considered in [EL]), an explanatory part dealing with the two-dimensional results, and examples. In Chapter 5, we investigate the sparsity patterns of the inverse component of the coarse grid, interpolation, and restriction operators. Chapter 6 discusses application of ILU(0) to the component to be inverted and also gives the results of using ILU(0) and truncation to obtain sparse versions of the inverse component. Next, Chapter 7 describes the numerical results of applying the wavelet multigrid method to a variety of problems, from elliptic problems to generalized problems involving only second order partial derivatives to the advection-diffusion equation with dominant advection. In Chapter 8, we discuss the results of applying the wavelet multigrid method to a reformulated version of Stokes equations and a reformulation of the incompressible Navier-Stokes equations. For all of the numerical results in this paper, the two-level V-cycle method is used with one Gauss-Seidel iteration in both the downswing (coarsening) and the upswing (correction) phases, unless otherwise specified.

CHAPTER 2

Multigrid Methods

The problem we are concerned with solving is the system of linear equations

$$Au = b \tag{2.1}$$

(where A and b arise from discretization of a differential equation on some grid Ω^h, where h represents the step size). Looking at basic iterative methods, such as Jacobi and Gauss-Seidel, we see that they are successful at reducing the oscillatory (or high-frequency) components in the error but not at reducing the smooth (or low-frequency) components. Thus, we see a rapid reduction of error in the early iteration steps, followed by a much slower error reduction. We observe, however, that smooth errors on a fine grid appear less smooth on a coarser grid. Multigrid methods take advantage of this observation by first performing a few smoothing iterations on a fine grid, then restricting the residual to the coarse grid and solving the error equation ($Ae = r$) on the coarse grid.

Before moving on and describing the standard multigrid methods in more detail, we must point out that the assumptions necessary for the standard method to work are not met by several types of problems. For problems with discontinuous or highly oscillatory coefficients or for advection-dominated problems, the multigrid methods are not as efficient. One problem is that small eigenvalues may not be associated with smooth eigenfunctions. For such problems, it is not as simple to approximate the smooth eigenfunctions on the coarse grids. The basic iterative methods only smooth out the high-frequency eigenmodes of the error and we need to approximate the low-frequency eigenmodes on the coarse grid. But, the homogenized problem approximates the eigenvalues and eigenvectors of the original problem [Kes79, SV93]. So, this leads to methods such as using homogenization to obtain a coarse grid operator, and, in more complicated cases, to obtain the interpolation and restriction operators in place of the standard operators [DE98, AEL97, EL96, EL97].

We need to find effective ways to perform the interpolation from the vectors on the coarse grid to vectors on the fine grid. In standard multigrid, this is done via an interpolation operator. Let $u^h_{i,j} = u(ih, jh)$ be the approximate solution to (2.1) on the fine grid. In two dimensions, the interpolation operator, defined by $u^h = I^h_{2h} u^{2h}$, may take the form of

$$\begin{aligned}
u^h_{2i,2j} &= u^{2h}_{i,j} \\
u^h_{2i+1,2j} &= \frac{u^{2h}_{i,j} + u^{2h}_{i+1,j}}{2} \\
u^h_{2i,2j+1} &= \frac{u^{2h}_{i,j} + u^{2h}_{i,j+1}}{2} \\
u^h_{2i+1,2j+1} &= \frac{u^{2h}_{i,j} + u^{2h}_{i+1,j} + u^{2h}_{i,j+1} + u^{2h}_{i+1,j+1}}{4}.
\end{aligned}$$

We generally obtain the coarse grid operator (our operator A defined on the coarse grid) in one of two ways. With the first approach, we determine the coarse grid operator using the step size $H = 2h$, where h is the step size from the original problem (if we are considering differential equations). Alternately, we may use what is known as the Galerkin approach. For this, we define A^{2h}, the coarse grid operator, as $A^{2h} = I^{2h}_h A^h I^h_{2h}$.

Represent our problem on the fine grid as

$$A^h u^h = b^h. \tag{2.2}$$

Now, given u^h on Ω^h, the fine grid, let $r^h = b^h - A^h u^h$ denote the residual on the fine grid. We need to represent r^h on Ω^{2h}, the coarse grid. To do this, we use a restriction operator I_h^{2h}, defined by $r^{2h} = I_h^{2h} r^h$. For the standard multigrid, the full-weighting restriction in two dimensions is given by

$$\begin{aligned} u_{i,j}^{2h} &= \frac{1}{16}[u_{2i+1,2j+1}^h + u_{2i+1,2j-1}^h + u_{2i-1,2j+1}^h + u_{2i-1,2j-1}^h \\ &\quad +2(u_{2i,2j+1}^h + u_{2i,2j-1}^h + u_{2i+1,2j}^h + u_{2i-1,2j}^h) + 4u_{2i,2j}^h]. \end{aligned}$$

The basic procedure follows. First, relax a few (usually one or two) steps on the fine grid Ω^h to get an initial guess u^h. Then, compute the residual $r^h = b^h - A^h u^h$ and restrict to the coarser grid Ω^{2h} : $r^{2h} = I_h^{2h} r^h$. Then, solve the residual equation $Ae = r$ on the coarser grid. Then, set $u^h = u^h + I_{2h}^h e^{2h}$ and relax again a few steps on the fine grid (usually one or two steps). This describes the two-level method.

With this in mind, we can recursively define the V-cycle multigrid scheme:

$u^h \leftarrow MG^h(u^h, b^h)$

Step 1: Relax ν_1 times on $A^h u^h = b^h$, starting with some initial guess u^h.

Step 2: If we are on the coarsest grid, go to Step 4.

(Alternately, solve exactly.)

Otherwise, set $r^h = b^h - A^h u^h,\ b^{2h} = I_h^{2h} r^h,\ u^{2h} \leftarrow 0,$

and $u^{2h} \leftarrow MG^{2h}(u^{2h}, b^{2h})$.

Step 3: Set $u^h \leftarrow u^h + I_{2h}^h u^{2h}$.

Step 4: Relax ν_2 times on $A^h u^h = b^h$ starting with initial guess u^h.

The V-cycle schemes are one of a class of μ-cycle schemes. In the μ-cycle schemes, Steps 1 through 4 of the V-cycle procedure are the same, with the exception of the last item in Step 2. In Step 2, we perform the operation $u^{2h} \leftarrow MG^{2h}(u^{2h}, b^{2h})$ μ times, not just once. In practice, the V-cycle (with $\mu = 1$) and the W-cycle (with $\mu = 2$) are used.

The procedure that combines the V-cycle scheme with nested iteration (which is using coarse grids to get better initial guesses to finer grids) is the full multigrid method. Basically, we restrict to the coarsest grid and do some iterations of the smoothing method, then we correct to the next finer grid and perform one (or more) V-cycles, and then interpolate to the next finer grid. We continue this process until we reach the finest grid, whereupon we perform one (or more) V-cycles.

The above procedures describe a linear multigrid approach. A good lower level introductory book on multigrid methods is [Bri87]. Another good reference is [McC87], which contains several articles on multigrid methods, including a brief introduction to standard multigrid methods and an article on linear multigrid methods [Wes87].

There are also nonlinear multigrid methods. Typically, these involve using a nested sequence of subspaces and orthogonal projection operators to get the restriction and interpolation operators. A variation on this involves using the finite element method in discretizing the differential equation and then using different levels of triangulation to get a multilevel algorithm. Basically, a nested sequence of subspaces is defined based on a nested sequence of triangulations. We suppose that the triangulation on the fine grid is constructed by a succession of refinements of triangulations from previous grids. For example, in two dimensions, we can form a coarse grid triangulation and then connect the midpoints of the triangles to get a finer triangulation. A finite element space is then defined for each level. For a more detailed discussion of nonlinear methods, see, for example, [Ban96, Xu92a, Xu97, BX95, BX94, CZ85, WCS98].

Defect-correction schemes are often used as well. These involve making a correction to the coarse grid equation to ensure that the coarse grid solution will represent the fine grid solution accurately. This has been applied to nonlinear problems.

The full approximation scheme (FAS) is another type of multigrid method. FAS is used for nonlinear problems because the error equation in the standard multigrid methods is only valid for linear problems. For nonlinear problems, we need to instead solve on the coarse grid for a function that represents the sum of a restriction to the coarse grid of the approximate solution, which is found by iterating on the fine grid, and the error function found in standard multigrid. The restriction operator used above should accurately represent the solution space on the coarse grid and need not be the same as the operator used to restrict the error. For a more detailed discussion of this scheme, see [Bra82].

Another type of multigrid scheme is algebraic multigrid, in which any underlying geometric structure to the problem is not used. Algebraic multigrid only uses the structure of the matrix in the problem to determine the coarsening process (choice of coarse grid and definition of interpolation/restriction operators). This process is performed in order to ensure that the range of interpolation approximates the errors not sufficiently reduced via relaxation. For a more detailed description of algebraic methods, see [RS87, Zee90, Den82, DE89, MDH98, BCF98, VMB96, Stu99]. Also, in [BX94], an algebraic multigrid formulation of the hierarchical basis multigrid method (a nonlinear multigrid method) is discussed and compared to an incomplete LU-factorization. Algebraic multigrid methods are of particular interest to us, in that they are the nearest methods to the approach taken in this paper. We will take a moment to describe the approach taken in [RS87], since that is one of the bases of comparison used later.

Ruge and Stüben's algebraic multigrid method, described in [RS87], has a somewhat adaptive nature, in that the restriction and interpolation procedures depend on the nature of the matrix defining the problem on the fine grid. Basically the size of a matrix entry, relative to other entries on a row, determines the strength of what the authors call "connections." Based on these connections, as well as certain other criteria, the set of coarse grid points is determined. Interpolation is defined by: if the point is already on the coarse grid, the value remains the same; if the point is on the fine grid, but not the coarse grid, a weighted sum of interpolation points is used. The Galerkin formulation of the coarse grid operator is used. This procedure is then recursively applied to obtain more than two levels.

Another interesting semi-algebraic multigrid method is discussed in [Rie93, Rie94, RWZ94]. In this case, Andreas Rieder (and his coauthors, in the latter paper) uses a choice of either the scaling operator or the wavelet operator for restriction/interpolation operators. The Galerkin procedure for obtaining the coarse grid operator is used. Also, a good survey of robust multigrid methods for elliptic equations can be found in [CW99].

It is expected that V-cycle schemes should converge at a rate independent of the step size, mainly due to the fact that the convergence rate of most relaxation schemes is independent of step size for the oscillatory modes. There are many articles discussing the convergence of multigrid methods applied to specific problems. For example, [Reu94] discusses max-norm convergence of multigrid methods for elliptic boundary value problems and [MM85] discusses a convergence estimate for the V-cycle method for symmetric positive definite operators. Research on convergence of multigrid methods for nonsymmetric problems includes [Man86, BPX88, Wan93a, Wan93b, ZV85].

In, for example, problems with highly oscillatory coefficients, this mesh-independent convergence is no longer true. New methods for restriction and prolongation (interpolation) or for treating the entire problem must be found.

CHAPTER 3

Wavelets

3.1 One-Dimensional Discussion

A brief description of wavelets follows. For more details, please refer to [Dau88, Dau92]. Wavelets basically separate data (or functions or operators) into different frequency components and analyze them by scaling. We can choose the wavelets to form a complete orthonormal basis of $L^2(\mathbb{R})$. And, by the very definition of the wavelet functions, they have time- or space-widths that are related to their frequency. At high frequencies, the wavelet functions are narrow and at low frequencies they are broader. This is due to the scaling of the wavelet functions. Therefore, they provide good localization of functions in both the frequency domain and physical space, and representation by wavelets seems natural to apply to analysis of fine and coarse scales as well as local properties of functions.

Basically, a multiresolution analysis (MRA) consists of a sequence of closed subspaces V_j of $L^2(\mathbb{R})$ that satisfy

$$(i) V_{j+1} \subset V_j, \ \forall j \in \mathbb{Z}, \tag{3.1}$$

$$(ii) \overline{\bigcup_{j\in\mathbb{Z}} V_j} = L^2(\mathbb{R}), \tag{3.2}$$

$$(iii) \bigcap_{j\in\mathbb{Z}} V_j = \{0\}. \tag{3.3}$$

If we define P_j to be the orthogonal projection onto V_j, then (3.1) and (3.2) ensure that $\lim_{j \to -\infty} P_j f = f,\ \forall f \in L^2(\mathbb{R})$. In order for the above to form an MRA, an additional requirement must be satisfied:

$$f \in V_j \Longleftrightarrow f(2^j \cdot) \in V_0,\ \forall j \in \mathbb{Z}. \tag{3.4}$$

Therefore, all the spaces V_j represent scalings of V_0.

Other properties an MRA must satisfy include:

$$i) \text{ if } f \in V_0, \text{then } f(x-n) \in V_0,\ \forall n \in \mathbb{Z}; \tag{3.5}$$

$ii)$ there exist scaling functions $\phi \in V_0$ such that

$$\{\phi(x-n) : n \in \mathbb{Z}\} \tag{3.6}$$

forms an orthonormal basis of V_0.

Note that this then implies that

$$\{\phi_{j,n} = 2^{\frac{-j}{2}}\phi(2^{-j}x - n) : n \in \mathbb{Z}\}$$

is an orthonormal basis for $V_j,\ \forall j \in \mathbb{Z}$.

The main idea is that given a set of closed subspaces satisfying (3.1) - (3.6), there exists an orthonormal wavelet basis

$$\{\psi_{j,k} : j, k \in \mathbb{Z}\}$$

of $L^2(\mathbb{R})$, where $\psi_{j,k} = 2^{\frac{-j}{2}}\psi(2^{-j}x - k)$, such that

$$P_{j-1}f = P_j f + \sum_{k \in \mathbb{Z}} < f, \psi_{j,k} > \psi_{j,k}$$

(recall that P_j is the projection onto V_j). These wavelets are constructed by considering, for every $j \in \mathbb{Z}$, the orthogonal complement of V_j in V_{j-1}. Let

$$W_j = \{f \in V_{j-1} :< f, g >= 0,\ \forall g \in V_j\}.$$

Then, $V_{j-1} = V_j \oplus W_j$ and $W_j \perp W_k$ for $j \neq k$. It follows that for $j < J$,

$$V_j = V_J \oplus \bigoplus_{k=0}^{J-j-1} W_{J-k}.$$

Then,

$$L^2(\mathbb{R}) = \bigoplus_{j\in\mathbb{Z}} W_j.$$

Also, W_j has the same scaling property (3.4) as the V_j, and we can find $\psi \in W_0$ (the mother wavelet) such that

$$\{\psi(\cdot - k) : k \in \mathbb{Z}\}$$

is an orthonormal basis for W_0. This implies that

$$\{\psi_{j,k}(x) = 2^{\frac{-j}{2}}\psi(2^{-j}x - k) : k \in \mathbb{Z}\}$$

is an orthonormal basis for W_j, for any $j \in \mathbb{Z}$.

The simplest example of an MRA is the Haar MRA:

$$\phi(x) = \begin{cases} 1 & \text{if } 0 \leq x < 1, \\ 0 & \text{otherwise,} \end{cases} \quad \psi(x) = \begin{cases} 1 & \text{if } 0 \leq x < \frac{1}{2}, \\ -1 & \text{if } \frac{1}{2} \leq x < 1, \\ 0 & \text{otherwise.} \end{cases}$$

For each j, V_j as defined by the Haar scaling functions is a space of piecewise constant functions. (Also, note that we can easily prove that any $f \in L^2(\mathbb{R})$ can be written as a linear combination of piecewise constant functions, e.g., step functions).

Now, to actually construct ψ, we examine some properties of ϕ and W_0. Since $\phi \in V_0 \subset V_{-1}$ and $\phi_{-1,n}$ form an orthonormal basis for V_{-1},

$$\phi = \sum_n h_n \phi_{-1,n}, \tag{3.7}$$

where

$$h_n = < \phi, \phi_{-1,n} > \text{ and } \sum_{n \in \mathbb{Z}} |h_n|^2 = 1. \tag{3.8}$$

We can rewrite (3.7) as

$$\phi(x) = \sqrt{2} \sum_n h_n \phi(2x - n) \tag{3.9}$$

or

$$\hat{\phi}(\xi) = \frac{1}{\sqrt{2}} \sum_n h_n e^{\frac{-in\xi}{2}} \hat{\phi}(\frac{\xi}{2}), \tag{3.10}$$

with convergence being in L^2. We can write (3.10) as

$$\hat{\phi}(\xi) = m_0(\frac{\xi}{2})\hat{\phi}(\frac{\xi}{2}), \tag{3.11}$$

where

$$m_0(\xi) = \frac{1}{\sqrt{2}} \sum_n h_n e^{-in\xi}, \tag{3.12}$$

with (3.11) being defined pointwise a.e.

From (3.8), we see that $m_0(\xi)$ is 2π-periodic in $L^2([0, 2\pi])$. The orthonormality of the $\phi(\cdot - n)$ leads to

$$|m_0(\xi)|^2 + |m_0(\xi + \pi)|^2 = 1 \text{ a.e.} \tag{3.13}$$

Using the characteristics of W_0, we come to the conclusion that the wavelet ψ can take the form of

$$\psi = \sum_n g_n \phi_{-1,n} \text{ with } g_n = (-1)^n h_{-n-1}. \tag{3.14}$$

For the Haar MRA,

$$h_n = \sqrt{2} \int \phi(x)\overline{\phi(2x - n)}dx = \begin{cases} \frac{1}{\sqrt{2}} & \text{if } n = 0, 1, \\ 0 & \text{otherwise.} \end{cases}$$

So,

$$g_n = (-1)^n h_{-n-1} = \begin{cases} \frac{1}{\sqrt{2}} & \text{if } n = 0, \\ \frac{-1}{\sqrt{2}} & \text{if } n = 1, \\ 0 & \text{otherwise.} \end{cases}$$

Now,

$$\begin{aligned} \phi_{j,k}(x) &= 2^{\frac{-j}{2}} \phi(2^{-j}x - k) \\ &= 2^{\frac{-j}{2}} \sum_n h_n 2^{\frac{1}{2}} \phi(2^{-j+1}x - 2k - n) \\ &= \sum_n h_n \phi_{j-1,2k+n}(x) \\ &= \sum_n h_{n-2k} \phi_{j-1,n}(x). \end{aligned} \tag{3.15}$$

Given $f^0 \in V_0 = V_1 \oplus W_1$, we can decompose f^0 into $f^0 = f^1 + \delta^1$. So, $f^0 = \sum_n < f, \phi_{0,n} > \phi_{0,n} = \sum_n c_n^0 \phi_{0,n}$, where $c_n^0 =< f, \phi_{0,n} >$. Then,

$$\begin{aligned} f^1 &= \sum_n < f, \phi_{1,n} > \phi_{1,n} = \sum_n c_n^1 \phi_{1,n}, \\ \delta^1 &= \sum_n < f, \psi_{1,n} > \psi_{1,n} = \sum_n d_n^1 \psi_{1,n}. \end{aligned}$$

It follows from (3.15) that

$$\begin{aligned} c_k^1 =< f, \phi_{1,k} > &= \sum_n \overline{h_{n-2k}} < f, \phi_{0,n} > \\ &= \sum_n \overline{h_{n-2k}} c_n^0. \end{aligned} \tag{3.16}$$

Similarly,

$$d_k^1 = \sum_n \overline{g_{n-2k}} c_n^0.$$

So, we define H_0 and G_0 to be the operators such that, letting $a = (a_k)_{k \in \mathbb{Z}}$, we have

$$c^1 = H_0 c^0 \text{ and } d^1 = G_0 c^0.$$

By induction on the above procedure, we can obtain

$$c^{j+1} = H_j c^j, \tag{3.17}$$

$$d^{j+1} = G_j c^j. \tag{3.18}$$

Now, to get f^j from f^{j+1} and δ^{j+1}, we have

$$\begin{aligned} f^j &= f^{j+1} + \delta^{j+1} \\ &= \sum_k c_k^{j+1} \phi_{j+1,k} + \sum_k d_k^{j+1} \psi_{j+1,k}. \end{aligned}$$

So,

$$\begin{aligned} c_n^j &=< f^j, \phi_{j,n} > \\ &= \sum_k c_k^{j+1} < \phi_{j+1,k}, \phi_{j,n} > + \sum_k d_k^{j+1} < \psi_{j+1,k}, \phi_{j,n} > \\ &= \sum_k [h_{n-2k} c_k^{j+1} + g_{n-2k} d_k^{j+1}]. \end{aligned}$$

Write this as

$$c^j = H_j^* c^{j+1} + G_j^* d^{j+1}. \tag{3.19}$$

So, H_j is defined by

$$(H_j a)_k = \sum_n \overline{h_{n-2k}} a_n$$

and G_j is defined by

$$(G_j a)_k = \sum_n \overline{g_{n-2k}} a_n.$$

The properties of H_j and G_j are

$$(i) H_j^* H_j + G_j^* G_j = I \tag{3.20}$$

(this follows from (3.17), (3.18), and (3.19)),

$$(ii) H_j G_j^* = G_j H_j^* = 0 \tag{3.21}$$

(this follows since $V_j \perp W_j$),

$$(iii) H_j H_j^* = G_j G_j^* = I \tag{3.22}$$

(this follows from (3.17), (3.18), and (3.19)).

We will assume that the elements of H_j and G_j are real, as we may choose them to be so. So, we may replace the conjugate transpose by the transpose. Now, define $\mathcal{W}_j : V_j \rightarrow V_{j+1} \oplus W_{j+1}$. This transformation maps $\{\phi_{j,k}\}$ into $\{\phi_{j+1,k}, \psi_{j+1,k}\}$. Then, by definition,

$$\mathcal{W}_j = \begin{pmatrix} H_j \\ G_j \end{pmatrix}.$$

Note that $\mathcal{W}_j$ is orthogonal, i.e.,

$$\mathcal{W}_j^T \mathcal{W}_j = \mathcal{W}_j \mathcal{W}_j^T = I.$$

The main point to observe here is that the discrete wavelet operators are computationally efficient. With respect to the Haar multiresolution analysis described above, application of the low-frequency operator to an $n \times n$ matrix involves only $2n$ operations. The same holds for the high-frequency operator. So, the application of the wavelet transform requires only $4n$ operations. In general, application of the wavelet transform requires $\mathcal{O}(n)$ operations, assuming a finite number of coefficients for the low- and high-frequency operators.

3.2 The Two-Dimensional Case

In two dimensions, we use the tensor product of one-dimensional multiresolution analyses. Now, define $\mathbf{V}_j$ by

$$\mathbf{V}_j = V_j \otimes V_j \text{ (tensor product) } = span\{\Phi(x,y) = \phi(x)\tilde{\phi}(y) : \phi, \tilde{\phi} \in V_j\}.$$

And, $F \in \mathbf{V}_j \Leftrightarrow F(2^j\cdot, 2^j\cdot) \in \mathbf{V}_0$. Then, the $\mathbf{V}_j$'s form an MRA in $L^2(\mathbb{R}^2)$ satisfying

$$(i)\mathbf{V}_{j+1} \subset \mathbf{V}_j, \ \forall j \in \mathbb{Z}, \tag{3.23}$$

$$(ii)\overline{\bigcup_{j\in\mathbb{Z}} \mathbf{V}_j} = L^2(\mathbb{R}^2), \tag{3.24}$$

$$(iii)\bigcap_{j\in\mathbb{Z}} \mathbf{V}_j = \{0\}. \tag{3.25}$$

Since the $\phi(\cdot - n), n \in \mathbb{Z}$, form an orthonormal basis for V_0, then

$$\{\Phi_{k,l}(x,y) = \phi(x-k)\phi(y-l) : k,l \in \mathbb{Z}\}$$

forms a basis for $\mathbf{V}_0$. Similarly,

$$\{\Phi_{j,k,l}(x,y) = \phi_{j,k}(x)\phi_{j,l}(y) = 2^{-j}\Phi(2^{-j}x - k, 2^{-j}y - l) : k,l \in \mathbb{Z}\}$$

forms an orthonormal basis of $\mathbf{V}_j$.

Now, for each $j \in \mathbb{Z}$, we define $\mathbf{W}_j$ to be the orthogonal complement of $\mathbf{V}_j$ in $\mathbf{V}_{j-1}$. So,

$$\begin{aligned}
\mathbf{V}_{j-1} &= V_{j-1} \otimes V_{j-1} \\
&= (V_j \oplus W_j) \otimes (V_j \oplus W_j) \\
&= (V_j \otimes V_j) \oplus [(W_j \otimes V_j) \oplus (V_j \otimes W_j) \oplus (W_j \otimes W_j)] \\
&= \mathbf{V}_j \oplus \mathbf{W}_j.
\end{aligned} \tag{3.26}$$

So, now, $\mathbf{W}_j$ is made up of three parts with orthonormal bases given by

$$\psi_{j,k}(x)\phi_{j,l}(y) \text{ for } W_j \otimes V_j,$$
$$\phi_{j,k}(x)\psi_{j,l}(y) \text{ for } V_j \otimes W_j, \text{ and}$$
$$\psi_{j,k}(x)\psi_{j,l}(y) \text{ for } W_j \otimes W_j.$$

Then, as in the one dimensional case, we define $\mathbf{H}_j$ to be

$$\mathbf{H}_j = H_j^{col} H_j^{row}$$

and $\mathbf{G}_j$ to be

$$\mathbf{G}_j = \begin{pmatrix} G_j^{col} H_j^{row} \\ H_j^{col} G_j^{row} \\ G_j^{col} G_j^{row} \end{pmatrix}.$$

Then, as before, define $\boldsymbol{\mathcal{W}}_j$ by

$$\boldsymbol{\mathcal{W}}_j = \begin{pmatrix} \mathbf{H}_j \\ \mathbf{G}_j \end{pmatrix}.$$

So, $\boldsymbol{\mathcal{W}}_j : \mathbf{V}_j \to \mathbf{V}_{j+1} \oplus \mathbf{W}_{j+1}$. Note that $\boldsymbol{\mathcal{W}}_j$ is orthogonal: $\boldsymbol{\mathcal{W}}_j^T \boldsymbol{\mathcal{W}}_j = \boldsymbol{\mathcal{W}}_j \boldsymbol{\mathcal{W}}_j^T = I$. This is because

$$\begin{aligned} \mathbf{H}_j \mathbf{H}_j^T &= (H_j^{col} H_j^{row})(H_j^{col} H_j^{row})^T \\ &= H_j^{col} H_j^{row} (H_j^{row})^T (H_j^{col})^T \\ &= I. \end{aligned}$$

Similarly, $\mathbf{G}_j \mathbf{G}_j^T = I$, $\mathbf{H}_j \mathbf{G}_j^T = \mathbf{G}_j \mathbf{H}_j^T = 0$, and, finally, it can be shown that $\mathbf{H}_j^T \mathbf{H}_j + \mathbf{G}_j^T \mathbf{G}_j = I$.

CHAPTER 4

Applying Wavelet Transform to Multigrid

4.1 One-Dimensional Case

We will here give an explanation and example of the wavelet multigrid method in the one-dimensional case, as discussed in [EL]. We do this simply for expository purposes, in order to contribute to the understanding of the procedure in the two-dimensional case. Note that here the fine grid operator is assumed to be symmetric. Given the problem

$$L_j U = F,$$

where L_j represents the operator on the fine grid obtained by discretization of a one-dimensional boundary value problem, we apply the wavelet transform to both sides of the equation. For example, consider the equation

$$\begin{aligned} -a\frac{d^2u(x)}{dx^2} &= f(x) \text{ in } (0,1) \\ u(0) &= u(1) = 0, \end{aligned} \tag{4.1}$$

which we discretize using the standard three-point discretization. Denote the step size by h, where $h = \frac{1}{N+1}$, N being the number of interior gridpoints, and $u_i = u(ih)$. Thus, we obtain the following difference problem:

$$a\frac{-u_{i+1} + 2u_i - u_{i-1}}{h^2} = f_i, \; i = 0, ..., N-1. \tag{4.2}$$

We may then represent this as the matrix problem $L_jU = F$, where F represents the vector $(f_0, \ldots, f_{N-1})^T$, U represents the vector $(u_0, \ldots, u_{N-1})^T$, and L_j denotes the operator determined from the above equation,

$$L_j = \frac{1}{h^2}\begin{pmatrix} 2a & -a & & & \\ -a & 2a & -a & & \\ & \ddots & \ddots & \ddots & \\ & & & -a & 2a \end{pmatrix}. \tag{4.3}$$

Let us use the Haar wavelet MRA, so that

$$H_j = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & \cdots & \cdots & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & \cdots & \cdots & \cdots & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix}$$

and

$$G_j = \begin{pmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & \cdots & \cdots & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & \cdots & \cdots & \cdots & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}.$$

Applying the wavelet transform to the equation yields

$$\begin{aligned} (\mathcal{W}_j L_j \mathcal{W}_j^T)\mathcal{W}_j U &= \mathcal{W}_j F \\ \Longrightarrow (\mathcal{W}_j L_j \mathcal{W}_j^T)\begin{pmatrix} U_L \\ U_H \end{pmatrix} &= \begin{pmatrix} F_L \\ F_H \end{pmatrix}, \end{aligned} \tag{4.4}$$

where $U_L,\ F_L \in V_j$ and $U_H,\ F_H \in W_j$. The subscripts L and H are used because the component H_j of the wavelet transform can be likened to a low-pass filter (i.e., only low frequency values can come in) and G_j is like a high-pass filter, allowing only high-frequency values. So, U_L and U_H consist of the low- and high-frequency parts of U, respectively, and similarly for F. And,

$$\mathcal{W}_j L_j \mathcal{W}_j^T = \begin{pmatrix} H_j L_j H_j^T & H_j L_j G_j^T \\ G_j L_j H_j^T & G_j L_j G_j^T \end{pmatrix}.$$

Then, we let

$$\begin{aligned} T_j &= H_j L_j H_j^T, \\ B_j &= H_j L_j G_j^T, \\ D_j &= G_j L_j G_j^T. \end{aligned}$$

Define $\tilde{L}_j$ as follows:

$$\tilde{L}_j = \mathcal{W}_j L_j \mathcal{W}_j^T = \begin{pmatrix} T_j & B_j \\ B_j^T & D_j \end{pmatrix}.$$

So, using (4.3), with $a = 1$ and $h = \frac{1}{9}$, we see

$$L_j = \begin{pmatrix} 162 & -81 & 0 & 0 & 0 & 0 & 0 & 0 \\ -81 & 162 & -81 & 0 & 0 & 0 & 0 & 0 \\ 0 & -81 & 162 & -81 & 0 & 0 & 0 & 0 \\ 0 & 0 & -81 & 162 & -81 & 0 & 0 & 0 \\ 0 & 0 & 0 & -81 & 162 & -81 & 0 & 0 \\ 0 & 0 & 0 & 0 & -81 & 162 & -81 & 0 \\ 0 & 0 & 0 & 0 & 0 & -81 & 162 & -81 \\ 0 & 0 & 0 & 0 & 0 & 0 & -81 & 162 \end{pmatrix}$$

and

$$\tilde{L}_j = \begin{pmatrix} 81 & -40.5 & 0 & 0 & 81 & -40.5 & 0 & 0 \\ -40.5 & 81 & -40.5 & 0 & -40.5 & 81 & -40.5 & 0 \\ 0 & -40.5 & 81 & -40.5 & 0 & -40.5 & 81 & -40.5 \\ 0 & 0 & -40.5 & 81 & 0 & 0 & -40.5 & 81 \\ 81 & -40.5 & 0 & 0 & 243 & 40.5 & 0 & 0 \\ -40.5 & 81 & -40.5 & 0 & 40.5 & 243 & 40.5 & 0 \\ 0 & -40.5 & 81 & -40.5 & 0 & 40.5 & 243 & 40.5 \\ 0 & 0 & -40.5 & 81 & 0 & 0 & 40.5 & 243 \end{pmatrix}.$$

The block UDL decomposition of $\tilde{L}_j$, where U is block upper triangular with unit diagonal, D is block diagonal, and L is block lower triangular with unit diagonal, is then

$$\tilde{L}_j = \begin{pmatrix} I & B_j D_j^{-1} \\ 0 & I \end{pmatrix} \begin{pmatrix} T_j - B_j D_j^{-1} B_j^T & 0 \\ 0 & D_j \end{pmatrix} \begin{pmatrix} I & 0 \\ D_j^{-1} B_j^T & I \end{pmatrix}. \qquad (4.5)$$

We seek $\tilde{L}_j^{-1}$:

$$\tilde{L}_j^{-1} = (\mathcal{W}_j L_j \mathcal{W}_j^T)^{-1} = \mathcal{W}_j L_j^{-1} \mathcal{W}_j^T.$$

The inverse of the factorization of $\tilde{L}_j$ is

$$\begin{aligned} &\begin{pmatrix} I & 0 \\ -D_j^{-1} B_j^T & I \end{pmatrix} \begin{pmatrix} (T_j - B_j D_j^{-1} B_j^T)^{-1} & 0 \\ 0 & D_j^{-1} \end{pmatrix} \begin{pmatrix} I & -B_j D_j^{-1} \\ 0 & I \end{pmatrix} \\ &= \begin{pmatrix} (T_j - B_j D_j^{-1} B_j^T)^{-1} & 0 \\ -D_j^{-1} B_j^T (T_j - B_j D_j^{-1} B_j^T)^{-1} & D_j^{-1} \end{pmatrix} \begin{pmatrix} I & -B_j D_j^{-1} \\ 0 & I \end{pmatrix} \\ &= \begin{pmatrix} (T_j - B_j D_j^{-1} B_j^T)^{-1} & -(T_j - B_j D_j^{-1} B_j^T)^{-1} B_j D_j^{-1} \\ -D_j^{-1} B_j^T (T_j - B_j D_j^{-1} B_j^T)^{-1} & D_j^{-1} B_j^T (T_j - B_j D_j^{-1} B_j^T)^{-1} B_j D_j^{-1} + D_j^{-1} \end{pmatrix}. \end{aligned}$$

Now, we know that $U = L_j^{-1}F$

$$\Longrightarrow \mathcal{W}_j U = \mathcal{W}_j L_j^{-1}\mathcal{W}_j^T\mathcal{W}_j F = \tilde{L}_j^{-1}\mathcal{W}_j F. \tag{4.6}$$

We obtain from (4.6), noting that $\mathcal{W}_j U = \begin{pmatrix} U_L \\ U_H \end{pmatrix}$ and similarly for $\mathcal{W}_j F$, and from the calculation of $\tilde{L}_j^{-1}$ that

$$\begin{pmatrix} U_L \\ U_H \end{pmatrix} = \tilde{L}_j^{-1}\begin{pmatrix} F_L \\ F_H \end{pmatrix}$$

$$= \begin{pmatrix} (T_j - B_jD_j^{-1}B_j^T)^{-1} & -(T_j - B_jD_j^{-1}B_j^T)^{-1}B_jD_j^{-1} \\ -D_j^{-1}B_j^T(T_j - B_jD_j^{-1}B_j^T)^{-1} & D_j^{-1}B_j^T(T_j - B_jD_j^{-1}B_j^T)^{-1}B_jD_j^{-1} + D_j^{-1} \end{pmatrix} *$$
$$\begin{pmatrix} F_L \\ F_H \end{pmatrix}$$

$$= \begin{pmatrix} (T_j - B_jD_j^{-1}B_j^T)^{-1} & -(T_j - B_jD_j^{-1}B_j^T)^{-1}B_jD_j^{-1} \\ -D_j^{-1}B_j^T(T_j - B_jD_j^{-1}B_j^T)^{-1} & D_j^{-1}B_j^T(T_j - B_jD_j^{-1}B_j^T)^{-1}B_jD_j^{-1} + D_j^{-1} \end{pmatrix} *$$
$$\begin{pmatrix} H_j \\ G_j \end{pmatrix} F$$

$$= \begin{pmatrix} (T_j - B_jD_j^{-1}B_j^T)^{-1}(H_j - B_jD_j^{-1}G_j) \\ -D_j^{-1}B_j^T(T_j - B_jD_j^{-1}B_j^T)^{-1}(H_j - B_jD_j^{-1}G_j) + D_j^{-1}G_j \end{pmatrix} F.$$

So, since $\begin{pmatrix} U_L \\ U_H \end{pmatrix} = \begin{pmatrix} H_j \\ G_j \end{pmatrix} U$ and $\mathcal{W}_j^T \mathcal{W}_j = I$,

$$U = \begin{pmatrix} H_j^T & G_j^T \end{pmatrix} \begin{pmatrix} (T_j - B_j D_j^{-1} B_j^T)^{-1}(H_j - B_j D_j^{-1} G_j) \\ -D_j^{-1} B_j^T (T_j - B_j D_j^{-1} B_j^T)^{-1}(H_j - B_j D_j^{-1} G_j) + D_j^{-1} G_j \end{pmatrix} F$$
$$= [H_j^T (T_j - B_j D_j^{-1} B_j^T)^{-1}(H_j - B_j D_j^{-1} G_j)$$
$$- G_j^T D_j^{-1} B_j^T (T_j - B_j D_j^{-1} B_j^T)^{-1}(H_j - B_j D_j^{-1} G_j) + G_j^T D_j^{-1} G_j] F$$
$$= [(H_j^T - G_j^T D_j^{-1} B_j^T)(T_j - B_j D_j^{-1} B_j^T)^{-1}(H_j - B_j D_j^{-1} G_j) + G_j^T D_j^{-1} G_j] F$$
$$= \begin{pmatrix} H_j^T - G_j^T D_j^{-1} B_j^T & G_j^T \end{pmatrix} \begin{pmatrix} (T_j - B_j D_j^{-1} B_j^T)^{-1} & 0 \\ 0 & D_j^{-1} \end{pmatrix} \begin{pmatrix} H_j - B_j D_j^{-1} G_j \\ G_j \end{pmatrix} F.$$

Denote

$$\begin{aligned} I_{2h}^h &= \sqrt{2}(H_j^T - G_j^T D_j^{-1} B_j^T) \text{ and} \\ I_h^{2h} &= \frac{1}{2}(I_{2h}^h)^T \end{aligned}$$

as our interpolation and restriction operators, respectively. Then,

$$U = I_{2h}^h (T_j - B_j D_j^{-1} B_j^T)^{-1} I_h^{2h} F + G_j^T D_j^{-1} G_j F. \tag{4.7}$$

We also note that in multigrid, we are working on the residual equation, i.e.,

$$e = I_{2h}^h (T_j - B_j D_j^{-1} B_j^T)^{-1} I_h^{2h} r + G_j^T D_j^{-1} G_j r.$$

If we assume that $G_j^T D_j^{-1} G_j r$ is small, i.e., r is almost in $Range(H_j^T)$, then we can approximate the error by

$$e = I_{2h}^h (T_j - B_j D_j^{-1} B_j^T)^{-1} I_h^{2h} r.$$

So,

$$(T_j - B_j D_j^{-1} B_j^T) e_{2h} = I_h^{2h} r_h.$$

The above assumption is good for most of the classical iterative methods, like Jacobi and Gauss-Seidel. Therefore, our coarse grid operator is

$$L_{j+1} = T_j - B_j D_j^{-1} B_j^T, \tag{4.8}$$

which is the Schur complement of D_j in $\tilde{L}_j$. For the above example, the coarse grid operator is

$$L_{j+1} = \begin{pmatrix} 73.8469 & -40.4659 & 6.9487 & -1.22624 \\ -40.4659 & 67.307 & -39.3759 & 6.9487 \\ 6.9487 & -39.3759 & 67.307 & -40.4659 \\ -1.22624 & 6.9487 & -40.4659 & 73.8469 \end{pmatrix}.$$

Now, although the matrices T_j, B_j, and D_j are as sparse as the original operator L_j, D_j^{-1} is not. But, we also observe that the fill-in that results from inverting is decaying exponentially as we move away from the original tridiagonal structure. This is evident above, in the structure of the coarse grid operator L_{j+1} (compare [BCR91]).

We can easily apply the above analysis again, in order to get a next coarser formulation of the original problem. The procedure may be repeatedly applied until the desired coarseness is reached. Although the level of fill-in in our D_j operator increases, the magnitude of the values decreases as we go away from the diagonal. One thing to be kept in mind is that the number of gridpoints used must be an even number, and the next to coarsest grid can have no fewer than four gridpoints.

4.2 Investigation into the Compressibility of D_j^{-1}

Here, we investigate the compressibility of D_j^{-1}. By this, we mean to investigate the decay of the values of the elements of D_j^{-1} on terms that are not on the main diagonal or on the diagonals above and below it. First, we will show that the matrix D_j has the same tridiagonal structure as the operator L_j, where L_j represents the operator formed by discretization of the one-dimensional differential equation

$$\begin{aligned} -\frac{d}{dx}(a(x)\frac{d}{dx}u(x)) + b(x)\frac{d}{dx}u(x) &= f(x),\ x \text{ in } \Omega \\ u(x) &= g,\ x \text{ on } \partial\Omega, \end{aligned} \tag{4.9}$$

where a is nonzero. The discretization of (4.9) takes the form

$$\frac{-a_{i+\frac{1}{2}}u_{i+1} + (a_{i+\frac{1}{2}} + a_{i-\frac{1}{2}})u_i - a_{i-\frac{1}{2}}u_{i-1}}{h^2} + \frac{b_i^- u_{i+1} + |b_i|u_i - b_i^+ u_{i-1}}{h}, \tag{4.10}$$

where

$$b_i^- = \frac{1}{2}(b_i - |b_i|), \tag{4.11}$$

$$b_i^+ = \frac{1}{2}(b_i + |b_i|). \tag{4.12}$$

Lemma 1. *Given L_j, the operator obtained from discretizing (4.9) using the three-point discretization with upwinding ((4.10), (4.11), and (4.12)), the matrix $D_j = G_j L_j G_j^T$, where G_j is the Haar wavelet operator, has the same tridiagonal structure as L_j.*

Proof. Given the discretization from (4.10), (4.11), and (4.12), L_j takes the form

$$L_j = \begin{pmatrix} \frac{a_{i+\frac{1}{2}}+a_{i-\frac{1}{2}}}{h^2} + \frac{|b_i|}{h} & -\frac{a_{i+\frac{1}{2}}}{h^2} + \frac{b_i^-}{h} & & \\ -\frac{a_{i-\frac{1}{2}}}{h^2} - \frac{b_i^+}{h} & \frac{a_{i+\frac{1}{2}}+a_{i-\frac{1}{2}}}{h^2} + \frac{|b_i|}{h} & -\frac{a_{i+\frac{1}{2}}}{h^2} + \frac{b_i^-}{h} & \\ & \ddots & \ddots & \ddots \\ & & -\frac{a_{i-\frac{1}{2}}}{h^2} - \frac{b_i^+}{h} & \frac{a_{i+\frac{1}{2}}+a_{i-\frac{1}{2}}}{h^2} + \frac{|b_i|}{h} \end{pmatrix}.$$

For simplicity, define

$$\alpha = \frac{a_{i+\frac{1}{2}} + a_{i-\frac{1}{2}}}{h^2} + \frac{|b_i|}{h}, \tag{4.13}$$

$$\beta = -\frac{a_{i+\frac{1}{2}}}{h^2} + \frac{b_i^-}{h}, \tag{4.14}$$

$$\gamma = -\frac{a_{i-\frac{1}{2}}}{h^2} - \frac{b_i^+}{h}. \tag{4.15}$$

Then, clearly, $\alpha = -\beta - \gamma$. So, we can write L_j as

$$L_j = \begin{pmatrix} \alpha & \beta & & \\ \gamma & \alpha & \beta & \\ & \ddots & \ddots & \ddots \\ & & \gamma & \alpha \end{pmatrix}. \tag{4.16}$$

Recall that

$$G_j = \begin{pmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & \cdots & \cdots & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & \cdots & \cdots & \cdots & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}.$$

Suppose L_j is a $4{\times}4$ matrix. Then,

$$L_j = \begin{pmatrix} \alpha & \beta & 0 & 0 \\ \gamma & \alpha & \beta & 0 \\ 0 & \gamma & \alpha & \beta \\ 0 & 0 & \gamma & \alpha \end{pmatrix}.$$

So,

$$
\begin{aligned}
D_j = G_j L_j G_j^T &= \frac{1}{2}\begin{pmatrix} 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{pmatrix}\begin{pmatrix} \alpha & \beta & 0 & 0 \\ \gamma & \alpha & \beta & 0 \\ 0 & \gamma & \alpha & \beta \\ 0 & 0 & \gamma & \alpha \end{pmatrix}\begin{pmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{pmatrix} \\
&= \frac{1}{2}\begin{pmatrix} \alpha-\gamma & \beta-\alpha & -\beta & 0 \\ 0 & \gamma & \alpha-\gamma & \beta-\alpha \end{pmatrix}\begin{pmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{pmatrix} \\
&= \frac{1}{2}\begin{pmatrix} 2\alpha-\gamma-\beta & -\beta \\ -\gamma & 2\alpha-\beta-\gamma \end{pmatrix} \\
&= \frac{1}{2}\begin{pmatrix} 3\alpha & -\beta \\ -\gamma & 3\alpha \end{pmatrix}.
\end{aligned}
$$

Then, clearly D_j is tridiagonal. Looking at the pattern in the multiplication process, we can see that we will still have a tridiagonal structure for L_j obtained from a discretization with more unknowns, since L_j and G_j have the same structure regardless of size. Therefore, the product $G_j L_j$ can have at most three terms in the first and last rows and four terms in the center rows. Then, when multiplying again by G_j^T, the resulting product is reduced to having two elements in the first and last rows and three in the central rows. $\square$

Theorem 1. *Define $D_j = G_j L_j G_j^T$ as in Lemma 4.2. Then, D_j^{-1} will be such that there exists constants $C > 0$ and $0 < \rho < 1$ such that*

$$|(D_j^{-1})_{ij}| \leq C\rho^{|j-i|}. \tag{4.17}$$

Proof. This proof is done by using a power series expansion to calculate D_j^{-1}. First, we must establish that $\frac{|\beta|}{|\alpha|} < 1$ and $\frac{|\gamma|}{|\alpha|} < 1$. Note that $\beta,\ \gamma \leq 0$ and $\alpha > 0$.

Now, $\beta = -\frac{a_{i+\frac{1}{2}}}{h^2} + \frac{b_i^-}{h}$ and $\alpha = \frac{a_{i+\frac{1}{2}}+a_{i-\frac{1}{2}}}{h^2} + \frac{|b_i|}{h}$. So,

$$\frac{|\beta|}{\alpha} = \frac{\left|-\frac{a_{i+\frac{1}{2}}}{h^2} + \frac{b_i^-}{h}\right|}{\frac{a_{i+\frac{1}{2}}+a_{i-\frac{1}{2}}}{h^2} + \frac{|b_i|}{h}},$$

which is clearly less than 1. A similar argument shows that $\frac{|\gamma|}{\alpha} < 1$. We know that

$$D_j = \frac{1}{2}\begin{pmatrix} 3\alpha & -\beta & & \\ -\gamma & 3\alpha & -\beta & \\ & \ddots & \ddots & \ddots \\ & & -\gamma & 3\alpha \end{pmatrix}.$$

This can be written as

$$D_j = \frac{1}{2}\left(\begin{pmatrix} 3\alpha & & & \\ & 3\alpha & & \\ & & \ddots & \\ & & & 3\alpha \end{pmatrix} + \begin{pmatrix} 0 & -\beta & & \\ -\gamma & 0 & -\beta & \\ & \ddots & \ddots & \ddots \\ & & -\gamma & 0 \end{pmatrix}\right).$$

Therefore, we may write

$$D_j^{-1} = \frac{2}{3\alpha}\left(\begin{pmatrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{pmatrix} + \begin{pmatrix} 0 & -\frac{\beta}{3\alpha} & & \\ -\frac{\gamma}{3\alpha} & 0 & -\frac{\beta}{3\alpha} & \\ & \ddots & \ddots & \ddots \\ & & -\frac{\gamma}{3\alpha} & 0 \end{pmatrix}\right)^{-1}.$$

Since we have established that $\frac{|\beta|}{\alpha} < 1$ and $\frac{|\gamma|}{\alpha} < 1$, a power series expansion may be used to represent the inverse. Let

$$A = \begin{pmatrix} 0 & -\frac{\beta}{3\alpha} & & \\ -\frac{\gamma}{3\alpha} & 0 & -\frac{\beta}{3\alpha} & \\ & \ddots & \ddots & \ddots \\ & & -\frac{\gamma}{3\alpha} & 0 \end{pmatrix}. \tag{4.18}$$

Then, we have

$$\left\| D_j^{-1} - \frac{2}{3a} \left(I - A + A^2 + ... + (-1)^{n-1} A^{n-1} \right) \right\| \le C \left(\frac{2^{n-1}}{3^n} \right).$$

The above error estimate is a worst case estimate that comes easily from the fact that, at worst, we are multiplying two entries and adding them to the product of two more to obtain A^2, and the same holds for all other products, since A has only two nonzero terms in each column. In reality, the error is somewhat less.

The norm of A can be bounded:

$$\|A\|_\infty = \|A\|_1 = -\frac{\beta + \gamma}{3\alpha},$$

so that $0 < \|A\| = \rho < 1$, where

$$\rho = -\frac{\beta + \gamma}{3\alpha}. \tag{4.19}$$

The first appearance of the $(ij)th$ entry in the estimate for D_j^{-1} occurs in the $|j-i|th$ term in the power series approximation. Further appearances occur in alternating terms of the expansion. This is obvious from the structure of A. This means

$$\begin{aligned} |(D_j^{-1})_{ij}| &\le \|A\|^{|j-i|} + \|A\|^{|j-i|+2} + \ldots \\ &\le \rho^{|j-i|} + \rho^{|j-i|+2} + \ldots \\ &= \rho^{|j-i|}(1 + \rho^2 + \ldots) \\ &= \rho^{|j-i|} \frac{1}{1-\rho^2} \\ &= C\rho^{|j-i|}, \end{aligned}$$

where

$$C = (1 - \rho^2)^{-1}. \tag{4.20}$$

Thus, the estimate (4.17) holds, with C given by (4.20) and ρ given by (4.19). $\square$

4.3 The Two-Dimensional Case

Given the problem

$$L_j = F, \tag{4.21}$$

where L_j represents the operator on the fine grid obtained by discretizing a two-dimensional partial differential equation, we apply the wavelet transform to both sides of the equation. Denote $\boldsymbol{\mathcal{W}}_j$ by $\mathcal{W}_j$ for simplicity.

$$\begin{aligned} (\mathcal{W}_j L_j \mathcal{W}_j^T)\mathcal{W}_j U &= \mathcal{W}_j F \\ \Longrightarrow (\mathcal{W}_j L_j \mathcal{W}_j^T) \begin{pmatrix} U_L \\ U_H \end{pmatrix} &= \begin{pmatrix} F_L \\ F_H \end{pmatrix}, \end{aligned} \tag{4.22}$$

where $U_L, F_L \in \mathbf{V}_j$ and $U_H, F_H \in \mathbf{W}_j$. Note that for simplicity, we will also let H_j denote $\mathbf{H}_j$ and G_j denote $\mathbf{G}_j$ Now,

$$\begin{aligned} &\mathcal{W}_j L_j \mathcal{W}_j^T \\ = &\begin{pmatrix} H_j^{col} H_j^{row} \\ G_j^{col} H_j^{row} \\ H_j^{col} G_j^{row} \\ G_j^{col} G_j^{row} \end{pmatrix} L_j \begin{pmatrix} (H_j^{col} H_j^{row})^T & (G_j^{col} H_j^{row})^T & (H_j^{col} G_j^{row})^T & (G_j^{col} G_j^{row})^T \end{pmatrix} \\ = &\begin{pmatrix} H_j^{col} H_j^{row} L_j \\ G_j^{col} H_j^{row} L_j \\ H_j^{col} G_j^{row} L_j \\ G_j^{col} G_j^{row} L_j \end{pmatrix} \begin{pmatrix} (H_j^{col} H_j^{row})^T & (G_j^{col} H_j^{row})^T & (H_j^{col} G_j^{row})^T & (G_j^{col} G_j^{row})^T \end{pmatrix}. \end{aligned}$$

So, $\mathcal{W}_j L_j \mathcal{W}_j^T =$

$$\left(\begin{array}{ll} H_j^{col} H_j^{row} L_j (H_j^{col} H_j^{row})^T & H_j^{col} H_j^{row} L_j (G_j^{col} H_j^{row})^T \\ G_j^{col} H_j^{row} L_j (H_j^{col} H_j^{row})^T & G_j^{col} H_j^{row} L_j (G_j^{col} H_j^{row})^T \\ H_j^{col} G_j^{row} L_j (H_j^{col} H_j^{row})^T & H_j^{col} G_j^{row} L_j (G_j^{col} H_j^{row})^T \\ G_j^{col} G_j^{row} L_j (H_j^{col} H_j^{row})^T & G_j^{col} G_j^{row} L_j (G_j^{col} H_j^{row})^T \end{array}\right. \tag{4.23}$$

$$\left.\begin{array}{ll} H_j^{col} H_j^{row} L_j (H_j^{col} G_j^{row})^T & H_j^{col} H_j^{row} L_j (G_j^{col} G_j^{row})^T \\ G_j^{col} H_j^{row} L_j (H_j^{col} G_j^{row})^T & G_j^{col} H_j^{row} L_j (G_j^{col} G_j^{row})^T \\ H_j^{col} G_j^{row} L_j (H_j^{col} G_j^{row})^T & H_j^{col} G_j^{row} L_j (G_j^{col} G_j^{row})^T \\ G_j^{col} G_j^{row} L_j (H_j^{col} G_j^{row})^T & G_j^{col} G_j^{row} L_j (G_j^{col} G_j^{row})^T \end{array}\right).$$

We also observe that in two-dimensions, as well, application of the wavelet transform only requires $\mathcal{O}(n)$ operations. See [BCR91] for more details regarding the fast wavelet transform.

Now, define

$$\begin{aligned} T_j &= H_j^{col} H_j^{row} L_j (H_j^{col} H_j^{row})^T, \\ B_j^1 &= H_j^{col} H_j^{row} L_j (G_j^{col} H_j^{row})^T, \\ B_j^2 &= H_j^{col} H_j^{row} L_j (H_j^{col} G_j^{row})^T, \\ B_j^3 &= H_j^{col} H_j^{row} L_j (G_j^{col} G_j^{row})^T, \\ C_j^1 &= G_j^{col} H_j^{row} L_j (H_j^{col} H_j^{row})^T, \\ C_j^2 &= H_j^{col} G_j^{row} L_j (H_j^{col} H_j^{row})^T, \\ C_j^3 &= G_j^{col} G_j^{row} L_j (H_j^{col} H_j^{row})^T, \end{aligned}$$

$$
\begin{aligned}
D_j^1 &= G_j^{col} H_j^{row} L_j (G_j^{col} H_j^{row})^T, \\
D_j^2 &= G_j^{col} H_j^{row} L_j (H_j^{col} G_j^{row})^T, \\
D_j^3 &= G_j^{col} H_j^{row} L_j (G_j^{col} G_j^{row})^T, \\
D_j^4 &= H_j^{col} G_j^{row} L_j (G_j^{col} H_j^{row})^T, \\
D_j^5 &= H_j^{col} G_j^{row} L_j (H_j^{col} G_j^{row})^T, \\
D_j^6 &= H_j^{col} G_j^{row} L_j (G_j^{col} G_j^{row})^T, \\
D_j^7 &= G_j^{col} G_j^{row} L_j (G_j^{col} H_j^{row})^T, \\
D_j^8 &= G_j^{col} G_j^{row} L_j (H_j^{col} G_j^{row})^T, \\
D_j^9 &= G_j^{col} G_j^{row} L_j (G_j^{col} G_j^{row})^T.
\end{aligned}
$$

So,

$$
\begin{aligned}
T_j &: \mathbf{V}_j \to \mathbf{V}_j, \\
B_j^1 &: V_j \otimes W_j \to \mathbf{V}_j, \\
B_j^2 &: W_j \otimes V_j \to \mathbf{V}_j, \\
B_j^3 &: W_j \otimes W_j \to \mathbf{V}_j, \\
C_j^1 &: \mathbf{V}_j \to V_j \otimes W_j, \\
C_j^2 &: \mathbf{V}_j \to W_j \otimes V_j, \\
C_j^3 &: \mathbf{V}_j \to W_j \otimes W_j,
\end{aligned}
$$

$$
\begin{aligned}
D_j^1 &: V_j \otimes W_j \to V_j \otimes W_j, \\
D_j^2 &: W_j \otimes V_j \to V_j \otimes W_j, \\
D_j^3 &: W_j \otimes W_j \to V_j \otimes W_j, \\
D_j^4 &: V_j \otimes W_j \to W_j \otimes V_j, \\
D_j^5 &: W_j \otimes V_j \to W_j \otimes V_j, \\
D_j^6 &: W_j \otimes W_j \to W_j \otimes V_j, \\
D_j^7 &: \mathbf{V}_j \to W_j \otimes W_j, \\
D_j^8 &: W_j \otimes V_j \to W_j \otimes W_j, \\
D_j^9 &: W_j \otimes W_j \to W_j \otimes W_j.
\end{aligned}
$$

Equivalently, let

$$
\begin{aligned}
T_j &= T_j, \\
B_j &= \begin{pmatrix} B_j^1 & B_j^2 & B_j^3 \end{pmatrix}, \\
C_j &= \begin{pmatrix} C_j^1 \\ C_j^2 \\ C_j^3 \end{pmatrix}, \\
D_j &= \begin{pmatrix} D_j^1 & D_j^2 & D_j^3 \\ D_j^4 & D_j^5 & D_j^6 \\ D_j^7 & D_j^8 & D_j^9 \end{pmatrix}.
\end{aligned}
$$

So, $T_j : \mathbf{V}_j \to \mathbf{V}_j,\ B_j : \mathbf{W}_j \to \mathbf{V}_j,\ C_j : \mathbf{V}_j \to \mathbf{W}_j,$ and $D_j : \mathbf{W}_j \to \mathbf{W}_j$. Then,

$$
\mathcal{W}_j L_j \mathcal{W}_j^T = \begin{pmatrix} T_j & B_j \\ C_j & D_j \end{pmatrix}.
$$

Define $\tilde{L}_j$ as follows:

$$
\tilde{L}_j = \mathcal{W}_j L_j \mathcal{W}_j^T = \begin{pmatrix} T_j & B_j \\ C_j & D_j \end{pmatrix}. \tag{4.24}
$$

The block UDL decomposition of $\tilde{L}_j$, where U is block upper triangular with unit diagonal, D is block diagonal, and L is block lower triangular with unit diagonal, is then

$$\tilde{L}_j = \begin{pmatrix} I & B_j D_j^{-1} \\ 0 & I \end{pmatrix} \begin{pmatrix} T_j - B_j D_j^{-1} C_j & 0 \\ 0 & D_j \end{pmatrix} \begin{pmatrix} I & 0 \\ D_j^{-1} C_j & I \end{pmatrix}. \qquad (4.25)$$

We seek $\tilde{L}_j^{-1}$:

$$\tilde{L}_j^{-1} = (\mathcal{W}_j L_j \mathcal{W}_j^T)^{-1} = \mathcal{W}_j L_j^{-1} \mathcal{W}_j^T.$$

The inverse of the factorization of $\tilde{L}_j$ is

$$\begin{aligned}
&\begin{pmatrix} I & 0 \\ -D_j^{-1} C_j & I \end{pmatrix} \begin{pmatrix} (T_j - B_j D_j^{-1} C_j)^{-1} & 0 \\ 0 & D_j^{-1} \end{pmatrix} \begin{pmatrix} I & -B_j D_j^{-1} \\ 0 & I \end{pmatrix} \\
&= \begin{pmatrix} (T_j - B_j D_j^{-1} C_j)^{-1} & 0 \\ -D_j^{-1} C_j (T_j - B_j D_j^{-1} C_j)^{-1} & D_j^{-1} \end{pmatrix} \begin{pmatrix} I & -B_j D_j^{-1} \\ 0 & I \end{pmatrix} \\
&= \begin{pmatrix} (T_j - B_j D_j^{-1} C_j)^{-1} & -(T_j - B_j D_j^{-1} C_j)^{-1} B_j D_j^{-1} \\ -D_j^{-1} C_j (T_j - B_j D_j^{-1} C_j)^{-1} & D_j^{-1} C_j (T_j - B_j D_j^{-1} C_j)^{-1} B_j D_j^{-1} + D_j^{-1} \end{pmatrix}.
\end{aligned}$$

Now, we know that $U = L_j^{-1} F$

$$\Longrightarrow \mathcal{W}_j U = \mathcal{W}_j L_j^{-1} \mathcal{W}_j^T \mathcal{W}_j F = \tilde{L}_j^{-1} \mathcal{W}_j F.$$

So,

$$\begin{pmatrix} U_L \\ U_{LH} \\ U_{HL} \\ U_{HH} \end{pmatrix} = \tilde{L}_j^{-1} \begin{pmatrix} F_L \\ F_{LH} \\ F_{HL} \\ F_{HH} \end{pmatrix}, \qquad (4.26)$$

where $\tilde{L}_j^{-1}$ is (4.23), with L_j replaced by L_j^{-1}. Then, let

$$\begin{pmatrix} U_L \\ U_H \end{pmatrix} = \begin{pmatrix} U_L \\ U_{LH} \\ U_{HL} \\ U_{HH} \end{pmatrix}$$

and similarly for $\begin{pmatrix} F_L \\ F_H \end{pmatrix}$. Then, solving for $\begin{pmatrix} U_L \\ U_H \end{pmatrix}$, replacing $\tilde{L}_j^{-1}$ by the inverse of the factorization of $\tilde{L}_j$, we see that

$$\begin{pmatrix} U_L \\ U_H \end{pmatrix} = \tilde{L}_j^{-1} \begin{pmatrix} F_L \\ F_H \end{pmatrix}$$

$$= \begin{pmatrix} (T_j - B_j D_j^{-1} C_j)^{-1} & -(T_j - B_j D_j^{-1} C_j)^{-1} B_j D_j^{-1} \\ -D_j^{-1} C_j (T_j - B_j D_j^{-1} C_j)^{-1} & D_j^{-1} C_j (T_j - B_j D_j^{-1} C_j)^{-1} B_j D_j^{-1} + D_j^{-1} \end{pmatrix} * \begin{pmatrix} F_L \\ F_H \end{pmatrix}$$

$$= \begin{pmatrix} (T_j - B_j D_j^{-1} C_j)^{-1} & -(T_j - B_j D_j^{-1} C_j)^{-1} B_j D_j^{-1} \\ -D_j^{-1} C_j (T_j - B_j D_j^{-1} C_j)^{-1} & D_j^{-1} C_j (T_j - B_j D_j^{-1} C_j)^{-1} B_j D_j^{-1} + D_j^{-1} \end{pmatrix} * \begin{pmatrix} H_j \\ G_j \end{pmatrix} F$$

$$= \begin{pmatrix} (T_j - B_j D_j^{-1} C_j)^{-1} (H_j - B_j D_j^{-1} G_j) \\ -D_j^{-1} C_j (T_j - B_j D_j^{-1} C_j)^{-1} (H_j - B_j D_j^{-1} G_j) + D_j^{-1} G_j \end{pmatrix} F.$$

So,

$$U = \begin{pmatrix} H_j^T & G_j^T \end{pmatrix} \begin{pmatrix} (T_j - B_j D_j^{-1} C_j)^{-1}(H_j - B_j D_j^{-1} G_j) \\ -D_j^{-1} C_j (T_j - B_j D_j^{-1} C_j)^{-1}(H_j - B_j D_j^{-1} G_j) + D_j^{-1} G_j \end{pmatrix} F$$

$$= [H_j^T (T_j - B_j D_j^{-1} C_j)^{-1}(H_j - B_j D_j^{-1} G_j)$$
$$- G_j^T D_j^{-1} C_j (T_j - B_j D_j^{-1} C_j)^{-1}(H_j - B_j D_j^{-1} G_j) + G_j^T D_j^{-1} G_j] F$$
$$= [(H_j^T - G_j^T D_j^{-1} C_j)(T_j - B_j D_j^{-1} C_j)^{-1}(H_j - B_j D_j^{-1} G_j) + G_j^T D_j^{-1} G_j] F$$
$$= \begin{pmatrix} H_j^T - G_j^T D_j^{-1} C_j & G_j^T \end{pmatrix} \begin{pmatrix} (T_j - B_j D_j^{-1} C_j)^{-1} & 0 \\ 0 & D_j^{-1} \end{pmatrix} \begin{pmatrix} H_j - B_j D_j^{-1} G_j \\ G_j \end{pmatrix} F.$$

Denote

$$I_{2h}^h = \sqrt{2}(H_j^T - G_j^T D_j^{-1} C_j) \text{ and} \tag{4.27}$$
$$I_h^{2h} = \frac{\sqrt{2}}{2}(H_j - B_j D_j^{-1} G_j) \tag{4.28}$$

as our interpolation and restriction operators, respectively. Note that if the fine grid operator L_j is symmetric, then $C_j = B_j^T$ and $I_h^{2h} = \frac{1}{2}(I_{2h}^h)^T$. Using the interpolation and restriction operators defined in (4.27) and (4.28), we have

$$U = I_{2h}^h (T_j - B_j D_j^{-1} C_j)^{-1} I_h^{2h} F + G_j^T D_j^{-1} G_j F. \tag{4.29}$$

We also note that in multigrid, we are working on the residual equation, i.e.,

$$e = I_{2h}^h (T_j - B_j D_j^{-1} C_j)^{-1} I_h^{2h} r + G_j^T D_j^{-1} G_j r.$$

If we assume that $G_j^T D_j^{-1} G_j r$ is small, i.e., r is almost in $Range(H_j^T)$, then we can approximate the error by

$$e = I_{2h}^h (T_j - B_j D_j^{-1} C_j)^{-1} I_h^{2h} r.$$

So,

$$(T_j - B_j D_j^{-1} C_j) e_{2h} = I_h^{2h} r_h.$$

The above assumption is good for most of the classical iterative methods, like Jacobi and Gauss-Seidel. Therefore, our coarse grid operator is

$$L_{j+1} = T_j - B_j D_j^{-1} C_j, \tag{4.30}$$

which is the Schur complement of D_j in $\tilde{L}_j$.

Notice that this operator is the same as the one we obtain if we solve for U_L in Equation 4.26. Solving for U_L yields

$$\begin{aligned} U_L &= (T_j - B_j D_j^{-1} C_j)^{-1} F_L - (T_j - B_j D_j^{-1} C_j)^{-1} B_j D_j^{-1} F_H \\ &= (T_j - B_j D_j^{-1} C_j)^{-1} (F_L - B_j D_j^{-1} F_H). \end{aligned}$$

Again, if the fine grid operator is symmetric, then the coarse grid operator is $T_j - B_j D_j^{-1} B_j^T$. We will denote the multigrid method thus formed as the wavelet multigrid method. We will take a moment now to point out that although the wavelet operator is periodic, this method may be applied to any problem, even those that are not periodic.

4.4 Examples

We consider the following problem:

$$\begin{aligned} -\nabla \cdot (a(x,y) \nabla u(x,y)) &= 0, \text{ in } \Omega \\ u(x,y) &= 0, \text{ on } \partial\Omega, \end{aligned} \tag{4.31}$$

where Ω is the unit square. We discretize on both 16×16 grids, leading to a 256×256 fine grid operator, and 32×32 grids, leading to a 1024×1024 fine grid operator. First, let us consider the problem with $a(x,y) = 1 + 0.8 \sin(10\sqrt{2}\pi x)$. The Haar MRA is used to obtain the wavelet transform, and Gauss-Seidel is the iterative smoother. The homogenized version of multigrid involves finding the homogenized equations for (4.31). These are given by

$$\begin{aligned} -\mu\frac{\partial^2 u}{\partial x^2} - \overline{a}\frac{\partial^2 u}{\partial y^2} &= 0, \text{ in } \Omega \\ u(x,y) &= 0, \text{ on } \partial\Omega, \end{aligned} \tag{4.32}$$

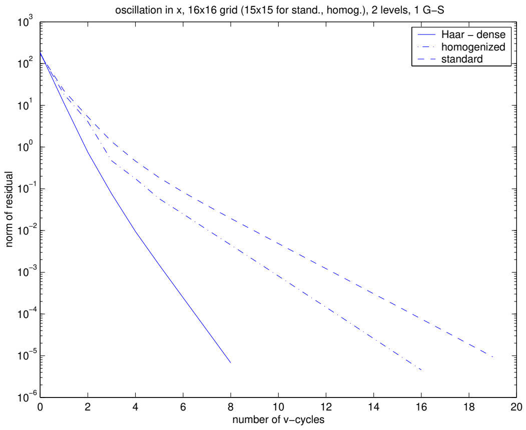
where

$$\mu = \left(\int_0^1 (1 + 0.8\sin(2\pi x))^{-1} dx\right)^{-1} \tag{4.33}$$

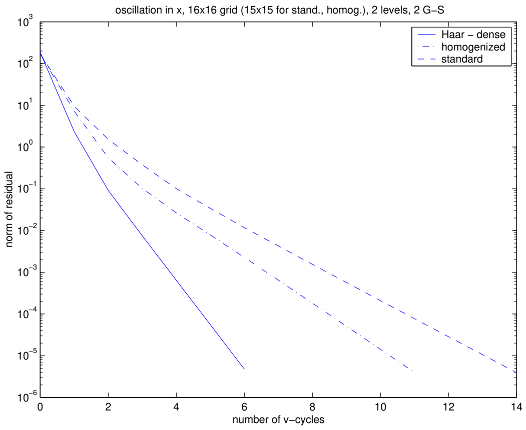
is the harmonic average and

$$\overline{a} = \int_0^1 (1 + 0.8\sin(2\pi x))dx \tag{4.34}$$

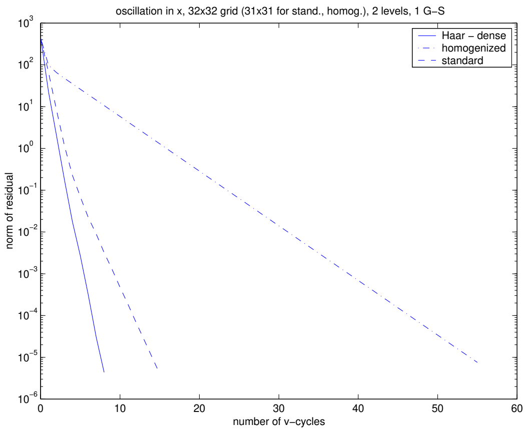
is the arithmetic average. Here, we have approximated μ numerically and calculated $\overline{a}$ exactly. We compare the convergence of the wavelet multigrid method (using Haar wavelets) with both the standard multigrid method and the method that uses (4.32) for the coarse grid operator with the standard interpolation and restriction operators. The initial solution in all cases is chosen to be $\mathbf{u} = \mathbf{1}$. The results (see Figures 4.1 and 4.2) show that the wavelet multigrid method is much more effective than standard multigrid (convergence to a residual with norm less than 10^{-5} is approximately twice as rapid). Also, the wavelet multigrid method has a convergence rate that is better than the standard homogenization approach. This result holds true regardless of the number of levels used or the number of Gauss-Seidel iterations, as well as regardless of the type of wavelet used.



(a)

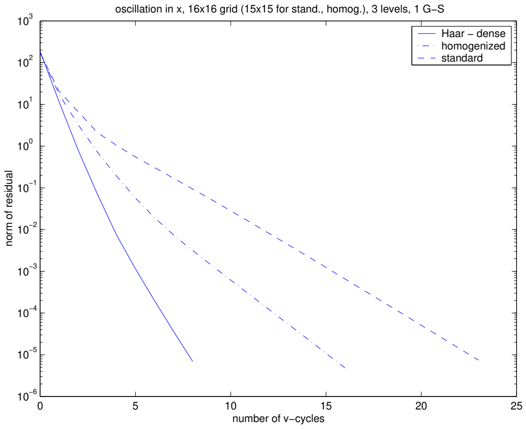


(b)

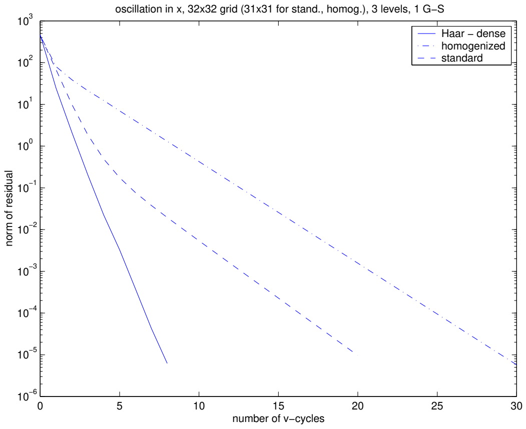


(c)

Figure 4.1: Comparison of results for oscillation in the x-direction, using Haar wavelets. (a), (b) show the results for a 16×16 grid, (a) using one Gauss-Seidel iteration and (b) using two. (c) shows the results for a 32×32 grid with one Gauss-Seidel iteration.



(a)



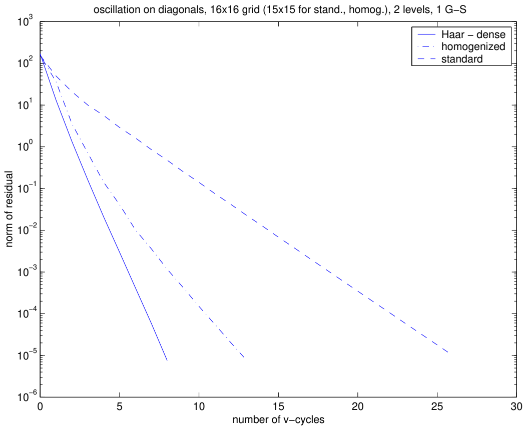
(b)

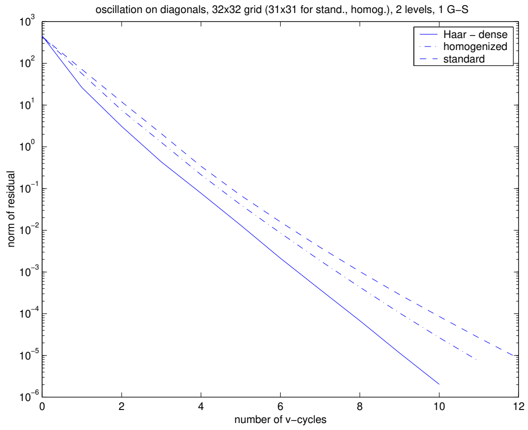
Figure 4.2: Comparison of results for oscillation in the x-direction, using Haar wavelets. Three levels and one Gauss-Seidel are used. (a) shows a 16×16 grid, (b) a 32×32 grid.

Next, consider $a(x,y) = 1+0.8\sin(10\sqrt{2}\pi(x-y))$. The homogenized equations for this problem take the form

$$\begin{aligned} -\frac{\mu+\overline{a}}{2}\frac{\partial^2 u}{\partial x^2} + (\overline{a}-\mu)\frac{\partial^2}{\partial x \partial y} - \frac{\mu+\overline{a}}{2}\frac{\partial^2 u}{\partial y^2} &= 0, \text{ in } \Omega \\ u(x,y) &= 0, \text{ on } \partial\Omega, \end{aligned} \tag{4.35}$$

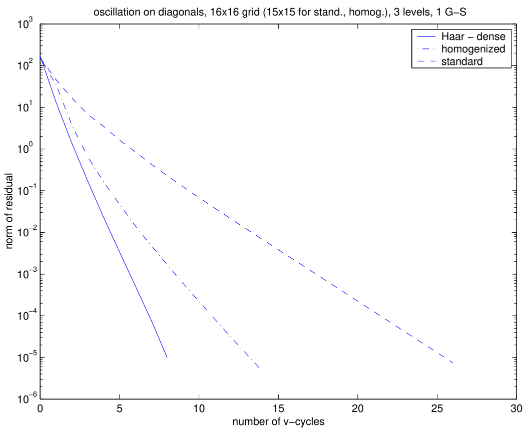
where μ and $\overline{a}$ are defined by (4.33) and (4.34). The wavelet multigrid method has a convergence rate that is comparable to the standard homogenization approach and is better than the standard multigrid. Figures 4.3 and 4.4 demonstrate that this result holds true regardless of the number of levels used or the number of Gauss-Seidel iterations.

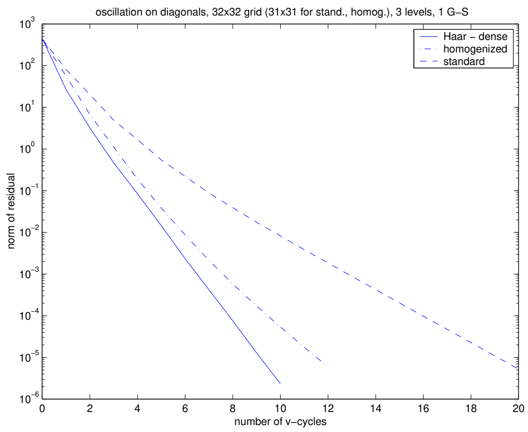




(a) (b)

Figure 4.3: Comparison of results for oscillation along diagonals, using Haar wavelets. (a) shows a 16×16 grid, (b) a 32×32 grid.

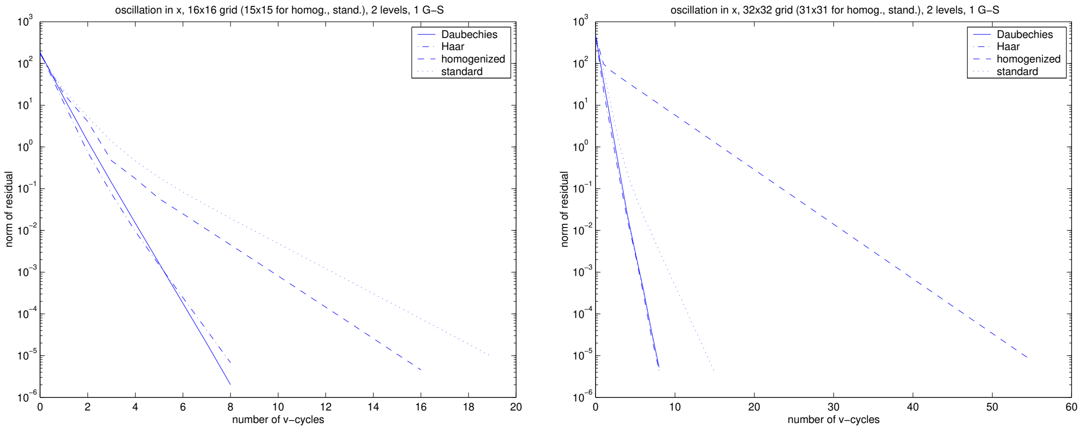




(a) (b)

Figure 4.4: Comparison of results for oscillation along diagonals, using Haar wavelets. Three levels and one Gauss-Seidel are used. (a) shows a 16×16 grid, (b) a 32×32 grid.

We also want to examine the results using wavelets that are more regular. This is because the application of such wavelet operators should give better compression results. We examine this more in later chapters. Comparison with the wavelet multigrid method using Daubechies wavelets in both of the problems discussed above yields the same good results (see Figures 4.5 - 4.7). Here, Daubechies 4 wavelets are used, i.e., with wavelet coefficients $c_0 = \frac{1+\sqrt{3}}{4\sqrt{2}}, c_1 = \frac{3+\sqrt{3}}{4\sqrt{2}}, c_2 = \frac{1-\sqrt{3}}{4\sqrt{2}}$, and $c_3 = \frac{3-\sqrt{3}}{4\sqrt{2}}$.



(a) (b)

Figure 4.5: Comparison of results for oscillation in the x-direction, using Daubechies wavelets. (a) shows a 16×16 grid, (b) a 32×32 grid.

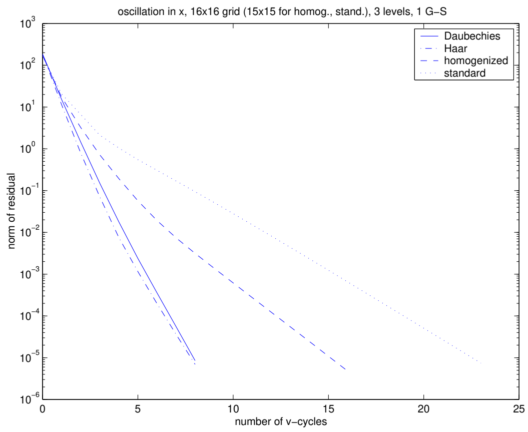
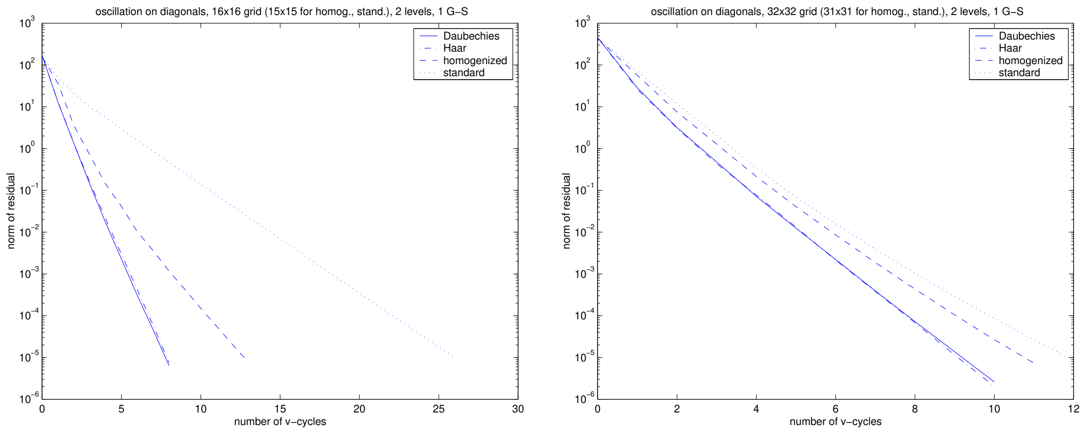


Figure 4.6: Comparison of results for oscillation in the x-direction, using Daubechies wavelets on a 16×16 grid. Three levels and one Gauss-Seidel are used.



(a) (b)

Figure 4.7: Comparison of results for oscillation along diagonals, using Daubechies wavelets. (a) shows a 16×16 grid, (b) a 32×32 grid.

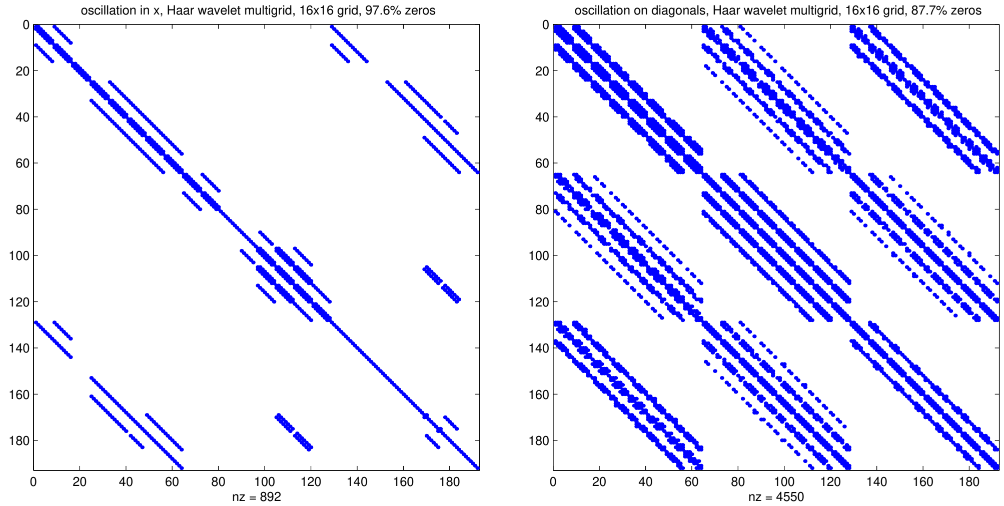
CHAPTER 5

Investigation into Sparsity Patterns of D_j^{-1} Via Thresholding

The main goal, now, is to make this procedure more efficient, so as to be practically useful. Now, although D_j is not dense (it is, in fact, a banded matrix), its inverse is dense due to fill-in. But, we observe that there is a great deal of decay of the values on certain diagonals. This leads us to believe that we may be able to increase the efficiency of the calculation in this area. This, then, would improve the efficiency of the overall algorithm, as this is the only area where the calculation and density of the matrix results in extra computational cost.

The first step in this investigation is to determine how sparse we can make D_j^{-1} after computing the inverse exactly. Also, we want to determine whether or not a banding pattern exists. To do this, we use a simple thresholding approach – all entries $D_j^{-1}(i,j)$ such that $|D_j^{-1}(i,j)| \leq \epsilon,\ \epsilon > 0$, are dropped after exact computation of D_j^{-1}. Then, the algorithm is allowed to proceed normally. The results of testing this approach on the problem with the coefficient function oscillating in the x-direction and the problem with the coefficient function oscillating along diagonals follow.

The sparsity pattern of D_j^{-1} at the optimum threshold to maintain the convergence properties of the method is shown in Figure 5.1 for the Haar wavelet multigrid method. We show both oscillation in x and oscillation along diagonals.



(a) (b)

Figure 5.1: Nonzero structure of D_j^{-1} after thresholding for (a) oscillation in the x-direction and (b) oscillation on diagonals.

Also pictured in Figure 5.2 is the convergence history for different threshold values.

For Daubechies wavelets, the nonzero structure for the same cases as before appears in Figure 5.3. The convergence history for different threshold values is shown in Figure 5.4

The observed results are quite good. Good convergence is observed for a sparsity level of over 80% zeros (and even greater than 90% zeros for oscillation in the x-direction or for a 32×32 grid with either type of oscillation). Convergence is almost exactly the same, or very close to, the convergence from using the full inverse, while at the same time remaining much better than either the standard or homogenized methods. We notice that the resulting structure is for the most part a banded structure.

These encouraging results prompt us to try a simple banding scheme. The

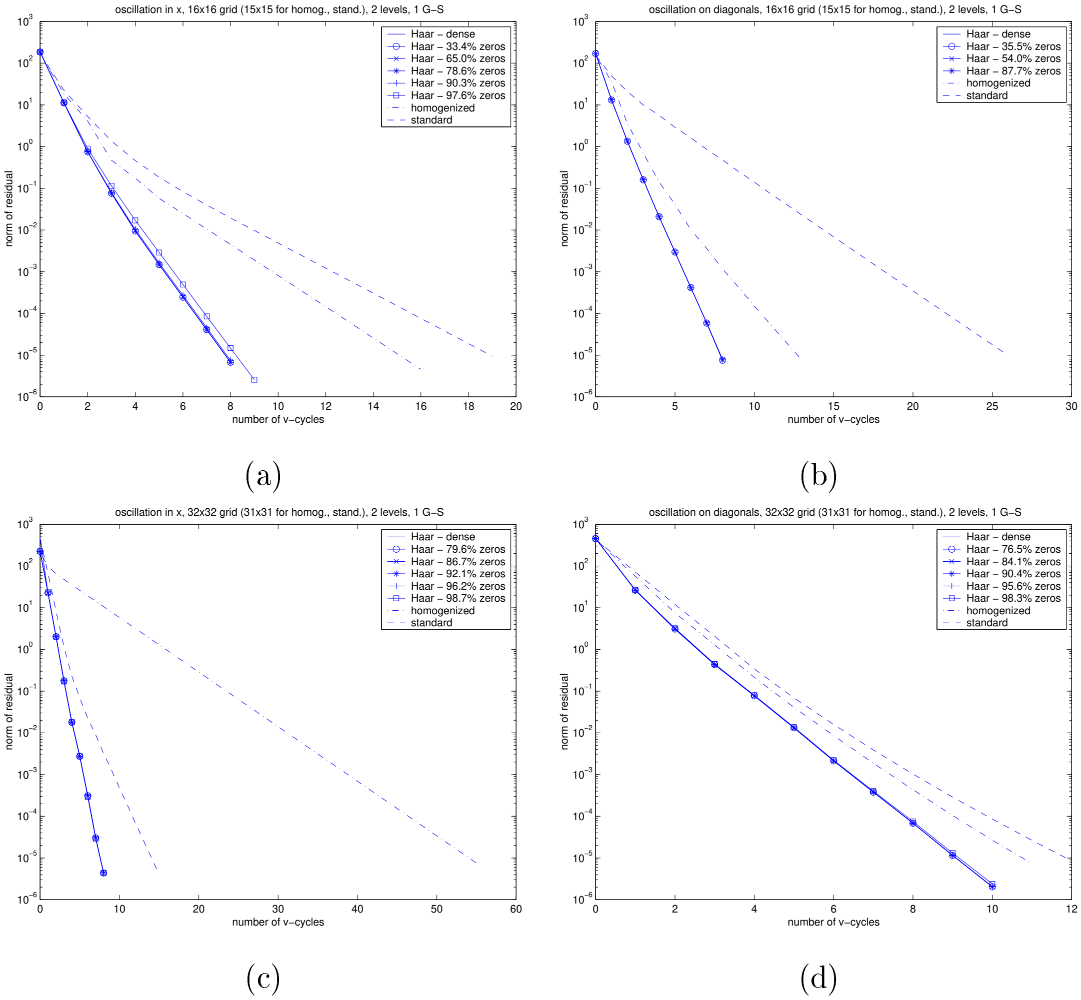
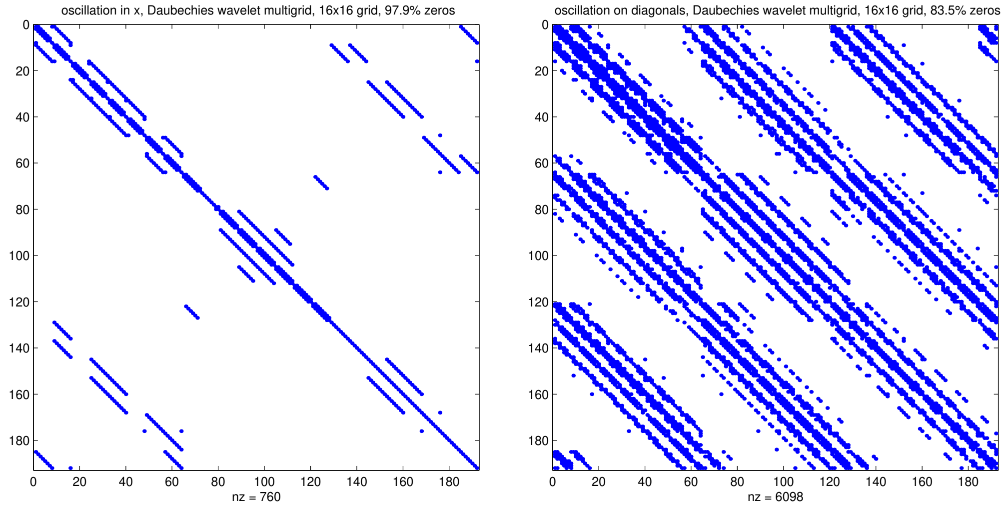


Figure 5.2: Using Haar wavelets. (a) shows oscillation in x, 16×16 grid; (b) shows oscillation on diagonals, 16×16 grid; (c) shows oscillation in x, 32×32 grid; and (d) shows oscillation on diagonals, 32×32 grid.



(a) (b)

Figure 5.3: Nonzero structure of D_j^{-1} after thresholding for (a) oscillation in the x-direction and (b) oscillation on diagonals.

procedure is as follows: for the 16×16 grid (i.e., a fine grid operator of size 256×256), we choose the sparsest matrix D_j^{-1} that gives results that are optimally comparable to those using the full matrix. Then, we determine a banding structure that approximates this and code the algorithm based upon the observed banding. Again, the results are good (see Figure 5.5). The convergence, as can be seen, is quite close (with respect to oscillation along diagonals, it is almost identical).

In regards to the above discussion, both the Haar and Daubechies 4 wavelets give very similar results. However, when considering the simple banding scheme, the results for the Daubechies 4 wavelets for the case of oscillation in the x-direction are superior – a higher degree of sparsity is achieved without convergence suffering. So, with respect to compression, Daubechies 4 wavelets prove superior in this case. For the case of oscillation along diagonals, the Haar wavelets yield a higher degree of sparsity than the Daubechies 4 wavelets.

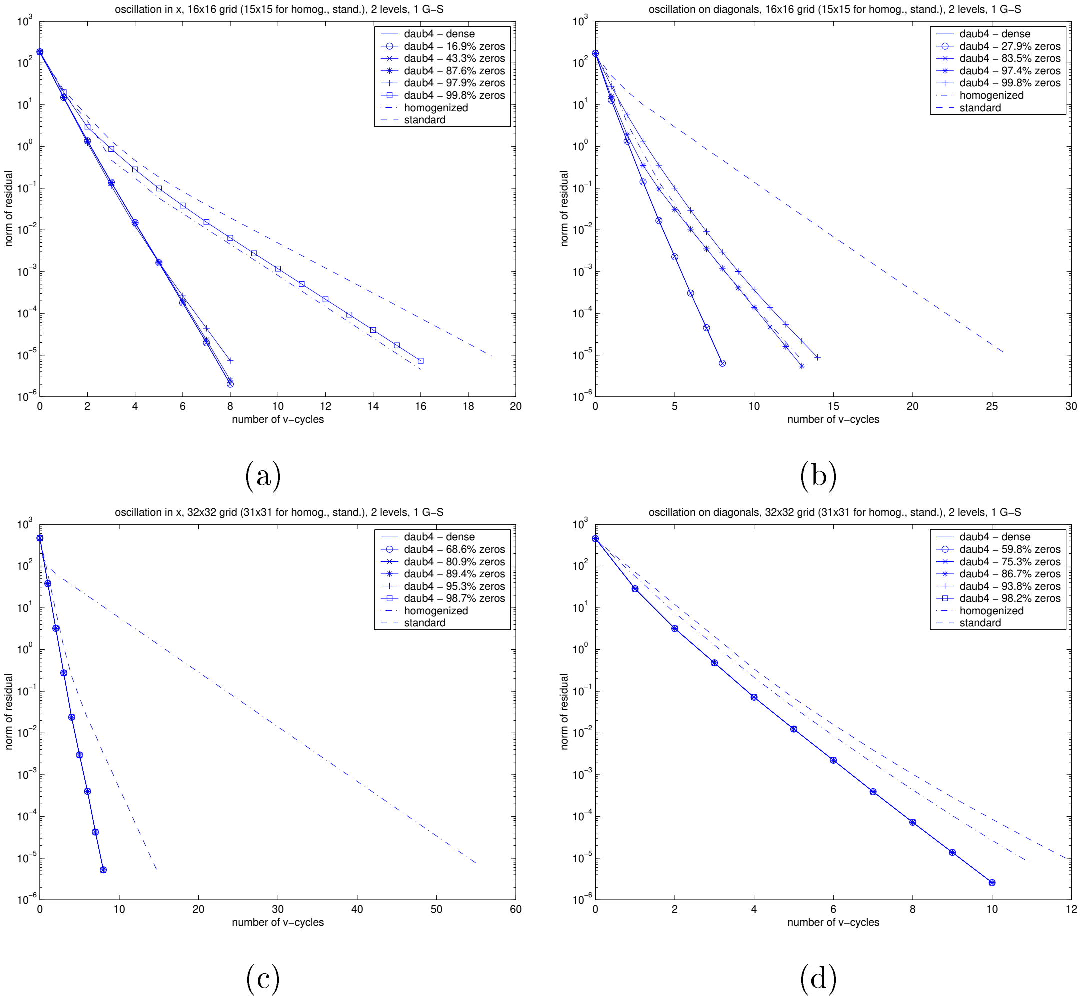


Figure 5.4: Using Daubechies 4 wavelets. (a) shows oscillation in x, 16×16 grid; (b) shows oscillation on diagonals, 16×16 grid; (c) shows oscillation in x, 32×32 grid; and (d) shows oscillation on diagonals, 32×32 grid.

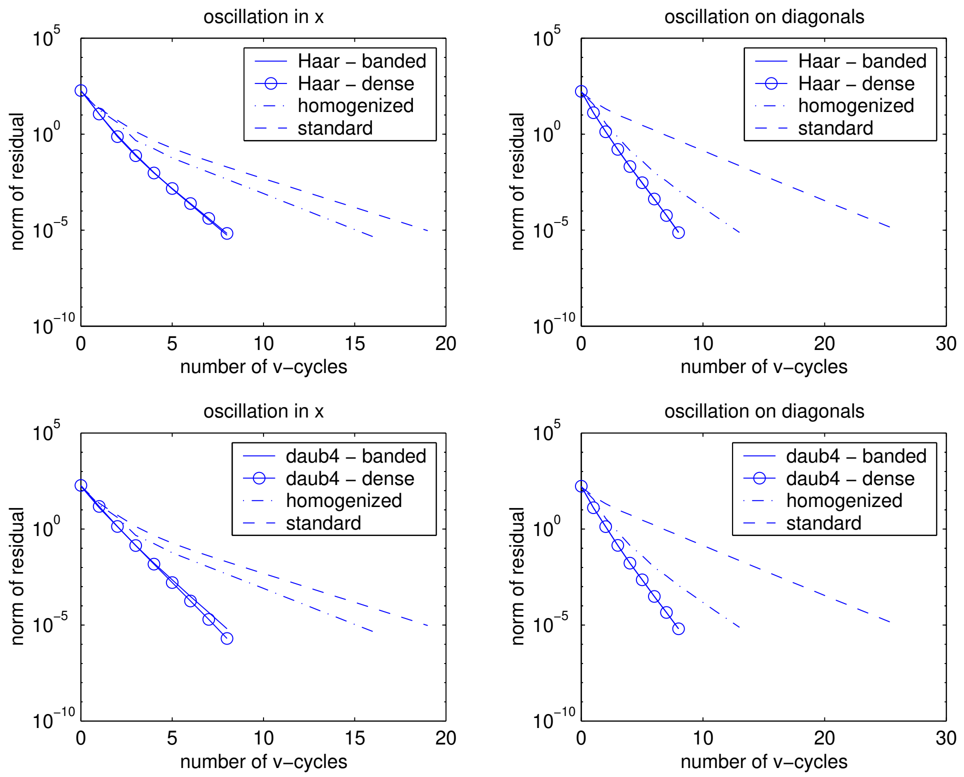
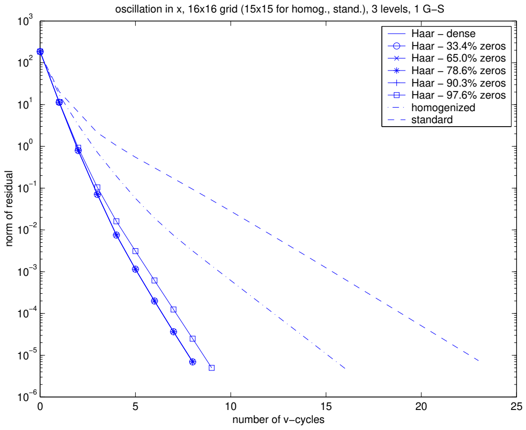


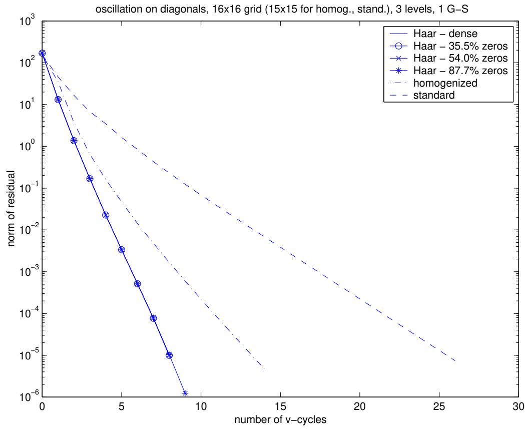
Figure 5.5: Results of the banding procedure compared to the original algorithm and standard and homogenized methods, 16×16 grid, two-level V-cycle, one Gauss-Seidel.

Now, since we achieve such good results with the two-level method, we want to determine if this carries through to V-cycles with more levels. The results are again positive. The above comments for the two-level method carry through to the three-level method. In Figure 5.6 we display the results for a 16×16 grid. Here, we have again applied the method of discarding entries of D_j^{-1} based on their magnitude, i.e., thresholding. As the figure demonstrates, convergence does not suffer from the thresholding procedure.

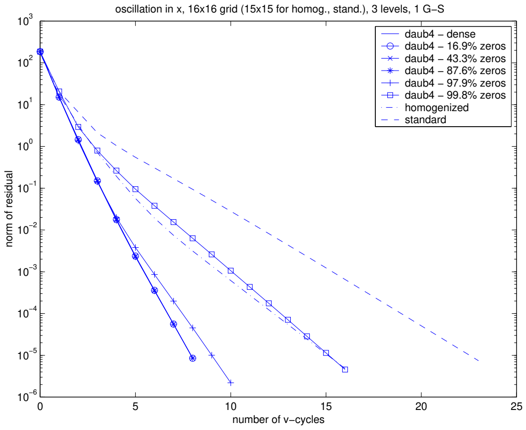
In practice, we simply assume that the matrix has a banded structure and apply an incomplete LU factorization to calculate the inverse.



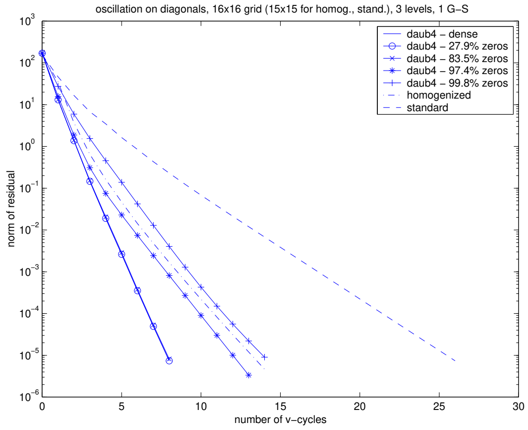
(a)



(b)



(c)



(d)

Figure 5.6: Using three-level V-cycle with one Gauss-Seidel on a 16×16 grid, the performance of Haar and Daubechies 4 wavelets with compression. All percentage zeros values represent number of zeros at the second level. (a) shows oscillation in x, Haar wavelets; (b) shows oscillation along diagonals, Haar wavelets; (c) shows oscillation in x, Daubechies 4 wavelets; and (d) shows oscillation along diagonals, Daubechies 4 wavelets.

CHAPTER 6

Improving Efficiency of the Wavelet Multigrid Method

6.1 Using ILU(0) to Improve Efficiency

After the encouraging results achieved by using the method of dropping entries via thresholding and by using banding to achieve sparsity in D_j^{-1}, we are prepared to aim for greater efficiency. The first goal is to avoid computing the inverse exactly. Instead, we propose using ILU(0) to perform an incomplete LU-factorization and then computing the inverse via a series of forward and backward substitutions.

First, a little background on incomplete ILU methods, in general, and ILU(0) specifically. Basically, we can derive an algorithm for incomplete LU-factorization (ILU) from the algorithm for Gaussian elimination by dropping elements in off-diagonal locations via some prescribed conditions. The determination of which elements to discard is governed by the amount of fill-in we plan to allow. If we will allow no fill-in (i.e., the LU decomposition of the matrix will have exactly the same structure as the original matrix), we are performing ILU(0), where the 0 indicates we are allowing no fill-in. It can be shown that for M-matrices ILU produces an incomplete factorization $A = LU - R$ of A. (Recall that A is a M-matrix if $a_{ii} \geq 0$ for $i = 1, ..., n$; $a_{ij} \leq 0$ for $i \neq j,\ i, j = 1, \ldots, n$; A nonsingular; and $A^{-1} \geq 0$. Alternately, one can see that, given A such that $a_{ii} \geq 0$ for $i = 1, \ldots, n$ and $a_{ij} \leq$

0 for $i \neq j,\ i,j = 1,\ldots,n$, then A is an M-matrix if and only if $\rho(I - D^{-1}A) \leq 1$, where $D = diag(a_{ii}),\ i = 1,\ldots,n$.) See [Saa96] for a more detailed description of ILU methods.

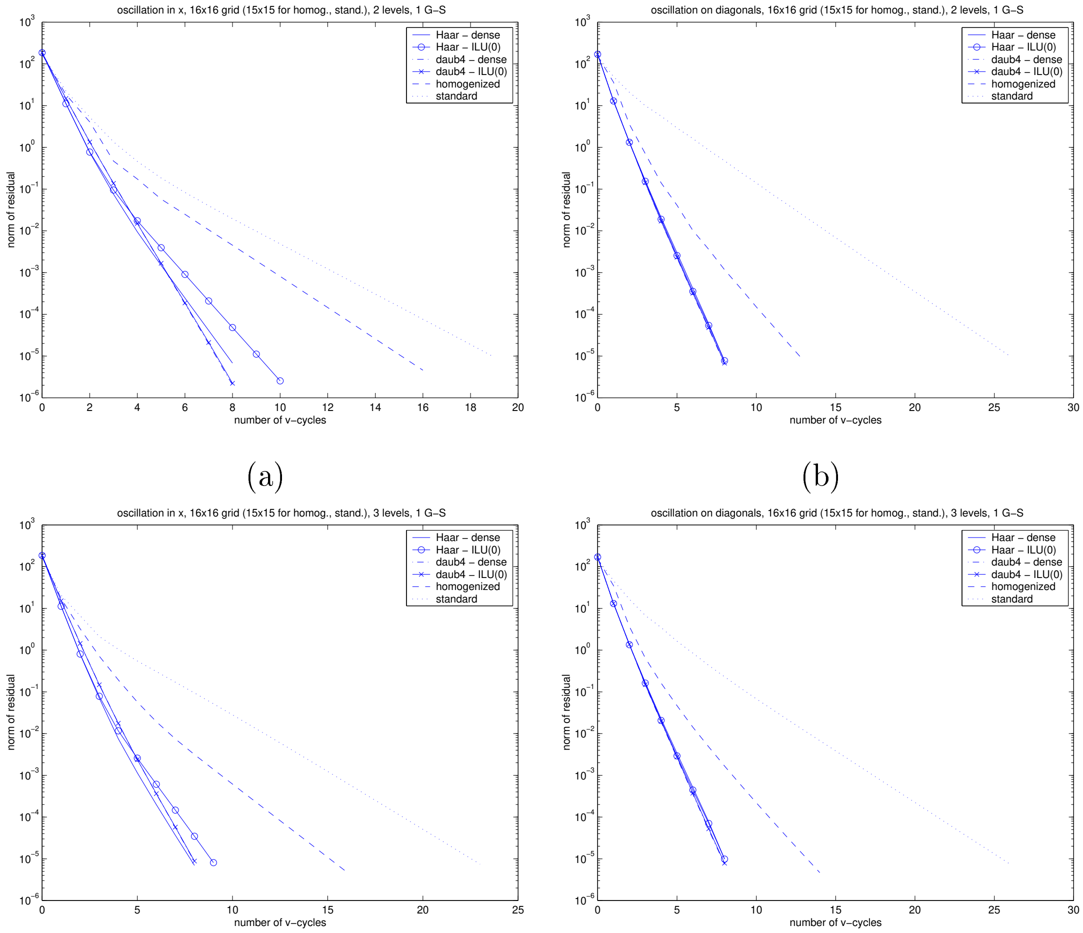
The algorithm for ILU(0) is as follows:

Let Z be the set of nonzero elements of A.

```
for i = 2,...,n do
    for k = 1,...,i-1 and for a_ik ∈ Z do
        a_ik = a_ik / a_kk
        for j = k+1,...,n and for a_ij ∈ Z do
            a_ij = a_ij - a_ik a_kj
        end do
    end do
end do
```

So, if the above ILU(0) algorithm works, we have indeed shown that the wavelet multigrid algorithm can be performed more efficiently. Figure 6.1 demonstrates that, for both the elliptic problem with oscillation in the x-direction and the elliptic problem with oscillation along diagonals, this method outperforms both the standard and homogenized multigrid methods, while having a convergence rate that is admirably close to that of the dense wavelet multigrid method. In this case, Daubechies 4 wavelets seem to suffer less decay in convergence rate than do the Haar wavelets. This, however, is to be expected due to the higher regularity of the Daubechies wavelets.

One thing to observe is that, although using ILU(0) reduces the computational complexity of calculating D_j^{-1} without hurting the convergence of the method, the resulting inverse is still a dense matrix. In practice, we would like to have a sparse matrix so that the coarse grid matrix is sparse.



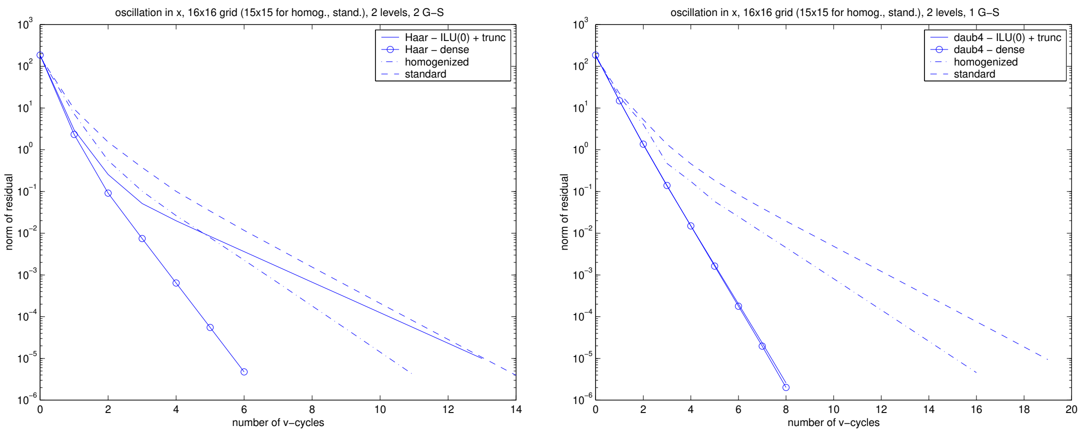
(a) (b) (c) (d)

Figure 6.1: Comparison of the performance of Daubechies 4 wavelets and Haar wavelets using ILU(0) in the wavelet multigrid method on a 16×16 grid, one Gauss-Seidel. (a) shows oscillation in x, 2 levels; (b) oscillation on diagonals, 2 levels; (c) oscillation in x, 3 levels; and (d) oscillation on diagonals, 3 levels.

6.2 Using ILU(0) Plus Truncation to Improve Efficiency

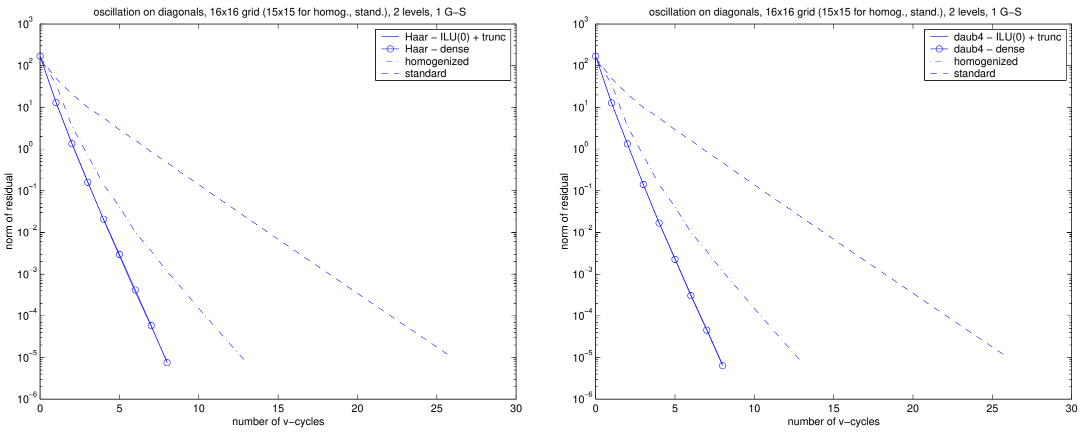
Now, we have increased the efficiency in the calculation of D_j^{-1}, but D_j^{-1} is still dense. This results in a formula for the coarse grid operator (as well as the restriction and interpolation operators) that is much denser than the fine grid operator. To avoid this, we use a very simple procedure – any values calculated in the inverse in locations that are zero in D_j are set to zero. Thus, we eliminate any fill-in. Happily, this procedure generally results in convergence rates that are very close to those of the original dense wavelet multigrid method. We will call this method the truncated wavelet multigrid method, for brevity, and we will refer to the original method as the dense or full wavelet multigrid method. The term wavelet multigrid method will refer to a generality applying to both versions.

This method is tested with a variety of problems, among them the elliptic problem with Dirichlet boundary conditions (4.31), using different functions for a. For the problem with coefficient a oscillating in the x-direction, even using two Gauss-Seidel sweeps for the truncated method with Haar wavelets on a 16×16 grid does not ensure convergence close to the homogenized method. Using ILU(0) followed by truncation to compute the inverse component of the coarse grid operator obviously degrades the convergence of the Haar wavelet multigrid method for such a coarse grid size. With Daubechies wavelets, however, the convergence with the truncated method is as good as that employing the full wavelet multigrid method. This is clear from Figure 6.2, which demonstrates the convergence of the wavelet multigrid method using both Haar and Daubechies wavelets. For the elliptic problem where a has oscillation along diagonals, however, all things are equal, as demonstrated in Figure 6.3, which again displays the convergence history of the wavelet multigrid method using both Haar and Daubechies wavelets.



(a) (b)

Figure 6.2: Comparison of truncated wavelet multigrid with dense version on a $16{\times}16$ grid for oscillation in x. Haar wavelets and two Gauss-Seidel are used in (a), Daubechies wavelets and one Gauss-Seidel in (b).



(a) (b)

Figure 6.3: Comparison of truncated wavelet multigrid with dense version on a $16{\times}16$ grid for oscillation along diagonals. Haar wavelets are used in (a), Daubechies wavelets in (b).

We will briefly discuss the complexity of the coarse grid operator for the truncated wavelet multigrid method. After calculating D_j^{-1} using the method of ILU(0) followed by truncation, for most of the examples that are discussed in Chapter 7, we find that the inverse is representative of a stencil that contains between seven and twenty-three elements. This leads to a coarse-grid operator $T_j - B_j D_j^{-1} C_j$ that corresponds to a stencil having between twenty-one and twenty-five elements. We observe, however, that the matrix T_j has the same structure as the fine grid operator, i.e., it corresponds to a stencil with five elements. The additional elements in the stencil, then, come solely from the product $B_j D_j^{-1} C_j$.

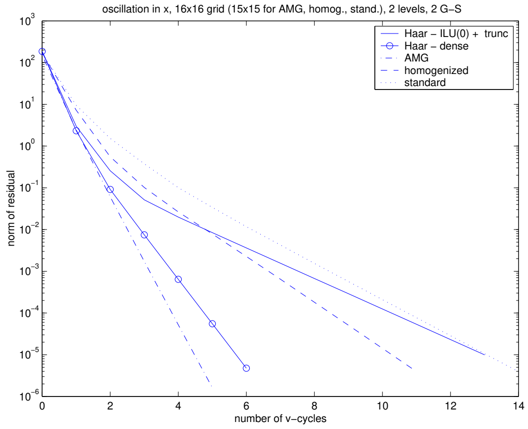
CHAPTER 7

Applications

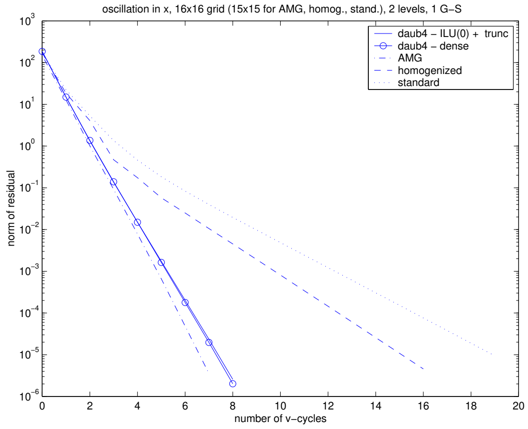
In this chapter, we display the results of applying the truncated wavelet multigrid method (the more efficient form, employing ILU(0) and truncation to obtain the inverse component). We compare the convergence with the full wavelet multigrid method, the algebraic multigrid method (AMG), the homogenization method (where appropriate), and the standard multigrid method. For all problems, unless otherwise specified, numerical results are analyzed using both a 16×16 grid, leading to a 256×256 fine grid operator, and a 32×32 grid, leading to a 1024×1024 fine grid operator.

7.1 Elliptic Problems from Previous Chapters Revisited

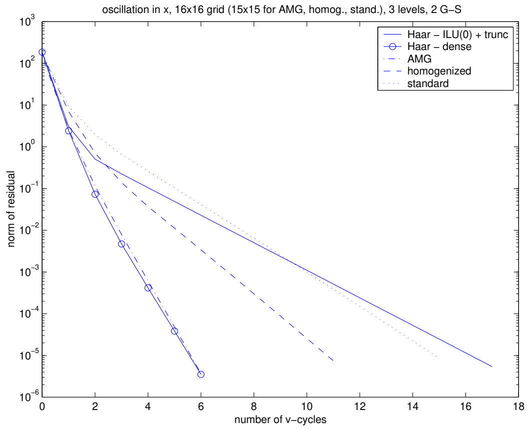
First, we look at the elliptic problem (4.31) with oscillatory coefficients. Here, we see that the full wavelet multigrid method actually has a convergence rate that is comparable to the algebraic multigrid method in both the case of oscillation in the x-direction and the case of oscillation along diagonals. Figures 7.1 through 7.4 demonstrate the results of these comparisons. In the case of oscillation in the x-direction, as mentioned in Chapter 6, ILU(0) coupled with truncation causes the Haar wavelet multigrid method applied on a 16×16 grid to have a convergence rate that is poor. In fact, it is much worse than the algebraic multigrid method. Using Daubechies wavelets, however, we see a convergence rate that is very close to that of algebraic multigrid. For a 32×32 grid, both wavelets yield approximately the same convergence rate as algebraic multigrid. For the case of oscillation along diagonals, using either type of wavelets produces convergence rates that are almost identical to those achieved by the algebraic multigrid method, regardless of the mesh size of the fine grid.



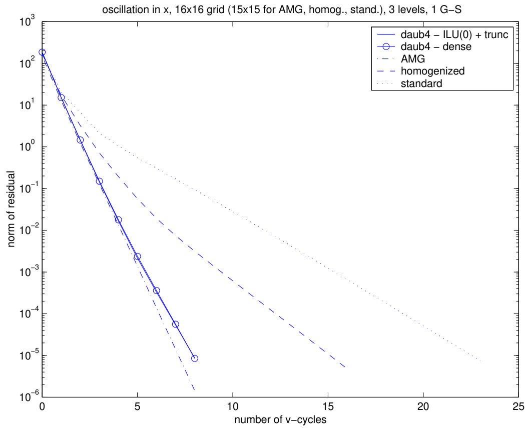
(a)



(b)

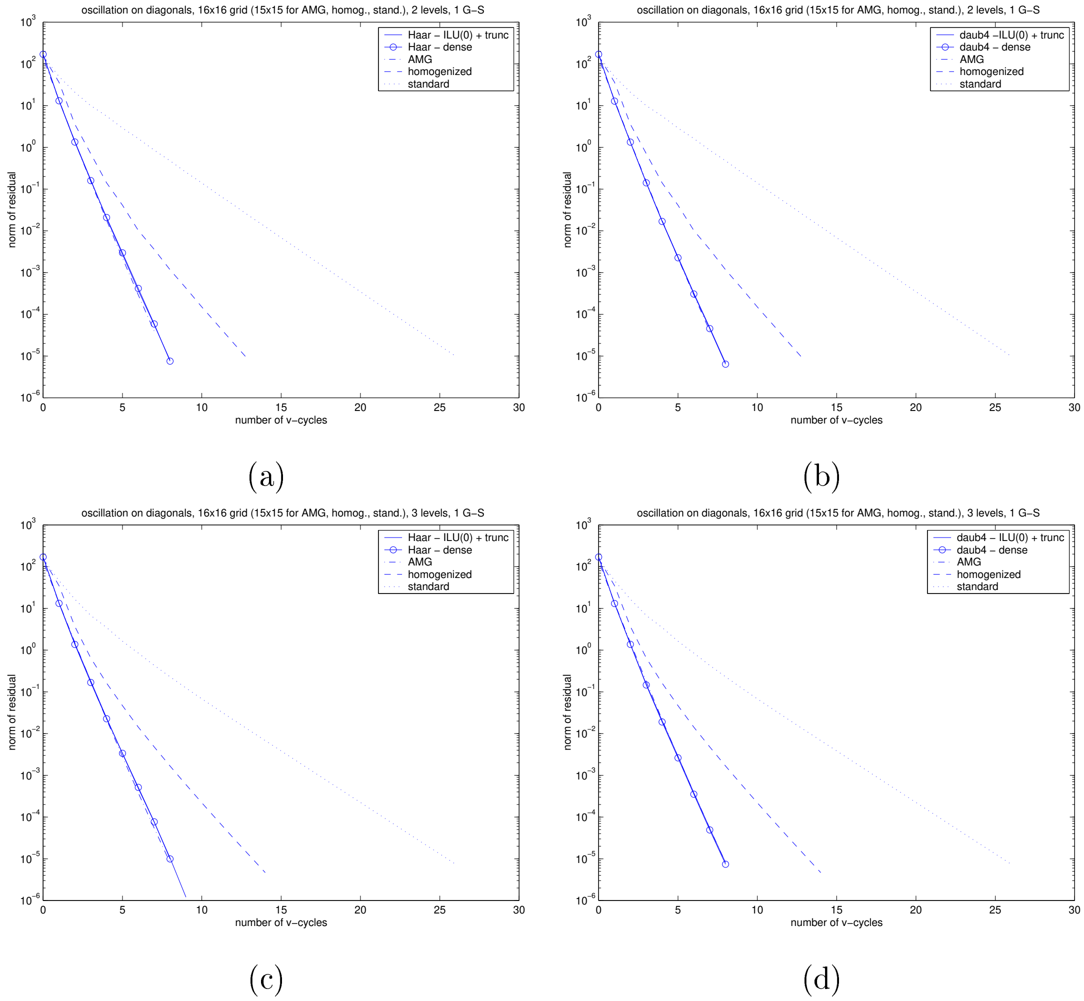


(c)



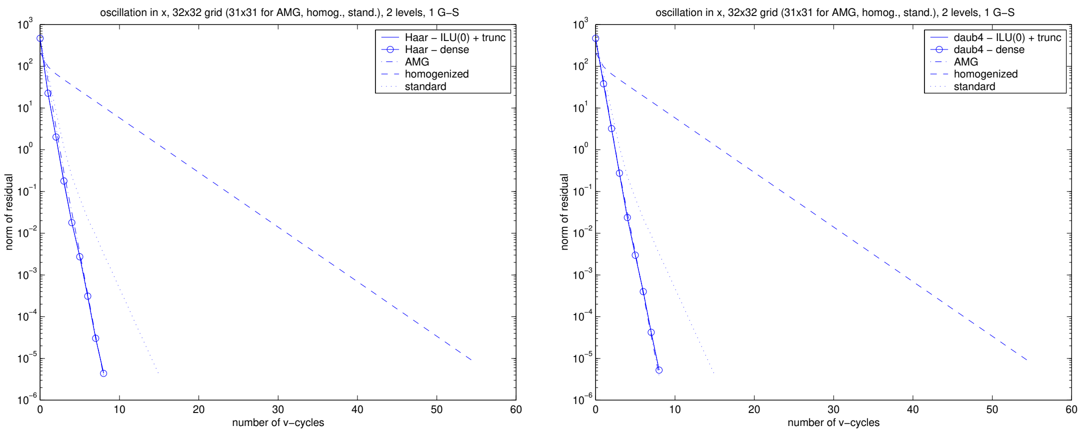
(d)

Figure 7.1: Oscillation in x-direction, 16×16 grid. Compare Haar and Daubechies wavelet multigrid with AMG, homogenized, and standard methods. (a) uses Haar wavelets, 2 levels, 2 Gauss-Seidel; (b) uses Daubechies wavelets, 2 levels, 1 Gauss-Seidel; (c) uses Haar wavelets, 3 levels, 2 Gauss-Seidel; and (d) uses Daubechies wavelets, 3 levels, 1 Gauss-Seidel.



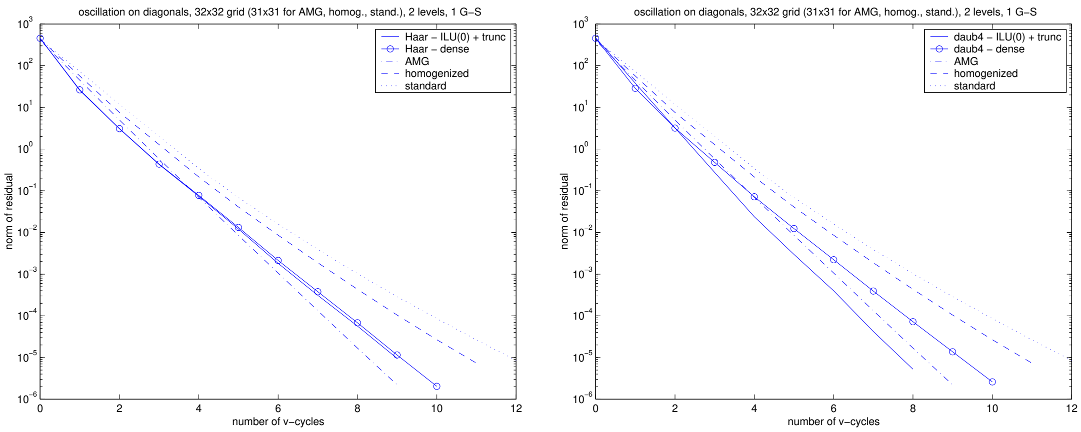
(a) (b) (c) (d)

Figure 7.2: Oscillation on diagonals, 16×16 grid. Compare Haar and Daubechies wavelet multigrid with AMG, homogenized, and standard methods. (a) uses Haar wavelets, 2 levels; (b) uses Daubechies wavelets, 2 levels; (c) uses Haar wavelets, 3 levels; and (d) uses Daubechies wavelets, 3 levels.



(a) (b)

Figure 7.3: Oscillation in x-direction, 32×32 grid. Compare Haar and Daubechies wavelet multigrid with AMG, homogenized, and standard methods. (a) uses Haar wavelets and (b) uses Daubechies wavelets.



(a) (b)

Figure 7.4: Oscillation on diagonals, 32×32 grid. Compare Haar and Daubechies wavelet multigrid with AMG, homogenized, and standard methods. (a) uses Haar wavelets and (b) uses Daubechies wavelets.

7.2 Other Elliptic Problems

For the elliptic problem (4.31) with a discontinuous coefficient, we see similar results. Here, we define the coefficient a as

$$a = \begin{cases} 10^5 & \text{if } 0.3 < x,\ y < 0.7, \\ 1 & \text{otherwise.} \end{cases}$$

In this case, the fine grid must contain sufficiently many grid points. A 48×48 grid ($h = \frac{1}{49}$) seems to be sufficiently fine to obtain good results. Note that this yields a 2304×2304 fine grid operator. The convergence rate of the truncated wavelet multigrid method is approximately the same as that obtained by using the algebraic multigrid method.

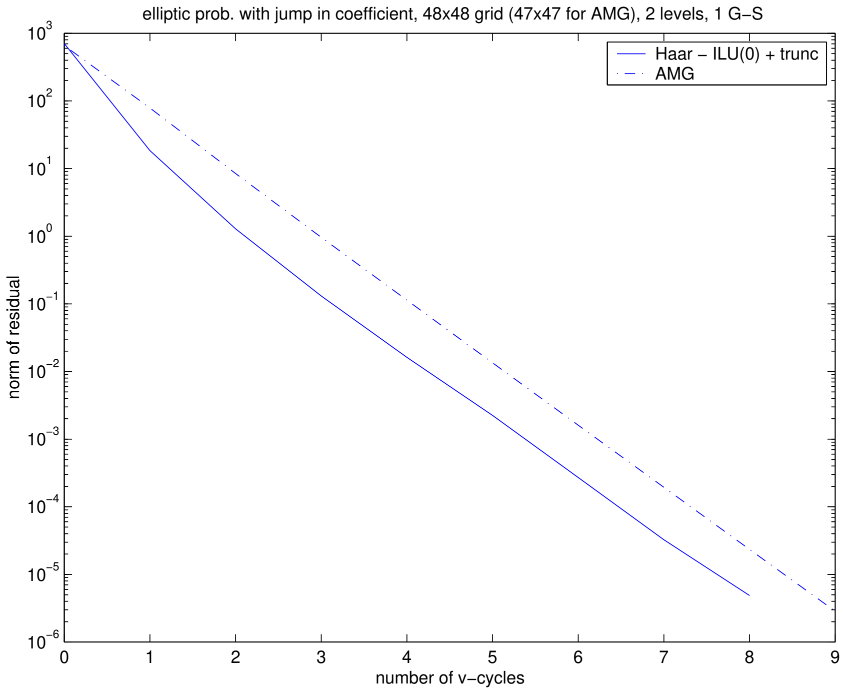


Figure 7.5: Problem with jump, 48x48 grid. Compare Haar wavelet multigrid with AMG.

Next, we look at the checkerboard problem, which is defined by (4.31) with

$$a = \begin{cases} 10^5 & \text{if } 0 < x, y < 0.5 \text{ or } 0.5 < x, y < 1, \\ 1 & \text{otherwise.} \end{cases} \tag{7.1}$$

The results for the checkerboard problem are quite good. The full Haar wavelet multigrid method has a convergence rate that is as good or better than the algebraic multigrid method. The truncated wavelet multigrid method also performs as well or better than the algebraic multigrid, except in the case where two iterations of Gauss-Seidel are used. In that case, the convergence rate of the algebraic multigrid method is slightly better. Both methods have a far better convergence rate than standard multigrid, which diverges for this problem. The results are shown in Figures 7.7 and 7.8

Having considered the simple 2 by 2 checkerboard problem, we expand our efforts. Now, we try to solve the 4 by 4 checkerboard problem using our wavelet multigrid method. For this problem, a is given in the figure below.

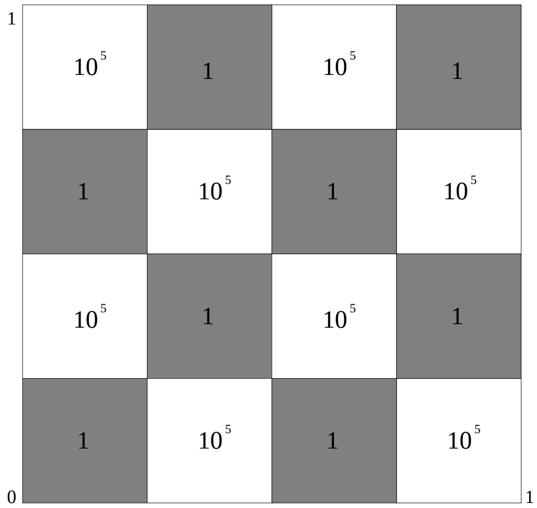
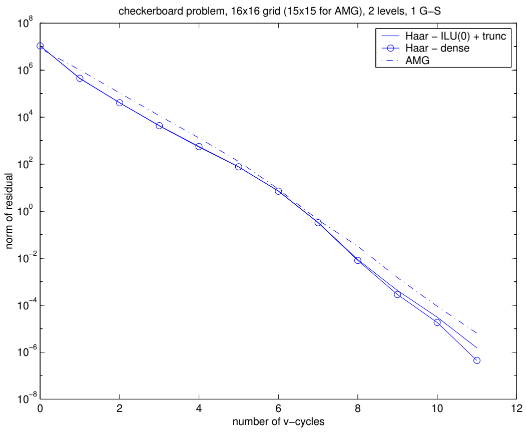
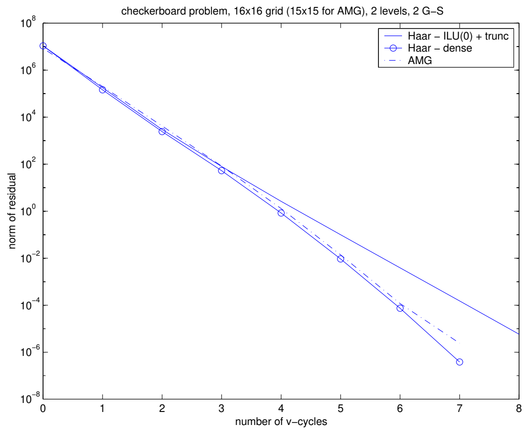


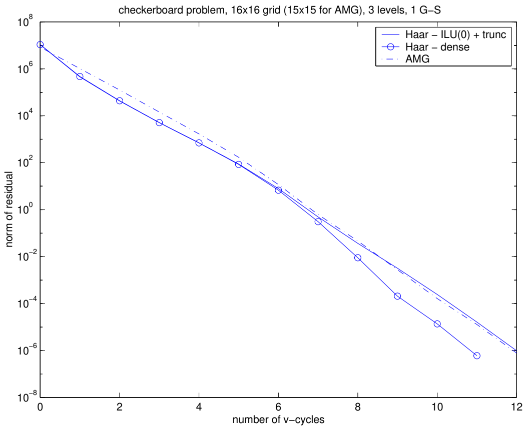
Figure 7.6: The values of a for the 4×4 checkerboard problem.



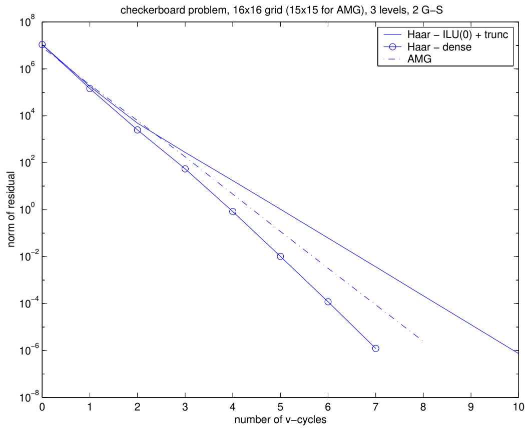
(a)



(b)

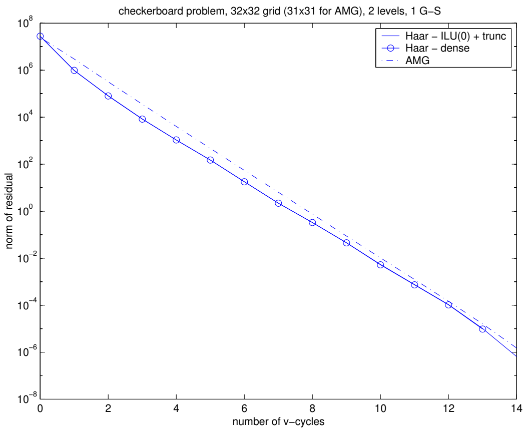


(c)

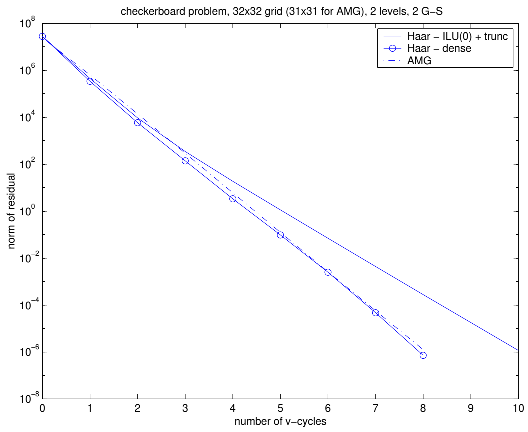


(d)

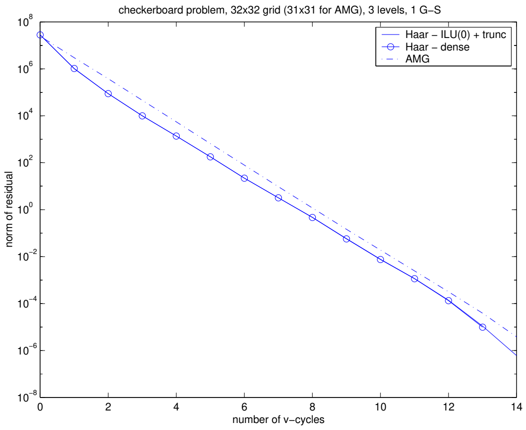
Figure 7.7: Checkerboard problem, 16×16 grid. Comparison of Haar wavelet multigrid method with AMG using (a) 2 levels, 1 Gauss-Seidel iteration; (b) 2 levels, 2 Gauss-Seidel iterations; (c) 3 levels, 1 Gauss-Seidel iteration; and (d) 3 levels, 2 Gauss-Seidel iterations.



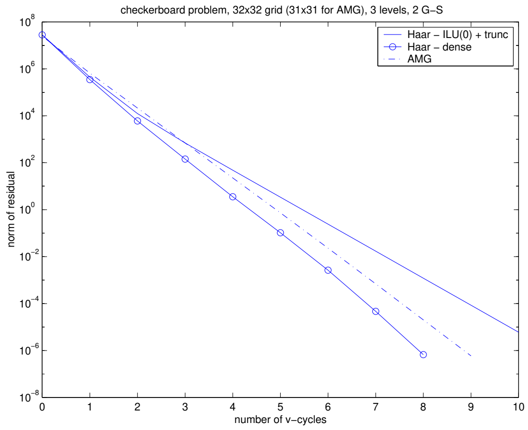
(a)



(b)



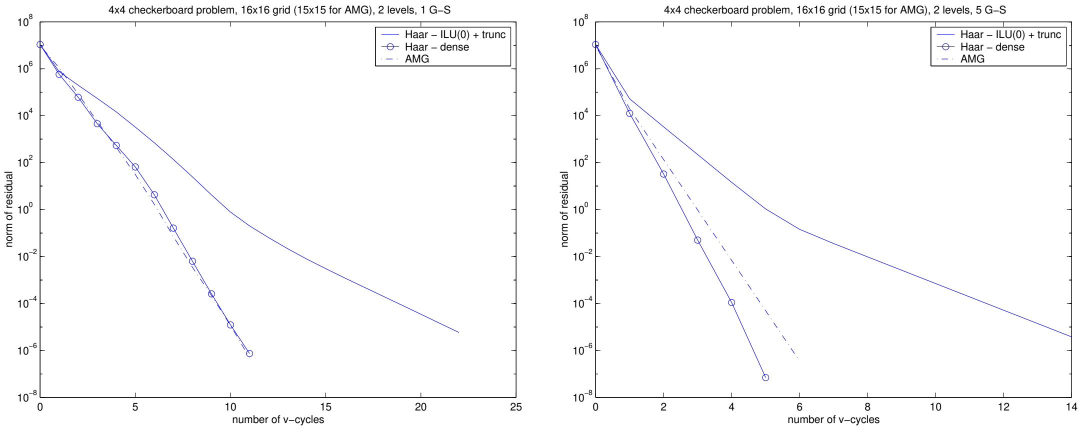
(c)



(d)

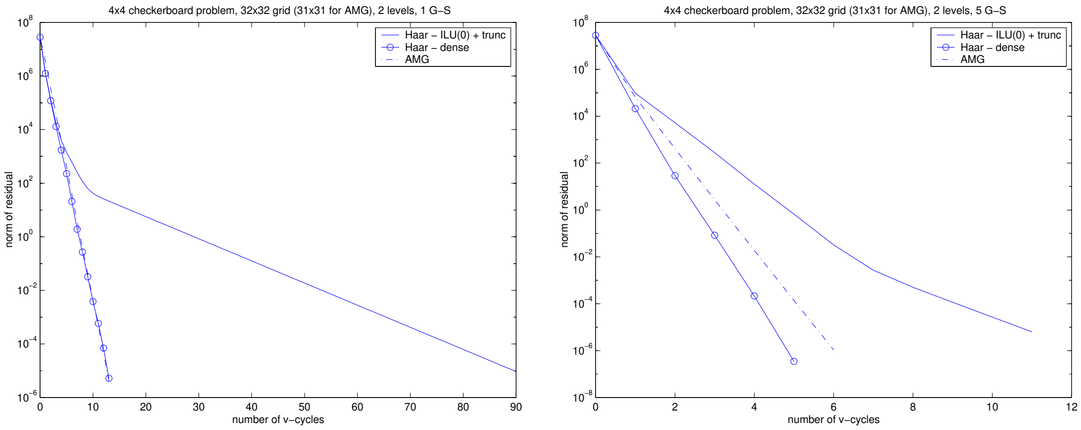
Figure 7.8: Checkerboard problem, 32×32 grid. Comparison of Haar wavelet multigrid method with AMG using (a) 2 levels, 1 Gauss-Seidel iteration; (b) 2 levels, 2 Gauss-Seidel iterations; (c) 3 levels, 1 Gauss-Seidel iteration; and (d) 3 levels, 2 Gauss-Seidel iterations.

Here, the dense wavelet multigrid method performs quite well. It's convergence rate is approximately the same as that for the algebraic multigrid method. The truncated method, however, is not quite as good. The convergence deteriorates to about half the rate of the dense method for the 16×16 grid and even worse for the 32×32 grid. Figures 7.9 and 7.10 demonstrate the results of applying the method with one Gauss-Seidel iteration on both the downswing and upswing and with five Gauss-Seidel iterations.



(a) (b)

Figure 7.9: 4×4 checkerboard problem, 16×16 grid. Comparison of Haar wavelet multigrid method with AMG using (a) 1 Gauss-Seidel iteration and (b) 5 Gauss-Seidel iterations.



(a) (b)

Figure 7.10: 4×4 checkerboard problem, 32×32 grid. Comparison of Haar wavelet multigrid method with AMG using (a) 1 Gauss-Seidel iteration and (b) 5 Gauss-Seidel iterations.

7.3 The Advection-Diffusion Problem

Here, we are investigating the problem

$$\begin{aligned} -\epsilon\triangle u + b\cdot u &= 0, \text{ in } \Omega \\ u &= f(x), \text{ on } \partial\Omega, \end{aligned} \tag{7.2}$$

where Ω is the unit square, and $||b|| >> \epsilon$.

In this problem, we encounter similar difficulties to the elliptic problem with highly oscillatory or discontinuous coefficients. Here, some of the oscillatory eigenmodes propagate along characteristics. So, moving to the coarse grid with the standard multigrid approach does not represent a good approximation to the problem on the coarse grid. See [YVB98, Yav98] for a brief discussion of this. We apply the wavelet multigrid method to these problems to overcome this difficulty, since application of the wavelet operator keeps the characteristics of the original problem.

To discretize, we use the usual five-point discretization for the diffusion term and a first order upwind scheme for the advection part of the equation. So, $-\epsilon\triangle u$ is discretized to be

$$\epsilon\frac{-u_{i+1,j}+2u_{i,j}-u_{i-1,j}}{h^2}+\epsilon\frac{-u_{i,j+1}+2u_{i,j}-u_{i,j-1}}{h^2}.$$

Letting $b=(b^x,b^y)$, and looking at the term $b^x u_x$, the first order upwind scheme looks like

$$b^x u_x(ih,jh)=\frac{b^{x,-}u_{i+1,j}+|b^x_{i,j}|u_{i,j}-b^{x,+}u_{i-1,j}}{h},$$

where $b^{x,-}=0.5(b^x_{i,j}-|b^x_{i,j}|)$ and $b^{x,+}=0.5(b^x_{i,j}+|b^x_{i,j}|)$, and similarly for b^y.

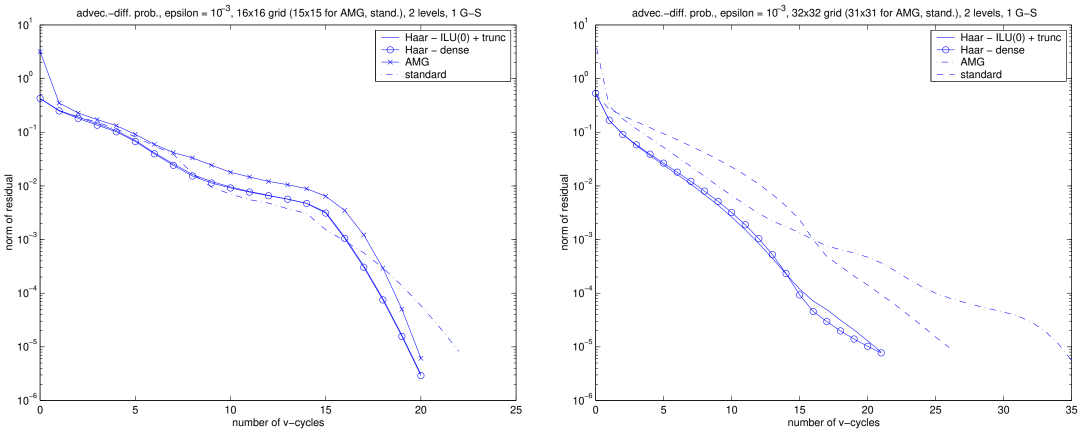
As a test case, we let $b^x=(2y-1)(1-x^2)$ and $b^y=2xy(y-1)$. We define the boundary conditions

$$f(x)=\begin{cases}1 & \text{if } x=0,\\ 0 & \text{otherwise.}\end{cases} \tag{7.3}$$

We vary ϵ to determine the effectiveness of the method when $||b|| >> \epsilon$, allowing $\epsilon=10^{-3}$ and 10^{-5}.

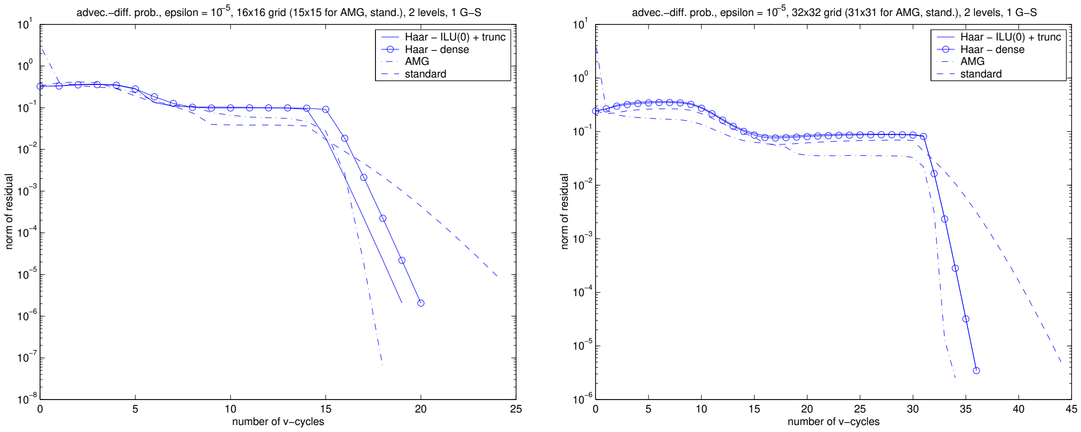
The results appear in Figures 7.11 and 7.12. We observe that the Haar wavelet multigrid method performs comparably to the algebraic multigrid method and outperforms the standard V-cycle method. One observes, however, a deterioration of convergence as the grid becomes finer for values of ϵ sufficiently small. For $\epsilon=10^{-5}$, convergence on a 32×32 grid is nearly twice as slow as that on a 16×16 grid.

Although this method performs as well as the algebraic multigrid method, its convergence rate is far below that which is desired, i.e., convergence depends on mesh size. The first matter, then, is to determine the reason for this deterioration in performance. This is done by testing a number of simpler problems using



(a) (b)

Figure 7.11: Comparison of standard V-cycle and AMG with Haar wavelet multigrid method. Here, $\epsilon = 10^{-3}$. (a) uses a 16×16 grid and (b) uses a 32×32 grid.

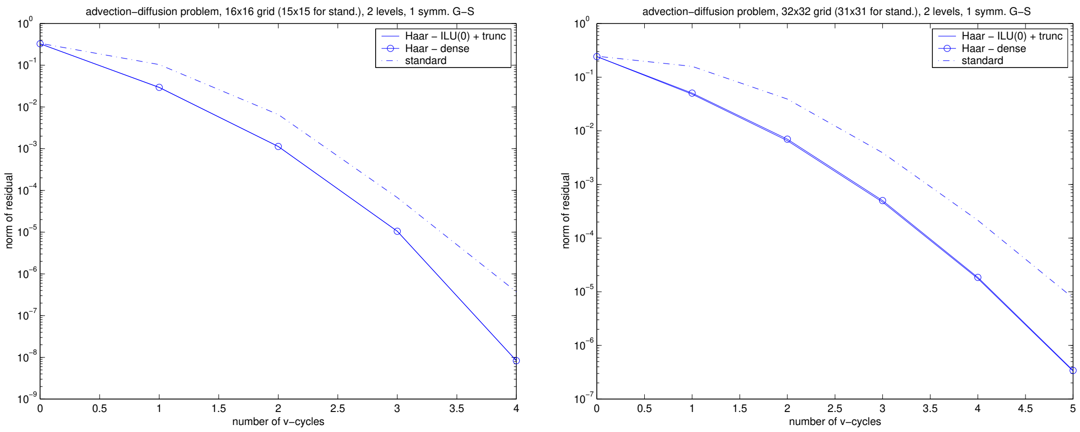


(a) (b)

Figure 7.12: Comparison of standard V-cycle and AMG with Haar wavelet multigrid method. Here, $\epsilon = 10^{-5}$. (a) uses a 16×16 grid and (b) uses a 32×32 grid.

$\epsilon = 10^{-5}$. The results of the tests convince us that the problem results from the sweeping of the Gauss-Seidel counter to the flow of the characteristics for at least part of the flow field. The solution becomes obvious – use symmetric Gauss-Seidel. This is expected to solve the problem because not only does it sweep in the forward direction, but this is followed by a sweep in the reverse direction. By doing this, we expect to capture all effects of the flow field. In fact, even applying this simple correction to the standard V-cycle method yields remarkable results.

First, we have a comparison of the methods for (7.2), where $b = ((2y-1)(1-x^2), 2xy(y-1))$ and $f(x)$ is defined in (7.3). We set $\epsilon = 10^{-5}$. The results are remarkable. Convergence has improved for both the standard multigrid method and the wavelet multigrid method, as Figure 7.13 demonstrates.

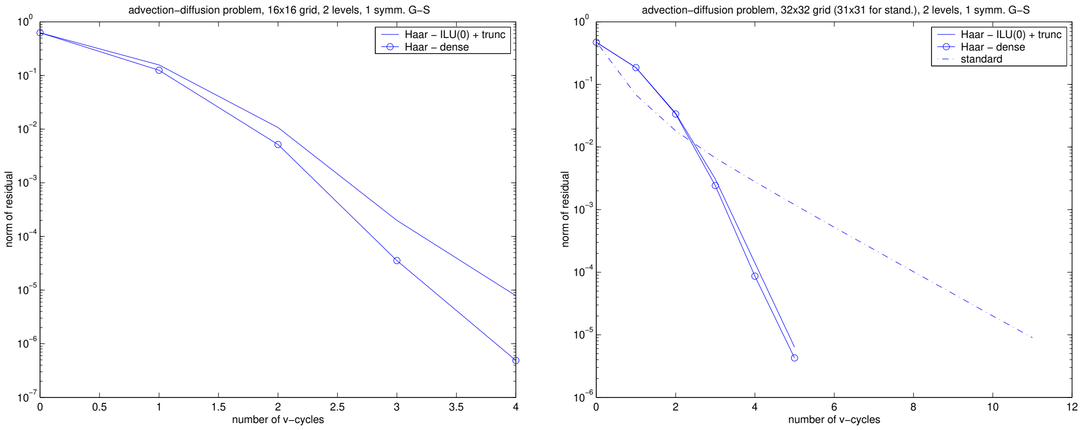


(a) (b)

Figure 7.13: Comparison of wavelet multigrid method with standard multigrid method, using symmetric Gauss-Seidel as the smoother. $\epsilon = 10^{-5}$ and $b = ((2y-1)(1-x^2), 2xy(y-1))$. In (a), a 16×16 grid is used and in (b), a 32×32 grid is used as the finest grid.

Next, we tackle a more difficult problem: a problem with recirculant flow. Here, $b = (4x(x-1)(1-2y), -4y(y-1)(1-2x))$ and $f(x)$ is defined as above.

For this particular case, the convergence is quite impressive. For the 16×16 grid, the standard multigrid method fails completely, although convergence does occur for a finer grid spacing. Here, too, however, the wavelet multigrid method outperforms the standard multigrid method with respect to convergence rate, as we can see from Figure 7.14. The contour plot of the solution, which shows the boundary layer is shown in Figure 7.15.



(a) (b)

Figure 7.14: Comparison of wavelet multigrid method with standard multigrid method, using symmetric Gauss-Seidel as the smoother. $\epsilon = 10^{-5}$ and flow is recirculant. In (a), a 16×16 grid is used and in (b), a 32×32 grid is used as the finest grid.

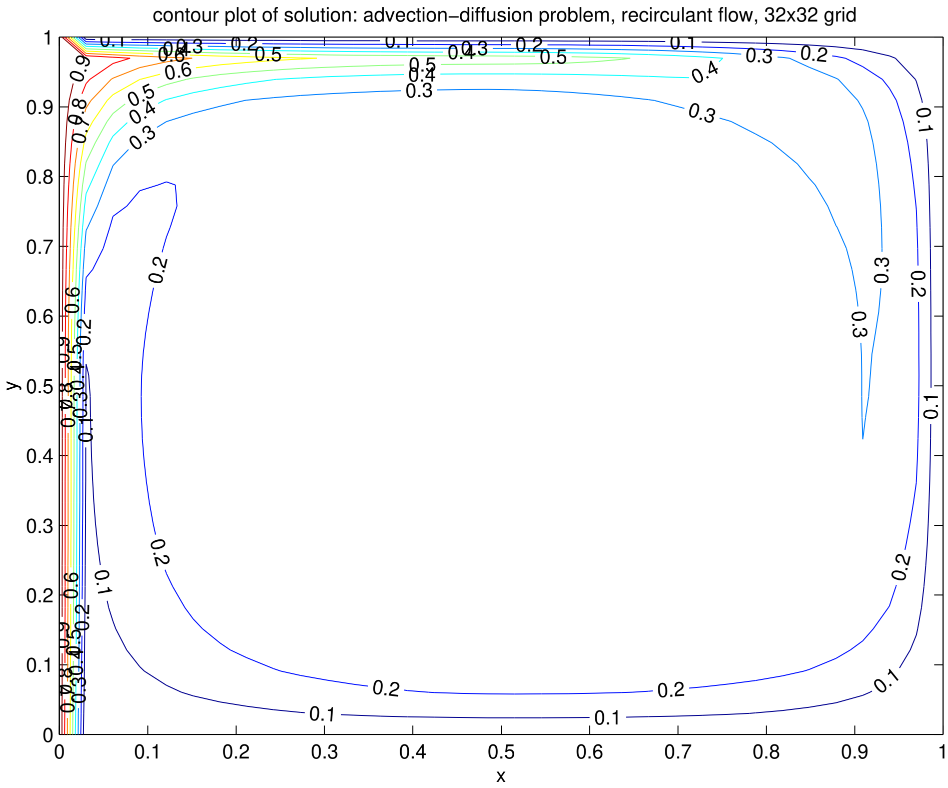


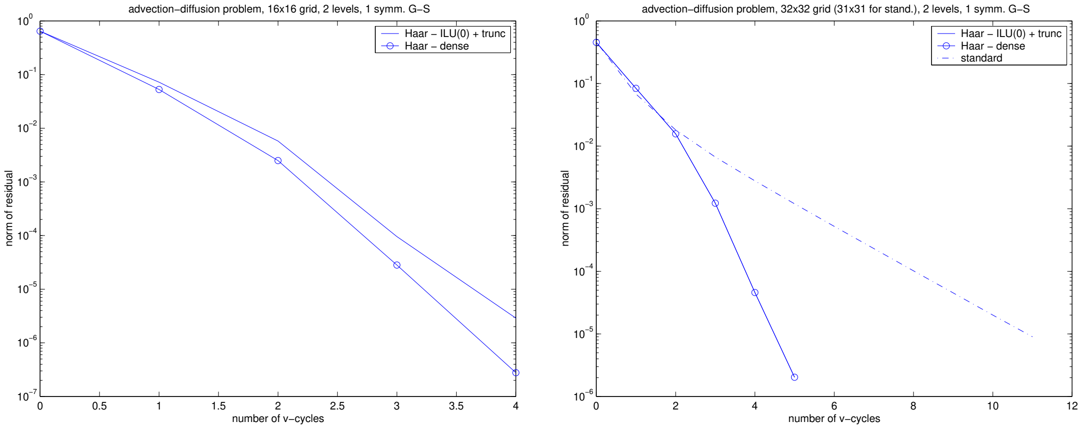
Figure 7.15: Contour plot of the solution of the advection-diffusion problem with recirculant flow. Results are shown for the 32×32 grid.

Next, the same advection and diffusion are used as in the above problem, but the boundary conditions are changed. Here, let

$$f(x) = \sin(\pi x) + \sin(13\pi x) + \sin(\pi y) + \sin(13\pi y).$$

Figure 7.16 demonstrates the superiority of the wavelet multigrid method in this case.

Finally, we use the boundary conditions from (7.3), but we change the advection component so that the characteristics are closed and the vorticity does not



(a) (b)

Figure 7.16: Comparison of wavelet multigrid method with standard multigrid method, using symmetric Gauss-Seidel as the smoother. $\epsilon = 10^{-5}$ with recirculant flow and sinusoidal boundary conditions. In (a), a 16×16 grid is used and in (b), a 32×32 grid is used.

line up with the axes. Here,

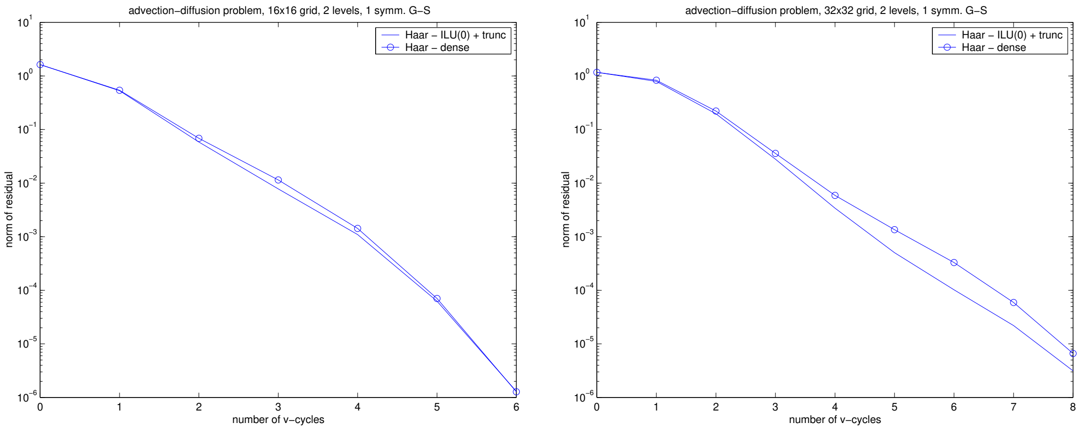
$$b = (\sin(\pi y_1)\cos(\pi x_1) + \sin(\pi y_2)\cos(\pi x_2),$$
$$- \cos(\pi y_1)\sin(\pi x_1) - \cos(\pi y_2)\sin(\pi x_2)),$$

where

$$x_1 = x^2 + 0.5,\ \ x_2 = (x-1)^2 + 0.5,\ \ y_1 = y^2 + 0.5,\ \ y_2 = (y-1)^2 + 0.5.$$

In this problem, the standard multigrid fails completely, but the wavelet multigrid method performs very well (see Figure 7.17). The contour plot of the solution, which shows the boundary layer is shown in Figure 7.18.

One thing to notice in the above examples is that, again, the wavelet multigrid method performs as the multigrid method should – convergence is essentially independent of the fine grid size.



(a) (b)

Figure 7.17: Comparison of wavelet multigrid method with standard multigrid method, using symmetric Gauss-Seidel as the smoother. $\epsilon = 10^{-5}$ and vorticity is skewed. In (a), a 16×16 grid is used and in (b), a 32×32 grid is used as the finest grid.

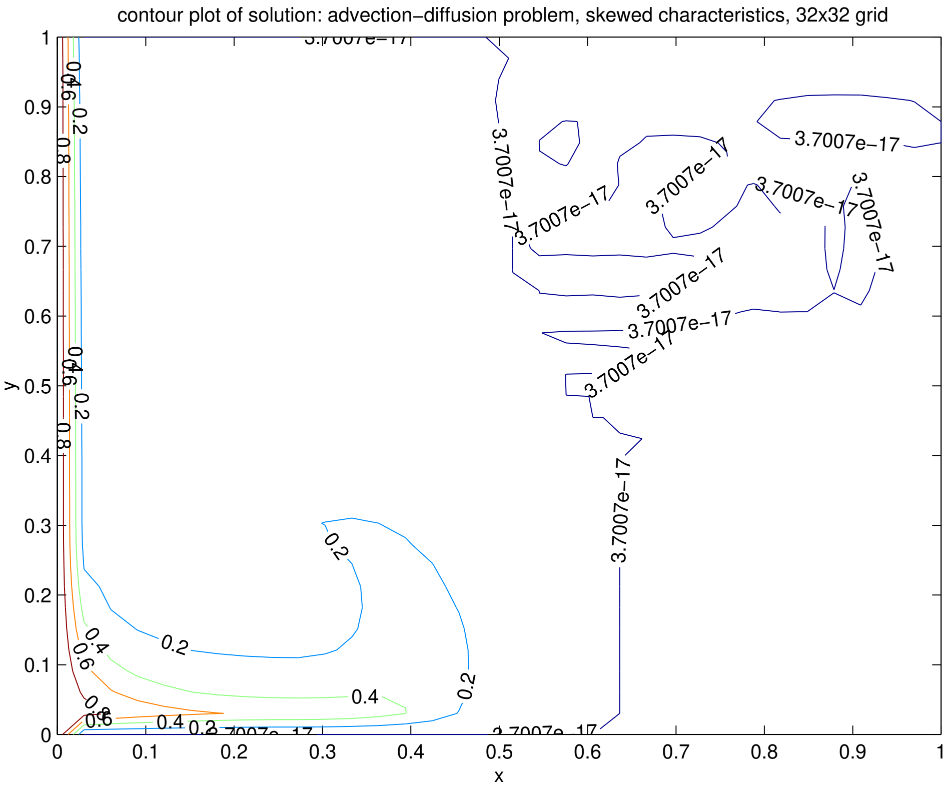


Figure 7.18: Contour plot of the solution of the advection-diffusion problem with skewed vorticity. Results are shown for the 32×32 grid.

Next, we look at the problem

$$\begin{aligned} -\nabla \cdot (A \nabla u) &= 0, \text{ in } \Omega \\ u &= 0, \text{ on } \partial\Omega, \end{aligned} \tag{7.4}$$

where Ω is the unit square and A is a positive definite 2×2 matrix:

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}.$$

For our test case, we set $a_{11} = \epsilon = 10^{-5}$ and $a_{22} = 1$, with $a_{12} = a_{21} = 0$. The wavelet multigrid procedure fails to produce rapid convergence. Why does it fail?

Suppose that the error has oscillatory components in the x-direction. When the averaging procedure is done (to get the coarse grid operator), these oscillatory components will not be represented on the coarse grid. So, the procedure will quickly smooth out the error in the y-direction, and, due to the anisotropy, the error in the x-direction will not be smoothed. Therefore, the method will fail to converge rapidly, beyond the order of ϵ.

CHAPTER 8

The Stokes and Incompressible Navier-Stokes Equations

8.1 The Stokes Equations

Here, we consider the Stokes equations

$$\begin{aligned} -\triangle u + p_x &= f^u \text{ in } \Omega \\ -\triangle v + p_y &= f^v \text{ in } \Omega \\ u_x + v_y &= 0 \text{ on } \overline{\Omega}. \end{aligned}$$

Now, as is done in [Sch90], summing the first two equations and using the continuity condition ($u_x + v_y = 0$), we obtain the following system of equations:

$$-\triangle u + p_x = f^u \text{ in } \Omega \tag{8.1}$$

$$-\triangle v + p_y = f^v \text{ in } \Omega \tag{8.2}$$

$$\triangle p = f^u_x + f^v_y \text{ in } \Omega \tag{8.3}$$

$$u_x + v_y = 0 \text{ on } \partial\Omega. \tag{8.4}$$

These equations are discretized using centered differencing for the first order terms (p_x, p_y, u_x, v_y) and the usual five-point discretization for the second order terms ($\triangle u, \triangle v, \triangle p$). Both f^u and f^v are assumed zero for the numerical calculations and Ω is the unit square. We use forward differencing (left-hand side and bottom of the square) and backward differencing (right-hand side and top of the square) to discretize the continuity equation (8.4). This discretization is used to obtain equations for the outermost interior values on the left and right boundaries of the square (for u) and on the top and bottom (for v). This can easily be done, because u and v are specified on the boundary of the unit square. So, we calculate v_y for $x = 0$ and $x = 1$, and then add this term to the right-hand side in our matrix equation and put the discretization for u_x into the matrix. The same holds true for the top and bottom, with u_x being calculated when $y = 0$ and $y = 1$ and v_y being discretized. The boundary values of p are obtained by using (8.1) for $x = 0$ and $x = 1$ and (8.2) for $y = 0$ and $y = 1$. Since the corner values of p do not appear directly in the discretization, we assign to them the value of the average of the adjacent boundary values for p. The V-cycle multigrid algorithm is followed, with Gauss-Seidel as the smoother, but fifty iterations are performed on the coarse grid in lieu of an exact solve (since the matrix is singular). Fewer iterations are necessary on the coarse grid if more than two levels are used (so that the coarse grid is actually sufficiently coarse). For the first sample problem, we assume boundary conditions of $u = 0$ and $v = 0$.

Now, for this problem, as well as for the incompressible Navier-Stokes problem, we have three unknowns, u, v, and p. Therefore, the wavelet multigrid method must be modified. For example, let H_u be the scaling operator that will be applied to the discrete values for u. Similarly, define H_v and H_p. We also define the wavelet operators G_u, G_v, and G_p accordingly. Let

$$\mathbf{H}_j = \begin{pmatrix} H_u & 0 & 0 \\ 0 & H_v & 0 \\ 0 & 0 & H_p \end{pmatrix}.$$

Define $\mathbf{G}_j$ in the same manner. Then, the wavelet transform $\boldsymbol{\mathcal{W}}_j$ must be defined as

$$\mathcal{W}_j = \begin{pmatrix} \mathbf{H}_j \\ \mathbf{G}_j \end{pmatrix}.$$

Then, the wavelet transform is orthogonal, and the new scaling and wavelet operators satisfy the conditions (3.20) - (3.22). Also, the wavelet multigrid method still follows as in Chapter 4, letting

$$U = \begin{pmatrix} u \\ v \\ p \end{pmatrix}$$

and

$$F = \begin{pmatrix} f^u \\ f^v \\ f^u_x + f^v_y \end{pmatrix} + \text{ boundary terms.}$$

As can be seen from Figure 8.1, the wavelet multigrid method is effective in solving the Stokes equations with zero boundary conditions. We also examine the effectiveness of the wavelet multigrid method in solving the Stokes equations with boundary conditions of

$$\begin{aligned} u &= \begin{cases} 1 & \text{for } y = 1, \\ 0 & \text{otherwise,} \end{cases} \\ v &= 0. \end{aligned}$$

Here, too, the results are good (see Figure 8.2).

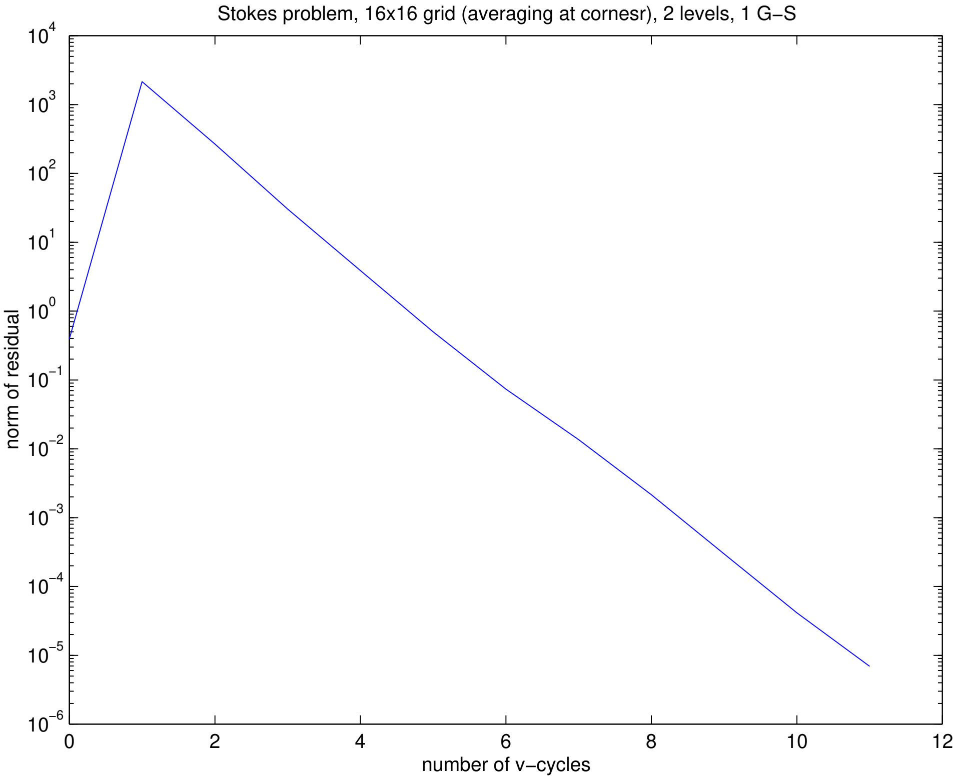


Figure 8.1: The Stokes equations on a 16×16 grid, $u = v = 0$ on the boundary

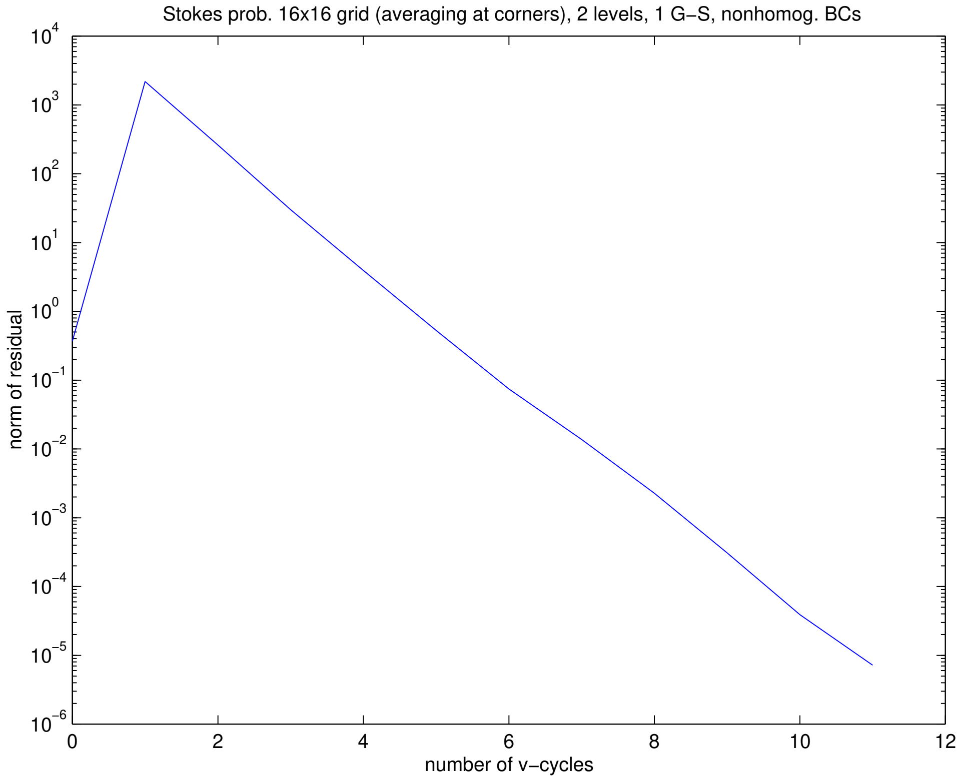


Figure 8.2: The Stokes equations on a 16×16 grid, nonhomogeneous boundary conditions.

8.2 The Incompressible Navier-Stokes Equations

Here, we consider the incompressible Navier-Stokes equations, given by

$$
\begin{aligned}
-\triangle u + Re(uu_x + vu_y) + p_x &= f^u \text{ in } \Omega && (8.5)\\
-\triangle v + Re(uv_x + vv_y) + p_y &= f^v \text{ in } \Omega && (8.6)\\
u_x + v_y &= 0 \text{ in } \overline{\Omega}. && (8.7)
\end{aligned}
$$

The following theorem allows us to rewrite the Navier-Stokes equations in a form that will allow our algorithm to be more efficient and eliminate the need for addition of an artificial stabilizing term. This theorem was stated and in a slightly different form in [Sch90].

Theorem 2. *Given Ω, a bounded domain in $\mathbb{R}^2$, and functions $u(x,y)$ and $v(x,y)$ in $C^3(\Omega)\bigcap C^1(\overline{\Omega})$, $p(x,y)$ in $C^2(\Omega)$, and $f^u, f^v \in C^1(\Omega)$. Then, the system of equations (8.5) - (8.7) is equivalent to the system*

$$
\begin{aligned}
-\triangle u + Re(uu_x + vu_y) + p_x &= f^u \textit{ in } \Omega && (8.8)\\
-\triangle v + Re(uv_x + vv_y) + p_y &= f^v \textit{ in } \Omega && (8.9)\\
\triangle p + 2Re(v_x u_y - u_x v_y) &= f^u_x + f^v_x \textit{ in } \Omega && (8.10)\\
u_x + v_y &= 0 \textit{ on } \partial\Omega. && (8.11)
\end{aligned}
$$

Proof.

$$\frac{\partial}{\partial x}(-\triangle u + Re(uu_x + vu_y) + p_x) = \frac{\partial}{\partial x} f^u$$

yields

$$-\triangle u_x + Re(-u_x v_y + v_x u_y + uu_{xx} + vu_{yx}) + p_{xx} = f^u_x,$$

after application of (8.7). Similarly,

$$\frac{\partial}{\partial y}(-\triangle v + Re(uv_x + vv_y) + p_y) = \frac{\partial}{\partial y} f^v$$

yields

$$-\triangle v_y + Re(u_y v_x - v_y u_x + u v_{xy} + v v_{yy}) + p_{yy} = f_y^v.$$

Adding these two equations and noting that $u_{yx} = u_{xy}$, we obtain

$$\begin{aligned} -\triangle(u_x + v_y) &+ 2Re(u_y v_x - u_x v_y) \\ &+ Re(u(u_x + v_y)_x + v(u_x + v_y)) = f_x^u + f_y^v. \end{aligned}$$

Applying the continuity condition (8.7) again, this reduces to (8.10). □

Equations (8.8) - (8.10) are discretized using the standard five-point discretization for the second order terms. So, for example

$$-\triangle u(ih, jh) = \frac{-u_{i+1,j} + 2u_{i,j} - u_{i-1,j}}{h^2} + \frac{-u_{i,j+1} + 2u_{i,j} - u_{i,j-1}}{h^2}.$$

The advection terms (u_x, u_y, v_x, v_y) are discretized using first order upwind differencing in (8.8) and (8.9). Centered differencing is employed for the first order terms in (8.10). Both f^u and f^v are assumed zero for the numerical calculations and Ω is the unit square. We use forward differencing (left-hand side and bottom of the square) and backward differencing (right-hand side and top of the square) to discretize the continuity equation (8.11). The equations for u and v at the outermost interior gridpoints are obtained from the discretization of the continuity equation in the same manner as described for the Stokes equations. The boundary values for p are calculated using (8.8) for $x = 0$ and $x = 1$ and using (8.9) for $y = 0$ and $y = 1$. Since the corner values of p do not appear directly in the discretization, we assign to them the value of the average of the adjacent boundary values for p, as we did for the Stokes problem. Newton's method is used to linearize (8.8) - (8.10). Basically, the procedure is as follows: let F_{NS}^u denote the discretization of (8.8), F_{NS}^v denote the discretization of (8.9), and F_{NS}^p

denote the discretization of (8.10). In each of these, we substitute $u+\tilde{u},\ v+\tilde{v}$, and $p+\tilde{p}$ for $u,\ v$, and p. Then, we expand, delete any terms involving products of $\tilde{u}$ and $\tilde{v}$, and put all remaining terms not involving $\tilde{u}$, $\tilde{v}$, and $\tilde{p}$ on the right-hand side. This gives us the approximation

$$(F^u_{NS})'(\tilde{u},\tilde{v},\tilde{p}) \approx f - F^u_{NS}(u,v,p),$$

for (8.8), and similarly for (8.9) and (8.10). We do the same for the continuity equation (8.11). Then, we write the above in matrix form

$$F'_{NS}\tilde{U}^{n+1} = f - F_{NS}U^n, \tag{8.12}$$

where F_{NS} is the matrix operator determined by the discretization of (8.8) - (8.11), and solve for

$$\tilde{U}^{n+1} = \begin{pmatrix} \tilde{u}^{n+1} \\ \tilde{v}^{n+1} \\ \tilde{p}^{n+1} \end{pmatrix}.$$

This, then, is the problem to be solved in the inner iterations, and the outer iteration is

$$U^{n+1} = U^n + \tilde{U}^{n+1}.$$

The matrix problem (8.12) is solved using the wavelet multigrid method, letting L_j be the matrix operator F_{NS}, $U=\tilde{U}^{n+1}$, and F the right-hand side in (8.12). The wavelet multigrid method is again applied as described in Section 8.1. The V-cycle multigrid algorithm is followed, with Gauss-Seidel as the smoother, but fifty iterations are performed on the coarse grid in lieu of an exact solve (since the matrix is singular). Fewer iterations are necessary on the coarse grid if more than two levels are used (so that the coarse grid is actually sufficiently coarse). Here, we assume that $u=0$ and $v=0$ on the boundary.

As can been seen from the following tables, the wavelet multigrid method performs well. Convergence is found to be almost independent of the size of the Reynold's number, after testing various values from $Re = 1$ to $Re = 100$.

Table 8.1: Convergence history of Navier-Stokes equations, $Re = 1$, on a 16×16 grid.

iter	k	norm of resid. in inner iter.	norm of resid. in outer iter.
0	0	7.905694e-01	3.952847e-01
1	1	7.008245e+03	
1	2	1.258527e+03	
1	3	1.446255e+02	
1	4	2.010187e+01	
1	5	2.677665e+00	
1	6	3.853789e-01	
1	7	6.580559e-02	
1	8	9.154482e-03	
1	9	1.247744e-03	
1	10	2.069859e-04	
1	11	3.254192e-05	
1	12	6.012843e-06	
1	13		6.007674e-06

Table 8.2: Convergence history of Navier-Stokes equations, $Re = 10$, on a 16×16 grid.

iter	k	norm of resid. in inner iter.	norm of resid. in outer iter.
0	0	7.905694e-01	3.952847e-01
1	1	8.893011e+03	
1	2	1.658027e+03	
1	3	1.913211e+02	
1	4	2.803854e+01	
1	5	3.623013e+00	
1	6	6.195620e-01	
1	7	1.117080e-01	
1	8	1.382316e-02	
1	9	2.560847e-03	
1	10	5.088335e-04	
1	11	6.377358e-05	
1	12	1.117035e-05	
1	13	2.300576e-06	
1	13		2.294850e-06

Table 8.3: Convergence history of Navier-Stokes equations, $Re = 50$, on a 16×16 grid.

iter	k	norm of resid. in inner iter.	norm of resid. in outer iter.
0	0	3.952847e-01	7.905694e-01
1	1	1.712370e+04	
1	2	3.322899e+03	
1	3	4.149911e+02	
1	4	6.536471e+01	
1	5	8.950878e+00	
1	6	2.245120e+00	
1	7	4.298050e-01	
1	8	6.098672e-02	
1	9	1.688919e-02	
1	10	3.284410e-03	
1	11	4.700841e-04	
1	12	1.266656e-04	
1	13	2.438572e-05	
1	14	3.551966e-06	
1	15		3.546535e-06

Table 8.4: Convergence history of Navier-Stokes equations, $Re = 100$, on a 16×16 grid.

iter	k	norm of resid. in inner iter.	norm of resid. in outer iter.
0	0	3.952847e-01	7.905694e-01
1	1	2.702935e+04	
1	2	5.345722e+03	
1	3	6.992212e+02	
1	4	1.125374e+02	
1	5	1.638484e+01	
1	6	4.608594e+00	
1	7	9.345795e-01	
1	8	1.388602e-01	
1	9	4.123109e-02	
1	10	8.631406e-03	
1	11	1.268998e-03	
1	12	3.643826e-04	
1	13	7.743688e-05	
1	14	1.135024e-05	
1	15	3.200787e-06	
1	16		3.198504e-06

For the Navier-Stokes problem with boundary conditions of

$$\begin{aligned} u &= \begin{cases} 1, & \text{y} = 1, \\ 0, & \text{otherwise}, \end{cases} \\ v &= 0, \end{aligned}$$

however, the results are not as promising. The difficulty here is that the Gauss-Seidel iterative method does not converge for this problem. Therefore, the multigrid method using this as a smoother does not converge. More investigation is required to determine a remedy.

CHAPTER 9

Conclusion and Future Research

As we have seen, the new multigrid method, called the wavelet multigrid method, has proven to be very useful in a wide variety of problems. In many of those problems where standard multigrid methods fail to converge independently of mesh size, the wavelet multigrid method does ensure such convergence. Also, due to the properties of wavelets, in many cases we can more efficiently apply the wavelet multigrid method through use of compression. For those problems where this method fails to work (the anisotropic second order problem and the incompressible Navier-Stokes with nonhomogeneous boundary conditions), more research is required to determine a remedy. With respect to solving the anisotropic second order problem, one approach is to use semicoarsening instead of full coarsening.

This work has opened the door to many other research options related to this method. One important direction of research involves improving the efficiency of the program. Although the truncated wavelet multigrid method does provide an advantage with respect to sparsity over the full method, a higher degree of sparsity in the coarse grid operator is desired in order to make the method of more practical value. Also, Professor Tony Chan at UCLA has been working with adaptive wavelet techniques, and these may be applied to the wavelet multigrid method. Use of other basis functions, as well, might be investigated. Investigation of computational fluid dynamics applications of the wavelet multigrid method is one other possible direction of research. In addition, the solution of problems with unstructured grids using this method may be examined. This may work because formally, at least, a matrix equation results after discretization of the problem.

References

[ABD81] R. E. Alcouffe, Achi Brandt, J. E. Dendy, Jr., and J. W. Painter. "The Multi-Grid Method for the Diffusion Equation with Strongly Discontinuous Coefficients." *SIAM Journal of Scientific Statistical Computing*, **2**(4):430–454, December 1981.

[AEL97] Ulf Andersson, Bjorn Engquist, Gunnar Ledfelt, and Olaf Runborg. "Wavelet-Based Subgrid Modeling: 1. Principles and Scalar Equations." CAM Report 97-13, UCLA, March 1997.

[Ban96] Randolph E. Bank. "Hierarchical Bases and the Finite Element Method." *Acta Numerica*, pp. 1–43, 1996.

[BCF98] M. Brezina, A. J. Cleary, R. D. Falgout, V. E. Henson, J. E. Jones, T. A. Manteuffel, S. F. McCormick, and J. W. Ruge. "Algebraic Multigrid Based on Element Interpolation (AMGe)." Technical Report UCRL-JC-131752, Lawrence Livermore National Laboratory, 1998.

[BCR91] G. Beylkin, R. Coifman, and V. Rokhlin. "Fast Wavelet Transforms and Numerical Algorithms I." *Communication on Pure and Applied Mathematics*, **44**:141–183, 1991.

[BFP89] C. Becker, J. H. Ferziger, M. Perić, and G. Scheuerer. "Finite Volume Multigrid Solutions of the Two-Dimensional Incompressible Navier-Stokes Equations." In *Robust Multigrid Methods (Kiel, 1988)*, volume 23 of *Notes on Numerical Fluid Mechanics*, pp. 37–47. Vieweg, Braunschweig, 1989.

[BPV98] James H. Bramble, Joseph E. Pasciak, and Apostol T. Vassilev. "Analysis of Non-Overlapping Domain Decomposition Algorithms with Inexact Solvers." *Mathematics of Computation*, **67**(221):1–19, 1998.

[BPX88] James H. Bramble, Joseph E. Pasciak, and Jinchao Xu. "The Analysis of Multigrid Algorithms for Nonsymmetric and Indefinite Elliptic Problems." *Mathematics of Computation*, **194**:389–414, October 1988.

[Bra82] Achi Brandt. "Guide to Multigrid Development." In *Multigrid Methods (Cologne, 1981)*, volume 960 of *Lecture Notes in Mathematics*, pp. 220–312. Springer-Verlag, Berlin-New York, 1982.

[Bri87] William L. Briggs. *A Multigrid Tutorial.* SIAM, Philadelphia, PA, 1987.

[BW95] Jürgen Bey and Gabriel Wittum. "Downwind Numbering: A Robust Multigrid Method for Convection Diffusion Problems on Unstructured Grids." In *Fast Solvers for Flow Problems (Kiel, 1994)*, volume 49 of *Notes on Numerical Fluid Mechanics*, pp. 63–73. Vieweg, Braunschweig, 1995.

[BX94] Randolph E. Bank and Jinchao Xu. "The Hierarchical Basis Multigrid Method and Incomplete LU Decomposition." In *Domain Decomposition Methods in Scientific and Engineering Computing (University Park, PA, 1993)*, volume 180 of *Contemporary Mathematics*, pp. 163–173. AMS, Providence, RI, 1994.

[BX95] Randolph E. Bank and Jinchao Xu. "A Hierarchical Basis Multigrid Method for Unstructured Grids." In *Fast Solvers for Flow Problems (Kiel, 1994)*, volume 49 of *Notes on Numerical Fluid Mechanics*, pp. 1–13. Vieweg, Braunschweig, 1995.

[CGM85] P. Concus, G. H. Golub, and G. Meurant. "Block Preconditioning for the Conjugate Gradient Method." *SIAM Journal on Scientific and Statistical Computing*, **6**(1):220–252, 1985.

[CM94] Tony F. Chan and Tarek P. Mathew. "Domain Decomposition Algorithms." *Acta Numerica*, pp. 61–143, 1994.

[CTW97] T. F. Chan, W. P. Tang, and W. L. Wan. "Wavelet Sparse Approximate Inverse Preconditioners." CAM Report 97-34, UCLA, August 1997.

[CW93] Xiao-Chuan Cai and Olof B. Widlund. "Multiplicative Schwarz Algorithms for Some Nonsymmetric and Indefinite Problems." *SIAM Journal of Numerical Analysis*, **30**(4):936–952, August 1993.

[CW99] Tony Chan and Wing-Lok Wan. "Robust Multigrid Methods for Elliptic Linear Systems." CAM Report 99-30, UCLA, September 1999.

[CZ85] A. W. Craig and O. G. Zienkiewicz. "A Multigrid Algorithm Using a Hierarchical Finite Element Basis." In D. J. Paddon and H. Holstein, editors, *Multigrid Methods for Integral and Differential Equations (Bristol, 1983)*, volume 3 of *Institute of Mathematics and its Applications Conference Series*, pp. 301–312. Oxford University Press, New York, NY, 1985.

[Dau88] Ingrid Daubechies. "Orthonormal Bases of Compactly Supported Wavelets." *Communication on Pure and Applied Mathematics*, **XLI**(7):909–996, 1988.

[Dau92] Ingrid Daubechies. *Ten Lectures on Wavelets*, volume 61 of *CBMS-NSF Series in Applied Mathematics*. SIAM, Philadelphia, PA, 1992.

[DE89] Wolfgang Dahmen and Ludwig Elsner. "Algebraic Multigrid Methods and the Schur Complement." In *Robust Multi-Grid Methods (Kiel, 1988)*, volume 23 of *Notes on Numerical Fluid Mechanics*, pp. 58–68. Vieweg, Braunschweig, 1989.

[DE98] Mihai Dorobantu and Bjorn Engquist. "Wavelet-Based Numerical Homogenization." *SIAM Journal of Numerical Analysis*, **35**(2):540–559, April 1998.

[Den82] J. E. Dendy, Jr. "Black Box Multigrid." *Journal of Computational Physics*, **48**:366–386, 1982.

[Dis99] Boris Diskin. "Solving Upwind-biased Discretizations II: Multigrid Solver Using Semicoarsening." ICASE Report 99-25, Institute for Computer Applications in Science and Engineering (ICASE), NASA Langley Research Center, July 1999.

[EL] Bjorn Engquist and Erding Luo. "The Multigrid Method Based on a Wavelet Transformation and Schur Complement." Manuscript.

[EL93] Bjorn Engquist and Erding Luo. "Multigrid Methods for Differential Equations with Highly Oscillatory Coefficients." In *Proceedings of the Sixth Copper Mountain Conference on Multigrid Methods*, pp. 175–190, 1993.

[EL96] Bjorn Engquist and Erding Luo. "New Coarse Grid Operators for Highly Oscillatory Coefficient Elliptic Problems." *Journal of Computational Physics*, **129**:296–306, 1996.

[EL97] Bjorn Engquist and Erding Luo. "Convergence of a Multigrid Method for Elliptic Equations with Highly Oscillatory Coefficients." *SIAM Journal of Numerical Analysis*, **34**(6):2254–2273, December 1997.

[GK95] M. Griebel and S. Knapek. "A Multigrid Homogenization Method." In *Modeling and Computation in Environmental Sciences (Stuttgart, 1995)*, volume 59 of *Notes on Numerical Fluid Mechanics*, pp. 187–202. Vieweg, Braunschweig, 1995.

[Gri94a] Michael Griebel. "A Domain Decomposition Method Using Sparse Grids." In *Domain Decomposition Methods in Scientific and Engineering Computing (Como, 1992)*, volume 157 of *Contemporary Mathematics*, pp. 255–260. AMS, Providence, RI, 1994.

[Gri94b] Michael Griebel. "Domain-Oriented Multilevel Methods." In *Domain Decomposition Methods in Scientific and Engineering Computing (University Park, PA, 1993)*, volume 180 of *Contemporary Mathematics*, pp. 223–229. AMS, Providence, RI, 1994.

[GRW96] Roland Glowinski, Andreas Rieder, Raymond Wells, Jr., and Xiaodong Zhou. "A Wavelet Multigrid Preconditioner for Dirichlet Boundary Value Problems in General Domains." *Modélisation Mathématique et Analyse Numérique (Mathematical Modeling and Numerical Analysis)*, **30**(6):711–729, 1996.

[JN98] M. Jung and S. V. Nepomnyaschikh. "Multilevel Preconditioning Procedures for Elliptic Problems." In *Large-Scale Scientific Computations of Engineering and Environmental Problems (Varna, 1997)*, volume 62 of *Notes on Numerical Fluid Mechanics*, pp. 78–90. Vieweg, Braunschweig, 1998.

[JV95] R. Jyotsna and S. P. Vanka. "A Pressure Based Multigrid Procedure for the Navier-Stokes Equations on Unstructured Grids." In *Proceedings of the Seventh Copper Mountain Conference on Multigrid Methods*, 1995.

[Kes79] Srinivasan Kesavan. "Homogenization of Elliptic Eigenvalue Problems: Part1." *Applied Mathematics and Optimization*, **5**:153–167, 1979.

[KL95] Andrei Kapurkin and G. Lube. "A Domain Decomposition Method for Singularly Perturbed Elliptic Problems." In *Fast Solvers for Flow Problems (Kiel, 1994)*, volume 49 of *Notes on Numerical Fluid Mechanics*, pp. 151–162. Vieweg, Braunschweig, 1995.

[KW93] R. Kornhuber and G. Wittum. "Discretization and Iterative Solution of Convection Diffusion Equation." In *Incomplete Decomposition (*ILU*) – Algorithms, Theory, and Applications (Kiel, 1992)*, volume 41 of *Notes on Numerical Fluid Mechanics*, pp. 67–77. Vieweg, Braunschweig, 1993.

[LC95] Wing Kam Liu and Yijung Chen. "Wavelet and Multiple Scale Reproducing Kernel Methods." *International Journal for Numerical Methods in Fluids*, **21**(10):901 – 931, 1995.

[Man86] Jan Mandel. "Multigrid Convergence for Nonsymmetric, Indefinite Variational Problems and One Smoothing Step." *Applied Mathematics and Computation*, **19**:201–216, 1986.

[McC87] Stephen McCormick, editor. *Multigrid Methods*. SIAM, Philadelphia, PA, 1987.

[MDH98] J. David Moulton, Joel E. Dendy, Jr., and James M. Hyman. "The Black Box Multigrid Numerical Homogenization Algorithm." *Journal of Computational Physics*, **142**:80–108, 1998.

[MM85] J. F. Maitre and F. Musy. "Algebraic Formalisation of the Multigrid Method in the Symmetric and Positive Definite Case – A Convergence Estimation for the V-Cycle." In D. J. Paddon and H. Holstein, editors, *Multigrid Methods for Integral and Differential Equations (Bristol, 1983)*, volume 3 of *Institute of Mathematics and its Applications Conference Series*, pp. 213–223. Oxford University Press, New York, NY, 1985.

[MMB87] J. Mandel, S. McCormick, and R. Bank. "Variational Multigrid Methods." In *Multigrid Methods*, Frontiers in Applied Mathematics, pp. 131–177. SIAM, Philadelphia, PA, 1987.

[MV98] Svetozar Margenov and Panayot S. Vassilevski. "Algebraic Two-Level Preconditioning of Nonconforming FEM Systems." In *Large-Scale Scientific Computations of Engineering and Environmental Problems (Varna, 1997)*, volume 62 of *Notes on Numerical Fluid Mechanics*, pp. 239–248. Vieweg, Braunschweig, 1998.

[NJW98] Nicolas Neuss, Willi Jäger, and Gabriel Wittum. "Homogenization and Multigrid." Preprint 98-04 (SFB 359), January 1998.

[Reu94] Arnold Reusken. "On Maximum Norm Convergence of Multigrid Methods for Elliptic Boundary Value Problems." *SIAM Journal of Numerical Analysis*, **31**(2):378–392, April 1994.

[Reu95] Arnold Reusken. "A New Robust Multigrid Method for 2D Convection-Diffusion Problems." In *Fast Solvers for Flow Problems (Kiel, 1994)*, volume 49 of *Notes on Numerical Fluid Mechanics*, pp. 229–240. Vieweg, Braunschweig, 1995.

[Rie93] Andreas Rieder. "Semi-algebraic multi-level methods based on wavelet decompositions I: Application to two-point boundary value problems." *Communication on Pure and Applied Mathematics*, **44**:141–183, 1993.

[Rie94] Andreas Rieder. "Multilevel Methods Based on Wavelet Decomposition." *East-West Journal of Numerical Mathematics*, **2**(4):313–330, 1994.

[RS87] J. W. Ruge and K. Stüben. "Algebraic Multigrid." In *Multigrid Methods*, Frontiers in Applied Mathematics, pp. 73–130. SIAM, Philadelphia, PA, 1987.

[RWZ94] Andreas Rieder, Raymond Wells, Jr., and Xiaodong Zhou. "On the Robustness of the Damped V-cycle of the Wavelet Frequency Decomposition Multigrid Method." *Computing Archives for Informatics and Numerical Computation*, **53**(2):155–171, 1994.

[Saa96] Yousef Saad. *Iterative Methods for Sparse Linear Systems.* PWS Publishing Company, Boston, MA, 1996.

[Sch90] A. Schüller. "A Multigrid Algorithm for the Incompressible Navier-Stokes Equations." In *Numerical Methods for Advection-Diffusion Problems (Kiel, 1989)*, volume 30 of *Notes on Numerical Fluid Mechanics*, pp. 124–133. Vieweg, Braunschweig, 1990.

[Sch98] Thomas Schlinkmann. "Multigrid and Multiscale Decompositions." In *Large-Scale Scientific Computations of Engineering and Environmental Problems (Varna, 1997)*, volume 62 of *Notes on Numerical Fluid Mechanics*, pp. 18–41. Vieweg, Braunschweig, 1998.

[Sta98] Gerhard Starke. "Multilevel Computation of Streamline Diffusion Finite Element Discretizations for Convection-Diffusion Problems." In *Large-Scale Scientific Computations of Engineering and Environmental Problems (Varna, 1997)*, volume 62 of *Notes on Numerical Fluid Mechanics*, pp. 281–289. Vieweg, Braunschweig, 1998.

[Stu99] K. Stüben. "Algebraic Multigrid (AMG): An Introduction with Applications." Technical Report 70, GMD, November 1999.

[SV93] Fadil Santosa and Michael Vogelius. "First Order Corrections to the Homogenized Eigenvalues of a Periodic Composite Medium." *SIAM Journal on Applied Mathematics*, **53**:1636–1668, 1993.

[SWZ85] P. Sonneveld, P. Wesseling, and P. M. de Zeeuw. "Multigrid and Conjugate Gradient Methods as Convergence Acceleration." In D. J. Paddon and H. Holstein, editors, *Multigrid Methods for Integral and Differential Equations (Bristol, 1983)*, volume 3 of *Institute of Mathematics and its Applications Conference Series*, pp. 117–167. Oxford University Press, New York, NY, 1985.

[TW98] Wei-Pai Tang and W. L. Wan. "Sparse Approximate Inverse Smoother for Multi-Grid." CAM Report 98-18, UCLA, March 1998.

[VMB96] P. Vaněk, J. Mandel, and M. Brezina. "Algebraic Multigrid by Smoothed Aggregation for Second and Fourth Order Elliptic Problems." *Computing*, **56**:179–196, 1996.

[VW95a] Panayot S. Vassilevski and Junping Wang. "Stabilizing the Hierarchical Basis by Approximate Wavelets, I: Theory." CAM Report 95-47, UCLA, December 1995.

[VW95b] Panayot S. Vassilevski and Junping Wang. "Stabilizing the Hierarchical Basis by Approximate Wavelets, II: Implementation and Numerical Results." CAM Report 95-48, UCLA, December 1995.

[Wan93a] Junping Wang. "Convergence Analysis of Multigrid Algorithms for Nonselfadjoint and Indefinite Elliptic Problems." *SIAM Journal of Numerical Analysis*, **30**(1):275–285, February 1993.

[Wan93b] Junping Wang. "Convergence Analysis of the Schwarz Algorithm and Multilevel Decomposition Iterative Methods II: Nonselfadjoint and Indefinite Elliptic Problems." *SIAM Journal of Numerical Analysis*, **30**(4):953–970, August 1993.

[Wan98] Wing Lok Wan. *Scalable and Multilevel Iterative Methods.* PhD thesis, UCLA, 1998.

[WCS98] Wing-Lok Wan, Tony F. Chan, and Barry Smith. "An Energy-Minimizing Interpolation for Robust Multigrid Methods." CAM Report 98-6, UCLA, February 1998.

[Web94] R. Webster. "An Algebraic Multigrid Solver for Navier-Stokes Problems." *International Journal for Numerical Methods in Fluids*, **18**(8):761–780, 1994.

[Web96] R. Webster. "An Algebraic Multigrid Solver for Navier-Stokes Problems in the Discrete Second-Order Approximation." *International Journal for Numerical Methods in Fluids*, **22**(11):1103–1123, 1996.

[Wes87] P. Wesseling. "Linear Multigrid Methods." In *Multigrid Methods*, Frontiers in Applied Mathematics, pp. 31–55. SIAM, Philadelphia, PA, 1987.

[Xu92a] Jinchao Xu. "Iterative Methods by Space Decomposition and Subspace Correction." *SIAM Review*, **34**(4):581–613, December 1992.

[Xu92b] Jinchao Xu. "A New Class of Iterative Methods for Nonselfadjoint or Indefinite Problems." *SIAM Journal of Numerical Analysis*, **29**(2):303–319, April 1992.

[Xu97] Jinchao Xu. "An Introduction to Multilevel Methods." In *Wavelets, Multilevel Methods, and Elliptic PDEs (Leicester, 1996)*, Numerical Mathematics and Scientific Computation, pp. 213–302. Oxford University Press, New York, NY, 1997.

[Yav98] Irad Yavneh. "Coarse-Grid Correction for Nonelliptic and Singular Perturbation Problems." *SIAM Journal on Scientific Computing*, **19**(5):1682–1699, September 1998.

[YVB98] Irad Yavneh, Cornelis H. Venner, and Achi Brandt. "Fast Multigrid Solution of the Advection Problem with Closed Characteristics." *SIAM Journal on Scientific Computing*, **19**(1):111–125, January 1998.

[Zee90] P. M. de Zeeuw. "Matrix-Dependent Prolongations and Restrictions in a Blackbox Multigrid Solver." *Journal of Computational and Applied Mathematics*, **33**:1–27, 1990.

[Zee93a] P. M. de Zeeuw. "Incomplete Line LU as Smoother and as Preconditioner." In *Incomplete Decomposition (*ILU*) – Algorithms, Theory, and Applications (Kiel, 1992)*, volume 41 of *Notes on Numerical Fluid Mechanics*, pp. 215–224. Vieweg, Braunschweig, 1993.

[Zee93b] P. M. de Zeeuw. "Multigrid and Advection." In *Numerical Methods for Advection-Diffusion Problems*, volume 45 of *Notes on Numerical Fluid Mechanics*, pp. 335–351. Vieweg, Braunschweig, 1993.

[ZV85] P. M. de Zeeuw and E. J. Van Asselt. "The Convergence Rate of Multi-Level Algorithms Applied to the Convection Diffusion Equation." *SIAM Journal of Scientific Statistical Computing*, **6**(2):492–502, April 1985.