Multigrid Acceleration Techniques and Applications to the Numerical Solution of Partial Differential Equations

By

Jun Zhang

B.S. July 1985, Chongqing University, China
M.Phil. May 1996, The George Washington University, U.S.A.

A Dissertation submitted to

The Faculty of

Columbian School of Arts and Sciences
of The George Washington University in partial satisfaction
of the requirements for the degree of Doctor of Philosophy

May 18, 1997

Dissertation directed by

Murli M. Gupta
Professor of Mathematics

Acknowledgments

I would like to acknowledge with gratitude various support received from the faculty members of the Department of Mathematics at the George Washington University during my three and a half years of Ph.D. studies. Special thanks are due to my dissertation advisor, Professor Murli M. Gupta, for his guidance and advice. It is him who introduced me to the wonderful multigrid world and provided every kinds of help and advice whenever needed. Support from the members of departmental Graduate Committee, especially from Professor Rodica Simion, is vital for me to be able to finish my Ph.D. studies in such a short time period.

Academic advice received from outside the George Washington University is gratefully acknowledged. Professor Donald G. M. Anderson at Harvard University has been continuously supporting me in my effort to publish my research work. His critiques on my manuscripts taught me how to write better research papers. Professor Scott R. Fulton at Clarkson University had been patiently answering my basic questions about the multigrid methods during the early stage of my studies and many of my multigrid computer codes were evolved from the multigrid control cycles obtained from him. Professor Homer F. Walker at the Utah State University provided me with advice on the minimal residual smoothing techniques and related sources of references, which turned out to be one of the key parts of my dissertation. Discussions with Dr. Irfan Altas at Charles Sturt University, Australia, were encouraging and fruitful.

The complete draft of this dissertation was carefully read by Professors Murli M. Gupta, Frank E. Baginski, Kevin G. Hockett, E. Arthur Robinson, Jr., Michael F. Moses, Dr. Irfan Altas and Mr. Jules Kouatchou. I thank them for their time and their critiques which improved the presentation of this dissertation.

Continuous encouragement from my former advisor, Dr. John A. Belward at the University of Queensland, Australia, and from Professors Yongwu Rong and Valentina Harizanov here at the George Washington University, has always been an assurance of confidence in the pursuit of my studies. They deserve a special thank-you.

My deepest gratitude goes to my wife, Yao Han, who has been sharing my joy and pain during the entire process of the studies. Without her understanding and full support, this dissertation would have never been finished. My son, Oliver, has been a source of relief when I was tired of debugging codes and proofreading manuscripts.

The never ending support from my parents has been the initial and final driving force to push me to overcome every obstacles to mature to be a scientist.

ABSTRACT OF DISSERTATION

Multigrid Acceleration Techniques and Applications to the Numerical Solution of Partial Differential Equations

Multigrid methods are extremely efficient for solving linear systems arising from discretized elliptic partial differential equations. For these problems, a few multigrid cycles are sufficient to obtain approximate solutions that are at least accurate to the level of the truncation error. However, difficulties associated with problems that are non-elliptic, or that have non-elliptic components, such as those described by the convection-diffusion equations and the incompressible Navier-Stokes equations with high Reynolds numbers, frequently cause a significant decrease in the efficiency of the standard multigrid methods. The convergence degradation gets worse when high-resolution discretization schemes are employed with the standard multigrid methods to obtain high accuracy numerical solutions.

The purpose of this study is to develop efficient multigrid acceleration techniques to speed up the convergence of the multigrid iteration process and to apply these techniques to obtain high accuracy numerical solutions of the partial differential equations in computational fluid dynamics. It is shown by analysis and numerical computations that standard multigrid methods can be significantly accelerated and yield highly improved convergence at negligible extra cost. Some acceleration techniques developed in this research even reduce the cost of the standard multigrid methods, in addition to providing satisfactory convergence acceleration. Other techniques have been shown to be essential for some problems to converge. One important feature that distinguishes these acceleration techniques from existing ones is that they do not require that the coefficient matrix be symmetric and positive definite and thus have the potential to be applied to a wider range of practical problems. Another feature of these techniques is that they can be parallelized.

Of particular importance to this work is the combination of these acceleration techniques with the high-order finite difference discretization schemes to construct stable multigrid solvers for obtaining fast and high accuracy numerical solutions of the convection-diffusion equations and of the incompressible Navier-Stokes equations with high Reynolds numbers.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

This research has been aimed at a better understanding of the difficulties associated with developing efficient acceleration techniques for the multigrid solvers for partial differential equations, especially for those equations that are non-elliptic or containing non-elliptic components.

Standard multigrid methods have been shown to be extremely efficient for solving linear systems that result from discretized elliptic partial differential equations, e.g., the Poisson type equations [6]. For these problems, a few standard multigrid cycles without any acceleration scheme are sufficient to obtain solutions with errors that are well below the truncation errors [7]. In recent years the multigrid techniques have been investigated by more and more researchers, and are employed in a much more general context and in a great number of fields. Several books have been published which are exclusively devoted to the multigrid [13, 34, 69].

However, difficulties which are associated with problems that are non-elliptic, or have non-elliptic components, such as those described by the convection-diffusion equations and the steady-state incompressible Navier-Stokes equations with high Reynolds numbers, frequently cause a significant decrease in the efficiency of the standard multigrid methods. This dissertation has been aimed at developing efficient multigrid acceleration techniques to speed up the convergence of the standard multigrid iteration process and to apply these techniques to obtain high accuracy numerical solution of the partial differential equations. It is shown by analysis and numerical computations that the standard multigrid methods can be significantly accelerated and yield highly improved convergence at negligible extra cost. Some acceleration techniques developed in this research even reduce the cost of the standard multigrid methods, in addition to provide satisfactory convergence acceleration. Other techniques have been shown to be essential for some problems to converge. One important feature which distinguishes our acceleration techniques from existing ones is that our acceleration techniques do not require that the coefficient matrix be symmetric and positive definite and thus have potential to be applied to a wider range of problems of practical importance. Most acceleration techniques developed in this dissertation aiming at the non-elliptic problems have also been shown to accelerate multigrid convergence of the elliptic problems.

Of particular importance of this research is to combine these acceleration techniques with the high-order finite difference discretization schemes to construct stable multigrid solvers for obtaining fast and high accuracy numerical solution of the convection-diffusion equations and of the steady-state incompressible Navier-Stokes equations with high Reynolds numbers.

We begin this dissertation with a description of basic concepts of the multigrid methods.

1.1 Multigrid Methods

Consider a large linear system of equations

$$A^h u^h = f^h \tag{1.1}$$

that arises from discretizing some differential or integro-differential equations on a discrete domain Ω^h with meshsize h. The superscript h in Eq. (1.1) may be dropped when there is no confusion on the discretized space in question.

The multigrid idea is based on the observation that classical iterative methods, such as the Gauss-Seidel relaxation and successive over-relaxation (SOR), strongly damp oscillatory error components but converge slowly for smooth error components. Hence, classical iterative methods work very well for the first few iterations. Inevitably, however, the convergence slows down and the entire iterative scheme appears to stall [13]. One effective way of removing the smooth error components is to approximate them on a coarser grid with larger meshsize H (usually $H = 2h$, but this is not necessary), on which the smooth errors become more oscillatory and can be removed quickly by relaxation methods. Solving the residual equations (the equations for the errors) on the coarse grid directly and interpolating the correction back to the fine grid give the two-level method. The two-level method is an important theoretical and debugging tool in developing the multi-level or multigrid method. However, since the number of equations to be solved on the coarse grid is proportional to the original number of equations, the direct solver employed to obtain the coarse grid solution will be relatively slow and costly. The multigrid method exploits the fact that the subproblem (residual equations) on the coarse grid has the same structure as the original problem on the fine grid, so the basic idea of the two-level method can be applied recursively, i.e., on the coarse grid, the subproblem (residual equations) is not solved exactly, instead, oscillatory error components are damped out and the smooth errors are projected to yet a coarser grid to be solved there. However, the two-level method contains all the basic ideas of the multigrid method, which in turn can be viewed as recursively using coarser grid solutions to approximate the direct solver required by the coarse grid in the two-level method. Hence, the main difference between the two-level method and the multigrid method is the computational efficiency, not the theoretical issue of accelerating basic convergence.

For a wide class of different problems, the multigrid method is found to be the most effective method and the cost of the multigrid method is independent of the grid meshsize [34, 69].

Briggs' book [13] is generally considered excellent as an introduction to basic multigrid concepts. Advanced multigrid techniques are presented in Brandt's guide [7]. For detailed treatments, we recommend the books of Hackbusch [34] and Wesseling [69].

1.2 Multigrid Acceleration Techniques

For elliptic problems such as the Poisson type equations it has been believed that there is no advantage of using any acceleration scheme for the standard multigrid methods. A few standard multigrid cycles are sufficient to bring the algebraic errors well below the level of truncation errors [6]. However, we have found that carefully designed acceleration schemes may accelerate the standard multigrid methods (see Chapter 2, Appendix A and [77, 78]). Although this will not arouse any excitement if we just solve a single Poisson equation, since the savings of the CPU time for a fast Poisson solver are not a big deal on modern computers. The acceleration becomes attractive if a fast Poisson solver is called repeatedly for many times as a subroutine in a major computational project, such as solving the steady-state incompressible Navier-Stokes equations, the accumulated savings in CPU time can be substantial.

For non-elliptic problems or problems containing non-elliptic components, acceleration schemes are generally required to obtain solution within reasonable CPU time, sometimes are even necessary for convergence. Efficient acceleration schemes must first be cost-effective, which means to accelerate the original multigrid iteration with reasonable (in most cases negligible) extra cost. There exist acceleration schemes that accelerate the multigrid convergence in some situations, but the cost of the acceleration is too prohibitive to be attractive for practical applications [46, 62]. Our acceleration schemes developed in this dissertation are all cost-effective, in most cases the acceleration cost is negligible.

The second criterion for an efficient acceleration scheme is the applicability. An efficient acceleration scheme must be applicable to a wide range of problems of practical importance. Existing acceleration schemes such as the over-correction scheme [62] and the steplength optimization technique [46] require that the coefficient matrix be symmetric and positive definite (SPD). The SPD requirement severely limits the application of these acceleration techniques. Usually the multigrid methods can solve many SPD problems very efficiently and this fact renders these acceleration techniques much less useful in practice. On the other hand, our acceleration schemes developed in this dissertation research do not require that the coefficient matrix be SPD and thus are much more applicable to solving practical problems.

1.3 Applications to Partial Differential Equations

The multigrid methods can be used to solve a large group of differential equations efficiently. In this dissertation, we concentrate our attention on obtaining stable, efficient and accurate numerical solution of the convection-diffusion equations and the steady-state incompressible Navier-Stokes equations, especially when the convection is the dominant phenomenon. These equations are important in modern scientific computation, especially in computational fluid dynamics. But obtaining stable and accurate numerical solution to these equations is a current research topic that many researchers are devoting their energy.

One of the non-trivial problems associated with the solution process is to efficiently solve the linear system (1.1) results from a discretized partial differential equation. For the convection-diffusion equation with large Reynolds number, the coefficient matrix of Eq. (1.1) is nonsymmetric and non-positive definite. Many classical iterative methods become divergent when they are employed to solve Eq. (1.1) if the cell Reynolds number is greater than a certain constant. Standard conjugate gradient type methods cannot be applied directly due to the lack of symmetry in the coefficient matrix. For other Krylov subspace methods, finding a suitable preconditioner for the convection-diffusion equations is not always easy.

Standard multigrid methods become divergent as a result of the divergence of the smoother. Multigrid methods with higher-order finite difference schemes usually converge slowly and acceleration schemes are therefore needed to obtain solutions in a reasonable time.

1.4 Finite Difference Schemes

Standard multigrid methods usually employ smoothers based on the second-order central difference schemes. These schemes may be efficient for the Poisson type equations. For problems like the convection-diffusion equations with large Reynolds numbers, standard multigrid methods diverge due to the divergence of the smoothers. Although there exist complicated higher-order upwind difference scheme, the popular trend is to use the first-order upwind difference scheme for the convective terms and the second-order central difference scheme for the diffusive terms. This modification guarantees convergence, but only gives solution of first-order accuracy. Hence, defect-correction or double discretization techniques are used by many researchers to obtain higher-order accuracy solution, in which the first-order upwind scheme is used for relaxation and the second-order central difference scheme is used for residual computation [3, 33]. The defect-correction multigrid methods may yield second-order accuracy solution for the convection-diffusion equations with smooth convective coefficients. According to our numerical experiments [81], for problems with highly oscillatory convective terms even the central difference scheme may not offer solution of second-order accuracy.

By employing the fourth-order compact finite difference schemes in the multigrid, we can design stable multigrid solvers for the convection-diffusion equations with highly oscillatory convective terms and with large Reynolds numbers. There is no need to use upwind scheme for stability. Furthermore, the fourth-order multigrid methods offer solution of fourth-order accuracy. The fourth-order multigrid method can further be accelerated by the acceleration techniques developed in this dissertation and the resulting accelerated fourth-order multigrid methods offer stable, fast and high accuracy solution for the convection-diffusion equations and the steady-state incompressible Navier-Stokes equations with large Reynolds numbers.

1.5 Main Conclusions

- Standard multigrid methods may be effectively accelerated.
- Most acceleration schemes are cheap and cost-effective.
- The convergence deterioration of the standard multigrid methods may be caused by the incorrect scale of the residuals that are projected to the coarse grid. Acceleration may be achieved by correctly scaling the residuals.
- When standard multigrid methods with the central difference scheme are unstable, the multigrid methods with the fourth-order compact finite difference schemes provide stability and efficiency.
- Classical relaxation methods can be turned into efficient smoothers by the acceleration schemes.
- High accuracy solution can come with cheap cost.

1.6 Summary and Structure of this Dissertation

This dissertation is composed of eight chapters and two appendices, each chapter starts with its introduction, with the exception of the first and the eighth chapters. Notations are introduced as the need arises and conclusions are given in each chapter separately.

Here are the summary of the major contributions of this dissertation:

- In Chapter 2, we explain the effect of the SOR relaxation on standard multigrid Poisson solver and develop two acceleration schemes for the five-point red-black Gauss-Seidel smoothing in the multigrid method for the Poisson equation. A long-time mis-understanding in multigrid community concerning the applicability of the SOR relaxation with the multigrid method for solving the Poisson equation is resolved.
- In Chapter 3, we unify the concept of the multigrid residual scaling techniques and prove the equivalence of the pre-scaling and post-scaling acceleration techniques. The equivalence proof clears the way for developing efficient and general-purpose pre-scaling acceleration schemes.

- In Chapter 4, we systematically introduce the minimal residual smoothing techniques to accelerate the multigrid convergence. Several application schemes are designed and numerical experiments are used to show the attractive acceleration rate.

- In Chapter 5, the minimal residual smoothing techniques introduced in Chapter 4 are analyzed and theoretical results are given. We show how the minimal residual smoothing accelerates the standard multigrid method.

- In Chapter 6, a fourth-order compact finite difference scheme is combined with the multigrid techniques to solve the convection-diffusion equations with high Reynolds numbers. A heuristic residual analysis is used to derive optimal residual injection factor for the high-order multigrid method.

- In Chapter 7, a fourth-order multigrid method is designed for solving the steady-state incompressible Navier-Stokes equations. Optimal residual scaling techniques are used to obtain stable, fast and high accuracy numerical solution of the driven cavity model problem for moderate to large Reynolds numbers.

- Chapter 8 contains the conclusions of this dissertation and the outlook for future research directions.

- In Appendix A, we give a practical example of the heuristic residual analysis technique and derive an optimal residual injection factor for the five-point multigrid Poisson solver.

- Appendix B contains some theoretical results on the stability of the fourth-order compact scheme and convergence of some iterative methods with this compact scheme.

Chapter 2

SOR Multigrid for Poisson Equation

2.1 Introduction

In this chapter, we concern ourself with the numerical solution of following two dimensional Poisson equation with Dirichlet boundary conditions, which we call the model problem,

$$\left.\begin{array}{rcll} -\Delta u(x,y) & = & f(x,y), & (x,y) \in \Omega, \\ u(x,y) & = & g(x,y), & (x,y) \in \partial\Omega, \end{array}\right\} \tag{2.1}$$

where

$$\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \tag{2.2}$$

is the Laplace operator. For simplicity, Ω is assumed to be the unit square, but other domains such as the rectangle are easily generalized. The discretization of Eq. (2.1) will be the usual accurate five-point central difference scheme, with a meshsize h and a truncation error of order h^2. The discretization generates a system of linear equations of the form (1.1), where A^h is a sparse, symmetric positive definite matrix.

An important part of the multigrid algorithms for the solution of discretized two dimensional Poisson-type problems is relaxation, whose purpose is to smooth the current error in the approximation, i.e., to efficiently reduce all error Fourier components that can not be approximated on the coarser grids. For various reasons such as good smoothing effect and obvious parallel potential, the red-black Gauss-Seidel method has been used extensively as a relaxation in the multigrid methods of many kinds [13].

It is well-known that, when used in an iterative method, a relaxation parameter may improve the convergence property. Relaxed Gauss-Seidel method is usually called the SOR (successive over relaxation) method, although under-relaxation is also included. Using an optimal relaxation parameter in SOR can accelerate the convergence dramatically in many cases, but finding the optimal parameter is not easy for most practical problems. In multigrid, the main role of relaxation is to smooth the error, not to reduce it. It has been observed that using relaxation parameters is not cost-effective in many situations, particularly when applied to Gauss-Seidel relaxation [7, 59]. For example, some slight improvement in the smoothing properties of Gauss-Seidel relaxation has been shown with slight over-relaxation in the case of the two dimensional Laplace operator, but the two additional floating-point operations required rendered this modification ineffective [59].

Probably due to the authoritative remarks made in [7, 59] that SOR is not cost-effective for multigrid for solving Laplace operator, research works in this part of the multigrid have been relatively inactive, comparing with others. Recently, there has been some interest in using acceleration schemes or SOR in multigrid [11, 73, 76]. In [11], Brandt and Mikulinsky investigated using recombining multigrid iterants, or "polynomial multigrid acceleration", to solve a group of problems with non-simply connected domains containing small "holes" or "islands". They used the acceleration on the coarse grids only, where the "island" cannot be presented directly; therefore it adds normally only a little work to the usual multigrid cycle. In [73], Yavneh proposed an over-relaxation for red-black Gauss-Seidel smoothing in multigrid for anisotropic elliptic partial differential operators. He showed that relaxation parameters are quite useful when the anisotropy is moderate and the same is true for isotropic operators in higher dimensions. In [76], four smoothing methods were compared for a multigrid method for the incompressible Navier-Stokes equations in general coordinates. Both under- and over-relaxation SORs were tested. It was shown that the under-relaxation parameter changes significantly as the problems change. For some problems, over-relaxation had to be used instead of under-relaxation. Furthermore, it seems that, in all the existing research works, relaxation parameters were used only when the convergence of the usual Gauss-Seidel method is very slow, and that using relaxation parameters for the model problem (2.1) has been carefully avoided.

In this chapter, we find the reasons why the multigrid V-cycle algorithm is not sensitive to SOR acceleration for Laplace operator in the conventional sense. This is because the first (restriction) half cycle is sensitive to under-relaxation only and the second (interpolation) half cycle to over-relaxation only. Based on this discovery, we present two acceleration schemes for the five-point red-black Gauss-Seidel smoother for the multigrid method to solve the model problem. The schemes, embedded in a multigrid V-cycle algorithm, have been tested by several numerical experiments and results in the acceleration of the average convergence rate by as much as 34% (see Table 2.6). The cost of implementing the algorithm is negligible, which is remarkably contrary to the remarks made in many monographs [6, 59]. Furthermore, the acceleration schemes are very simple, one can just add several lines to the existing codes.

We briefly introduce the multigrid V-cycle algorithm and the red-black Gauss-Seidel smoothing in Section 2.2. The relaxation analysis is conducted in Section 2.3. In Section 2.4, we analyze the suitability and cost of accelerating the multigrid V-cycle algorithm, and design two acceleration algorithms. In Section 2.5, we tested our schemes against the usual red-black Gauss-Seidel smoother by several numerical experiments. Some conclusions and remarks are given in Section 2.6.

2.2 Multigrid and Smoother

2.2.1 Multigrid V-cycle

We only give a brief sketch of a typical multigrid V-cycle algorithm. Detailed descriptions of some more general multigrid algorithms can be found in [7] and [34]. [13] is generally considered as an excellent introduction to multigrid.

Algorithm 2.2.1 *One iteration of a usual multigrid V-cycle for solving the linear system (1.1).*

```
Program $MG(f^h, u^h, h)$
        if ($h = coarsest$ ) then $u \leftarrow (A^h)^{-1} f^h$
        else
            $u \leftarrow relax(f, u, h)$
            $r \leftarrow f - A^h u$
            $\bar{r} \leftarrow Rr$            ! $R$ is a projection operator
            $v \leftarrow 0$
            $v \leftarrow MG(\bar{r}, v, 2h)$
            $u \leftarrow u + Pv$        ! $P$ is an interpolation operator
            $u \leftarrow relax(f, u, h)$
        end if
```

One iteration of a simple multigrid V-cycle consists of smoothing the error using a relaxation technique, solving an approximation to the smooth error equation on a coarse grid, interpolating the error correction to the fine grid, and finally adding the error correction into the approximation. An important aspect of the multigrid method is that the coarse grid solution can be approximated by recursively using the multigrid idea. That is, on the coarse grid, relaxation is performed to reduce high frequency errors followed by the projection of a correction equation on yet a coarser grid, and so on. Thus, the multigrid method requires a series of different (but similar) problems to be solved on a hierarchy of grids with different meshsizes. A multigrid V-cycle algorithm is the computational process that goes from the finest grid down to the coarsest grid and back from the coarsest up to the finest. We summarize one iteration of this procedure in Algorithm 2.2.1, where $MG(f^h, u^h, f^h)$ is a recursive procedure and $relax(f, u, h)$ is a relaxation procedure (smoother). A common variation of the V-cycle algorithm is to do two correction cycles at each level before returning to the next higher level; this is the W-cycle algorithm. However, in this chapter, we restrict our attention to the multigrid V(1,1)-cycle algorithm, which performs one relaxation sweep before projecting the residual to the coarse grid and one after finishing the coarse grid correction step. The W-cycle algorithm will be discussed in subsequent chapters. Interested readers may also consult the listed references for more details on other multigrid cycles [6, 13, 34, 69].

The well-behaved model problem may require a number of V-cycles to satisfy a prescribed tolerance, denoted by ϵ. Other problems such as the convection-diffusion equation [23, 30, 31] (also see Chapters 6 and 7) require many V-cycles (or even diverge). Therefore, the number of multigrid V-cycles, denoted by $\mathcal{MV}$, required to satisfy a given ϵ is an important factor of testing the computational efficiency of an algorithm in experiments. The average residual reduction factor, or the contraction number, κ, is defined as

$$\kappa = \left(\frac{||r_n||_2}{||r_0||_2}\right)^{\frac{1}{\mathcal{MV}}},$$

which is a more accurate convergence indicator. Here $||r_0||_2$ is the initial discrete residual in L_2 norm, $||r_n||_2$ is the final discrete residual.

The smoothing factor, defined below, is mainly used to predict the performance of a multigrid algorithm in theory.

2.2.2 Five-Point Red-Black Gauss-Seidel Relaxation

For each internal grid point (i, j) in the discretized Ω, (1.1) is a system of equations of the form

$$u_{i,j} = \frac{1}{4}\left[u_{i-1,j} + u_{i,j-1} + u_{i,j+1} + u_{i+1,j} + h^2 f_{i,j}\right].$$

This means that, in each relaxation sweep, the value on a grid point is updated by the average of its four immediate neighbor's values. If we re-arrange the grids in an alternative red and black order (see Figure 2.1), the relaxation can be carried out simultaneously on red points and black points independently. The red-black Gauss-Seidel may therefore be considered as two Jacobi relaxations, each carried out on roughly half of the grid points independently. This idea certainly benefits the parallel computers, but it has been shown that, even on serial computers, when used in multigrid, the red-black ordering is slightly superior to the natural ordering [24, p. 369]. This idea (pattern relaxation) can be generalized to multiple colors immediately.

2.3 Relaxation Analysis

2.3.1 Conventional Analysis

Pattern relaxation methods, such as red-black Gauss-Seidel have been studied and analyzed extensively in the context of multigrid algorithms [59]. It has been shown that when the operator is essentially isotropic (the coefficient terms are equal, like the Laplace operator), the most cost-effective known smoother (at least for low dimension) is probably the point Gauss-Seidel relaxation in red-black ordering [72]. To avoid confusing readers by using different notations and formulas with almost identical meanings, we try to use formulas of [72] in this subsection and describe the smoothing analysis here briefly. To this end, we need to normalize the Laplace operator in (2.1) slightly to

$$L = \frac{1}{2}\Delta = \frac{1}{2}\frac{\partial}{\partial x^2} + \frac{1}{2}\frac{\partial}{\partial y^2},$$

so that the sum of the coefficients is a unit.

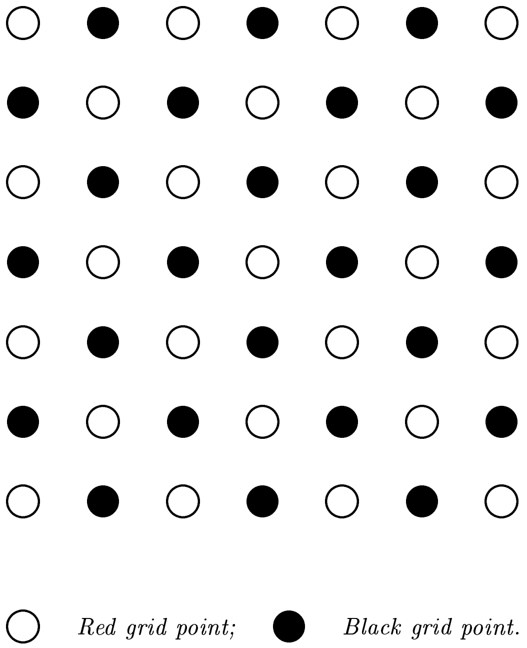

Figure 2.1: A 7 × 7 grid points marked in red-black ordering. Boundary points are not included.

For Gauss-Seidel relaxation of five-point operators (in two dimensions) in red-black ordering, Fourier components are not eigenfunctions, and two-dimensional subspaces of error Fourier components are invariant [7, 39, 59, 69]. This means that the red-black relaxation couples each Fourier component $\exp(\mathrm{i}\mathbf{k} \cdot \theta)$ only with $\exp(\mathrm{i}\mathbf{k} \cdot \bar{\theta})$, where

$$\theta = (\theta_1, \theta_2), \qquad \bar{\theta} = (\bar{\theta}_1, \bar{\theta}_2),$$

$$-\pi < \theta_i \leq \pi,$$

and

$$\bar{\theta}_i = \theta_i - \mathrm{sign}(\theta_i)\pi, \qquad i = 1, 2. \tag{2.3}$$

Here sign(0) is defined as -1. θ_i may take on discrete values only, however, for convenience in employing analytic techniques, we assume that θ_i can take on any value in $(-\pi, \pi]$ in the sequel.

An element θ_i of mode θ is called smooth if $-\pi/2 < \theta_i \leq \pi/2$. Otherwise it is rough. A Fourier mode θ is called a high frequency if at least one of its elements θ_i is rough. Otherwise it is a low frequency.

We denote the subspace of high frequency θ's by Θ_{high} and the subspace of low frequency θ's by Θ_{low}. Corresponding to this definition, the coarse grid space is split into an orthogonal sum of Θ_{high} and Θ_{low} [59, p. 121].

Fourier modes that are high frequencies on the fine grid are not "visible" on the coarse grid, in the sense that they can not be approximated on the immediate coarser grid that is assumed to provide the coarse-grid correction, since they alias with other components. Hence they need to be smoothed efficiently by the relaxation on the fine grid. Evidently, all pairs $(\theta, \bar{\theta})$ consist of either two high frequencies or one high and one low frequencies [73]. Without loss of generality, we assume that $\bar{\theta}$ is always a high frequency.

The relaxation operator

$$R(\theta, \bar{\theta}) = \begin{bmatrix} r_{11}(\theta, \bar{\theta}) & r_{12}(\theta, \bar{\theta}) \\ r_{21}(\theta, \bar{\theta}) & r_{22}(\theta, \bar{\theta}) \end{bmatrix},$$

which gives the amplitude of error Fourier components $(\exp(\mathbf{ik} \cdot \theta), \exp(\mathbf{ik} \cdot \bar{\theta}))^T$ after one full relaxation sweep, when multiplied by their amplitudes before the sweep. It can be obtained by multiplying the operators of two weighted-Jacobi half-sweeps performed over the red points (whose index-sums are even) and the black points (whose index-sums are odd). A sweep over the red points amplifies components of red-point errors by the weighted-Jacobi symbol r_ω, without affecting the black-point errors, and vice versa. Since red-point and black-point error components can be expressed as sums and differences of the pairs of Fourier components, each half sweep operator can be written as a two by two matrix [59, 73]

$$\frac{1}{2}\begin{bmatrix} r_\omega(\theta) + 1 & r_\omega(\bar{\theta}) - 1 \\ r_\omega(\theta) - 1 & r_\omega(\bar{\theta}) + 1 \end{bmatrix} \qquad \text{and} \qquad \frac{1}{2}\begin{bmatrix} r_\omega(\theta) + 1 & -r_\omega(\bar{\theta}) + 1 \\ -r_\omega(\theta) + 1 & r_\omega(\bar{\theta}) + 1 \end{bmatrix}$$

for the red-point and black-point relaxation half-sweeps, respectively. R is given by their product:

$$R = \frac{1}{4}\begin{bmatrix} (r_\omega(\theta)+1)^2 + (r_\omega(\bar{\theta})-1)(1-r_\omega(\theta)) & (r_\omega(\theta)+1)(1-r_\omega(\bar{\theta})) + r_\omega(\bar{\theta})^2 - 1 \\ (r_\omega(\bar{\theta})+1)(1-r_\omega(\theta)) + r_\omega(\theta)^2 - 1 & (r_\omega(\bar{\theta})+1)^2 + (r_\omega(\bar{\theta})-1)(1-r_\omega(\theta)) \end{bmatrix}. \tag{2.4}$$

Here, the weighted Jacobi relaxation symbols are given by

$$r_\omega(\theta) = 1 - \omega(1-c), \qquad r_\omega(\bar{\theta}) = 1 - \omega(1+c), \tag{2.5}$$

where

$$c = \frac{1}{2}[\cos(\theta_1) + \cos(\theta_2)], \tag{2.6}$$

and ω is the relaxation parameter.

In relaxation operators for which the Fourier components are eigenfunctions (such as the damped Jacobi relaxation for the present problem), R is a scalar, and the smoothing factor $\bar{\mu}$ is conventionally defined as the largest absolute value of R over the space of high frequencies. $\bar{\mu}$ must give some information on the asymptotic reduction of high frequency error components by the relaxation, given other parts of the multigrid cycle are idealized. This motivation led to the following highly successful extension to general R, introduced in [7] and [59, p. 91]: apply R and then annihilate the low frequencies, while leaving the high frequencies unchanged, by projecting on the space of high frequencies. When R is a two by two matrix, the projection operator P, which acts as the idealized coarse-grid operator, is given by

$$P = \begin{bmatrix} p(\theta) & 0 \\ 0 & p(\bar{\theta}) \end{bmatrix}, \tag{2.7}$$

where $p = 0$ for a low frequency argument and 1 otherwise. (Hence, our high frequency $\bar{\theta}$ assumption implies $p(\bar{\theta}) = 1$). The smoothing factor $\bar{\mu}$, when a single relaxation sweep is performed between successive coarse-grid corrections, is conventionally defined by

$$\bar{\mu} = \sup_h \max_\theta \rho(PR), \tag{2.8}$$

where ρ denotes spectral radius. The implication of the supremum is that θ is allowed any value in $(-\pi, \pi]$. This will henceforth be implicitly assumed, and $\sup_h$ will be omitted for brevity. Let

$$\bar{\mu}^{low} = \max_{\theta \in \Theta_{low}} \rho(PR),$$

and

$$\bar{\mu}^{high} = \max_{\theta \in \Theta_{high}} \rho(PR).$$

Hence

$$\bar{\mu} = \max(\bar{\mu}^{low}, \bar{\mu}^{high}) \tag{2.9}$$

in the conventional sense. The optimal relaxation parameter, $\bar{\omega}_{optimal}$, is the one that minimizes $\bar{\mu}$, yielding $\bar{\mu}_{optimal}$. It has been established [59, 69, 73] that $\bar{\mu}_{optimal}$ satisfies

$$\bar{\mu}_{optimal} \leq \bar{\mu}_{\omega=1} = \frac{1}{4}. \tag{2.10}$$

2.3.2 New Approach

The conventional smoothing analysis does not distinguish the different reduction effect on the low and high frequencies by relaxation (only consider the reduction of high frequency). In multigrid, low frequency and high frequency are affected differently by restriction and interpolation. Therefore, it makes sense to consider μ^{low} and μ^{high} separately, and only regard the conventional $\bar{\mu}$ as a reference. To distinguish μ^{low} and μ^{high} from the conventional context, we call them the reduction factors of low and high frequencies respectively. Since these reduction factors are obviously a function of ω, we will make this relation clear in the following sections of this chapter.

For $\theta \in \Theta_{low}$, $p(\theta)$ vanishes as noted, leaving

$$\mu^{low}(\omega) = \max_{\theta \in \Theta_{low}} |r_{22}|. \tag{2.11}$$

Now, $\theta \in \Theta_{low}$ implies by (2.6) that c can take on any value between 0 (when all components of θ equal $\frac{\pi}{2}$), and 1 (when all components of θ equal zero). Hence (2.4), (2.5) and (2.11) yield after rearrangement

$$\mu^{low}(\omega) = \frac{1}{2} \max_{c \in [0,1]} |\omega^2 c(1+c) - 2\omega(1+c) + 2|. \tag{2.12}$$

In order to find the maximum of the right-hand side of (2.12) over c in the relevant range, we must check the end-points, $c = 0$ and $c = 1$, and the point at which the derivative with respect to c vanishes. Setting $c = 0$ yields $|\omega - 1|$. This implies that

$$0 < \omega < 2. \tag{2.13}$$

Setting $c = 1$ yields $(\omega - 1)^2$, which is smaller than $|\omega - 1|$ in this range of ω. Equating the derivative with respect to c to zero yields after rearrangement

$$\mu^{low}(\omega) = \frac{1}{8}|\omega^2 + 4\omega - 4| \tag{2.14}$$

at

$$c = \frac{2-\omega}{2\omega}.$$

This c is in $[0, 1]$ so long as

$$\frac{2}{3} \le \omega \le 2. \tag{2.15}$$

We do not want to make the assumption to restrict $\omega \in (1, 2)$ as Yavneh did and reached an over-relaxation scheme for anisotropic operators in [73]. Instead, we try to find an ω_{low} that minimizes the low frequency errors. From (2.10) and (2.14), we have the following inequality

$$\mu^{low}(\omega) = \frac{1}{8}|\omega^2 + 4\omega - 4| < \frac{1}{8} = \mu^{low}(1). \tag{2.16}$$

Solving inequality in the center of (2.16) gives us a favorable range of ω

$$\sqrt{7} - 2 < \omega < 1, \tag{2.17}$$

which reduces the low frequency errors better than $\omega = 1$. Combining (2.13), (2.15) and (2.17) gives the relevant range of ω favoring the reduction of low frequency errors

$$\frac{2}{3} \le \omega_{low} < 1. \tag{2.18}$$

Hence, we have the following proposition:

Proposition 2.3.1 *Under-relaxation with $\omega \in [\frac{2}{3}, 1)$ accelerates the reduction of low frequency errors.*

In the range of (2.18), we find that

$$\omega_{low} = 2(\sqrt{2} - 1) \approx 0.8284271$$

minimizes μ^{low}, giving $\mu^{low}_{optimal} = 0$, ideally. This suggests that, without the presence of high frequency errors, one step coarse grid correction could remove all low frequency errors.

For $\theta \in \Theta_{high}$, P is the identity matrix, so $PR = R$, whose eigenvalues are given by [73]

$$\lambda^{high}_{1,2}(\omega) = \frac{1}{2}\left[r_{11} + r_{22} \pm \sqrt{(r_{11} - r_{22})^2 + 4r_{12}r_{21}}\right]. \tag{2.19}$$

Substitution from (2.4) and (2.5) yields

$$\begin{aligned} r_{11}+r_{22} &= \omega^2 c^2 - 2(\omega-1), \\ r_{11}-r_{22} &= \omega c(2-\omega), \\ 4r_{12}r_{21} &= c^2\omega^4(c^2-1). \end{aligned}$$

It follows that

$$\lambda_{1,2}^{high}(\omega) = \frac{1}{2}\left[\omega^2c^2 - 2(\omega-1) \pm \omega c\sqrt{\omega^2c^2 - 4(\omega-1)}\right]. \tag{2.20}$$

Furthermore, $\theta \in \Theta_{high}$ implies that at least one of θ's components is high frequency (by definition), but also that at least one of its components is low frequency, otherwise $\bar{\theta}$ would be a low frequency, by (2.3), in contradiction to our assumption. Hence, by (2.6), the relevant range of c in (2.20) is given by

$$|c| \leq \frac{1}{2}.$$

This bound is achieved when all of θ's components vanish, except that corresponds to $\theta = -\pi/2$. Note in (2.20) that $\lambda_1^{high}(c) = \lambda_2^{high}(-c)$, so it suffices to consider nonnegative values of c, and this will be assumed henceforth for convenience.

We then define $D_\omega(c) = \omega^2c^2 - 4(\omega-1)$. We find that if $D_\omega < 0$, $\lambda_{1,2}^{high}$ are complex, and $|\lambda_1^{high}| = |\lambda_2^{high}| = |\omega - 1|$, independent of c. This is also hold for $D_\omega = 0$. Moreover, $\omega^2c^2 - 2(\omega-1) \geq 0$, if $D_\omega(c) \geq 0$. Hence, only the "+" sign in front of the square root in (2.20) (corresponding to λ_1^{high}) needs to be considered when seeking the root that is larger in absolute value [73]. Thus we only consider $\mu^{high}(\omega) = \lambda_1^{high}(\omega)$. Moreover, in this case the corresponding root is evidently a monotonically increasing function of c [73, p. 188], so c reaches its upper bound 1/2. We then define

$$\mu^{high}(\omega) = \frac{1}{8}\left[\omega^2 - 8(\omega-1) + \omega\sqrt{\omega^2 - 16(\omega-1)}\right]. \tag{2.21}$$

As in the case of $\mu^{low}(\omega)$, we find that $\mu^{high}(1) = 1/4$, which agrees with (2.10). Furthermore, by solving the inequality $\mu^{high}(\omega) < \mu^{high}(1)$ for ω in the relevant range $(0, 2)$, we have an interval which favors the reduction of high frequency errors

$$1 < \omega < \frac{5}{4}. \tag{2.22}$$

Proposition 2.3.2 *Over-relaxation with $\omega \in (1, \frac{5}{4})$ accelerates the reduction of high frequency errors.*

Taking the derivative with respect to ω in (2.21), we find a critical point

$$\omega_{high} = 4(2-\sqrt{3}) \approx 1.0717968,$$

at which $\mu^{high}(\omega)$ reaches its minimum

$$\mu_{optimal}^{high} = 7 - 4\sqrt{3} \approx 0.0717968.$$

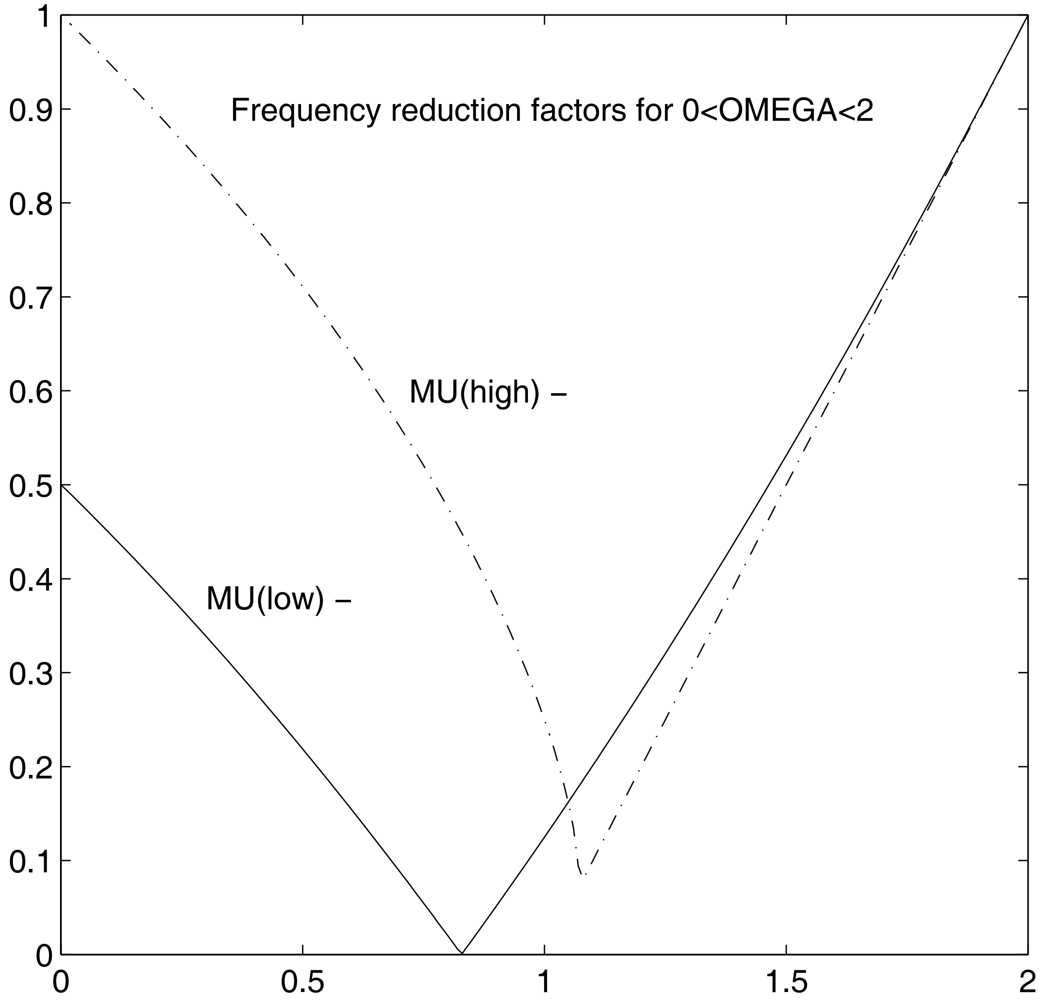

Figure 2.2: The graphs of $\mu^{high}(\omega)$ and $\mu^{low}(\omega)$.

This choice of ω minimizes the high frequency errors solely.

For a better understanding the behaviors of $\mu^{high}(\omega)$ and $\mu^{low}(\omega)$, we plot their graphs against $\omega \in (0, 2)$ (see Figure 2.2).

It is clear from the above analysis and from Figure 2.2 that we are in a dilemma to choose an optimal relaxation parameter. Under-relaxation reduces the low frequencies, but excites the high frequencies. On the other hand, over-relaxation smoothes the high frequencies, but amplifies the low frequencies. There may exist two solutions. One is to use two different parameters, which will be referred to as two-way acceleration. Another is to accelerate one half cycle only, which will be referred to as one-way acceleration.

This analysis shows the reason why the early applications of SOR in multigrid for Laplace operator could not yield satisfactory results. They failed to distinguish low and high frequencies, but treated them equally by using SOR as in the usual iterative methods on fixed grids. Thus any parameter may accelerate one half cycle, but deteriorates another half, yields virtually no acceleration. Moreover, the above analysis also implies that relaxation parameter should be used after the low and high frequencies are distinguished. In other words, SOR should be avoided on the finest grid of the first half cycle, i.e., before the application of restriction operator.

2.4 Analysis and Design of Algorithms

2.4.1 Suitability Analysis

We have removed the first finest grid from our candidate list for applying SOR. What about other grids?

From Figure 2.2, we note that the choice of a good parameter will be a compromise between minimization of high and low frequency errors. From $\omega = 1$ to $\omega = \omega_{low}$, $\mu^{low}(\omega)$ decreases monotonically while $\mu^{high}(\omega)$ increases monotonically. This suggests that the employment of an under-relaxation parameter benefits mainly the reduction of low frequency errors. This analysis basically agrees with conclusions of Stüben and Trottenberg [59, p. 128] that "the reduction of low frequencies is improved by a parameter, the reduction of high frequencies (and by that the smoothing factor) usually becomes even worse". We point out that Stüben and Trottenberg's smoothing factor is the conventional one $\bar{\mu}$, which considers high frequencies only.

From Figure 2.2, we find $\mu_{low}(\omega)$ and $\mu_{high}(\omega)$ are smooth for $\omega \in (\omega_{low}, \omega_{high})$. By carefully choosing a parameter, we may be able to reduce the low frequencies and restrict the high frequencies, or to smooth the high frequencies and control the low frequencies, at the same time. From Figure 2.2, it is obvious that $\omega_1 \in [\omega_{low}, 1)$ and $\omega_2 \in (1, \omega_{high}]$.

Hence, a parameter may not be used on the finest level of the first half V-cycle, where the main effort is to smooth the high frequency errors. The employment of a parameter might reduce the effect of smoothing, and the computational cost of doing some work on the finest grid is always expensive. On the other hand, a parameter may not be used on the very coarse levels. On those levels, the magnitudes of the residuals are already small and an acceleration helps little. Therefore, an under-relaxation parameter should be used on the levels where the the major low frequencies are removed and the high frequencies are hidden. These are the second and the third finest levels of the first (restriction) half cycle, as the restriction operator hides high frequencies. An over-relaxation parameter should be used on the levels where the major high frequencies emerge. These are the first and the second finest levels of the second (interpolation) half cycle, as the interpolation operator excites the high frequencies [13, p. 73].

2.4.2 Cost Estimate

The main computational cost of a multigrid V-cycle is on the finest level. It is usual to measure the computational cost in terms of the work unit, defined to be the computational cost of a relaxation sweep on the finest level. The work on each level is roughly proportional to the number of unknowns. The work units for a V-cycle algorithm is about 2.6667 [24, p. 366]. The inter-grid transferring adds additional $15 - 30\%$ to this amount for a total of 3.4667 work units. One relaxation on the second level of a half cycle costs about 0.25 work units.

The computation of the discrete error in L_2 norm requires about three floating-point operations. The five-point Gauss-Seidel has five floating-point operations, a parameter adds two more, if the parameter is stored and the code is modified properly. Hence one-way acceleration on the second finest level costs 0.0625 more work units than the usual Gauss-Seidel. This is about 1.8% of a V-cycle. One-way acceleration on the finest level is about 7.2% of a V-cycle. Two-way acceleration on the same level doubles the cost. For acceleration on the second finest level, the cost is negligible for most practical applications.

On parallel machines, the acceleration involve only local operations and no communication with other processors is needed. The relative cost of the acceleration is even lower.

2.4.3 Design of Algorithms

Based on our foregoing analyses, we propose two acceleration schemes. The one-way acceleration scheme is to embed an under-relaxation procedure, with a parameter $\omega_1 < 1$, in the five-point red-black Gauss-Seidel, on the second finest level of the first (restriction) half V-cycles. The two-way acceleration scheme, based on the one-way acceleration scheme, is to embed another over-relaxation procedure, with a parameter $\omega_2 > 1$, on the second finest level of the second (interpolation) half V-cycles.

The two-way acceleration multigrid V-cycle algorithm is given by Algorithm 2.4.1. The one-way acceleration multigrid V-cycle algorithm can be viewed as choosing $\omega_2 = 1$ in the two-way acceleration algorithm. The usual (unaccelerated) multigrid V-cycle can be viewed as choosing $\omega_1 = \omega_2 = 1$.

Algorithm 2.4.1 *One iteration of a modified multigrid V-cycle algorithm with two-way acceleration for solving the linear system (1.1).*

```
Program $MG(f^h, u^h, h)$
        if ($h =$ coarsest ) then $u \leftarrow {A^{-1}}^h f^h$
        else
            $\bar{u} \leftarrow relax(f, u, h)$
            if ($h =$ second finest and first half cycle) then
                $u \leftarrow \omega_1 \bar{u} + (1 - \omega_1) u$        ! one-way acceleration
            else
                $u \leftarrow \bar{u}$                                ! no acceleration
            end if
            $r \leftarrow f - Au$
            $\bar{r} \leftarrow Rr$                ! R is a projection operator
            $v \leftarrow 0$
            $v \leftarrow MG(\bar{r}, v, 2h)$
            $u \leftarrow u + Pv$        ! $P$ is an interpolation operator
            $\bar{u} \leftarrow relax(f, u, h)$
            if ($h$ = second finest and second half cycle) then
                $u \leftarrow \omega_2\bar{u} + (1 - \omega_2)u$      ! two-way acceleration
            else
                $u \leftarrow \bar{u}$                          ! no acceleration
            end if
        end if
```

A two parameter SOR method was first investigated by Golub and de Pillis, they used singular value decomposition (SVD) to improve SOR for the case that the coefficient matrix is symmetric [22]. The determination of the optimal parameters ω_1, ω_2 for their mixed strategy is still an open question. Their results have been generalized by Moussavi, who also considered a special nonsymmetric case [43]. More recently, Prager considered using two parameter iterative method to solve algebraic systems of domain decomposition type [45]. In multigrid, Stüben and Trottenberg mentioned that different parameters may be used for different Jacobi ω-relaxation steps, when more than one such relaxation steps are employed [59, p. 41]. Since Jacobi ω-relaxation is not efficient comparing with red-black relaxations, their remark was primarily intended for the completion of their discussions. We are not aware of any practical attempt to use two parameter SOR in multigrid in the form of what we proposed here for accelerating the reduction of the high and the low frequency errors separately.

Our simple single relaxation sweep analysis in Section 2.3.2 only gives a rough prediction of the range (upper and lower bounds) of the relaxation parameters. More sophisticated analytic method may be employed to obtain more accurate estimate of the optimal parameters. However, for a particular operator, the optimal parameters are fixed (independent of meshsize but dependent on a particular level) and therefore may be found by numerical methods. For our model problem (2.1), we find that $\omega_1 = 0.9$ and $\omega_2 = 1.06$ are optimal choices for our one-way and two-way acceleration schemes respectively. Although they may not be the best for every problem (any numerical method tolerates some variations), our numerical experiments show that they are in a small neighborhood of the best parameters and yield best results for $\mathcal{MV}$ and very near-best for κ for all of our test problems.

2.5 Numerical Experiments

We first define some notations. In addition to h, ϵ, $\mathcal{MV}$, κ, ω_1 and ω_2 defined above, we define $N = 1/h$ and $(N-1)^2$ is the number of unknowns on the finest grid. We also define $S_{\mathcal{MV}}$ and S_κ as the reduction rates in $\mathcal{MV}$ and κ respectively, for a particular acceleration scheme, with respect to the unaccelerated red-black Gauss-Seidel multigrid.

We experiment our algorithm by several test problems. For simplicity, the domain is the unit square for all tests. The definition of $f(x,y)$ and $g(x,y)$ are referred to the model problem (2.1) in the Section 2.1, here $g(x,y)$ is also the exact solution.

The original multigrid V-cycle Fortran 77 control routine, which uses a half-injection operator and a bi-linear interpolation operator, was provided by Scott Fulton at Clarkson University. The five-point red-black Gauss-Seidel with or without acceleration is used as the smoother wherever applicable. Unless otherwise indicated explicitly, all experiments are done on a SUN SPARCstation using double precision. All tests use the initial guess $u(x, y) = 0$ except on the boundary it takes the prescribed values of $g(x, y)$. Different meshsizes are employed. The program set an upper bound of 20 as the maximum number of V-cycles allowed. Therefore the cases that $\mathcal{MV}$s equal 21 should be considered as equivalent to very slow convergence or divergence.

Test Problem 2.5.1

$$\begin{cases} f(x,y) &= -x^2(1-x^2)(2-12y^2) - y^2(1-y^2)(2-12x^2); \\ g(x,y) &= x^2y^2(1-x^2)(1-y^2). \end{cases}$$

Test Problem 2.5.2

$$\begin{cases} f(x,y) &= -(x^2+y^2)\exp(xy); \\ g(x,y) &= \exp(xy). \end{cases}$$

Test Problem 2.5.3

$$\begin{cases} f(x,y) &= 52\cos(4x+6y); \\ g(x,y) &= \cos(4x+6y). \end{cases}$$

To show the overall behavior of $\mathcal{MV}$ and κ as functions of ω_1 and ω_2, we experiment Test Problems 2.5.1 to 2.5.3 by using our one-way acceleration algorithm for all $\omega_1 \in (0, 2)$ (fixing $\omega_2 = 1$). We also test the case in which a parameter is placed in the second (interpolation) half cycles for all $\omega_2 \in (0, 2)$ (fixing $\omega_1 = 1$). We compute, for all different $\omega_1 \in (0, 2)$ and $\omega_2 \in (0, 2)$, $\mathcal{MV}$ (multigrid V-cycle or iteration number) needed to achieve the prescribed tolerance $\epsilon = 10^{-8}$ and κ (contraction number). For all three problems, we use $N = 128$. The acceleration or SOR algorithms are embedded on different levels (from the first to the third finest) of the multigrid V-cycles.

Figures 2.3 to 2.5 show the change of $\mathcal{MV}$ and κ, as functions of ω_1 or ω_2, for Test Problems 2.5.1 to 2.5.3. For each figure, (a) shows $\mathcal{MV}$ against ω_1 (with $\omega_2 = 1$), (b) shows κ against ω_1 (with $\omega_2 = 1$), (c) shows $\mathcal{MV}$ against ω_2 (with $\omega_1 = 1$), (d) shows κ against ω_2 (with $\omega_1 = 1$). All tests show that under-relaxation in the first (restriction) half cycles with ω_1 in some range achieves better convergence than no acceleration, while over-relaxation in the first (restriction) half cycles deteriorates the convergence (see subfigures (a) and (b) in each figure). The figures also show that the acceleration on the second finest level is better than on the first finest level of the first (restriction) half cycle, which support our analysis made in Section 2.4 that acceleration is more efficient after restriction hides high frequency errors. Similarly, over-relaxation in the second (interpolation) half cycles with ω_2 in some range accelerates the convergence, while under-relaxation usually makes it worse. We also note that acceleration on the first finest level achieves slightly better results than on the second finest level of the

	Gauss-Seidel	One-Way Acceleration		Two-Way Acceleration	
N	$\mathcal{MV}$	$\mathcal{MV}$	$S_{\mathcal{MV}}$	$\mathcal{MV}$	$S_{\mathcal{MV}}$
32	10	10	0.00%	10	0.00%
64	11	10	9.09%	10	9.09%
128	11	10	9.09%	10	9.09%
256	11	10	9.09%	10	9.09%
512	12	10	16.67%	10	16.67%

Table 2.1: Comparison of the multigrid V(1,1)-cycle number for Test Problem 2.5.1.

second half (interpolation) cycle. This also supports our analysis that acceleration is more efficient on the second (interpolation) half cycle after interpolation excites the high frequency errors.

Next, we test $\mathcal{MV}$ and κ for the two-way acceleration scheme by fixing $w_1 = 0.9$ and letting w_2 change in $(0, 2)$. Test Problem 1 is not sensitive to this scheme (see Tables 2.1 and 2.2 below), so we only record the results for Test Problems 2.5.2 and 2.5.3. We use $N = 128$, $\epsilon = 10^{-9}$ and 30 as the maximum V-cycle number allowed. Figure 2.6 shows the tested results. For Test Problem 2, (a) shows $\mathcal{MV}$ against ω_2, (b) shows κ against ω_2; for Test Problem 2.5.3, (c) shows $\mathcal{MV}$ against ω_2, (d) shows κ against ω_2. It is clear from Figure 2.6 that the two-way acceleration accelerates the convergence. Subfigures (a) and (b) of Figure 2.6 show that one additional V-cycle is reduced by employing interpolation acceleration after restriction acceleration.

We further compare the one- and two-way acceleration algorithms with the unaccelerated red-black Gauss-Seidel in multigrid, for all three test problems, for various N, with a fixed $\epsilon = 10^{-9}$. Tables 2.1 to 2.6 detail the comparisons. We note that, in all test problems, the employment of our chosen parameters helps reduce the multigrid V(1,1)-cycles by one to three, which amounts to as much as 18% reduction in iteration. The reduction of contraction number is even more significant, up to 34%. These data clearly demonstrate that the use of suitable parameters does pay for the cost. We also show that the acceleration parameters are independent of the discretization parameter N.

It is interesting to note that the acceleration schemes do not have much effect on Test Problem 2.5.1, especially the two-way acceleration which yields no improvement over the one-way acceleration. This may be due to the fact that the solution of Test Problem 2.5.1 is a polynomial, which has been approximated very well by the unaccelerated algorithm. Its $\mathcal{MV}$s are smaller than those of Test Problems 2.5.2 and 2.5.3. This suggests that our acceleration schemes be more effective when the convergence of the usual multigrid V(1,1)-cycle algorithm is slow. This fact is also demonstrated by the decrease of the meshsize which increases the number of multigrid V(1,1)-cycles due to the fact that the magnitude of the residual norm increases as the meshsize decreases. The acceleration rate increases as the number of iteration increases.

	Gauss-Seidel	One-Way Acceleration		Two-Way Acceleration	
N	κ	κ	S_κ	κ	S_κ
32	0.112	0.102	8.93%	0.103	8.04%
64	0.127	0.108	14.96%	0.108	14.96%
128	0.136	0.110	19.12%	0.110	19.12%
256	0.143	0.111	23.38%	0.111	23.38%
512	0.150	0.112	25.33%	0.112	25.33%

Table 2.2: Comparison of the contraction number for Test Problem 2.5.1.

	Gauss-Seidel	One-Way Acceleration		Two-Way Acceleration	
N	$\mathcal{MV}$	$\mathcal{MV}$	$S_{\mathcal{MV}}$	$\mathcal{MV}$	$S_{\mathcal{MV}}$
32	13	12	7.69%	12	7.69%
64	14	13	7.14%	12	12.29%
128	15	13	13.33%	12	20.00%
256	15	14	6.67%	13	13.33%
512	16	14	12.50%	13	18.75%

Table 2.3: Comparison of the multigrid V(1,1)-cycle number for Test Problem 2.5.2.

	Gauss-Seidel	One-Way Acceleration		Two-Way Acceleration	
N	κ	κ	S_κ	κ	S_κ
32	0.116	0.094	18.97%	0.088	24.14%
64	0.124	0.098	24.19%	0.086	30.65%
128	0.130	0.099	23.85%	0.087	33.08%
256	0.131	0.102	22.14%	0.087	33.59%
512	0.135	0.103	23.70%	0.089	34.00%

Table 2.4: Comparison of the contraction number for Test Problem 2.5.2.

	Gauss-Seidel	One-Way Acceleration		Two-Way Acceleration	
N	$\mathcal{MV}$	$\mathcal{MV}$	$S_{\mathcal{MV}}$	$\mathcal{MV}$	$S_{\mathcal{MV}}$
32	12	11	8.33%	11	8.33%
64	13	12	7.69%	12	7.69%
128	14	13	7.14%	12	14.29%
256	15	13	13.33%	13	13.33%
512	16	14	12.50%	13	18.75%

Table 2.5: Comparison of the multigrid V(1,1)-cycle number for Test Problem 2.5.3.

	Gauss-Seidel	One-Way Acceleration		Two-Way Acceleration	
N	κ	κ	S_κ	κ	S_κ
32	0.104	0.086	17.31%	0.082	21.15%
64	0.117	0.093	20.51%	0.088	24.79%
128	0.125	0.098	21.60%	0.087	30.40%
256	0.131	0.099	24.43%	0.087	33.59%
512	0.134	0.102	23.88%	0.088	34.33%

Table 2.6: Comparison of the contraction number for Test Problem 2.5.3.

2.6 Conclusions and Remarks

We have proposed a new approach to the relaxation analysis for the five-point red-black Gauss-Seidel smoothing in multigrid method for two dimensional Poisson equations. We have shown that under-relaxation is effective for the restriction half cycle and over-relaxation for the interpolation half cycle. Two acceleration schemes based on the these observations have been designed, recorded as much as 34% acceleration in the convergence rate and 18% reduction in iteration ($\mathcal{MV}$). The additional cost of implementing the acceleration schemes is negligible. The result corrects a long-time misunderstanding in multigrid that relaxation parameter for Gauss-Seidel method used as a smoother in multigrid for Laplace operator does not pay. Although the 34% acceleration in convergence rate still may not be substantial enough to attract practical use. It offers a new way of searching for better SOR (both under- and over-relaxation) smoothers in multigrid.

It may be possible that even better algorithms may come up for our model problem as the result of this research. Analytical optimal (but may not be the best for all problems in practice) parameters may be obtained by some analytic means and may further accelerate the convergence a little bit for specific problems. As we stated above, the best parameters should lie in a small neighborhood of what we suggested, and the best parameters should achieve the same results so long as the reduction of $\mathcal{MV}$ is concerned.

Although the analysis in Section 2.3 is based on the five-point operators, the idea of using different treatments for the high and the low frequencies is independent of any particular operators and relaxation methods. It is in fact an intrinsic property of general multigrid method. Therefore, we expect that similar one- or two-way acceleration algorithms can be developed for other isotropic and anisotropic elliptic operators. For the cases with slow convergence, our idea will provide a good means for acceleration.

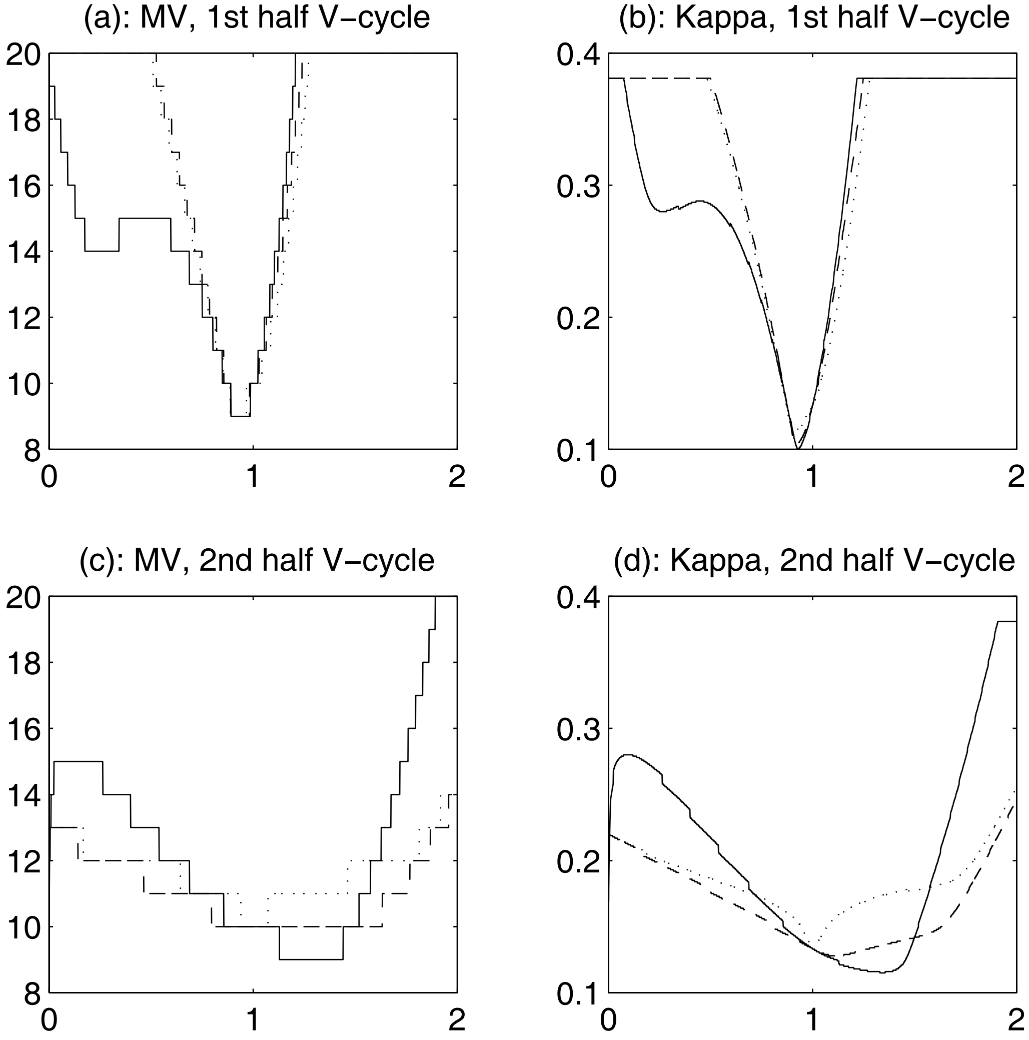

Figure 2.3: Different acceleration schemes for Test Problem 2.5.1. $N = 128$, $\epsilon = 10^{-8}$. Solid line is on the finest level, dashed line on the second finest level, dotted line on the third level. (a): $\mathcal{MV}$ vs ω_1 in the first half cycle; (b): κ vs ω_1 in the first half cycle; (c): $\mathcal{MV}$ vs ω_2 in the second half cycle; (d): κ vs ω_2 in the second half cycle.

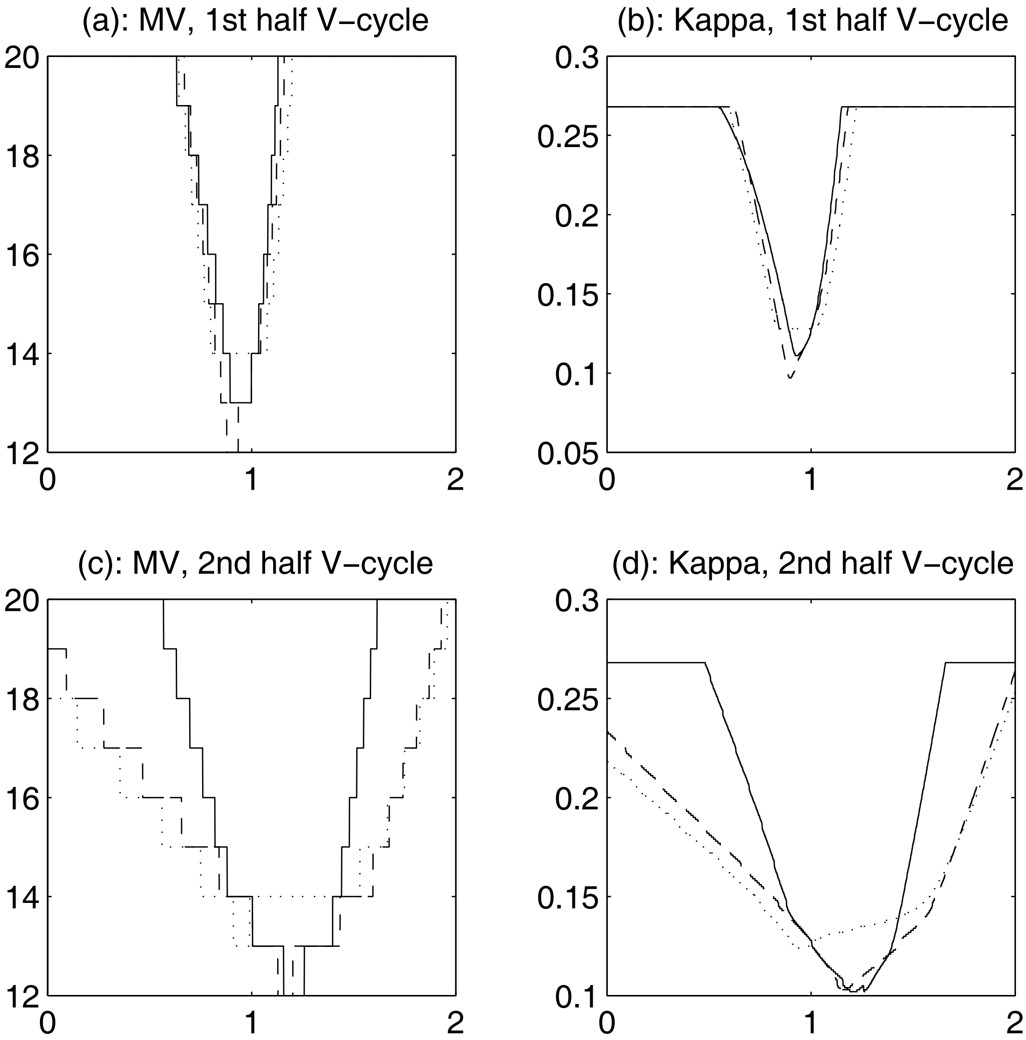

Figure 2.4: Different acceleration schemes for Test Problem 2.5.2. $N = 128$, $\epsilon = 10^{-8}$. Solid line is on the finest level, dashed line on the second finest level, dotted line on the third level. (a): $\mathcal{MV}$ vs ω_1 in the first half cycle; (b): κ vs ω_1 in the first half cycle; (c): $\mathcal{MV}$ vs ω_2 in the second half cycle; (d): κ vs ω_2 in the second half cycle.

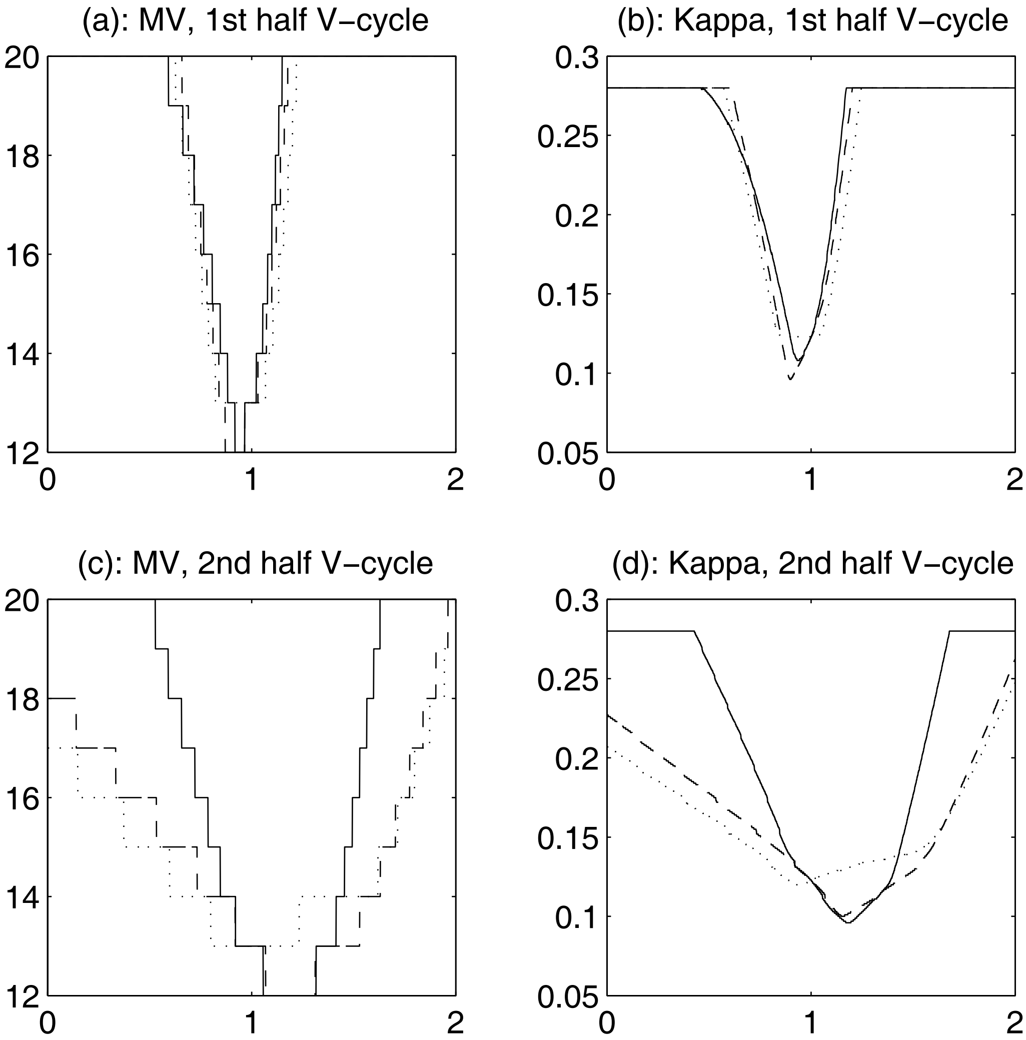

Figure 2.5: Different acceleration schemes for Test Problem 2.5.3. $N = 128$, $\epsilon = 10^{-8}$. Solid line is on the finest level, dashed line on the second finest level, dotted line on the third level. (a): $\mathcal{MV}$ vs ω_1 in the first half cycle; (b): κ vs ω_1 in the first half cycle; (c): $\mathcal{MV}$ vs ω_2 in the second half cycle; (d): κ vs ω_2 in the second half cycle.

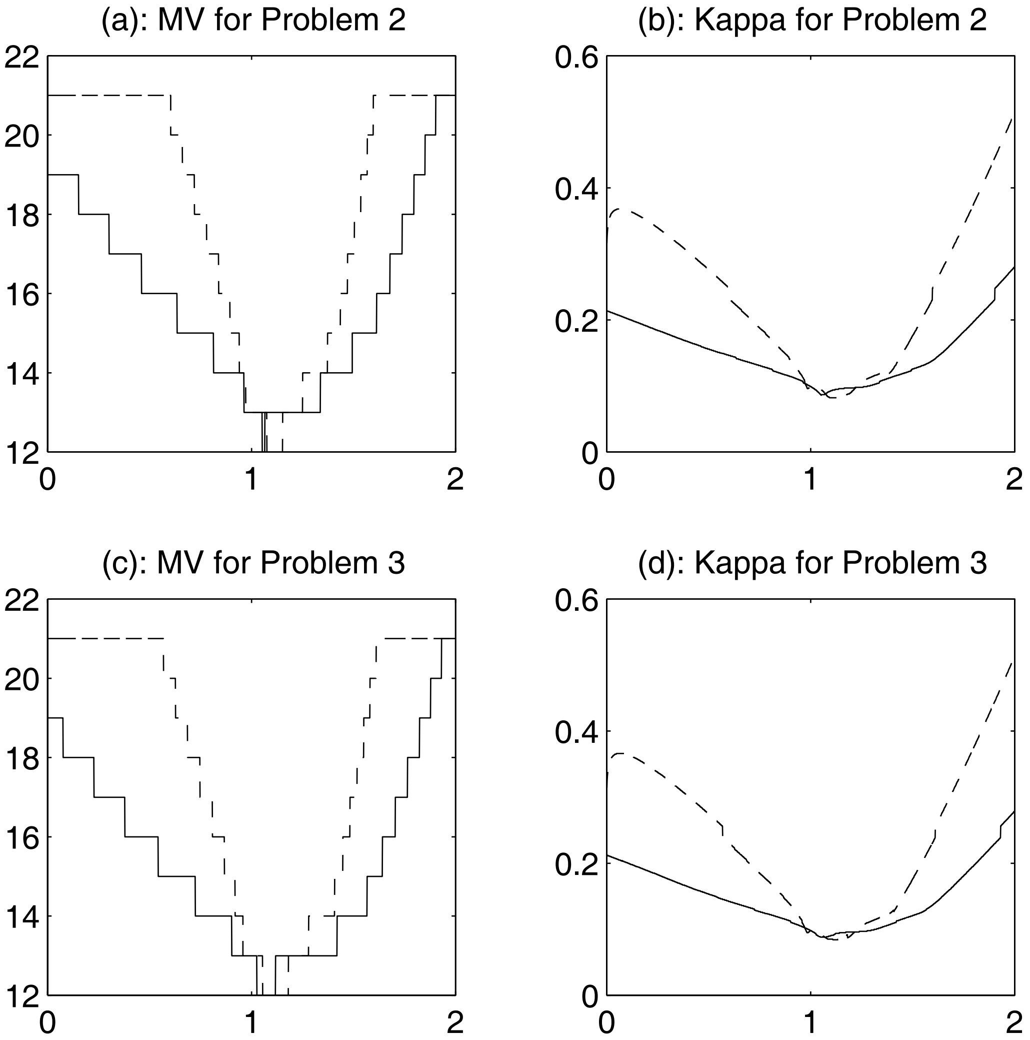

Figure 2.6: Two way acceleration scheme for Test Problems 2.5.2. and 2.5.3. $N = 128$ and $\epsilon = 10^{-9}$. Solid line is on the finest level, dashed line on the second finest level of the second (interpolation) half cycle. (a): $\mathcal{MV}$ vs ω_2 for Test Problem 2.5.2. (b): κ vs ω_2 for Test Problem 2.5.2. (c): $\mathcal{MV}$ vs ω_2 for Test Problem 2.5.3. (d): κ vs ω_2 for Test Problem 2.5.3.

Chapter 3

Multigrid Residual Scaling Techniques

3.1 Introduction

In Appendix A, we developed a heuristic residual analysis to obtain an optimal residual injection parameter for the five-point RBGS multigrid method (also see [78]). The resulting multigrid method with the optimal residual injection operator achieves convergence rate even better than the two-way acceleration scheme introduced in [77]. Both methods give near-optimal performance as claimed in Appendix A and in [78]. The optimal residual injection multigrid method results in about 40% reduction in CPU time with respect to the standard multigrid method with full-weighting (see Appendix A and [78] for numerical results).

The technique used in Appendix A and in [78] belongs to a more general category of multigrid acceleration techniques which we will refer to as the *residual scaling techniques*. Residual scaling techniques are concerned with the techniques that optimize the scale of the residuals which are projected to the coarse grid to form the coarse-grid-correction subproblem. Residual scaling techniques have been used by several investigators to accelerate the convergence of the standard multigrid methods in a variety of applications [9, 35, 42, 46, 62, 78]. They have been given different names by different authors in different contexts. It was termed as the "over-weighted residual" technique by Brandt and Yavneh [9] and was used to accelerate the convergence of the multigrid method for solving the convection-diffusion equations with high Reynolds numbers. Míka and Vaněk [42, 62] called it the "over-correction" technique and employed it in developing an algebraic multigrid method. Reusken [46] named it the "steplength optimization" technique and used it to accelerate the convergence of both linear and nonlinear multigrid methods. We called it an "under-injection" residual transfer operator [78]. In this chapter, we will prove that, under certain conditions, all of these techniques are mathematically equivalent, and we therefore give them a unified terminology as the residual scaling techniques.

The minimal residual smoothing techniques proposed by us in Chapter 4 and [80] may also be considered as an (indirect) residual scaling techniques.

This chapter is organized as follows: In Section 3.2 the residual scaling techniques are discussed and analyzed with respect to the two-level method. We prove the equivalence of two special examples of the residual scaling techniques and generalize the results to prove the mathematical equivalence of the pre-scaling acceleration techniques and the post-scaling acceleration techniques. Conclusions and some remarks are given in Section 3.3.

Practical applications of the residual scaling techniques and some numerical experiments are discussed in Chapters 6 and 7, Appendix A and [80].

3.2 Analysis of Residual Scaling Techniques

The rate of convergence of the standard multigrid algorithm is strongly dependent on the properties of the vector which is the correction of the error obtained on the coarse grid. This correction usually approximates the error of the solution very well in the sense of its “progress”, but not in the sense of its “size” [42]. Hence, standard multigrid algorithm may be accelerated when we multiply the correction vector by a suitable scalar factor. Several acceleration schemes based on this idea have been designed and employed in a number of applications [35, 42, 46, 62].

Let A^H be the coarse grid operator on Ω^H. A^H must be nonsingular, but its exact nature is not important in our current discussion. It may be an Ω^H version of A^h or it may be constructed by using the Galerkin technique [13].

Let $(\cdot,\cdot)_h$ and $(\cdot,\cdot)_H$ denote the usual inner products on Ω^h and Ω^H, respectively. We also denote $|\cdot|_h = (\cdot,\cdot)_h^{1/2}$ and $|\cdot|_H = (\cdot,\cdot)_H^{1/2}$ the associated Euclidean norms.

Definition 3.2.1 *The energetic inner product with respect to a symmetric and positive definite matrix Z on Ω^h is defined as*

$$\langle\cdot,\cdot\rangle_Z = (Z\cdot,\cdot)_h. \tag{3.1}$$

Furthermore, we define the corresponding energy norm with respect to Z as

$$\|\cdot\|_Z = \langle\cdot,\cdot\rangle_Z^{1/2}. \tag{3.2}$$

The energetic inner product and the corresponding energy norm on Ω^H may be defined similarly.

Remark 3.2.2 *If $Z = I$ (the identity matrix), then $\|\cdot\|_I$ reduces to the Euclidean norm.*

Let there be a regular splitting of A^h and let M and N be nonsingular square matrices on Ω^h satisfying the consistency condition

$$M + NA^h = I, \tag{3.3}$$

where I is the identity matrix on Ω^h.

Definition 3.2.3 *The smoothing iterative method is defined as of the form*

$$S(v^h) = Mv^h + Nf^h, \tag{3.4}$$

where $v^h \in \Omega^h$. For any integer $\nu > 1$, we recursively define

$$S^\nu(v^h) = S(S^{\nu-1}(v^h)). \tag{3.5}$$

For notational convenience, we denote

$$S^0(v^h) = I(v^h) = v^h,$$

where $I(\cdot)$ is the identity operator.

M is sometimes called the iteration matrix of the smoothing iterative method (3.4).

Definition 3.2.4 *Let $\nu > 0$ be any integer and $Q(x)$ be a polynomial of degree $\nu - 1$*

$$Q^\nu(x) = x^{\nu-1} + x^{\nu-2} + \cdots + x + 1,$$

we define the matrix polynomial of degree $\nu - 1$ with respect to the iteration matrix M as

$$Q^\nu(M) = M^{\nu-1} + M^{\nu-2} + \cdots + M + I. \tag{3.6}$$

Lemma 3.2.5 *For any integer $\nu > 0$, the following identity is valid*

$$Q^{\nu+1}(M) = MQ^\nu(M) + I. \tag{3.7}$$

Proof. The proof is trivial. □

Remark 3.2.6 *The consistency condition (3.3) implies*

$$S(u^h) = u^h, \tag{3.8}$$

where $u^h \in \Omega^h$ is the exact solution of the linear system (1.1) (on the fine grid), i.e.

$$u^h = A^{h^{-1}} f^h.$$

Lemma 3.2.7 *For any integer $\nu > 0$ and any approximate solution $v^h \in \Omega^h$, we have the following identity*

$$S^\nu(v^h) = M^\nu v^h + Q^\nu(M)Nf^h. \tag{3.9}$$

Proof. We prove Lemma 3.2.7 by induction on the integer ν.

The identity (3.9) obviously holds for $\nu = 1$ with $Q(M) = I$ by Definition 3.2.3.

Suppose the Lemma is true for ν.

From Definition 3.2.3 and Lemma 3.2.5, we have

$$\begin{array}{rcll} S^{\nu+1}(v^h) & = & S(S^\nu(v^h)) & \text{by (3.5)} \\ & = & S(M^\nu v^h + Q^\nu(M)Nf^h) & \text{by induction assumption} \\ & = & M[M^\nu v^h + Q^\nu(M)Nf^h] + Nf^h & \text{by (3.4)} \\ & = & M^{\nu+1}v^h + [MQ^\nu(M) + I]Nf^h & \\ & = & M^{\nu+1}v^h + Q^{\nu+1}(M)Nf^h & \text{by Lemma 3.2.5.} \end{array}$$

This completes the proof of Lemma 3.2.7. □

Definition 3.2.8 *Given an approximate solution $v^h \in \Omega^h$, we define the error as*

$$e^h = e^h(v^h) = u^h - v^h. \tag{3.10}$$

Definition 3.2.9 *Given an approximate solution $v^h \in \Omega^h$, we define the residual as*

$$r^h = f^h - A^h v^h. \tag{3.11}$$

Remark 3.2.10 *It is clear that $e^h \in \Omega^h$ and $r^h \in \Omega^h$. We also have the following error (residual) equation*

$$A^h e^h = r^h. \tag{3.12}$$

Eq. (3.12) is the basis for the coarse grid equation.

Components of the error which are not effectively removable by smoothing, i.e.

$$Me^h \approx e^h$$

will be called smooth components. We try to represent these smooth components in Ω^H using the interpolation operator P. The error $e^h \in \Omega^h$ can be represented in Ω^H if there exists some $e^H \in \Omega^H$ such that

$$e^h = Pe^H. \tag{3.13}$$

Eq. (3.13) means that the error e^h lies entirely in the range of the interpolation operator P. This requirement is usually not satisfied in reality, otherwise solving the Ω^H error (residual) equation exactly and doing coarse-grid-correction would give the exact solution [13].

3.2.1 Standard Two-Level Method

One iteration of the standard two-level method is as follows:

Algorithm 3.2.11 *One Iteration of the Standard Two-level Algorithm.*

```
Step 0:    Given an initial guess $v^h_0$.
Step 1:    Pre-smoothing     $v^h_1 = S^{\nu_1}(v^h_0)$.
Step 2:    Compute           $r^h = f^h - A^h v^h_1$.
Step 3:    Restrict          $r^H = Rr^h$.
Step 4:    Solve             $e^H = A^{H^{-1}} r^H$.
Step 5:    Correct           $v^h_2 = v^h_1 + Pe^H$.
Step 6:    Post-smoothing    $v^h_3 = S^{\nu_2}(v^h_2)$.
```

Lemma 3.2.12 *If the correction step in Algorithm 3.2.11 is*

$$v^h_2 = v^h_1 + Pe^H, \tag{3.14}$$

then the following equality holds

$$S^{\nu_2}(v^h_2) = S^{\nu_2}(v^h_1) + M^{\nu_2} Pe^H. \tag{3.15}$$

Proof. From Lemma 3.2.7 and the definition of the smoothing operator (3.4) we have

$$\begin{aligned} S^{\nu_2}(v_2^h) &= M^{\nu_2}v_2^h + Q^{\nu_2}(M)Nf^h && \text{by (3.9)} \\ &= M^{\nu_2}(v_1^h + Pe^H) + Q^{\nu_2}(M)Nf^h && \text{by (3.14)} \\ &= M^{\nu_2}v_1^h + M^{\nu_2}Pe^H + Q^{\nu_2}(M)Nf^h && \\ &= S^{\nu_2}(v_1^h) + M^{\nu_2}Pe^H && \text{by (3.9).} \end{aligned}$$

This proves Lemma 3.2.12. □

Corollary 3.2.13 *For Algorithm 3.2.11, the following equality holds for any integer* $\nu_2 > 0$

$$v_3^h = S^{\nu_2}(v_1^h) + M^{\nu_2}Pe^H. \tag{3.16}$$

Lemma 3.2.14 *Let* $v_1^h \in \Omega^h$ *be an approximate solution to* u^h *and* $\nu_2 > 0$ *be an integer, then*

$$M^{\nu_2}e^h(v_1^h) = e^h(S^{\nu_2}(v_1^h)). \tag{3.17}$$

Proof. From Definition 3.2.8 and Remark 3.2.6, we have

$$\begin{aligned} M^{\nu_2}e^h(v_1^h) &= M^{\nu_2}(u^h - v_1^h) && \text{by (3.10)} \\ &= M^{\nu_2}u^h + Q^{\nu_2}(M)Nf^h - M^{\nu_2}v_1^h - Q^{\nu_2}(M)Nf^h && \\ &= S^{\nu_2}(u^h) - S^{\nu_2}(v_1^h) && \text{by (3.9)} \\ &= u^h - S^{\nu_2}(v_1^h) && \text{by (3.8)} \\ &= e^h(S^{\nu_2}(v_1^h)) && \text{by (3.10).} \end{aligned}$$

This proves Lemma 3.2.14. □

From Definition 3.2.8 and Corollary 3.2.13, we have the following lemma immediately:

Lemma 3.2.15 *For any integer* $\nu_2 > 0$*, the following identity is valid*

$$e^h(v_3^h) = e^h(S^{\nu_2}(v_1^h)) - M^{\nu_2}Pe^H. \tag{3.18}$$

From Lemma 3.2.14 and Lemma 3.2.15, we have the following lemma:

Lemma 3.2.16 *For any integer* $\nu_2 > 0$*, the following identity is valid*

$$e^h(v_3^h) = M^{\nu_2}[e^h(v_1^h) - Pe^H]. \tag{3.19}$$

The quality of the coarse-grid-correction of Algorithm 3.2.11 may be measured by the ratio of the errors in some norm before and after the coarse-grid-correction.

Definition 3.2.17 *Let* Z *be a symmetric and positive definite matrix, we define the convergence rate* τ *as:*

$$\tau = \frac{\|e^h(v_3^h)\|_Z}{\|e^h(v_0^h)\|_Z}, \tag{3.20}$$

where $\|\cdot\|_Z$ *is the energy norm with respect to* Z*.*

3.2.2 Over-Correction Technique

The convergence rate τ of the standard two-level method may be accelerated by choosing a scalar α such that the scale of the correction is optimized, i.e., in Algorithm 3.2.11, we replace the correction Step 5 by

$$v_2^h = v_1^h + \alpha P e^H \tag{3.21}$$

and α is chosen so that the convergence rate τ in (3.20) is minimized.

Remark 3.2.18 *Since v_0^h and ν_1 are supposed to be given and u^h is fixed (the unknown exact solution), minimizing τ is equivalent to minimizing $\|e^h(v_3^h)\|_Z$ in (3.20).*

Lemma 3.2.19 *The minimization problems (3.20) and (3.21) is equivalent to the following minimization problem*

$$\|M^{\nu_2}[e^h(v_1^h) - \alpha P e^H]\|_Z = \min_{\bar{\alpha} \in \mathbf{R}} \|M^{\nu_2}[e^h(v_1^h) - \bar{\alpha} P e^H]\|_Z, \tag{3.22}$$

where $\mathbf{R}$ is the set of all real numbers.

Proof. Since

$$M^{\nu_2}[\alpha P e^H] = \alpha M^{\nu_2} P e^H.$$

The proof follows from Lemma 3.2.16 with Pe^H being replaced by $\alpha P e^H$ and Remark 3.2.18. □

Lemma 3.2.20 *If Z is symmetric and positive definite, the minimization problem (3.22) is solved by*

$$\alpha = \frac{\langle M^{\nu_2} e^h(v_1^h), M^{\nu_2} P e^H \rangle_Z}{\|M^{\nu_2} P e^H\|_Z^2} \qquad \textit{for} \qquad M^{\nu_2} P e^H \neq 0. \tag{3.23}$$

If $M^{\nu_2} P e^H = 0$, an arbitrary α fulfills the minimization condition (3.22).

Proof. If $M^{\nu_2} P e^H = 0$, the minimization condition (3.22) is obviously satisfied with an arbitrary α.

Let $M^{\nu_2} P e^H \neq 0$ and let

$$f(\bar{\alpha}) = \|M^{\nu_2}[e^h(v_1^h) - \bar{\alpha} P e^H]\|_Z^2. \tag{3.24}$$

From Definition 3.2.1 we have

$$\begin{aligned} f(\bar{\alpha}) &= \langle M^{\nu_2}[e^h(v_1^h) - \bar{\alpha} P e^H], M^{\nu_2}[e^h(v_1^h) - \bar{\alpha} P e^H] \rangle_Z \\ &= \bar{\alpha}^2 \langle M^{\nu_2} P e^H, M^{\nu_2} P e^H \rangle_Z - 2\bar{\alpha} \langle M^{\nu_2} e^h(v_1^h), M^{\nu_2} P e^H \rangle_Z \\ &\quad + \langle M^{\nu_2} e^h(v_1^h), M^{\nu_2} e^h(v_1^h) \rangle_Z. \end{aligned} \tag{3.25}$$

$f(\bar{\alpha})$ is minimized if and only if $\bar{\alpha}$ satisfy the equation

$$\frac{\mathrm{d} f(\bar{\alpha})}{\mathrm{d} \bar{\alpha}} = 0. \tag{3.26}$$

From (3.25) and (3.26), together with the assumption that $M^{\nu_2}Pe^H \neq 0$, we obtain the unique minimizer

$$\begin{aligned}\alpha &= \frac{\langle M^{\nu_2}e^h(v_1^h), M^{\nu_2}Pe^H\rangle_Z}{\langle M^{\nu_2}Pe^H, M^{\nu_2}Pe^H\rangle_Z} \\ &= \frac{\langle M^{\nu_2}e^h(v_1^h), M^{\nu_2}Pe^H\rangle_Z}{\|M^{\nu_2}Pe^H\|_Z^2}. \qquad (3.27)\end{aligned}$$

We prove Lemma 3.2.20 by virtue of Definition 3.2.1 (3.2). □

Remark 3.2.21 *$M^{\nu_2}Pe^H = 0$ implies that no coarse-grid-correction is needed.*

Although the minimization condition (3.22) is solved by Lemma 3.2.20, the scaling parameter α can not be computed from formula (3.23) for an arbitrary matrix Z, it may be computed in a special case when Z is chosen to be A^h and A^h is symmetric and positive definite. The last condition puts a severe limitation on the application of this kind of acceleration technique to practical problems.

Lemma 3.2.22 *Suppose that $Z = A^h$ and A^h is symmetric and positive definite, for any integer $\nu_2 > 0$, let*

$$\begin{aligned}\bar{e}^h &= M^{\nu_2}Pe^H, \qquad (3.28)\\ \bar{v}_2^h &= S^{\nu_2}(v_1^h). \qquad (3.29)\end{aligned}$$

The optimal over-correction factor α may be computed from

$$\begin{aligned}\alpha &= \frac{(f^h - A^h\bar{v}_2^h, \bar{e}^h)_h}{(A\bar{e}^h, \bar{e}^h)_h} \\ &= \frac{(f^h - A^h\bar{v}_2^h, \bar{e}^h)_h}{\|\bar{e}^h\|_{A^h}^2}. \qquad (3.30)\end{aligned}$$

Proof. We want to show

$$\langle M^{\nu_2}e^h(v_1^h), M^{\nu_2}Pe^H\rangle_{A^h} = (f^h - A^h\bar{v}_2^h, \bar{e}^h)_h. \qquad (3.31)$$

By Definitions 3.2.1, 3.2.8 and Lemma 3.2.14 we have

$$\begin{aligned}\langle M^{\nu_2}e^h(v_1^h), M^{\nu_2}Pe^H\rangle_{A^h} &= (A^h[M^{\nu_2}e^h(v_1^h)], M^{\nu_2}Pe^H)_h && \text{by (3.1)}\\ &= (A^h[e^h(S^{\nu_2}(v_1^h))], M^{\nu_2}Pe^H)_h && \text{by (3.17)}\\ &= (A^h[e^h(\bar{v}_2^h)], \bar{e}^h)_h && \text{by (3.28), (3.29)}\\ &= (f^h - A^h\bar{v}_2^h, \bar{e}^h)_h\end{aligned}$$

by (3.11) and (3.12). □

Note that all the quantities in equations (3.28) and (3.29) are computable and the optimal over-correction parameter α can therefore be computed from formula (3.30).

The modified two-level method with the over-correction technique is as follows:

Algorithm 3.2.23 *One Iteration of the Two-level Algorithm with Over-Correction.*

```
Step 0:   Given an initial guess $v_0^h$.
Step 1:   Pre-smoothing   $v_1^h = S^{\nu_1}(v_0^h)$.
Step 2:   Compute         $r^h = f^h - A^h v_1^h$.
Step 3:   Restrict        $r^H = Rr^h$.
Step 4:   Solve           $e^H = A^{H^{-1}} r^H$.
Step 5:   Smoothing       $\bar{e}^h = M^{\nu_2} P e^H$.
Step 6:   If   $\bar{e}^h = 0$   then   $\bar{v}_3^h = v_1^h$,   else
Step 7:   Smoothing       $\bar{v}_2^h = S^{\nu_2}(v_1^h)$.
Step 8:   Compute         $\alpha = (f^h - A^h \bar{v}_2^h, \bar{e}^h)_h / \|\bar{e}^h\|_{A^h}^2$.
Step 9:   Correct         $\bar{v}_3^h = \bar{v}_2^h + \alpha \bar{e}^h$.
```

Remark 3.2.24 *If we put $\alpha = 1$, Algorithm 3.2.23 is mathematically equivalent to Algorithm 3.2.11, although Algorithm 3.2.23 is computationally much more expansive. Lemma 3.2.19 guarantees that the convergence rate of Algorithm 3.2.23 is at least as good as the convergence rate of Algorithm 3.2.11.*

Acceleration techniques like Algorithm 3.2.23 compute the scaling parameter α after the process of the coarse-grid-correction and were termed as post-optimization by Reusken [46] for this reason. Algorithm 3.2.23 is the so-called over-correction acceleration scheme due to Míka and Vaněk [42, 62]. These techniques optimize the computed correction and may accelerate the convergence substantially if the original computed correction is not in an appropriate scale, albeit at a heavy computational cost. In addition to the cost of the standard two-level method (Algorithm 3.2.11), Algorithm 3.2.23 requires at least the following additional cost: two inner product computations, one residual computation, one vector-matrix multiplication, one vector-vector addition and one scalar-vector multiplication. Usually, a large number of pre-smoothing sweeps are needed for the effect of the over-correction to be significant. In Vaněk's test problem for solving an anisotropic Poisson equation [62], he used 7 pre-smoothing and 2 post-smoothing sweeps (see [62]). The cost is high and the algorithm is not computationally efficient for accelerating the standard multigrid method. Furthermore, the validity of the algorithm depends on the symmetry and positive definiteness of the coefficient matrix A^h. In many applications of practical interests, such as the convection-diffusion equation with large convection coefficients, the coefficient matrix A^h is nonsymmetric and non-positive definite and Algorithm 3.2.23 can not be applied.

Reusken's approach [46, 48] is similar to the over-correction Algorithm 3.2.23, but he considered the problem in a different point of view and viewed the correction step as an optimization process to search for the steplength parameter α along the direction defined by the correction vector Pe^H. In Reusken's numerical test for solving a Poisson equation, it was found that the convergence rate of a V-cycle algorithm accelerated by the steplength optimization (over-correction) is somewhat equivalent to a standard W-cycle algorithm.

Bounds for the convergence rates of the two-level methods with and without the over-correction acceleration have been obtained by Míka and Vaněk [42, 62]. More quantitative bounds were obtained by Reusken [46].

3.2.3 Weighted Residual Technique

Let us consider a different approach to optimize the coarse-grid-correction process. We may modify the residual vector r^h by multiplying it with a scalar β before it is projected to the coarse grid to form the coarse grid subproblem. Hence, if the original multigrid coarse-grid-correction is not optimal in "scale", we may choose the weighting parameter β so that the "scale" of the coarse-grid-correction is optimized. We refer to this technique as the weighted residual technique.

The following lemma states the relation between the scale of the residuals and the solution of the error (residual) equation.

Lemma 3.2.25 *Suppose that A^H is nonsingular and β is a scalar. If e^H is the solution of the linear system $A^H e^H = r^H$ and $\hat{r}^H = \beta r^H$, then $\hat{e}^H = \beta e^H$ is the solution of the scaled linear system $A^H \hat{e}^H = \hat{r}^H$.*

Proof. The proof is trivial. □

The following theorem follows immediately from Lemma 3.2.25.

Theorem 3.2.26 *In the standard two-level method, the scale of the correction vector is linearly dependent on the scale of the residual vector. The quality of the correction vector may therefore be improved by properly modifying the scale of the residual vector.*

Suppose that we have some way to find β to satisfy some conditions so that the residuals are better "scaled" in some sense, then we can modify the standard multigrid Algorithm 3.2.11 as follows

Algorithm 3.2.27 *One Iteration of the Two-level Algorithm with Weighted Residual*

```
Step 0:   Given an initial guess $v_0^h$.
Step 1:   Pre-smoothing    $v_1^h = S^{\nu_1}(v_0^h)$.
Step 2:   Compute          $r^h = f^h - A^h v_1^h$.
Step 3:   Restrict         $\hat{r}^H = \beta R r^h$.
Step 4:   Solve            $\hat{e}^H = {A^{2h}}^{-1} \hat{r}^H$.
Step 5:   Correct          $v_2^h = v_1^h + P\hat{e}^H$.
Step 6:   Post-smoothing   $v_3^h = S^{\nu_2}(v_2^h)$.
```

Weighted residual techniques in different forms have been used by Brandt and Yavneh [9], and Zhang [78, 80] to accelerate the multigrid convergence in different situations.

As Brandt and Yavneh [9] remarked that, when solving the convection-diffusion equations with high-Reynolds number, the error is dominated by smooth components. Hence, instead of increasing the number of pre-smoothing sweeps on the fine grid, they concentrated their efforts on improving the coarse-grid-correction because in many cases, the coarse grid solution fails to approximate that of the fine grid. They developed an over-weighted residual algorithm to accelerate the convergence of the red-black Distributive Gauss-Seidel relaxation method to solve the convection-diffusion equations and the Navier-Stokes equations. For example, the two-level convergence factor of the convection-diffusion equation (with vanishing diffusion coefficients) employing the first-order discretization improves from 0.5 to 0.33 with a residual scaling factor $\beta = 4/3$. Since the residual scaling factor $\beta > 1$, they called their technique the over-weighted residual technique.

In Appendix A and [78], we used a heuristic residual analysis, based on the analysis of the geometry of the grid points and the relaxation pattern, to derive a residual scaling factor $\beta = 0.4634$ to inject the residuals from the fine grid to the coarse grid for the five-point RBGS relaxation. Since $\beta < 0.5$ (0.5 is the standard half-injection factor), we called the technique the under-injection residual transfer technique. The numerical experiments conducted in [78] showed that the weighted residual injection operator converges faster and is more computationally cost-effective (with about 40% reduction in CPU time) than the standard multigrid method (Algorithm 3.2.11).

In the applications of the weighted residual techniques mentioned above, there is no requirement that the coefficient matrix A^h be symmetric and positive definite.

We will use a rather different form of the weighted residual technique named the minimal residual smoothing technique in Chapter 4 (and in [80]) to smooth the residuals before they are projected to the coarse grid. The new residuals are obtained as an optimal linear combination of the current and the previous residuals. The optimality is satisfied when the new residuals in Euclidean norm are minimized.

From the above examples, it is clear that the residual weighting factor β is different in different situation with different relaxation method. However, we note that, if β can be pre-determined by some kind of residual analysis, there is no additional cost for the weighted residual algorithm (Algorithm 3.2.27) over the standard algorithm (Algorithm 3.2.11). This is the primary incentive for us to investigate the weighted residual technique because it is computationally efficient.

Remark 3.2.28 *We point out that, in Algorithm 3.2.27, if the residuals need to be scaled before they are projected to the coarse grid, then we may choose a suitable scaling factor so that it scales the residuals result from the injection operator instead of from the averaging (full-weighting) operator. If we use the injection operator, we save about* 3/4 *of the residual projection cost.*

Next, we will prove the main theorem of this chapter which states that, under certain conditions, Algorithm 3.2.23 and Algorithm 3.2.27 are mathematically equivalent.

Theorem 3.2.29 *Suppose that A^h is symmetric and positive definite, then the two-level method with over-correction acceleration and the two-level method with weighted residual acceleration have the same convergence rate if and only if the over-correction parameter α in Algorithm 3.2.23 equals the residual weighting parameter β in Algorithm 3.2.27.*

Proof. Given any initial value v_0^h, suppose that we choose the residual weighting parameter $\beta = \alpha$ in Algorithm 3.2.27, the solution of the coarse-grid-correction reads

$$\hat{e}^H = A^{H^{-1}} \alpha R r^h. \tag{3.32}$$

The approximate value after the coarse-grid-correction is

$$v_2^h = v_1^h + P\hat{e}^H = v_1^h + \alpha P {A^H}^{-1} R r^h. \tag{3.33}$$

By the definition of the smoothing operator (3.4) and Lemma 3.3, after the post-smoothing, the approximate solution of the two-level method with weighted residual is

$$\begin{aligned}
v_3^h &= S^{\nu_2}(v_2^h) \\
&= S^{\nu_2}(v_1^h) + M^{\nu_2}\alpha P {A^H}^{-1} R r^h \\
&= S^{\nu_2}(v_1^h) + \alpha M^{\nu_2} P {A^H}^{-1} R r^h \\
&= \bar{v}_2^h + \alpha \bar{e}^h \\
&= \bar{v}_3^h.
\end{aligned}$$

Since the approximate solution from Algorithm 3.2.27 is the same as that from Algorithm 3.2.23, we conclude that these two algorithms are mathematically equivalent and have the same convergence rate.

On the other hand, if Algorithm 3.2.23 and Algorithm 3.2.27 have the same convergence rate, then

$$v_3^h = \bar{v}_3^h. \tag{3.34}$$

We have

$$\begin{aligned}
v_3^h &= S^{\nu_2}(v_2^h) \\
&= S^{\nu_2}(v_1^h + P {A^H}^{-1} \beta R r^h) \\
&= S^{\nu_2}(v_1^h) + \beta M^{\nu_2} P {A^H}^{-1} R r^h. \qquad (3.35)
\end{aligned}$$

And

$$\begin{aligned}
\bar{v}_3^h &= \bar{v}_2^h + \alpha \bar{e}^h \\
&= S^{\nu_2}(v_1^h) + \alpha M^{\nu_2} P {A^H}^{-1} R r^h. \qquad (3.36)
\end{aligned}$$

It follows from (3.34), (3.35) and (3.36) that $\beta = \alpha$. □

Because of Theorem 3.2.29, we can search for alternative methods to estimate the over-correction parameter α. Hereinafter we refer to parameters α and β as the residual scaling parameter α. Theorem 3.2.29 states that it is the scale of the residuals that really matters the convergence of the two-level method. If we can develop some methods to estimate the appropriate magnitude of the residual scaling parameter and if these methods do not rely on the symmetry and the positive definiteness assumption of the coefficient matrix, Algorithm 3.2.27 is suitable for a larger area of applications and is computationally more efficient than Algorithm 3.2.23. Fortunately these alternative techniques do exist and we will discuss some of them in Chapters 4 to 7 and Appendix A.

Definition 3.2.30 *We define the acceleration techniques that scale the residual before it is projected to the coarse grid as the pre-scaling acceleration techniques.*

Definition 3.2.31 *We define the acceleration techniques that scale the computed correction after it is interpolated back to the fine grid as the post-scaling acceleration techniques.*

From Theorems 3.2.26 and 3.2.29, we have the following corollary:

Corollary 3.2.32 *In a two-level method, the pre-scaling acceleration techniques and the post-scaling acceleration techniques are mathematically equivalent if and only if their scaling factors are equal.*

In parallel implementation, the injection operator is clearly advantageous because injection is a local process and requires no communication with neighboring processors. The full-weighting operator (or any other weighting scheme) requires communication with eight neighboring grid points which may be stored in different processors.

3.3 Conclusions and Remarks

We have developed and unified the ideas of residual scaling techniques and proved the equivalence of two residual scaling techniques and the mathematical equivalence of pre-scaling and post-scaling acceleration techniques. These theoretical results clear the way for developing efficient pre-scaling acceleration techniques for practical applications. In Chapter 6 and Appendix A, we will introduce the concept of *heuristic residual analysis* technique and use that technique to derive optimal residual scaling factors for particular applications. The optimal residual injection operators with the optimal residual scaling factors will be used in standard multigrid method as well as high-order multigrid method. We will show that the resulting multigrid methods achieve convergence rate faster than the multigrid methods with the full-weighting and half-injection operators in the sense of computational cost-effectiveness for solving the Poisson equations. For some convection-diffusion equations, the residual injection is one way to retain convergence, when the full-weighting operator diverges for highly convection-dominated problems.

Chapter 4

Minimal Residual Smoothing in Multigrid

4.1 Introduction

In this chapter, we investigate the feasibility and the efficiency of employing some minimal residual smoothing techniques to accelerate the convergence of the multigrid method (MG). For reasons which will become clear later, the acceleration scheme discussed in this chapter is applied on the finest level only. Hence, our discussion will be focused on the two-level method (Algorithm 3.2.11), because it contains all the basic ideas of the multigrid method and because the multigrid method can be viewed as using recursively defined two-level method to solve the coarse grid subproblem.

Suppose that a sequence $\{u_k\}$ is generated by some iterative method with the associated residual sequence $\{r_k\}$. Since it is generally not possible to measure the convergence of the error directly, the quality of the iteration is usually judged by the behavior of the residual norm sequence $\{\|r_k\|\}$, where $\|\cdot\|$ is some norm, e.g., the Euclidean norm. Usually, it is desirable that $\{\|r_k\|\}$ converges "smoothly" to zero. In many classical iterative methods, residuals are not effectively utilized in the iteration process, they are usually used to measure the convergence only.

Let us take a detour to review another category of iterative methods which utilize residual techniques extensively.

In the widely used generalized minimal residual (GMRES) method [50], each u_k is characterized by

$$\|f - Au_k\|_2 \ = \min_{u \in u_0 + \mathcal{K}_k(r_0, A)} \|f - Au\|_2,$$

where $\|\cdot\|_2$ is the Euclidean norm and the *Krylov subspace* $\mathcal{K}_k(r_0, A)$ is defined by

$$\mathcal{K}_k(r_0, A) = \text{span}\{r_0, Ar_0, \ldots, A^{k-1}r_0\}.$$

For GMRES, $\{\|r_k\|_2\}$ converges to zero optimally among all Krylov subspace methods, for which $u_k \in u_0 + \mathcal{K}_k(r_0, A)$. Other comparable methods, such as biconjugate gradient (BCG) [18, 40], and conjugate gradient squared (CGS) [56], have certain advantages over GMRES, but often exhibit very irregular residual-norm behavior [86]. This irregular residual-norm behavior has provided an incentive for the development of methods that have similar advantages but produce better behaved residual norms, such as the biconjugate gradient stabilized (Bi-CGSTAB) methods [32, 64] and methods based on the quasi-minimal residual (QMR) approach [14, 15, 19, 20].

Another approach to generating well-behaved residual norms has been proposed by Schönauer [54] and investigated extensively by Weiss [66]. In this approach, an auxiliary sequence $\{v_k\}$ is generated from $\{u_k\}$ by a relation

$$\begin{aligned} v_0 &= u_0, \\ v_k &= (1-\beta_k)v_{k-1} + \beta_k u_k, \qquad k = 1, 2, \ldots, \end{aligned} \tag{4.1}$$

in which each β_k is chosen to minimize $\|f - A[(1-\beta)v_{k-1} + \beta u_k]\|_Z$ over $\beta \in \mathbf{R}$, i.e.

$$\beta_k = -\frac{\langle s_{k-1}, r_k - s_{k-1}\rangle_Z}{\|r_k - s_{k-1}\|_Z^2}, \tag{4.2}$$

where $s_{k-1} = f - Av_{k-1}$. Here $\langle \cdot, \cdot \rangle_Z$ is the energetic inner product given by (3.1) and $\|\cdot\|_Z$ is the energy norm given by (3.2) with respect to a symmetric and positive definite matrix Z (Definition 3.2.1). The resulting residuals s_k obviously have non-increasing energy norms, i.e., $\|s_k\|_Z \leq \|s_{k-1}\|_Z$ and $\|s_k\|_Z \leq \|r_k\|_Z$ for each k.

Remark 4.1.1 *The energy norm used in the definition of MRS is for theoretical completeness. In practical computations, the Euclidean norm (corresponding to $Z = I$) is usually used.*

In the context of the Krylov subspace methods, Weiss [66] explored and analyzed the residual smoothing technique of the form (4.1) extensively, which was referred to by Zhou and Walker [86] as the *minimal residual smoothing* (MRS) technique. Weiss showed that applying MRS to an orthogonal residual method results in a minimal residual method. Zhou and Walker extended MRS to a *quasi-minimal residual smoothing* (QMRS) technique applicable to any iterative method [65, 86]. They also showed that QMRS can be used to derive a QMS-type method from any given method. In their numerical experiments, it was found that MRS residual norms were often, although not always, slightly smaller than the QMRS residual norms and, in some cases, tended to remain a little more stable in the final iterations. They have some preference for MRS over QMRS for general use. In this dissertation, we choose MRS to smooth the residuals generated by the multigrid method (MG). In our numerical experiments using MRS and QMRS to smooth and to accelerate the MG sequence, the MRS sequence is better behaved than the QMRS sequence when both are used as smoothing techniques. However, when both are used as acceleration techniques, MRS is far better than QMRS. The later is actually inefficient in our experiments when it is used as an acceleration method for MG. Hence, discussion of applying QMRS to the multigrid method will not be pursued here.

More general forms of residual smoothing techniques are considered by Brezinski and Redivo-Zaglia [12]. But we will limit our attention to MRS of the form (4.1) in this dissertation.

Most existing research has been focused on employing MRS techniques as smoothing methods to smooth the residuals generated by the Krylov subspace methods. Although some numerical experiments are reported to show that sequence generated by classical iterative methods such Jacobi and Gauss-Seidel methods can also be smoothed [12], we are not aware of discussion on the practical implementation of employing MRS techniques to accelerate the classical iterative methods. The major reason is probably that residuals are not necessarily calculated in the classical iterative methods, and the employment of MRS techniques requires residual computation. This may render the cost of implementing MRS techniques prohibitive in the classical iterative methods.

However, in the two-level (multigrid) method, residuals are computed automatically and used to form the subproblem on the coarse level. This advantage certainly reduces the cost of implementing MRS techniques. Furthermore, since the subproblem on the coarse-grid is a residual equation and the smoothness of the residuals are essential for the solution of the coarse-grid subproblem to approximate that of the fine grid problem so that a good coarse-grid-correction may be provided to the fine grid [6]. This gives the primary incentive to use MRS techniques to smooth the residuals before they are projected to the coarse grid and the reason that this approach will accelerate the convergence of the original two-level (multigrid) iterative process.

In this chapter, we emphasize the practical implementation of employing MRS in the two-level (multigrid) method to accelerate the convergence of the original method. Rigorous analysis is postponed in Chapter 5.

In Section 4.2, We briefly elaborate on MRS techniques and design algorithms to employ MRS as smoothing techniques in the two-level iteration process. In Section 4.3, we develop MRS acceleration scheme and algorithms that feed the sequence with "smoothed" residuals generated by MRS back to the two-level and multigrid iteration processes. Numerical tests are employed in Section 4.4 to show the remarkable acceleration rate and negligible cost of the MRS acceleration scheme used in the two-level and multigrid methods. Conclusions are included in Section 4.5. Suggestions on future research are given in Section 4.6.

4.2 Minimal Residual Smoothing

4.2.1 MRS Techniques

Assume that we have some iterative method which generates a sequence of iterates $\{u_k\}$ with the corresponding residual sequence $\{r_k\}$, we formulate the MRS technique of [54] and [66] as follows:

Algorithm 4.2.1 *Minimal residual smoothing (MRS) [54, 66]*

Initialize $s_0 = r_0$ *and* $v_0 = u_0$.

```
For $k = 1, 2, \ldots,$ do:
    Compute  $u_k$ and $r_k$.
    Compute  $\beta_k = -\langle s_{k-1}, r_k - s_{k-1}\rangle_Z / \|r_k - s_{k-1}\|_Z^2$.
    Set      $s_k = s_{k-1} + \beta_k(r_k - s_{k-1})$;
    and      $v_k = v_{k-1} + \beta_k(u_k - v_{k-1})$.
```

4.2.2 Two-Level Method and MRS

Algorithm 4.2.1 may be applied to any step of the standard two-level method (Algorithm 3.2.11) where there is an update of current values to generate one or more sequences of iterates with smoothed residuals. However, application of MRS requires the values of the residuals which are not generally computed at each step of TLM. Normally, the computation of the residuals is equivalent to one relaxation on that grid level and should be avoided whenever possible. Thus, we would like to use MRS to smooth the residuals calculated in TLM before they are weighted and projected to the coarse grid. Algorithm 4.2.2 is a procedure that incorporates MRS in TLM to generate a sequence with "smoothed" residuals.

Algorithm 4.2.2 *Two-level method and MRS*

```
Given $u_0^h$ and $r_0^h = f^h - A^h u_0^h$, set $v_0 = u_0^h$ and $s_0 = r_0^h$.
For $k = 0, 1, 2, \ldots,$ do:
    Relax $\nu_1$ times on $A^h u_k^h = f^h$ with the given initial guess $u_k^h$.
    Compute   $r_k^h = f^h - A^h u_k^h$.
    Compute   $\beta_{k+1} = -\langle s_k, r_k^h - s_k\rangle_Z / \|r_k^h - s_k\|_Z^2$.
    Set       $s_{k+1} = s_k + \beta_{k+1}(r_k^h - s_k)$;
    and       $v_{k+1} = v_k + \beta_{k+1}(u_k^h - v_k)$.
    Restrict  $r_k^H = R r_k^h$.
    Solve     $e_k^H = (A^H)^{-1} r_k^H$.
    Correct   $u_{k+1}^h = u_k^h + P e_k^H$.
    Relax $\nu_2$ times on $A^h u_{k+1}^h = f^h$ with the initial guess $u_{k+1}^h$.
```

Algorithm 4.2.2 generates a TLM sequence $\{u_k^h\}$ with the associated residual sequence $\{r_k^h\}$ and an MRS sequence $\{v_k\}$ with the smoothed residual sequence $\{s_k\}$. Note that v_k and s_k are generated before u_k^h and r_k^h for $k \geq 1$. We have $\|s_k\|_Z \leq \|r_{k-1}^h\|_Z$ for all $k \geq 1$. Furthermore, in most classical iterative methods used as relaxation schemes in TLM, r_0^h is not usually calculated. One exception is that when the initial guess is taken as zero, i.e., $u_0^h = 0$, then $r_0^h = f^h$. In general, we may not want to compute the initial residuals just for MRS. Algorithm 4.2.3 is a slight modification of Algorithm 4.2.2, only v_0 and s_0 are generated after the first round of smoothing sweeps on the fine grid.

Algorithm 4.2.3 *Two-level method and MRS*

```
Given any initial guess $u_0^h$.
For $k = 0, 1, 2, \ldots,$ do:
```

```
Relax $\nu_1$ times on $A^h u_k^h = f^h$ with the given initial guess $u_k^h$.
Compute    $r_k^h = f^h - A^h u_k^h$.
If $k = 0$, then
   Set $v_0 = u_0^h$ and $s_0 = r_0^h$.
Else
   Compute    $\beta_k = -\langle s_{k-1}, r_k^h - s_{k-1}\rangle_Z / \|r_k^h - s_{k-1}\|_2^2$.
   Set        $s_k = s_{k-1} + \beta_k (r_k^h - s_{k-1})$;
   and        $v_k = v_{k-1} + \beta_k (u_k^h - v_{k-1})$.
End if.
Restrict   $r_k^H = R r_k^h$.
Solve      $e_k^H = (A^H)^{-1} r_k^H$.
Correct    $u_{k+1}^h = u_k^h + P e_k^H$.
Relax $\nu_2$ times on $A^h u_{k+1}^h = f^h$ with the initial guess $u_{k+1}^h$.
```

Algorithm 4.2.3 generates an auxiliary sequence $\{v_k\}$ with the associated residual sequence $\{s_k\}$ which satisfies $\|s_k\|_Z \le \|r_k\|_Z$ for all k. The new residual norm sequence $\{\|s_k\|_Z\}$ is better behaved than the original residual norm sequence $\{\|r_k\|_Z\}$. We actually have [67]

$$\|s_k\|_Z \ \le \min\{\|r_0\|_Z, \|r_1\|_Z, \dots, \|r_k\|_Z\}.$$

This implies that if $\{\|r_k\|_Z\}$ is not monotonically decreasing, $\{\|s_k\|_Z\}$ does, since $\|s_k\|_Z \le \|s_{k-1}\|_Z$ for all k.

In our following discussions and numerical experiments, we use Algorithm 4.2.3 to generate MRS sequence.

4.3 MRS Acceleration Schemes

4.3.1 Two-Level Method with MRS Acceleration

The sequence $\{v_k\}$ generated by Algorithm 4.2.3 is guaranteed to have non-increasing residual norms $\{\|r_k\|_Z\}$. Most existing research works employ MRS techniques as a means to stabilize the residual sequence generated by some iterative method (usually some Krylov subspace methods). These approaches utilize the smoothing property of the MRS techniques and have been reported by many investigators [65, 66, 67, 86]. However, as Zhou and Walker remarked in [86], having a smoothly decreasing or monotonically decreasing residual norm may be of real importance or just a nicety, dependent on the particular application. Employment of MRS techniques as a means of acceleration has not been pursued extensively outside the context of the Krylov subspace methods. On the other hand, using MRS techniques to accelerate classical iterative methods still faces the cost of residual computation, which may render potential MRS acceleration scheme very expensive. If MRS techniques are applied to the sequences generated by classical iterative methods such as Jacobi or Gauss-Seidel methods, unless the original sequence diverges or behaves very irregularly, there is a serious question about the cost of generating such a new sequence against doing one more iteration using the original iterative method. Because the cost of generating the new sequence normally is larger than the computation of the original iteration due to the residual computation. It would be better if we could feed the new sequence, which has a better behaved residual norms, back to the original iteration, to accelerate the convergence of the original iteration. In return, the accelerated original sequence will help MRS generate a better new sequence with even more "smoothed" residuals. However, this is not always possible. For instance, if the original sequence is generated by some Krylov subspace method, as the cases usually discussed in MRS context, this approach would destroy some properties of the original sequence, e.g., mutual orthogonality, which is essential for the original sequence to converge. In particular, Schönauer originally developed MRS to smooth the sequence generated by the conjugate gradient-type methods and explicitly mentioned that $\{s_k\}$ is computed without feedback to the original sequence [54, p. 261]. Only in a later chapter, when he discussed the development of the PRES20 method, which is a pseudo-residual method, he restarted the original iteration from the smoothed sequence after 20 iterations [53]. He also claimed that a restart from the smoothed sequence pays.

However, in TLM, the situation is different. Since the new iterate is not solely resulted from the old values by the relaxation scheme (it involves the correction from the coarse grid), there is no specific property that the original sequence must obey. Thus we may expect that a feedback of the new sequence generated by MRS will accelerate the convergence of the original iteration. A further advantage of the two-level methods is that the residuals of the original sequence are calculated even there is no MRS involved. This advantage, if exploited properly, reduces the cost of implementing MRS acceleration schemes significantly.

As we noted before, there are many places in a two-level method where we may insert the MRS procedure. But we would like to do it in a cheapest way. i.e., to use existing information as much as possible and to minimize computation of quantities which are only useful in MRS. This leads us to insert MRS procedure just after the residuals on the finest grid being computed and before they are projected to the coarse level grid. At each major iteration, we replace the original TLM iterate u_k^h and its residual iterate r_k^h by the MRS iterate v_k and the associated residual iterate s_k. We then project the residual s_k to the coarse-level grid to form a coarse-level grid subproblem. (Note that we must replace both the TLM iterate u_k^h and its residual r_k^h at the same time, otherwise the coarse-grid subproblem would provide a wrong correction to the fine grid.) In this way, we give the coarse grid smoothed residuals which are essential for the coarse grid to provide a good coarse-grid-correction to the fine grid [6]. Therefore, we expect that the acceleration rate is favorable. The following Algorithm 4.3.1 is parallel to Algorithm 4.2.3, without unnecessarily computing the initial residuals before the MRS acceleration scheme is applied.

Algorithm 4.3.1 *Two-level method with MRS feedback*

```
Given any initial guess $u_0^h$.
For $k = 0, 1, 2, \dots,$ do:
```

```
Relax $\nu_1$ times on $A^h u_k^h = f^h$ with the given initial guess $u_k^h$.
Compute   $r_k^h = f^h - A^h u_k^h$.
If $k = 0$, then
   Set $v_0 = u_0^h$ and $s_0 = r_0^h$.
Else
   Compute $\beta_k = -\langle s_{k-1}, r_k^h - s_{k-1}\rangle_Z / \|r_k^h - s_{k-1}\|_Z^2$.
   Set       $s_k = s_{k-1} + \beta_k (r_k^h - s_{k-1})$;
   and       $v_k = v_{k-1} + \beta_k (u_k^h - v_{k-1})$.
   Set       $u_k^h = v_k$ and $r_k^h = s_k$.
End if.
Restrict  $r_k^H = R r_k^h$.
Solve     $e_k^H = (A^H)^{-1} r_k^H$.
Correct   $u_{k+1}^h = u_k^h + P e_k^H$.
Relax $\nu_2$ times on $A^h u_{k+1}^h = f^h$ with the initial guess $u_{k+1}^h$.
```

We refer to the acceleration scheme of replacing the TLM (MG) sequence and its residual sequence by the MRS sequence and its residual sequence as the *MRS acceleration scheme* or *MRS feedback*. The role of the MRS acceleration scheme in the two-level (multigrid) method is to accelerate the convergence of the coarse-grid-correction by reducing the norm of the residuals (smoothing the residuals). Since high frequency components of the errors have already been effectively removed by the pre-smoothing sweeps, the MRS acceleration scheme primarily reduces the norm of the residuals associated with the low frequency components of the errors.

On the other hand, we may think the MRS acceleration scheme in a different way. MRS may be used repeatedly to smooth the high frequency errors after the initial iterate is generated by a particular relaxation scheme. This will be cheaper than using more pre-smoothing sweeps, especially when A^h is complicated, since the MRS acceleration scheme is independent of the original operator A^h.

4.3.2 Multigrid with MRS Acceleration

We have developed algorithms to accelerate the convergence of the two-level method and by the heuristic arguments given above we predict that the acceleration effect will be favorable. One question left in Section 4.1 is that why we are only interested in using the MRS acceleration scheme on the finest level grid. In a multigrid method, one may think that the MRS acceleration scheme developed above may be used on the coarse level grids as well as on the finest grid. This is indeed possible if we are solely interested in generating a smoothed sequence. Because MRS will generate a new sequence with much smoother residuals regardless of the quality and the nature of the original sequence. However, generating such a “well-behaved” sequence is meaningless in the multigrid method. It will not be used to accelerate the convergence of the original sequence. This new sequence can not be feeded back into the original iteration process in the way we discussed above for the finest level (an exception is discussed in Section 4.6). Because in each major MG iteration, different residuals are projected to the coarse level and thus the subproblem on the coarse level is different. The iterates (the coarse-grid-correction) generated by a particular coarse-grid iteration have no direct connection, they are used to correct the fine grid solution at each iteration and have no usefulness after the coarse-grid-correction process.

Hence, this explains why we only use the MRS acceleration scheme on the finest level. The multigrid method (MG) may be considered as a two-level method but solving the coarse-grid subproblem by recursively employing coarser grids to obtain coarse-grid-corrections.

Algorithm 4.3.2 is the formal multigrid method with the MRS acceleration scheme, which is parallel to Algorithm 4.3.1 of the two-level method.

Algorithm 4.3.2 *Multigrid method with MRS acceleration scheme*

```
    $u^h \leftarrow MG(u^h, f^h)$

Given any initial guess $u_0^h$.
For $k = 0, 1, 2, \dots,$ do:
   If $\Omega^h$ = the coarsest grid, then
      Solve      $u_k^h = (A^h)^{-1} f^h$.
   Else
      Relax $\nu_1$ times on $A^h u_k^h = f^h$ with the given initial guess $u_k^h$.
      Compute  $r_k^h = f^h - A^h u_k^h$.
      If $\Omega^h$ = the finest grid, then
         If $k = 0$, then
            Set   $v_0 = u_0^h$ and $s_0 = r_0^h$.
         Else
            Set   $\beta_k = -\langle s_{k-1}, r_k^h - s_{k-1}\rangle_Z / \|r_k^h - s_{k-1}\|_Z^2$.
            Set   $s_k = s_{k-1} + \beta_k (r_k^h - s_{k-1})$;
            and   $v_k = v_{k-1} + \beta_k (u_k^h - v_{k-1})$.
            Set   $u_k^h = v_k$ and $r_k^h = s_k$.
         End if.
      End if.
      Restrict   $r_k^{2h} = R r_k^h$.
      Set        $f^{2h} = r_k^{2h}$.
      Set        $u_k^{2h} = 0$.
      Do         $u_k^{2h} \leftarrow MG(u_k^{2h}, f^{2h})$ $\mu$ times.
      Correct    $u_{k+1}^h = u_k^h + P u_k^{2h}$.
      Relax $\nu_2$ times on $A^h u_{k+1}^h = f^h$ with the initial guess $u_{k+1}^h$.
   End if.
```

The parameter μ in Algorithm 4.3.2 is set to control the number of times of the cycle to visit the coarse grid before it returns to the fine grid. In practice, only $\mu = 1$ and $\mu = 2$ are usually used. This corresponds to the so-called V-cycle and W-cycle schemes respectively (see [7, 69] for more details on definition of various multigrid cycling schemes).

4.4 Numerical Examples

Numerical examples are set to test the convection-diffusion equation with the Dirichlet boundary condition (6.1) on a unite square $(0,1) \times (0,1)$.

Specifically, we take $p(x,y) = Mx$ and $q(x,y) = -My$. M is a positive constant to be chosen. The boundary conditions are given so that the exact solution is $u(x,y) = xy(1-x)(1-y)\exp(x+y)$.

The high-order compact finite difference scheme discussed in Chapter 6 is used to discretize equation (6.1) which results in a linear system with a compact nine-point formula. The reason for choosing this high-order discretization formula is that it is stable for all M so that we do not worry about the magnitude of M which might cause divergence if a lower-order, e.g., five-point formula, were employed.

The coefficient matrix A^h on all levels is not stored, but computed in the process of iterations. For the numerical experiments of the two-level method, the fine grid mesh-size is 1/64 and the coarse grid mesh-size is 1/32. For the multigrid method, the coarsest grid contains only one unknown (with mesh-size 1/2). For all computation, we apply one pre-smoothing and one post-smoothing sweeps ($\nu_1 = \nu_2 = 1$). The discrete grid space is naturally (lexicographically) ordered and the point Gauss-Seidel relaxation is used as the smoothing method. The residual projection operator R is the full-weighting scheme and the interpolation operator P is the bi-linear interpolation (see [69]). As stated in Remark 4.1.1, unless otherwise indicated explicitly, the norm used in the numerical experiments is the Euclidean norm.

The computations are done on a Cray-90 vector machine at the Pittsburgh Supercomputing Center using Cray Fortran 77 in single precision (equivalent to double precision on serial machines such as SUN workstations).

4.4.1 Two-Level Method with MRS Acceleration

In this subsection, we test Algorithms 4.2.3 and 4.3.1, i.e., the two-level method with and without MRS feedback (acceleration). In the two-level method, the coarse-grid subproblem is not solved exactly. Instead, five relaxation sweeps are carried out on the coarse grid.

We have four sequences generated by: TLM without MRS, MRS without feedback, TLM with MRS feedback (acceleration) and MRS with feedback.

First, we choose $M \equiv 0$ and Eq. (6.1) reduces to the Poisson equation. The convergence histories of the residual norm sequences are depicted in Figure 4.1. Since TLM is not a very efficient Poisson solver, it takes more than 400 iterations to converge to the limit of residual reduction. In this dissertation, the limit of residual reduction (LRR) is defined to be the limit of reducing residual norm by an algorithm on a given computer using finite-precision computation. The sequence generated by MRS alone converges faster than the sequence generated by TLM alone (without feedback). This shows the smoothing property of the MRS technique. When TLM is combined with MRS with feedback (acceleration), both sequences converge very fast and almost reduce the iterations by half to reach LRR. This demonstrates the acceleration property of the MRS technique.

Next, we take $M = 128$, the cell Reynolds number (Re) on the finest grid is 1. In this case, the two-level method is more efficient (the spectral radius of A^h is smaller when Re = 128 than when Re = 0.) The convergence histories of the residual norm sequences are graphed in Figure 4.2. The sequence generated by MRS without feedback converges faster than the sequence generated by TLM without feedback from MRS. This again proves that the minimal residual smoothing property of the MRS algorithm. The convergence histories of sequences generated by TLM with MRS feedback and MRS with feedback are overlapped, because they actually accelerate each other. However, since the original sequence (generated by TLM without feedback) converges quite satisfactorily, the acceleration by the MRS feedback is not as much as it did for the Poisson equation (see Figures 4.1). Hence, when the original sequence converges fast, the benefit of the MRS acceleration is somewhat less. We emphasize that the MRS acceleration scheme is still efficient.

A convection-dominated case is studied next. We take $M = 1.28 \times 10^5$. The cell Reynolds number on the finest grid is 1000. The initial residual norm is increased significantly. It is well-known that the magnitude of the cell Reynolds number affects the convergence of the numerical methods inversely. The convergence histories of the residual norm sequences are depicted in Figure 4.3. It takes TLM without MRS feedback about 2500 iterations to reach LRR. The sequence generated by MRS converges faster, but is restricted by the TLM sequence, i.e., these two sequences are bounded together by MRS.

Again, TLM with MRS feedback (acceleration) and MRS with feedback accelerate mutually and their convergence histories are virtually the same. Both are remarkably better than TLM and MRS sequences without feedback. The numbers of the iterations are halved. This again supports the claim made earlier that TLM with MRS feedback (acceleration) works very well when the original sequence converges slowly.

One might doubt the efficiency of the MRS acceleration scheme, because it may be viewed as one sequence with double iterates. It will not be advantageous if the double iterates generated at the same cost and reduce the iterations by half. The matter of fact here is that the sequence generated by MRS is very cheap, comparing with that generated by TLM. Because MRS does not involve the coefficient matrix whose realization is very expensive for the variable coefficient equations. Therefore, the cost of the MRS acceleration scheme is minimal, if not negligible. More about the cost of the MRS acceleration scheme are discussed in subsection 4.4.3.

4.4.2 Multigrid with MRS Acceleration

In this subsection, we experiment with the multigrid method and the MRS acceleration scheme. In Algorithm 4.3.2, μ is chosen to be 1, which is a V-cycle scheme. Also, we use a true V-cycle algorithm. Specifically, with $h = 1/64$ on the finest level, there are 6 levels and the coarsest level just contains one unknown. One relaxation sweep is carried out on the coarsest grid. We mention explicitly that the MRS acceleration scheme is only applied on the finest level.

Again, we first consider the Poisson equation with $M = 0$. It is well-known that the true multigrid method (MG) is one of the most efficient Poisson solver. The convergence histories of the residual norm sequences are depicted in Figure 4.4. The sequence generated by the standard MG reaches LRR in about 12 iterations. The sequence generated by MRS without feedback converges slightly faster. On the other hand, the sequences generated by MG with MRS acceleration and MRS with feedback mutually accelerate the convergence and reach LRR in fewer iterations. This means that MRS acceleration scheme is efficient even when the original method converges fast, such as the true multigrid method, although the acceleration rate is somewhat low.

We again consider the case when $M = 128$, the convergence histories of the residual norm sequences are contained in Figure 4.5. Again, the true multigrid method is very efficient in this case. The sequence generated by MRS converges slightly faster than the sequence generated by MG without feedback. The sequence generated by MG with MRS acceleration converges faster than the sequences generated by MG and MRS separately. The same is true for the sequence generated by MRS with feedback. By comparing Figure 4.5 with Figure 4.4, we note that the acceleration rate is increased.

Next, we consider the multigrid method for problem with large Reynolds number, here $M = 1.28 \times 10^5$. The convergence histories are depicted in Figure 4.6. Since the convergence of the true MG is slow for the convection-diffusion equations with large Reynolds number, Figure 4.6 is similar to Figure 4.5, except that the acceleration rate is more attractive. MG with MRS feedback is a powerful method to accelerate the true MG in this case, with almost one-half reduction in the number of the iterations.

Our conclusion from these test examples is that MRS acceleration scheme accelerates the convergence of the multigrid method regardless of the convergence of the original method. The MRS acceleration rate is more attractive when the original method converges slowly,

4.4.3 Computational Cost and Computed Accuracy

In this subsection, we test the computational cost of the MRS acceleration scheme. We also consider the two-level and multigrid methods separately. The computations are terminated when the residuals in L_2 norm on the finest level are reduced by 10^{10}. We record the iteration numbers, the CPU time in seconds and the computed accuracy in L_∞ norm.

Table 4.1 contains data for comparison of iterations and computed accuracy for the two-level methods with or without MRS acceleration scheme (feedback). Table 4.2 is the same comparison of CPU time in seconds.

From Table 4.1, we again note that TLM with MRS feedback accelerates the convergence of the original TLM considerably, especially when the original TLM converges slowly ($M = 0$ and $M = 1.28 \times 10^5$). The rates of reduction are very attractive. In the case of slow convergence, the rates of reduction are more than 40%. Furthermore,

	TLM without MRS		TLM with MRS		
M	Iteration	Error	Iteration	Error	Rate of Reduction
0	358	5.57(-9)	195	5.57(-9)	45.53%
128	23	1.84(-6)	21	1.84(-6)	9.52%
128000	1982	1.32(-4)	1111	1.32(-4)	43.95%

Table 4.1: Comparison of iterations and computed accuracy for two-level methods.

M	TLM without MRS	TLM with MRS	Rate of Reduction
0	27.98	15.28	45.39%
128	1.83	1.67	8.74%
128000	155.74	87.48	43.83%

Table 4.2: Comparison of CPU time in seconds for two-level methods.

comparing the rates of reduction in CPU time in Table 4.2 with those in iteration in Table 4.1, we find that the cost of the MRS acceleration is truly negligible, less than 1% of the TLM cost. This means that the MRS acceleration scheme is a very cost-effective acceleration scheme for TLM. Even when $M = 128$, the acceleration is not very much, but it comes with negligible cost.

The computed accuracy is the same with or without the MRS acceleration scheme (see Table 4.1), this means that the MRS acceleration scheme maintains the computed accuracy of the original TLM method.

Next, we consider the multigrid method with MRS feedback acceleration scheme. Parallel to Tables 4.1 and 4.2, the data from the numerical experiments are given in Tables 4.3 and 4.4.

Table 4.3 gives the similar results as Table 4.1. The rates of reduction are very attractive in all cases. The average rate of reduction is even better than that for the two-level method. The acceleration cost is negligible. The acceleration scheme does not change the computed accuracy of the original multigrid method.

We note that in Table 4.4, when $M = 0$, the rate of reduction in CPU time is

	MG without MRS		MG with MRS		
M	Iteration	Error	Iteration	Error	Rate of Reduction
0	12	5.59(-9)	10	5.59(-9)	16.67%
128	30	1.84(-6)	20	1.84(-6)	33.33%
128000	2036	1.32(-4)	1117	1.32(-4)	45.14%

Table 4.3: Comparison of iterations and computed accuracy in multigrid methods.

M	MG without MRS	MG with MRS	Rate of Reduction
0	0.788	0.656	16.75%
128	1.963	1.328	32.35%
128000	132.67	74.99	43.48%

Table 4.4: Comparison of CPU time in seconds in multigrid methods.

slightly larger than that in iteration in Table 4.3. This is possibly caused by the nature of the vector computer and the timing function. Since many users use the computer at the same time and the overhead may be large with respect to the actual computing time in this case.

From these tests, it is clear that MRS acceleration scheme is an efficient technique for accelerating the convergence of the multigrid method. When the linear system results from discretized partial differential equations with variable coefficients, the complexity of the coefficient matrix renders the cost of implementing MRS acceleration scheme negligible.

4.5 Conclusions

We have shown that the minimal residual smoothing (MRS) is an effective way of generating sequence with “smoothed” residual norms in both two-level method (TLM) and multigrid method (MG). We explored acceleration property of the MRS techniques and constructed algorithms that effectively employ MRS acceleration scheme to accelerate the convergence of both TLM and MG. Hence, MRS techniques are not only effective smoothing techniques to stabilize the residuals of the original sequence, but also cost-effective acceleration techniques to accelerate the convergence of the original sequence. The MRS acceleration scheme developed in this chapter is independent of the original operator A^h and of the relaxation method employed. Our numerical experiments demonstrated that the acceleration rate is remarkable and the cost of the acceleration scheme is negligible. We also showed that the benefit of the MRS acceleration is large when the original TLM or MG converge slowly. But the acceleration is achieved regardless of the convergence of the original method.

4.6 Suggestions on Future Research

For those sequences which converge very slowly, such as those generated by the TLM or MG in solving the convection-diffusion equations with variable coefficients and high-Reynolds numbers, there may be a reason to use MRS techniques to accelerate the original multigrid method more than once. One way of doing this is to smooth the residual sequence repeatedly. Another way is to smooth the sequence when it finishes the coarse-grid-correction cycle. The second way requires additional computation of residuals. However, if the MRS acceleration scheme is more efficient than the relaxation sweep to remove the high frequency errors, this approach is viable as a substitute for the inefficient relaxation schemes.

The more general minimal residual smoothing techniques discussed by Brezinski and Redivo-Zaglia [12] may be employed to accelerate the convergence of the multigrid method. Also, there is an approach of defining the new sequence by minimizing the residual norm in a different sense, e.g., in the energy norm with respect to A^h, if A^h is symmetric and positive definite. The energy norm involves the coefficient matrix A^h and therefore may cost more when A^h is complicated.

In Section 4.4.2, we mentioned that the MRS techniques are applicable only to the finest level. On the coarse levels, each iteration solves a different subproblem projected from the fine level and thus no continuous MRS acceleration like that for the fine level makes sense on the coarse levels. However, MRS techniques may still be used to smooth the residuals generated on the coarse level before and after they are projected to an even coarser level. We may expect that this approach will give a better residual sequence to the coarser level. In this way, only a short MRS sequence is generated in each MG iteration.

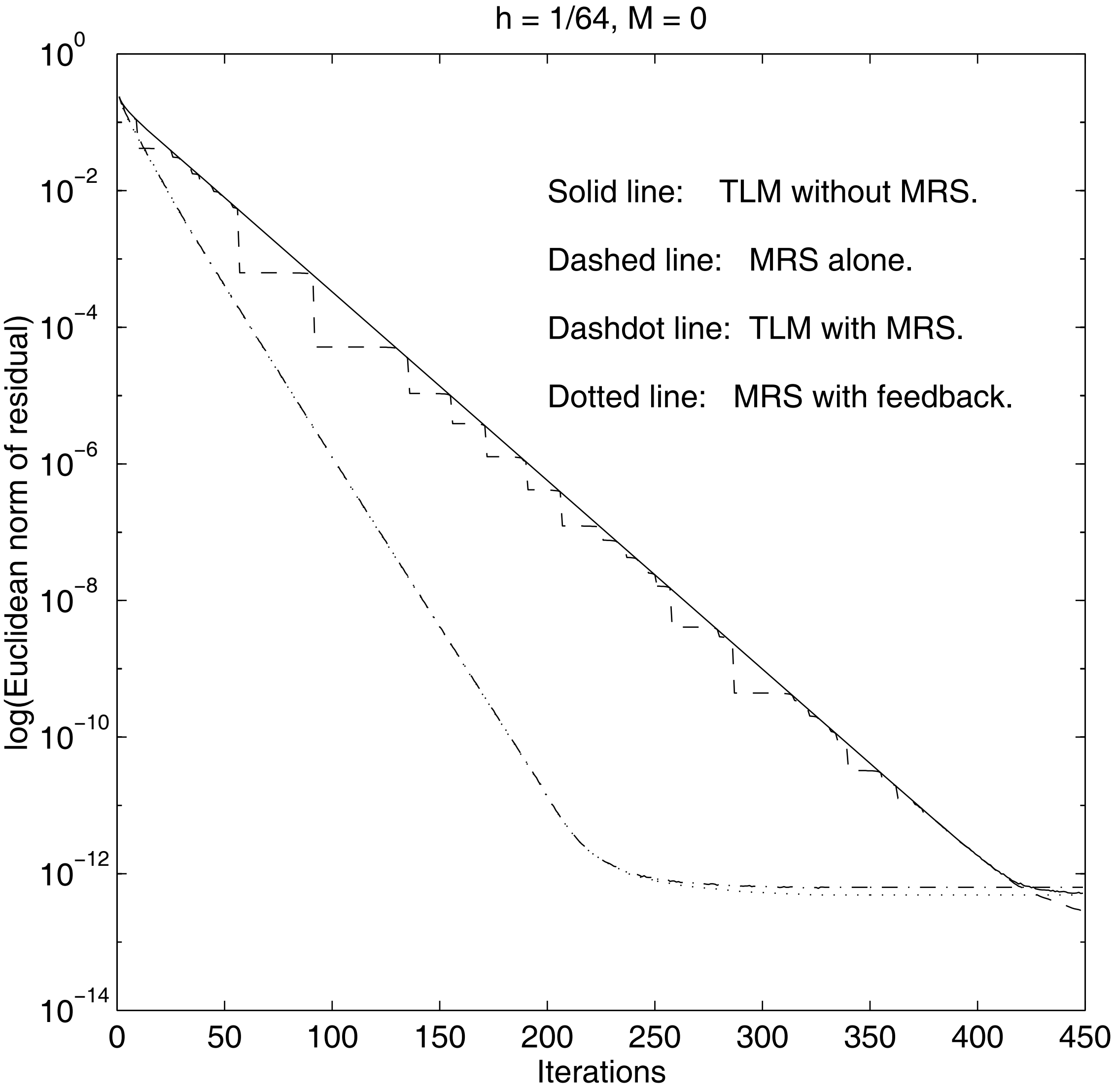

Figure 4.1: The convergence histories of the residual norms (in log scale) of TLM-MRS with $M = 0$.

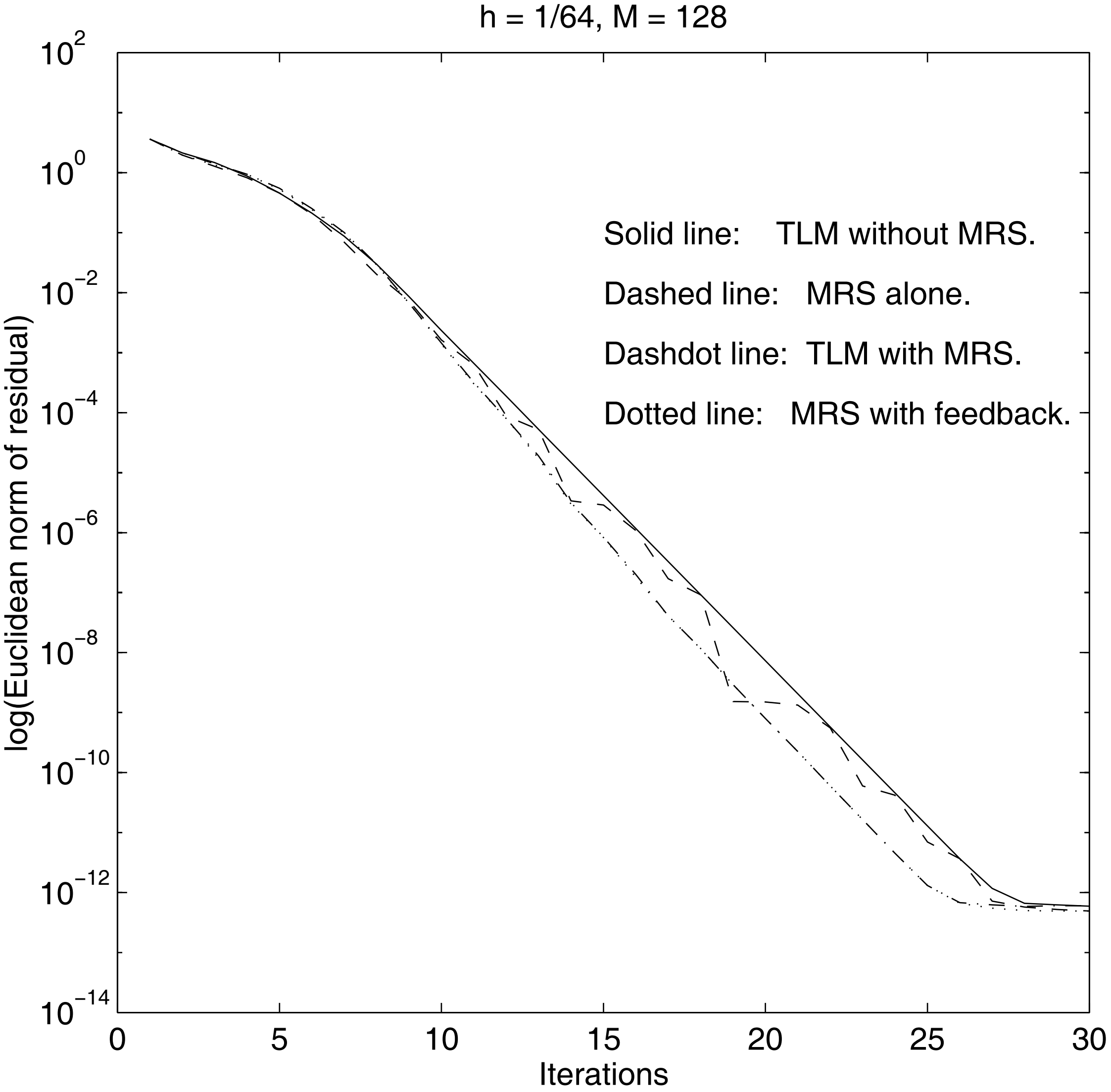

Figure 4.2: The convergence histories of the residual norms (in log scale) of TLM-MRS with $M = 128$.

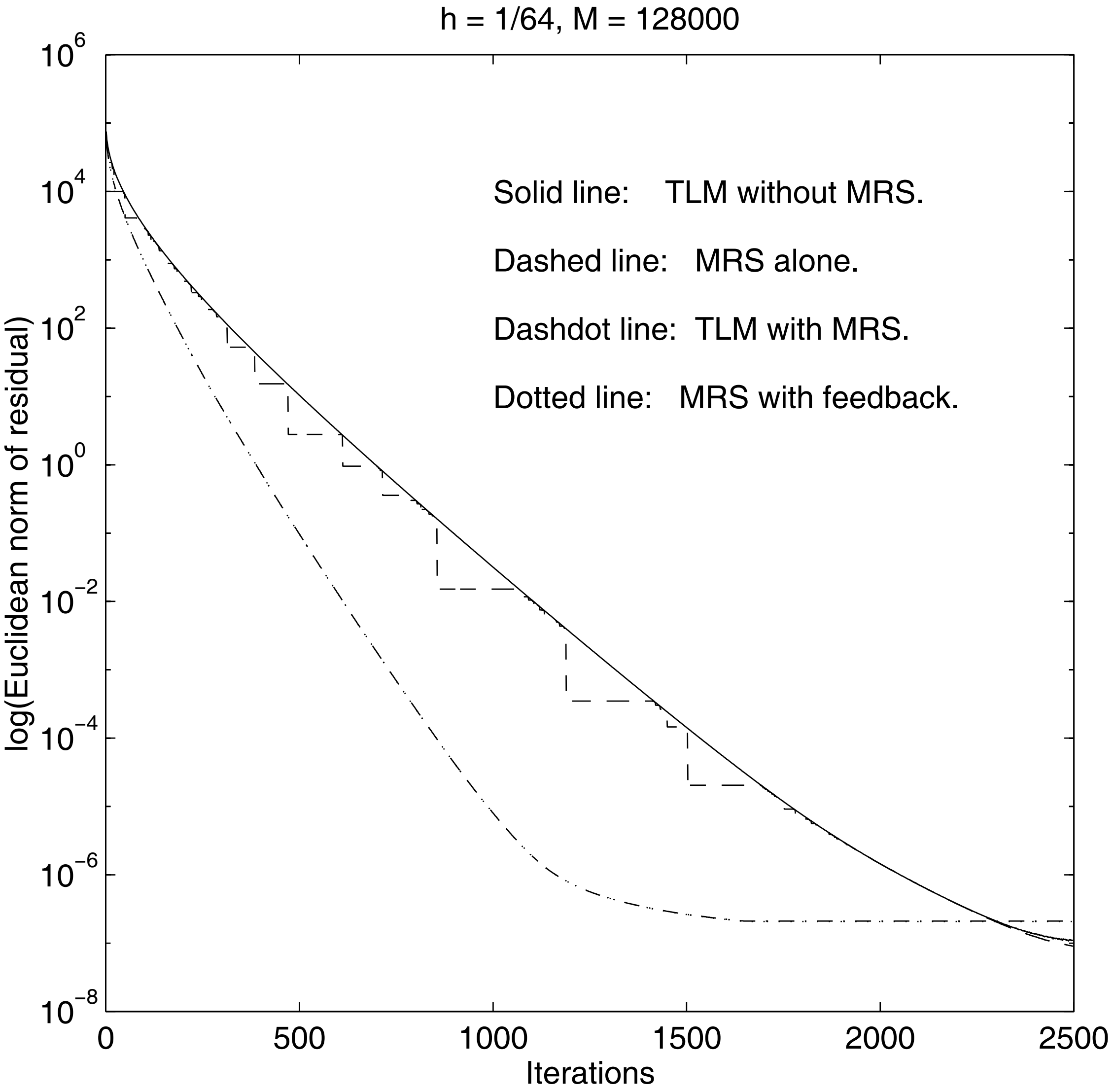

Figure 4.3: The convergence of the residual norms (in log scale) of TLM-MRS with $M = 1.28 \times 10^5$.

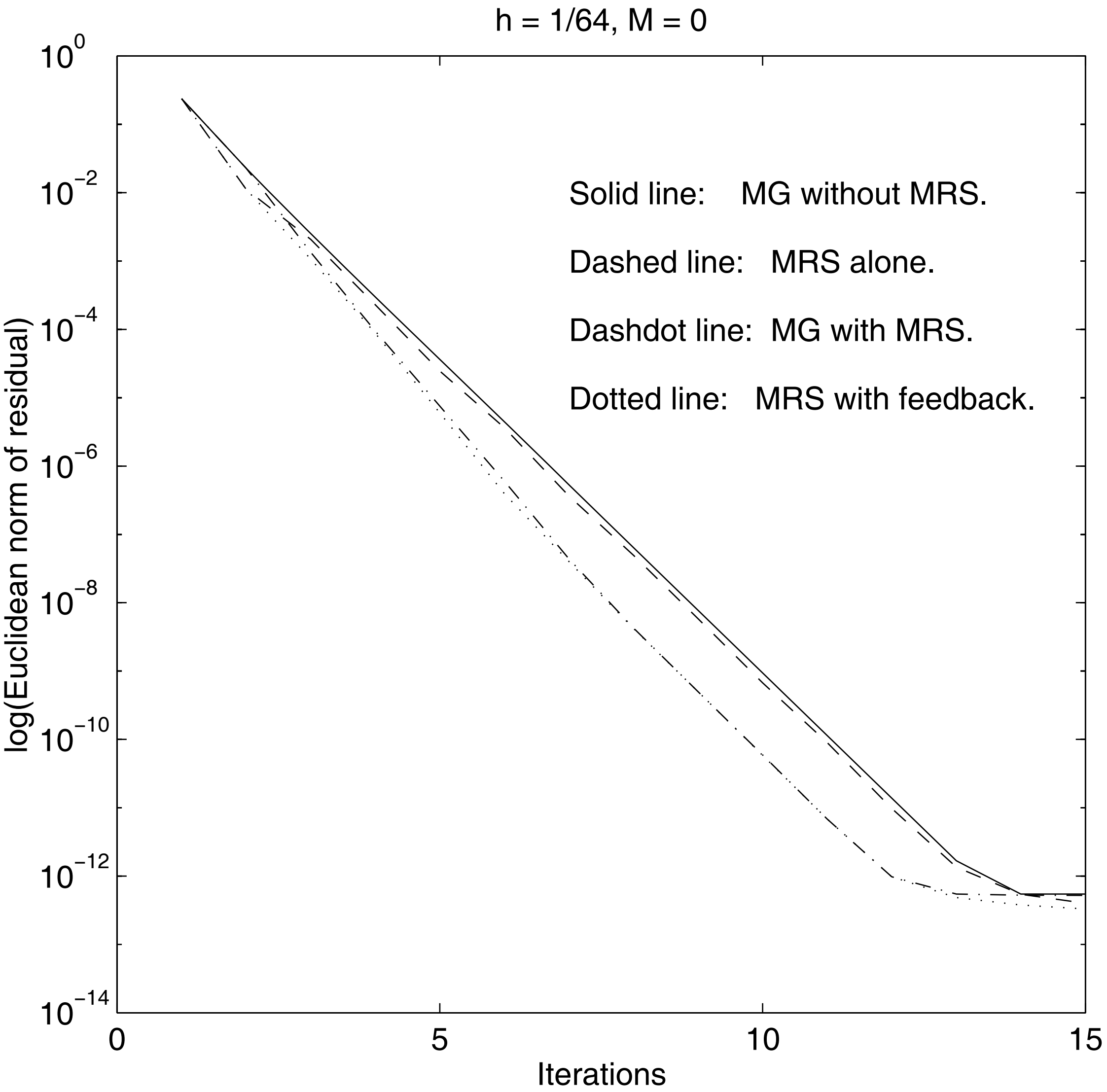

Figure 4.4: The convergence histories of the residual norms (in log scale) of MG-MRS with $M = 0$.

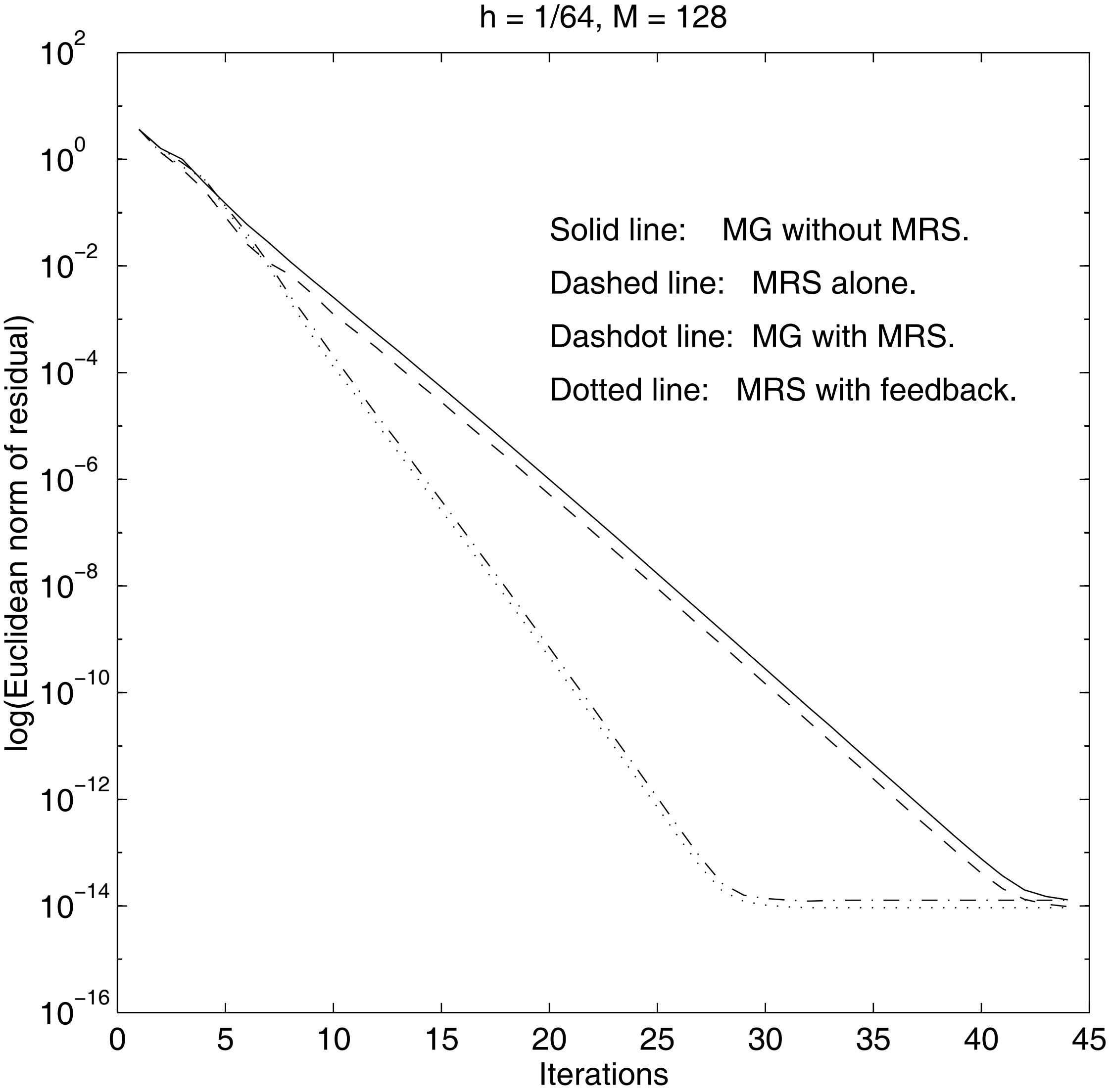

Figure 4.5: The convergence histories of the residual norms (in log scale) of MG-MRS with $M = 128$.

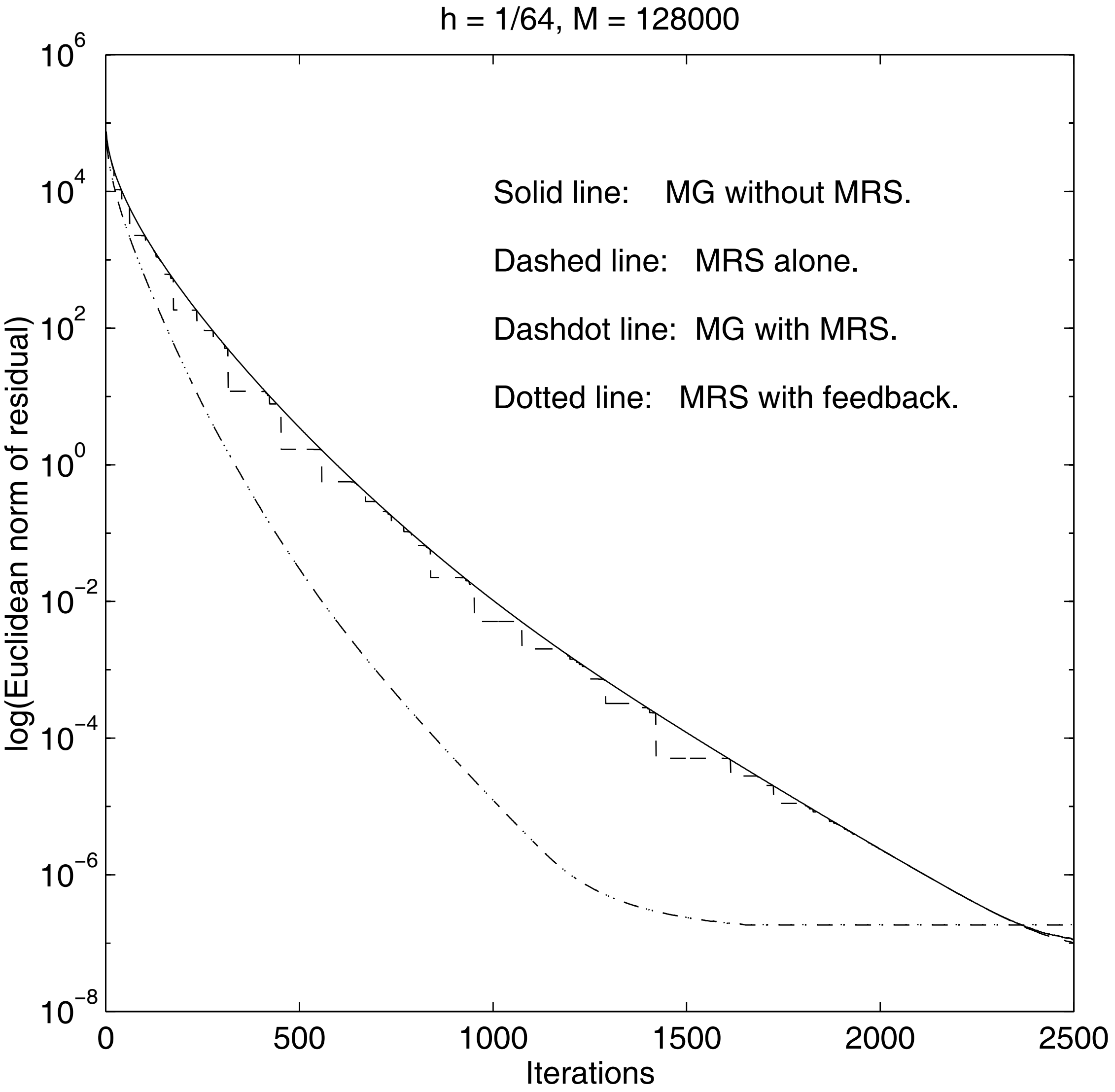

Figure 4.6: The convergence histories of the residual norms (in log scale) of MG-MRS with $M = 1.28 \times 10^5$.

Chapter 5

Analysis of Multigrid-MRS

5.1 Introduction

In this chapter, we analyze multigrid-MRS and try to show how this technique works. We also discuss some limitations of this acceleration technique and ideas of overcoming these limitations.

This chapter is organized as follows: In Section 5.2 we prove that the MRS procedure is a semi-iterative method with respect to the two-level method (TLM) and that TLM-MRS is a polynomial acceleration of first order. We explain why TLM-MRS does not work very well for the Poisson equation. The error and residual iteration matrices for the MRS accelerated coarse-grid-correction operator and for the two-level operator are obtained in Section 5.3. In Section 5.4 we prove some technical lemmas. In Section 5.5 we give preliminary analysis on TLM-MRS. In Section 5.6 we formulate some reasonable assumptions and give bounds for residual reduction rates of TLM-MRS. Conclusions and some remarks are included in Section 5.7.

5.2 MRS as a Semi-Iterative Method

Definition 5.2.1 *Let Q be an operator (matrix) on Ω^h, we define the operator norm of Q with respect to a symmetric, positive definite matrix Z as*

$$\|Q\|_Z = \sup_{0 \neq v \in \Omega^h} \frac{\|Qv\|_Z}{\|v\|_Z}, \tag{5.1}$$

where $v \neq 0$ is any non-zero vector on Ω^h.

We will analyze TLM-MRS of Algorithm 4.3.1, multigrid-MRS can be analyzed similarly. In Algorithm 4.3.1 the residual equation on the coarse grid is assumed to be solved exactly by a direct solver.

Theorem 5.2.2 *The MRS technique in Algorithm 4.3.1 is a semi-iterative method with respect to the standard two-level (multigrid) method. The TLM-MRS Algorithm 4.3.1 (and multigrid-MRS) is a polynomial acceleration of the first order.*

Proof. Let

$$\{u_0, u_1, u_2, \dots, u_k, \dots\} \tag{5.2}$$

be the sequence generated by the two-level iteration process after the pre-smoothing sweeps. Let

$$\{v_0, v_1, v_2, \dots, v_k, \dots\}$$

be the sequence generated by the MRS scheme from the TLM sequence (5.2). Hence, at the kth iteration, we have

$$v_k = v_{k-1} + \beta_k(u_k - v_{k-1})$$

by the definition of the MRS acceleration (see Algorithm 4.3.1).

We define a new sequence $\{z_k\}$ by

$$\left.\begin{array}{rcl} z_{2k} & = & u_k \\ z_{2k+1} & = & v_k \end{array}\right\} k = 0, 1, \dots.$$

It is obvious that the sequence $\{z_k\}$ is formed as

$$\{u_0, v_0, u_1, v_1, u_2, v_2, \dots, u_k, v_k, \dots\}.$$

It is easy to see that the sequence $\{z_k\}$ is actually the iterates of Algorithm 4.3.1, each new iterate z_k is generated by the procedure

$$z_k = z_{k-1} + \beta_k(\bar{z}_k - z_{k-1}) \tag{5.3}$$

with

$$z_{k-1} = v_k \qquad \text{and} \qquad \bar{z}_k = u_k.$$

Iteration procedure (5.3) is called by Varga [61] the *semi-iterative method* with respect to the basic two-level iterative method. The combined TLM-MRS is therefore the so-called *polynomial acceleration of the first order* due to Hageman and Young [37, p. 40]. □

If the relaxation used in TLM-MRS is the Gauss-Seidel, iteration procedure (5.3) is reminiscent of an SOR acceleration step. The only difference is that z_{k-1} is not the value of the previous Gauss-Seidel iteration, it is instead the value of the previous TLM-MRS iteration.

If TLM-MRS (or multigrid-MRS) using Gauss-Seidel is considered as TLM using SOR-type relaxation, we can expect that TLM-MRS (or multigrid-MRS) will not be efficient in solving the Poisson equation when the grid space is ordered in a red-black fashion and the discretization is the five-point second-order central difference scheme.

Remark 5.2.3 *By Propositions 2.3.1 and 2.3.2 in Chapter 2, if the Poisson equation is discretized by the standard five-point second-order central difference scheme and the red-black SOR relaxation is used in multigrid method, two different relaxation parameters are necessary to achieve efficient acceleration. An under-relaxation parameter ($\omega_1 < 1$) should be used in the projection half cycle and an over-relaxation ($\omega_2 > 1$) should be used in the interpolation half cycle.*

It has long been observed that the SOR acceleration is not effective to accelerate the nine-point Gauss-Seidel multigrid method [59]. Remark 5.2.3 may explain the numerical results that we observed in Chapter 4 that the efficiency of MRS acceleration scheme is reduced when it is used to accelerate the nine-point Poisson solver.

Brandt and Mikulinsky [11] considered some combining multigrid iterates to accelerate the convergence of problems with "small island." But they did not allow a continuous iterates to be formed like we did for the MRS sequence in Chapter 4 (and in [80]). What they formed there is some short sequence and the feature of the MRS technique was not fully exploited. They also suggested to apply their acceleration scheme only once on the coarse grid to reduce the cost of the acceleration.

5.3 Convergence Analysis

5.3.1 MRS with Coarse-Grid-Correction Operator

Let us first assume that there is no smoothing, i.e., $\nu_1 = \nu_2 = 0$, in order to analyze the effect of the MRS acceleration on the coarse-grid-correction operator. The coarse-grid-correction operator with respect to residual is given by [69, p. 90]

$$C = I - AP(A^H)^{-1}R. \tag{5.4}$$

For the standard two-level method (TLM), suppose that at the kth iteration, the residual is r_k, then after the kth coarse-grid-correction the residual changes to

$$r_{k+1} = Cr_k. \tag{5.5}$$

If TLM is accelerated by MRS (Algorithm 4.3.1), the residual after the kth MRS accelerated iteration reads

$$s_k = (1 - \beta_k)s_{k-1} + \beta_k r_k, \tag{5.6}$$

where β_k is given by Algorithm 4.3.1.

Hence, after the kth coarse-grid-correction, the new residual is

$$r_{k+1} = Cs_k = (1 - \beta_k)Cs_{k-1} + \beta_k Cr_k. \tag{5.7}$$

Since we have replaced the TLM residual r_{k-1} by the MRS residual s_{k-1} at the $(k-1)$th iteration, we have

$$r_k = Cs_{k-1} \tag{5.8}$$

by virtue of (5.5) with k being replaced by $k - 1$.

Substituting (5.8) into (5.7), we have the residual after the kth MRS accelerated coarse-grid-correction

$$\begin{aligned} r_{k+1} &= (1 - \beta_k)r_k + \beta_k Cr_k \\ &= [(1 - \beta_k)I + \beta_k C]r_k \\ &= [I - \beta_k AP(A^H)^{-1}R]r_k \end{aligned} \tag{5.9}$$

by using (5.4) and (5.8).

Therefore, the error after the kth MRS accelerated coarse-grid-correction is

$$e_{k+1} = A^{-1}[I - \beta_k AP(A^H)^{-1}R]r_k. \tag{5.10}$$

Theorem 5.3.1 *At the kth TLM-MRS iteration without smoothing, the error iteration matrix is given by*

$$E_k = I - \beta_k P(A^H)^{-1}RA \tag{5.11}$$

and the residual iteration matrix by

$$T_k = I - \beta_k AP(A^H)^{-1}R \tag{5.12}$$

with β_k given by Algorithm 4.3.1.

Proof. The residual iteration matrix (5.12) is obtained directly by (5.9).

At the kth iteration, the residual r_k and the corresponding error e_k satisfy the following error (residual) equation (see Remark 3.2.10 and equation (3.12))

$$Ae_k = r_k. \tag{5.13}$$

Substituting (5.13) into (5.10), we obtain (5.11) as the error iteration matrix of Algorithm 4.3.1 at the kth iteration without smoothing sweep. □

Comparing the MRS accelerated coarse-grid-correction residual operator (5.12) with the standard coarse-grid-correction residual operator (5.4), we have the following corollary:

Corollary 5.3.2 *The acceleration rate of the MRS acceleration scheme with the coarse-grid-correction operator at the kth iteration is given by*

$$\tau_k = \frac{\|I - \beta_k P(A^H)^{-1}RA\|_Z}{\|I - P(A^H)^{-1}RA\|_Z} \tag{5.14}$$

with β_k being the MRS parameter given by Algorithm 4.3.1. The norm $\|\cdot\|_Z$ is the operator norm defined in (5.1).

Although the acceleration rate is theoretically given by (5.14), computing τ_k by (5.14) for each k is still a much involved task. Direct computation of residuals from the MRS accelerated coarse-grid-correction residual operator and the standard coarse-grid-correction residual operator at the kth iteration as an estimate of τ_k may be more economic.

The MRS acceleration scheme may be heuristically viewed as scaling the operator $P(A^H)^{-1}RA$ by MRS parameter β_k so that τ_k is smaller than 1 (hopefully). Note here that (5.14) does not guarantee that $\tau_k < 1$.

From (5.14) it seems that the optimal scaling factor β for accelerating the convergence is the one that fulfills the following minimization problem

$$\|I - \beta P(A^H)^{-1}RA\|_Z = \min_{\bar{\beta}\in\mathbf{R}} \left\{ \sup_{0\neq v\in\Omega^h} \frac{\|[I - \bar{\beta}P(A^H)^{-1}RA]v\|_Z}{\|v\|_Z} \right\}, \tag{5.15}$$

where **R** is the set of all real numbers. However, solving minimization problem (5.15) is by no means realistic. In [46, 62], Reusken and Vaněk discussed the so-called over-correction technique that solves a minimization problem similar to (5.15) to optimize the computed correction (after the coarse-grid-correction procedure) with the assumption that $Z = A$ (symmetric and positive definite) and the number of post-smoothing sweeps is non-zero. As we remarked earlier, the requirement that A be symmetric and positive definite severely limits the application of the over-correction technique.

5.3.2 MRS with Two-Level Operator

Now we consider the case that the number of smoothing sweeps is not zero, i.e., $\nu_1+\nu_2 > 0$.

Let the coefficient matrix A be split as M and N satisfying the consistency condition (3.3) and the smoothing iterative method S be defined by Definition 3.2.3 as in (3.4).

Definition 5.3.3 *Denote the residual iteration matrix of the smoothing iterative method (3.4) by (see [69, p. 90])*

$$\tilde{M} = AMA^{-1} = I - AN.$$

For any integer $\nu > 1$, we recursively define

$$\tilde{M}^{\nu} = \tilde{M}\tilde{M}^{\nu-1}.$$

Lemma 5.3.4 *For any integer $\nu > 0$, the following identities are valid:*

$$\begin{aligned} \tilde{M}^{\nu} &= AM^{\nu}A^{-1}, & (5.16)\\ A^{-1}\tilde{M}^{\nu}A &= M^{\nu}, & (5.17)\\ A^{-1}\tilde{M}^{\nu} &= M^{\nu}A^{-1}, & (5.18)\\ \tilde{M}^{\nu}A &= AM^{\nu}. & (5.19) \end{aligned}$$

Proof. (5.16) is proved by induction on ν. (5.17) follows immediately from (5.16) and Definition 5.3.3. (5.18) and (5.19) are special cases of (5.17). □

The two-level residual iteration operator with ν_1 pre-smoothing and ν_2 post-smoothing sweeps is given by [69, p. 90]

$$\tilde{C} = \tilde{M}^{\nu_2}C\tilde{M}^{\nu_1}, \tag{5.20}$$

where C is the coarse-grid-correction residual operator (5.4).

The residual after the kth TLM-MRS iteration is

$$\begin{aligned} r_{k+1} &= \tilde{C}s_k\\ &= \tilde{M}^{\nu_2}C\tilde{M}^{\nu_1}[(1-\beta_k)s_{k-1} + \beta_k r_k]\\ &= (1-\beta_k)\tilde{M}^{\nu_2}C\tilde{M}^{\nu_1}s_{k-1} + \beta_k\tilde{M}^{\nu_2}C\tilde{M}^{\nu_1}r_k. \qquad (5.21) \end{aligned}$$

By the definitions of the MRS acceleration and of the standard TLM residual iteration operator (5.20), we have

$$r_k = \tilde{C} s_{k-1}. \tag{5.22}$$

Substituting (5.22) into (5.21), after the kth TLM-MRS iteration, we obtain the new residual

$$\begin{aligned} r_{k+1} &= [(1-\beta_k)I + \beta_k \tilde{M}^{\nu_2} C \tilde{M}^{\nu_1}] r_k \\ &= \{(1-\beta_k)I + \beta_k \tilde{M}^{\nu_2}[I - AP(A^H)^{-1}R]\tilde{M}^{\nu_1}\} r_k. \end{aligned} \tag{5.23}$$

Theorem 5.3.5 *At the kth TLM-MRS iteration, the error iteration matrix is given by*

$$\tilde{E}_k = I - \beta_k[I - M^{\nu_1+\nu_2} + M^{\nu_2}P(A^H)^{-1}RAM^{\nu_1}] \tag{5.24}$$

and the residual iteration matrix by

$$\tilde{T}_k = (1-\beta_k)I + \beta_k \tilde{C} \tag{5.25}$$

$$= I - \beta_k[I - AM^{\nu_1+\nu_2}A^{-1} + AM^{\nu_2}P(A^H)^{-1}RAM^{\nu_1}]. \tag{5.26}$$

Proof. The residual iteration matrices (5.25) and (5.26) are obtained directly from (5.23) and Lemma 5.3.4.

The proof of (5.24) follows from (5.13), (5.23) and Lemma 5.3.4:

$$\begin{aligned} \tilde{E}_k &= (1-\beta_k)I + \beta_k A^{-1}\tilde{M}^{\nu_2}[I - AP(A^H)^{-1}R]\tilde{M}^{\nu_1}A \\ &= (1-\beta_k)I + \beta_k M^{\nu_2}A^{-1}[I - AP(A^H)^{-1}R]AM^{\nu_1} \\ &= I - \beta_k[I - M^{\nu_1+\nu_2} + M^{\nu_2}P(A^H)^{-1}RAM^{\nu_1}]. \end{aligned}$$

This finishes the proof of Theorem 5.3.5. □

5.4 Some Technical Lemmas and Notations

Definition 5.4.1 *Let B be a set, the cardinality of B is defined as*

$$\mathrm{card}(B) = \textit{the number of elements contained in } B.$$

Definition 5.4.2 *For each integer $k \geq 1$, let P_k denote a set of 2^k ordered set pairs*

$$\Big\{\{M_1^k, N_1^k\}, \{M_2^k, N_2^k\}, \ldots, \{M_i^k, N_i^k\}, \ldots, \{M_{2^k}^k, N_{2^k}^k\}\Big\}$$

with the element pair M_i^k and N_i^k defined as

$$M_i^k = \{m_i : m_i \textit{ are natural numbers}, 1 \leq m_i \leq k\}$$

and

$$N_i^k = \{n_i : n_i \textit{ are natural numbers}, 1 \leq n_i \leq k\}$$

such that the following two conditions are satisfied:

$$\mathrm{card}(M_i^k) + \mathrm{card}(N_i^k) = k \tag{5.27}$$

and

$$\mathrm{card}(M_i^k \cup N_i^k) = k. \tag{5.28}$$

Remark 5.4.3 *For each integer $k \geq 1$, the elements of M_i^k are necessarily different from the elements of N_i^k for each $1 \leq i \leq 2^k$, i.e.*

$$M_i^k \cap N_i^k = \emptyset \quad \textit{(the empty set)}$$

due to conditions (5.27) and (5.28). The union of M_i^k and N_i^k contains all natural numbers less than and equal to k.

Remark 5.4.4 *For any integer $k \geq 1$, if the equality*

$$\mathrm{card}(M_i^k) = k$$

holds for some $1 \leq i \leq 2^k$, then

$$N_i^k = \emptyset$$

and vice versa.

Remark 5.4.5 *For any integers $k \geq 1$ and $1 \leq i \leq 2^k$,*

$$\{M_i^k, N_i^k\} \neq \{N_i^k, M_i^k\},$$

because each element of P_k is an ordered set pair.

Remark 5.4.6 *For each integer $k \geq 1$, the elements of P_k are formed by dividing all natural numbers from 1 to k into two sets. P_k contains all possible divisions.*

Lemma 5.4.7 *For any integer $k \geq 1$, the set P_{k+1} may be formed from the set P_k by the union of two auxiliary sets:*

$$P_{k+1}^1 = \left\{\{\bar{M}_1^{k+1}, N_1^k\}, \{\bar{M}_2^{k+1}, N_2^k\}, \ldots, \{\bar{M}_i^{k+1}, N_i^k\}, \ldots, \{\bar{M}_{2^k}^{k+1}, N_{2^k}^k\}\right\}$$

and

$$P_{k+1}^2 = \left\{\{M_1^k, \bar{N}_1^{k+1}\}, \{M_2^k, \bar{N}_2^{k+1}\}, \ldots, \{M_i^k, \bar{N}_i^{k+1}\}, \ldots, \{M_{2^k}^k, \bar{N}_{2^k}^{k+1}\}\right\},$$

where for each $1 \leq i \leq 2^k$

$$\begin{aligned} \bar{M}_i^{k+1} &= M_i^k \cup \{k+1\}, \\ \bar{N}_i^{k+1} &= N_i^k \cup \{k+1\}. \end{aligned}$$

Proof. By Remark 5.4.5, each element of P_k is ordered and it follows that

$$P_{k+1}^1 \cap P_{k+1}^2 = \emptyset. \tag{5.29}$$

Hence,

$$\mathrm{card}(P_{k+1}^1 \cup P_{k+1}^2) = \mathrm{card}(P_{k+1}^1) + \mathrm{card}(P_{k+1}^2) = 2^k + 2^k = 2^{k+1} = \mathrm{card}(P_{k+1}). \tag{5.30}$$

It is clear that each element of P_{k+1}^1 and P_{k+1}^2 is formed by dividing the natural numbers from 1 to $(k+1)$ into two sets. It follows from Remark 5.4.6, (5.29) and (5.30) that

$$P_{k+1} = P_{k+1}^1 \cup P_{k+1}^2.$$

□

Definition 5.4.8 *For any integer $k \geq 1$ and a set of k elements*

$$\{x_1, x_2, \ldots, x_j, \ldots, x_k\}, \tag{5.31}$$

let M_i^k be an index set containing at most k indices of set (5.31), for example, for some $1 \leq l \leq k$

$$M_i^k = \{m_{i_1}, m_{i_2}, \ldots, m_{i_l}\},$$

where

$$1 \leq m_{i_s} \leq k, \qquad for \qquad s = 1, 2, \ldots, l.$$

We define the product of all elements of set (5.31) whose indices are contained in set M_i^k by the symbol

$$\prod_{m \in M_i^k} x_m = x_{m_{i_1}} x_{m_{i_2}} \cdots x_{m_{i_{l-1}}} x_{m_{i_l}}.$$

If $M_i^k = \emptyset$, we define

$$\prod_{m \in \emptyset} x_m = 1.$$

Lemma 5.4.9 *For any integer $k \geq 1$, let*

$$\{x_1, x_2, \ldots, x_j, \ldots, x_k\},$$
$$\{y_1, y_2, \ldots, y_j, \ldots, y_k\}$$

be two sets, each contains k elements, then the following identity holds:

$$\prod_{j=1}^{k} (x_j + y_j) = \sum_{i=1}^{2^k} \left(\prod_{m \in M_i^k} x_m \right) \left(\prod_{n \in N_i^k} y_n \right), \tag{5.32}$$

where for each $1 \leq i \leq 2^k$

$$\{M_i^k, N_i^k\} \in P_k.$$

Proof. We prove Lemma 5.4.9 by induction on k.

The lemma is true for $k = 1$, since

$$\begin{aligned} x_1 + y_1 &= \left(\prod_{m \in \{1\}} x_m \right) \left(\prod_{n \in \emptyset} y_n \right) + \left(\prod_{m \in \emptyset} x_m \right) \left(\prod_{n \in \{1\}} y_n \right) \\ &= \sum_{i=1}^{2} \left(\prod_{m \in M_i^1} x_m \right) \left(\prod_{n \in N_i^1} y_n \right), \end{aligned}$$

and

$$\{M_i^1, N_i^1\} \in P_1, \qquad \text{for} \qquad i = 1, 2.$$

Let the lemma be true for k, then

$$\begin{aligned}
\prod_{j=1}^{k+1}(x_j+y_j) &= (x_{k+1}+y_{k+1})\prod_{j=1}^{k}(x_j+y_j) \\
&= (x_{k+1}+y_{k+1})\left[\sum_{i=1}^{2^k}\left(\prod_{m\in M_i^k} x_m\right)\left(\prod_{n\in N_i^k} y_n\right)\right] \\
&= x_{k+1}\left[\sum_{i=1}^{2^k}\left(\prod_{m\in M_i^k} x_m\right)\left(\prod_{n\in N_i^k} y_n\right)\right] + y_{k+1}\left[\sum_{i=1}^{2^k}\left(\prod_{m\in M_i^k} x_m\right)\left(\prod_{n\in N_i^k} y_n\right)\right] \\
&= \sum_{i=1}^{2^k}\left[\left(\prod_{m\in M_i^k} x_m\right)x_{k+1}\right]\left(\prod_{n\in N_i^k} y_n\right) + \sum_{i=1}^{2^k}\left(\prod_{m\in M_i^k} x_m\right)\left[\left(\prod_{n\in N_i^k} y_n\right)y_{k+1}\right] \\
&= \sum_{i=1}^{2^k}\left(\prod_{m\in M_i^k\cup\{k+1\}} x_m\right)\left(\prod_{n\in N_i^k} y_n\right) + \sum_{i=1}^{2^k}\left(\prod_{m\in M_i^k} x_m\right)\left(\prod_{n\in N_i^k\cup\{k+1\}} y_n\right) \\
&= \sum_{i=1}^{2^k}\left(\prod_{m\in \bar{M}_i^{k+1}} x_m\right)\left(\prod_{n\in N_i^k} y_n\right) + \sum_{i=1}^{2^k}\left(\prod_{m\in M_i^k} x_m\right)\left(\prod_{n\in \bar{N}_i^{k+1}} y_n\right) \\
&= \sum_{i=1}^{2^{k+1}}\left(\prod_{m\in M_i^{k+1}} x_m\right)\left(\prod_{n\in N_i^{k+1}} y_n\right).
\end{aligned}$$

The last equality is valid due to Lemma 5.4.7, since

$$\left.\begin{array}{lll} \{\bar{M}_i^{k+1}, N_i^k\} & \in & P_{k+1}^1 \\ \{M_i^k, \bar{N}_i^{k+1}\} & \in & P_{k+1}^2 \end{array}\right\}, \qquad \text{for each} \qquad 1 \le i \le 2^k,$$

we have

$$\{M_i^{k+1}, N_i^{k+1}\} \in P_{k+1}, \qquad \text{for each} \qquad 1 \le i \le 2^{k+1}.$$

This finishes the proof of Lemma 5.4.9. □

5.5 Preliminary Results

The following assumption is motivated by the work of Brandt and Mikulinsky [11].

Assumption 5.5.1 *Let $\tilde{C}$ be the residual iteration matrix of the two-level method given by (5.20). Let any initial residual r_0 be decomposed into one possibly slow component $r_0^{(s)}$ and a remainder $r_0^{(f)}$ made up of fast components, i.e.*

$$r_0 = r_0^{(s)} + r_0^{(f)},$$

where

$$\tilde{C} r_0^{(s)} = \varepsilon_s r_0^{(s)}$$

and

$$\|\tilde{C} r_0^{(f)}\|_Z \le \varepsilon_f \|r_0^{(f)}\|_Z.$$

ε_s and ε_f measure the convergence of the slow and fast components, respectively. We also assume that the following condition holds

$$0 < \varepsilon_f \ll 0.5 \ll |\varepsilon_s| \le 1. \tag{5.33}$$

Remark 5.5.2 *The residual after the kth two-level cycle is (see [11])*

$$r_{k+1} = \tilde{C}^{k+1}(r_0^{(s)} + r_0^{(f)}) = \varepsilon_s^{k+1} r_0^{(s)} + r_{k+1}^{(f)},$$

where

$$\|r^{(f)}\|_Z \le \varepsilon_f^{k+1}\|r_0^{(f)}\|_Z. \tag{5.34}$$

Now, we prove the main lemma of this section:

Lemma 5.5.3 *Let the initial residual r_0 satisfy all conditions of Assumption 5.5.1, then after the kth TLM-MRS iteration, the slow and fast components of the residual $\tilde{r}_{k+1}$satisfy:*

$$\begin{aligned} \tilde{r}_{k+1}^{(s)} &= \left[\prod_{j=1}^{k}(1-\beta_j+\beta_j\varepsilon_s)\right]\ \varepsilon_s r_0^{(s)}, & (5.35)\\ \|\tilde{r}_{k+1}^{(f)}\|_Z &\le \left[\prod_{j=1}^{k}(|1-\beta_j|+|\beta_j|\varepsilon_f)\right]\ \varepsilon_f\|r_0^{(f)}\|_Z. & (5.36)\end{aligned}$$

Proof. Let $\tilde{T}_j, j = 0, 1, 2, \cdots$, be the residual iteration matrices of the TLM-MRS algorithm given by Theorem 5.3.5 (see (5.25) and (5.26)). We first note that $\tilde{T}_0 = \tilde{C}$ because $\beta_0 = 1$ by the definition of Algorithm 4.3.1.

The proof of Lemma 5.5.3 follows from Remark 5.5.2 and Lemma 5.4.9 with

$$\left.\begin{aligned} x_j &= (1-\beta_j)I \\ y_j &= \beta_j\tilde{C} \end{aligned}\right\}, \qquad \text{for each} \qquad 1 \le j \le k.$$

$$\begin{aligned} \tilde{r}_{k+1} &= \prod_{j=0}^{k}\tilde{T}_k r_0 \\ &= \left(\prod_{j=1}^{k}\tilde{T}_k\right)\tilde{T}_0 r_0 \\ &= \prod_{j=1}^{k}\left[(1-\beta_j)I+\beta_j\tilde{C}\right]\tilde{C}r_0 \\ &= \sum_{i=1}^{2^k}\left[\prod_{m\in M_i^k}(1-\beta_m)I\prod_{n\in N_i^k}(\beta_n\tilde{C})\right]\tilde{C}r_0 \\ &= \sum_{i=1}^{2^k}\left[\prod_{m\in M_i^k}(1-\beta_m)I\left(\prod_{n\in N_i^k}\beta_n\right)\tilde{C}^{\mathrm{card}(N_i^k)}\ \tilde{C}r_0\right] \end{aligned}$$

$$\begin{aligned}
&= \sum_{i=1}^{2^k} \left[\prod_{m \in M_i^k} (1-\beta_m) I \left(\prod_{n \in N_i^k} \beta_n \right) \tilde{C}^{\mathrm{card}(N_i^k)+1} (r_0^{(s)} + r_0^{(f)}) \right] \\
&= \sum_{i=1}^{2^k} \left[\prod_{m \in M_i^k} (1-\beta_m) I \left(\prod_{n \in N_i^k} \beta_n \right) \left(\varepsilon_s^{\mathrm{card}(N_i^k)+1} r_0^{(s)} + r^{(f)}_{\mathrm{card}(N_i^k)+1} \right) \right] \\
&= \sum_{i=1}^{2^k} \left[\prod_{m \in M_i^k} (1-\beta_m) I \left(\prod_{n \in N_i^k} \beta_n \right) \left(\varepsilon_s^{\mathrm{card}(N_i^k)+1} r_0^{(s)} \right) \right] \\
&\quad + \sum_{i=1}^{2^k} \left[\prod_{m \in M_i^k} (1-\beta_m) I \left(\prod_{n \in N_i^k} \beta_n \right) \left(r^{(f)}_{\mathrm{card}(N_i^k)+1} \right) \right] \\
&= \tilde{r}^{(s)}_{k+1} + \tilde{r}^{(f)}_{k+1}.
\end{aligned}$$

Hence,

$$\begin{aligned}
\tilde{r}^{(s)}_{k+1} &= \sum_{i=1}^{2^k} \left[\prod_{m \in M_i^k} (1-\beta_m) I \left(\prod_{n \in N_i^k} \beta_n \right) \left(\varepsilon_s^{\mathrm{card}(N_i^k)+1} r_0^{(s)} \right) \right] \\
&= \sum_{i=1}^{2^k} \left[\prod_{m \in M_i^k} (1-\beta_m) \left(\prod_{n \in N_i^k} \beta_n \right) \left(\varepsilon_s^{\mathrm{card}(N_i^k)} \ \varepsilon_s r_0^{(s)} \right) \right] \\
&= \sum_{i=1}^{2^k} \left[\prod_{m \in M_i^k} (1-\beta_m) \left(\prod_{n \in N_i^k} \beta_n \right) \left(\varepsilon_s^{\mathrm{card}(N_i^k)} \right) \right] \ \varepsilon_s r_0^{(s)} \\
&= \sum_{i=1}^{2^k} \left[\prod_{m \in M_i^k} (1-\beta_m) \prod_{n \in N_i^k} (\beta_n \varepsilon_s) \right] \ \varepsilon_s r_0^{(s)} \\
&= \left[\prod_{j=1}^{k} (1-\beta_j + \beta_j \varepsilon_s) \right] \ \varepsilon_s r_0^{(s)}.
\end{aligned}$$

The last equality follows from Lemma 5.4.9 with

$$\left.\begin{aligned} x_j &= 1-\beta_j \\ y_j &= \ \ \beta_j \varepsilon_s \end{aligned}\right\}, \qquad \text{for each} \qquad 1 \leq j \leq k.$$

This proves equality (5.35).

Furthermore,

$$\tilde{r}^{(f)}_{k+1} = \sum_{i=1}^{2^k} \left[\prod_{m \in M_i^k} (1-\beta_m) I \left(\prod_{n \in N_i^k} \beta_n \right) \left(r^{(f)}_{\mathrm{card}(N_i^k)+1} \right) \right].$$

It follows from (5.34) that

$$\|\tilde{r}^{(f)}_{k+1}\|_Z \leq \sum_{i=1}^{2^k} \left[\prod_{m \in M_i^k} |1-\beta_m| \left(\prod_{n \in N_i^k} |\beta_n| \right) \left(\|r^{(f)}_{\mathrm{card}(N_i^k)+1}\|_Z \right) \right]$$

$$
\begin{aligned}
&\leq \sum_{i=1}^{2^k}\left[\prod_{m\in M_i^k}|1-\beta_m|\left(\prod_{n\in N_i^k}|\beta_n|\right)\left(\varepsilon_f^{\mathrm{card}(N_i^k)+1}\|r_0^{(f)}\|_Z\right)\right]\\
&= \sum_{i=1}^{2^k}\left[\prod_{m\in M_i^k}|1-\beta_m|\left(\prod_{n\in N_i^k}|\beta_n|\right)\left(\varepsilon_f^{\mathrm{card}(N_i^k)}\right)\right]\varepsilon_f\|r_0^{(f)}\|_Z\\
&= \sum_{i=1}^{2^k}\left[\prod_{m\in M_i^k}|1-\beta_m|\prod_{n\in N_i^k}|\beta_n\varepsilon_f|\right]\varepsilon_f\|r_0^{(f)}\|_Z\\
&= \prod_{j=1}^{k}\left(|1-\beta_j|+|\beta_j|\varepsilon_f\right)\varepsilon_f\|r_0^{(f)}\|_Z.
\end{aligned}
$$

The last equality follows from Lemma 5.4.9 with

$$
\left.\begin{aligned} x_j &= |1-\beta_j| \\ y_j &= |\beta_j|\varepsilon_f \end{aligned}\right\}, \qquad \text{for each} \qquad 1\leq j\leq k.
$$

This proves inequality (5.36) and we finish the proof of Lemma 3.2.15. □

Corollary 5.5.4 *After the kth TLM-MRS iteration, the residual norm satisfies*

$$
\begin{aligned}
\|\tilde{r}_{k+1}\|_Z &= \|\prod_{j=0}^{k}\tilde{T}_j r_0\|_Z\\
&\leq |\Theta_k(\varepsilon_s)|\ \|r_0^{(s)}\|_Z+\Theta_k(\varepsilon_f)\ \|r_0^{(f)}\|_Z, \qquad (5.37)
\end{aligned}
$$

where

$$
\Theta_k(\varepsilon_s)=\left[\prod_{j=1}^{k}(1-\beta_j+\beta_j\varepsilon_s)\right]\varepsilon_s \qquad (5.38)
$$

and

$$
\Theta_k(\varepsilon_f)=\left[\prod_{j=1}^{k}(|1-\beta_j|+|\beta_j|\varepsilon_f)\right]\varepsilon_f. \qquad (5.39)
$$

$|\Theta_k(\varepsilon_s)|$ and $\Theta_k(\varepsilon_f)$ measure the reduction rates of the slow and fast components of the residual with TLM-MRS, respectively. $|\Theta_k(\varepsilon_s)|^{1/(k+1)}$ and $\Theta_k(\varepsilon_f)^{1/(k+1)}$ are the average reduction factors of the slow and fast residual components for the first $(k+1)$ TLM-MRS iterations.

5.6 Bounds of Residual Reduction Rates

Acceleration is achieved by speeding up the convergence of the slow component, this leads to the following assumption:

Assumption 5.6.1 *Let there exist some* $0<\hat{\varepsilon}_s<1$ *such that* $\hat{\varepsilon}_s<|\varepsilon_s|$ *and*

$$
|1-\beta_k+\beta_k\varepsilon_s|\leq\hat{\varepsilon}_s, \qquad \textit{for all} \qquad k\geq 1. \qquad (5.40)
$$

$\hat{\varepsilon}_s$ is an upper bound of the residual reduction rate of TLM-MRS.

Lemma 5.6.2 *If Assumption 5.6.1 holds, then the MRS parameter β_k satisfies:*

$$\beta_k \geq \frac{1-\hat{\varepsilon}_s}{1-\varepsilon_s} = \tilde{\beta} \tag{5.41}$$

or

$$\beta_k \leq \frac{1+\hat{\varepsilon}_s}{1-\varepsilon_s} = \bar{\beta}, \tag{5.42}$$

for each $k \geq 1$.

Proof. Inequalities (5.41) and (5.42) are the solutions of inequality (5.40). □

Lemma 5.6.3 *Let Assumptions 5.5.1 and 5.6.1 hold, then*

$$|\Theta_k(\varepsilon_s)| \leq |\varepsilon_s|\hat{\varepsilon}_s^k \tag{5.43}$$

holds for any $k \geq 1$.

Proof. Inequality (5.43) follows from (5.38) and (5.40). □

Lemma 5.6.4 *Let Assumptions 5.5.1 and 5.6.1 and the inequality*

$$-1 \leq \varepsilon_s < -1/2 \tag{5.44}$$

hold. We define

$$\varepsilon_f(\varepsilon_s, \hat{\varepsilon}_s, \varepsilon_f)_k \overset{\text{def}}{=} |1-\beta_k| + |\beta_k|\varepsilon_f. \tag{5.45}$$

If

$$\frac{\varepsilon_f - \varepsilon_s}{2-\varepsilon_s-\varepsilon_f} \leq \hat{\varepsilon}_s \tag{5.46}$$

holds, then

$$\varepsilon_f(\varepsilon_s, \hat{\varepsilon}_s, \varepsilon_f)_k \leq \hat{\varepsilon}_s.$$

Proof. By assumption (5.44) and Lemma 5.6.2 (5.42), we have

$$0 < |\beta_k| \leq \bar{\beta} = \frac{1+\hat{\varepsilon}_s}{1-\varepsilon_s} < 1 \tag{5.47}$$

because $\hat{\varepsilon}_s < |\varepsilon_s|$ leads to $\hat{\varepsilon}_s < -\varepsilon_s$.

It follows from (5.45), (5.46) and (5.47) that, for each $k \geq 1$,

$$\begin{aligned}
\varepsilon_f(\varepsilon_s, \hat{\varepsilon}_s, \varepsilon_f)_k &\leq 1-\bar{\beta}+\bar{\beta}\varepsilon_f \\
&= \frac{\varepsilon_f - \varepsilon_s - (1-\varepsilon_f)\hat{\varepsilon}_s}{1-\varepsilon_s} \\
&\leq \frac{\varepsilon_f-\varepsilon_s}{2-\varepsilon_s-\varepsilon_f} \\
&\leq \hat{\varepsilon}_s.
\end{aligned}$$

This proves Lemma 5.6.4. □

Lemma 5.6.5 *Under the conditions of Lemma 5.6.4, the following bounds hold:*

$$\frac{1}{5} < \frac{\varepsilon_f - \varepsilon_s}{2 - \varepsilon_s - \varepsilon_f} < \frac{3}{5}.$$

Proof. The lower bound is obtained as $\varepsilon_f \to 0$ and $\varepsilon_s \to -1/2$; the upper bound is obtained when $\varepsilon_f \to 1/2$ and $\varepsilon_s \to -1$. □

Remark 5.6.6 *Lemmas 5.6.4 and 5.6.5 imply that under the conditions of Lemma 5.6.4, MRS may not provide any acceleration if the residual reduction rate of the underlying TLM is smaller than 1/5. In other words, TLM-MRS may not be better than TLM if the latter converges very fast.*

Lemma 5.6.7 *Under the conditions of Lemma 5.6.4, we have*

$$\Theta_k(\varepsilon_f) \leq \varepsilon_f \ \hat{\varepsilon}_s^k. \tag{5.48}$$

Proof. The proof follows from (5.39) and Lemma 5.6.4. □

Theorem 5.6.8 *Under the conditions of Lemma 5.6.4, at the kth TLM-MRS iteration the residual satisfies*

$$\|r_{k+1}\|_Z \leq \alpha \ \hat{\varepsilon}_s^k$$

with

$$\hat{\varepsilon}_s^k \to 0, \qquad as \qquad k \to \infty,$$

where

$$\alpha = |\varepsilon_s| \ \|r_0^{(s)}\|_Z + \varepsilon_f \ \|r_0^{(f)}\|_Z$$

is a constant independent of k.

Proof. By Corollary 5.5.4, Lemma 5.6.3 and Lemma 5.6.7, we have at the kth TLM-MRS iteration:

$$\begin{array}{rcl}
\|r_{k+1}\|_Z & \leq & |\varepsilon_s| \ \hat{\varepsilon}_s^k \ \|r_0^{(s)}\|_Z + \varepsilon_f \ \hat{\varepsilon}_s^k \ \|r_0^{(f)}\|_Z \\
& = & (|\varepsilon_s| \ \|r_0^{(s)}\|_Z + \varepsilon_f \ \|r_0^{(f)}\|_Z) \ \hat{\varepsilon}_s^k \\
& = & \alpha\hat{\varepsilon}_s^k \\
& \to & 0, \qquad \text{as} \qquad k \to 0.
\end{array}$$

This proves Theorem 5.6.8. □

Lemma 5.6.9 *Let Assumptions 5.5.1 and 5.6.1 and the inequality*

$$1/2 < \varepsilon_s \leq 1$$

hold, then

$$\hat{\varepsilon}_s \geq \frac{\varepsilon_f + \varepsilon_s}{2 + \varepsilon_f - \varepsilon_s}. \tag{5.49}$$

Proof. By Lemma 5.6.2 (5.41), we have

$$\beta_k \geq \tilde{\beta} = \frac{1-\hat{\varepsilon}_s}{1-\varepsilon_s} > 1$$

because $\hat{\varepsilon}_s < \varepsilon_s$.

If bound (5.49) fails, we have

$$\begin{array}{rcl} \varepsilon_f(\varepsilon_s, \hat{\varepsilon}_s, \varepsilon_f)_k & \geq & \hat{\beta} - 1 + \hat{\beta}\varepsilon_f \\ & = & \dfrac{\varepsilon_f + \varepsilon_s - (1+\varepsilon_f)\hat{\varepsilon}_s}{1-\varepsilon_s} \\ & \geq & \dfrac{\varepsilon_f + \varepsilon_s}{2+\varepsilon_f - \varepsilon_s} \\ & \geq & \hat{\varepsilon}_s, \end{array}$$

which is impossible because the reduction rate of the fast components should be smaller than the reduction rate of the slow component. Hence bound (5.49) must hold. □

Unlike Lemma 5.6.4, Lemma 5.6.9 does not state that MRS will provide any acceleration. It only sets an lower bound for the possible residual reduction rate.

Lemma 5.6.10 *Under the conditions of Lemma 5.6.9, the following bounds hold:*

$$\frac{1}{3} < \frac{\varepsilon_f + \varepsilon_s}{2+\varepsilon_f - \varepsilon_s} \leq 1.$$

Proof. The lower bound is obtained when $\varepsilon_f \to 0$ and $\varepsilon_s \to 1/2$; the upper bound is obtained when $\varepsilon_s \to 1$, independent of ε_f. □

5.7 Conclusions and Remarks

We have proved that the minimal residual smoothing (MRS) technique is a semi-iterative method with respect to the original two-level or multigrid method, and that the TLM-MRS or multigrid-MRS method is a polynomial acceleration of the first order. We also give the error and residual iteration matrices of the MRS accelerated coarse-grid-operator and TLM-MRS. Under our assumptions, we obtained quantitative lower and upper bounds for the TLM-MRS residual reduction rates which support our numerical results in Chapter 4. We gave heuristic arguments that TLM-MRS and multigrid-MRS may not be efficient to solve diffusion-dominated (Poisson-type) problems.

We note that many analytical results can be generalized to multigrid-MRS straightforwardly by using the recursive idea. Some results, e.g., Theorem 5.2.2, can be generalized to any underlying method accelerated by MRS acceleration scheme.

Chapter 6

High Accuracy Solution of the Convection-Diffusion Equation

6.1 Introduction

Numerical simulation of the convection-diffusion equations plays a very important role in modern large scale scientific computation, especially in computational fluid dynamics. The general convection-diffusion equation is of the form

$$\left.\begin{array}{rcll} u_{xx}+u_{yy}+p(x,y)u_x+q(x,y)u_y & = & f(x,y), & (x,y)\in\Omega, \\ u(x,y) & = & g(x,y), & (x,y)\in\partial\Omega. \end{array}\right\} \tag{6.1}$$

where $p(x,y)$ and $q(x,y)$ are continuously differentiable functions of x and y. Ω is a convex domain and $\partial\Omega$ is the boundary of Ω.

This equation often appears in the description of transport phenomena, especially in those described by the incompressible Navier-Stokes equations. The magnitudes of $p(x,y)$ and $q(x,y)$ determine the ratio of the convection to diffusion. In many problems of practical interest the convective terms dominate the diffusion. Many numerical simulations of (6.1) become increasingly difficult (converge slowly or even diverge) as the ratio of convection to diffusion increases.

For convenience, we define the cell Reynolds number as

$$\mathrm{Re}=\max(\sup_{(x,y)\in\Omega}|p(x,y)|,\ \sup_{(x,y)\in\Omega}|q(x,y)|)\ \ h/2,$$

where h is the uniform grid spacing. For $\mathrm{Re}\le 1$, we say that Eq. (6.1) is diffusion-dominated. Otherwise it is convection-dominated.

When Eq. (6.1) is discretized using central differences, the resulting scheme is a five-point formula ($\mathcal{FPF}$) with a truncation error of order h^2. In the case of the $\mathcal{FPF}$ scheme, iterative methods for solving the resulting system of linear equations do not converge when the convective terms dominate and Re is greater than a certain constant.

Some attempts have been made to solve the convection-diffusion equation with iterative methods based on $\mathcal{FPF}$. Recently, Brandt and Yavneh [9] used $\mathcal{FPF}$ with added dissipation terms to solve (6.1) with high Reynolds numbers. They proposed an over-weighted residual technique to accelerate the multigrid convergence for high-Reynolds number flows. de Zeeuw and van Asselt [74], de Zeeuw [75] developed a black-box multigrid solver with some matrix-dependent prolongations and restrictions. A multigrid method based on the Schur complement of the coefficient matrix and the matrix-dependent prolongation operator was recently proposed by Reusken [47, 48]. Elman and Golub [17] proposed methods that consist of applying one step of cyclic reduction resulting in a reduced system of half the order of the original discrete problem. Golub and Tuminaro [23] suggested a multigrid solution in conjunction with one step of cyclic reduction to solve the convection-diffusion equation.

Gupta et al. [25, 26] proposed a fourth-order compact nine-point finite difference scheme ($\mathcal{NPF}$) for Eq. (6.1) which was shown to be both accurate and cost-effective. It is also stable with the classical iterative methods (e.g., Gauss-Seidel, SOR) for large values of $p(x,y)$ and $q(x,y)$. The current work in this chapter is to merge the multigrid technique with the $\mathcal{NPF}$ scheme to develop a general convection-diffusion equation solver.

In Section 6.2, we present the nine-point compact finite difference discretization scheme for (6.1). A heuristic residual analysis is presented in Section 6.3 to obtain an optimal residual injection operator for diffusion-dominated problems. The $\mathcal{NPF}$ multigrid solver for the convection-diffusion equation is designed in Section 6.4. In Section 6.5, numerical experiments are employed to show the stability and the effectiveness of the $\mathcal{NPF}$ multigrid solver. Some conclusions and remarks are included in Section 6.6.

6.2 Finite Difference Scheme

The approximate value of a function $u(x,y)$ at a mesh point (x,y) is denoted by u_0. The approximate values at its eight immediate neighboring points are denoted by $u_i, i = 1, 2, \ldots, 8$, with the following computational stencil:

$$\begin{pmatrix} u_6 & u_2 & u_5 \\ u_3 & u_0 & u_1 \\ u_7 & u_4 & u_8 \end{pmatrix} \tag{6.2}$$

The discretized values of p_i, q_i and $f_i, i = 0, 1, \ldots, 4$, have their obvious meanings. The compact finite difference formula for the mesh point (x,y) involves the nearest eight neighboring mesh points with the mesh spacing h (see [26] for details):

$$\sum_{j=0}^{8} \alpha_j u_j = \frac{h^2}{2}[8f_0 + f_1 + f_2 + f_3 + f_4] + \frac{h^3}{4}[p_0(f_1 - f_3) + q_0(f_2 - f_4)]. \tag{6.3}$$

The coefficients $\alpha_i, i = 0, 1, \ldots, 8$, are

$$\alpha_1 = 4 + \frac{h}{4}[4p_0 + 3p_1 - p_3 + p_2 + p_4] + \frac{h^2}{8}[4p_0^2 + p_0(p_1 - p_3) + q_0(p_2 - p_4)],$$

$$
\begin{aligned}
\alpha_2 &= 4 + \frac{h}{4}[4q_0 + 3q_2 - q_4 + q_1 + q_3] + \frac{h^2}{8}[4q_0^2 + p_0(q_1 - q_3) + q_0(q_2 - q_4)], \\
\alpha_3 &= 4 - \frac{h}{4}[4p_0 - p_1 + 3p_3 + p_2 + p_4] + \frac{h^2}{8}[4p_0^2 - p_0(p_1 - p_3) - q_0(p_2 - p_4)], \\
\alpha_4 &= 4 - \frac{h}{4}[4q_0 - q_2 + 3q_4 + q_1 + q_3] + \frac{h^2}{8}[4q_0^2 - p_0(q_1 - q_3) - q_0(q_2 - q_4)], \\
\alpha_5 &= 1 + \frac{h}{2}(p_0 + q_0) + \frac{h}{8}(q_1 - q_3 + p_2 - p_4) + \frac{h^2}{4}p_0 q_0, \\
\alpha_6 &= 1 - \frac{h}{2}(p_0 - q_0) - \frac{h}{8}(q_1 - q_3 + p_2 - p_4) - \frac{h^2}{4}p_0 q_0, \\
\alpha_7 &= 1 - \frac{h}{2}(p_0 + q_0) + \frac{h}{8}(q_1 - q_3 + p_2 - p_4) + \frac{h^2}{4}p_0 q_0, \\
\alpha_8 &= 1 + \frac{h}{2}(p_0 - q_0) - \frac{h}{8}(q_1 - q_3 + p_2 - p_4) - \frac{h^2}{4}p_0 q_0, \\
\alpha_0 &= -[20 + h^2(p_0^2 + q_0^2) + h(p_1 - p_3) + h(q_2 - q_4)]. \qquad (6.4)
\end{aligned}
$$

The results of the numerical experiments in [26] show that this scheme converges for any values of $p(x, y)$ and $q(x, y)$ when classical iterative methods such as SOR are used. Some limited stability results for this scheme with constant coefficients are discussed in Appendix B and [83].

When Re $\equiv 0$, Eq. (6.1) reduces to the Poisson equation, and Eq. (6.3) reduces to the well-known Mehrstellen formula [27]. Multigrid applications of the Mehrstellen formula have been investigated by Schaffer [51, 52]. Recently, we [29] made some comparisons between the nine-point and five-point multigrid Poisson solvers on serial and vector machines and showed that $\mathcal{NPF}$ is more cost-effective than $\mathcal{FPF}$.

In [30], we investigated the multigrid solution of Eq. (6.1) when $p(x, y)$ and $q(x, y)$ are constants and observed the cost-effectiveness of employing a residual injection operator for convection-dominated problems. We found that the $\mathcal{NPF}$ multigrid converges for any value of Re, while the $\mathcal{FPF}$ multigrid becomes divergent when Re > 1.

6.2.1 Multigrid Implementation

The discretized grid space is usually naturally (lexicographically) ordered. We may rearrange the grids in an alternative red and black order in a checkerboard fashion. The relaxation (smoothing) can be carried out simultaneously on red points and black points (independently for $\mathcal{FPF}$, but not independently for $\mathcal{NPF}$). This idea is certainly beneficial on the parallel computers. But it has been shown that, even on serial computers, the red-black ordering is superior to the natural ordering for both $\mathcal{FPF}$ and $\mathcal{NPF}$ multigrid algorithms for solving the Poisson equation [29]. The Gauss-Seidel relaxation with the red-black ordering is referred to as the RBGS relaxation (smoothing).

The bi-linear interpolation will be used in all our algorithms to interpolate the coarse grid correction to the fine grid. The full-weighting restriction or the injection operators (defined below) will be used to transfer the residuals from the fine grid to the coarse grid.

In the context of the multigrid method, the right-hand side as it appears in (6.3) is only evaluated once on the finest grid when the initialization of data (boundary conditions) is performed. We may define F at each reference point by

$$F_0 = \frac{h^2}{2}[8f_0 + f_1 + f_2 + f_3 + f_4] + \frac{h^3}{4}[p_0(f_1 - f_3) + q_0(f_2 - f_4)].$$

Now (6.3) becomes

$$\sum_{i=0}^{8} \alpha_i u_i = F_0.$$

The computation of F_0 for grid points close to the boundary requires the values of $f(x, y)$ on the boundary. We assume that $f(x, y)$ is extended naturally on to $\partial\Omega$.

To utilize the computer's memory more efficiently, practical multigrid solvers usually use a single long array to store the discretized values of u and f (here F) for all grid levels. On the coarse grids, the locations of u and f are used to store coarse grid correction and residual respectively.

It is also economical to pre-compute the values of $p(x, y)$ and $q(x, y)$ on each grid points. We use another long array to store these values at each grid point and at each grid level. This is similar to the long array used above to store values of the approximate solution $u(x, y)$ and the right-hand side $f(x, y)$. The same pointers may be used for all of the arrays.

From now on, we refer to the algorithms using the multigrid W(1,1)-cycle (see Section 6.4) and the $\mathcal{NPF}$ smoother as $\mathcal{NPF}$-MG; those use the $\mathcal{FPF}$ smoothers are referred to as $\mathcal{FPF}$-MG.

6.3 Residual Transfer Analysis

The methods to carry out residual transfer (projection) typically fall into two categories. One is the direct injection of the fine grid residuals to the corresponding coarse grid points weighted by a constant factor. In practice, the factor 1/2 is used for $\mathcal{FPF}$-MG when RBGS is used as the smoother (see Appendix A). If the lexicographic Gauss-Seidel is used as the smoother, the factor 1 is used to inject the residuals and is called the full-injection operator.

For $\mathcal{NPF}$-MG, the grid space is not completely de-coupled by the RBGS smoothing and the half-injection is not accurate. On the other hand, the full-injection may result in divergence for many diffusion-dominated problems.

Another category of projection operators is to weight the residuals at all fine grid points and to project a weighted quantity to the coarse grid. The full-weighting scheme is to weight the residual at the nearest nine points by formula (A.1).

For the full-weighting, we evaluate the residuals at all fine grid points and weight them to the corresponding coarse grid points. On the other hand, the injection operator needs only the residuals corresponding to the coarse grid points. There is no cost for weighting the neighboring residuals. The cost of evaluating residuals on a grid space is equivalent to one full relaxation on that grid. Hence, computational cost of using the injection operator is less than a quarter of the cost of using the full-weighting operator. For $\mathcal{NPF}$-MG, the employment of the injection operator may result in 25% savings in CPU time if the convergence of $\mathcal{NPF}$-MG does not deteriorate.

Our numerical experiments with the constant coefficients in [30] show that the injection with a factor 1/2 or 1 is more cost-effective than the full-weighting for $\mathcal{NPF}$-MG when Re is greater than 1. More interesting is the fact that the convergence is not very sensitive to the injection factor. However, when Re < 1 (diffusion-dominated), the half-injection deteriorates the convergence considerably and the full-injection causes divergence for some problems.

6.3.1 Diffusion-Dominated Residual Injection

Our goal in this section is to find a residual injection scaling factor that will improve the convergence of the diffusion-dominated problems (Re ≤ 1). To this end, we investigate the full-weighting scheme (A.1).

The heuristic residual analysis is similar to that we use in Appendix A to derive an optimal residual injection scaling factor for the standard five-point Poisson solver.

In (A.1), the weight assigned to each point is determined by the involvement of that point in the number of coarse grid point computation. For example, $r_{i+1,j+1}$ is weighted into the calculation of four coarse grid points at $(i/2, j/2), (i/2, j/2+1), (i/2+1, j/2)$ and $(i/2+1, j/2+1)$. The weights in (A.1) correctly reflect these correlations. But they do not reflect the geometric correlation of $r_{i,j}$ and $r_{i+1,j+1}$, and the red-black relaxation pattern.

To find the optimal injection operator is to find the optimal scaling factor α to represent $\bar{r}_{i/2,j/2}$ in terms of $r_{i,j}$, as accurately as possible, i.e.

$$\bar{r}_{i/2,j/2} = \alpha r_{i,j}. \tag{6.5}$$

It is very difficult to give a precise representation of the residuals on each grid space. However, with some assumptions, a reasonable estimate may be obtained. We assume that the solution is not highly oscillating and the high frequency error components are removed before the residuals are injected to the coarse grid. (If the smoothing condition is not satisfied, several relaxation sweeps may be needed to smooth the high frequency error components before the residuals are projected to the coarse grid [84, 85].) Hence, the residual at each grid point is not supposed to differ by a large magnitude from those at its nearest eight neighboring points. We assume that the residuals are locally equal for each of the nine points involved in a particular $\mathcal{NPF}$ relaxation step.

A half Jacobi sweep is carried out on the red points first without updating the black points. Except for those points close to the boundaries, a red point is updated by two previously updated (new) red points, two un-updated (old) red points and four un-updated (old) black points (see Figure A.1). The update of a red point uses new values at only two points and the residual at that point is supposed to be large (relative to those at the neighboring black points). A subsequent half Jacobi sweep on the black points updates each black point by four new red points, two new black points and two old black points. Hence, the residual at each black point is presumably smaller than those at the nearest neighboring red points.

If we assume that, after one RBGS sweep on the entire grid, the residuals at all the black points are zero, i.e.

$$r_{i+1,j} = r_{i-1,j} = r_{i,j+1} = r_{i,j-1} = 0, \tag{6.6}$$

and residuals at all red points are equal (not zero, otherwise we would have reached convergence and have no need to transfer the residuals),

$$r_{i-1,j-1} = r_{i+1,j-1} = r_{i-1,j+1} = r_{i+1,j+1} = r_{i,j}. \tag{6.7}$$

Substituting (6.5), (6.6) and (6.7) into (A.1), we obtain $\alpha = 1/2$, which is the half-injection. Since this α results from the assumption that the residuals at the black points are zero, this should be the lower bound of α. We denote $\alpha_{lower} = 1/2$.

In practice, the residuals at the black points may not be zero. Since updating a black point uses three times as much new information as updating a red point, we assume that after one red-black full sweep on the entire grid, the residual $r_{i,j}$ at any particular red point (i,j) is three times as large as the residuals at its four immediate neighboring black points $(i,j-1),(i,j+1),(i-1,j),(i+1,j)$, i.e.

$$r_{i+1,j} = r_{i-1,j} = r_{i,j+1} = r_{i,j-1} = \frac{1}{3} r_{i,j}. \tag{6.8}$$

We also assume that the four neighboring red points $(i-1,j-1),(i+1,j-1),(i-1,j+1)$ and $(i+1,j+1)$ have the residuals of the same magnitude as $r_{i,j}$. But their geometric positions are $\sqrt{2}$ unit away from the reference point (i,j) and their influence on the weighting scheme should be scaled by $\sqrt{2}$, i.e.

$$r_{i-1,j-1} = r_{i+1,j-1} = r_{i-1,j+1} = r_{i+1,j+1} = \frac{1}{\sqrt{2}} r_{i,j} \tag{6.9}$$

Substituting (6.5), (6.8) and (6.9) into (A.1), we have

$$\alpha = \frac{10 + 3\sqrt{2}}{24} \approx 0.5934. \tag{6.10}$$

This would be the upper bound of the residual injection factor, we denote it by α_{upper}.

The optimal factor $\alpha_{optimal}$ lies between α_{upper} and α_{lower}. There exists $\xi \in (0,1)$ such that

$$\alpha_{optimal} = \xi \alpha_{upper} + (1-\xi)\alpha_{lower}. \tag{6.11}$$

In absence of further information to justify any better weighting scheme, we take $\xi = 1/2$, (6.11) yields

$$\alpha_{optimal} = \frac{\alpha_{upper} + \alpha_{lower}}{2} = \frac{22 + 3\sqrt{2}}{48} \approx 0.5467.$$

Our numerical experiments on several test problems show that this choice of α for the residual injection operator indeed gives better performance than both the full-weighting and the half-injection operators. However, a slight variation of $\alpha_{optimal}$ may give even better results for particular test problems.

6.3.2 Convection-Dominated Residual Injection

When the cell Reynolds number Re increases, the smooth components of the errors dominate [9]. This is characterized by the relation [10]

$$\| \tilde{r}^h \|_2 \ll \| e^h \|_2, \tag{6.12}$$

where $\tilde{r}^h$ and e^h are the normalized residuals and errors on the fine grid, respectively (see [10] for definition of the normalized residuals). This relation states that the norms of the smooth components of the errors are much larger than the norm of the normalized residuals. Increasing relaxation on the fine grid reduces the magnitude of the high frequency errors, but does not reduce magnitude of the low frequency errors. On the other hand, due to the inconsistency between the fine and coarse grid approximations, the residual equation on the coarse grid converges to a solution which is not the error correction required by the fine grid. When this wrong coarse grid correction is interpolated back to the fine grid and is added to the fine grid solution, the latter tends to diverge.

Therefore, the residuals on the fine grid must be scaled before they are projected to the coarse grid. To modify the full-weighting operator so that it may reflect the fine grid residuals more closely, Brandt and Yavneh [9] proposed to over-weight the residuals to accelerate the $\mathcal{FPF}$ based multigrid methods with added dissipation terms for recirculating flows with high Reynolds number. Their idea is to improve the coarse grid correction to the error in the fine grid approximation by multiplying the residuals that are transferred to the coarse grid by some constant η between one and two. For example, the two-level convergence factor of the advection-diffusion equation (with vanishing coefficients) employing the first-order discretization improves from 0.5 to 0.33 with $\eta = 4/3$.

The residual injection operator may reflect some behavior of the fine grid residuals. If the residuals need to be scaled before they are injected to the coarse grid, we can scale the residual injection factor so that we keep the cost of transferring residuals low by using injection instead of full-weighting.

Moreover, when $p(x, y)$ and $q(x, y)$ are oscillatory rapidly in Ω, the direction of the convection is complicated. In particular, when Ω contains stagnation points, where the convection coefficient vanishes and the velocity is zero, the convection changes direction around the stagnation points and equation (6.1) represents a recirculating flow. The full-weighting operator usually mis-represents the closed characteristics of the flow around the stagnation points. By projecting residuals with mis-represented characteristics to the coarse-grid, the coarse-grid sub-problem fails to approximate that of the fine-grid at all and causes divergence on the fine-grid for high Reynolds number recirculating flows. On the other hand, the injection may maintain the characteristics. These properties have been clearly simulated by our numerical examples presented in Section 6.5.

Of course, the best residual injection factor will generally change from that we obtained above for the diffusion-dominated problem. Since the smooth components of the errors increase as Re increases to infinity, the scaling factor α may be increased to reflect this fact. Although there is no absolute guarantee that any single factor will work for all practical problems, we find that, for most problems, the residual injection factor is indeed an increasing function of Re, and it approaches a constant when Re tends to infinity. This constant may be problem-dependent, but it is usually around 1, i.e., for high Reynolds number problems, the scaled residual injection for RBGS relaxation tends to the full injection.

6.4 Design of $\mathcal{NPF}$-MG Solver

We design our $\mathcal{NPF}$-MG solver as follows:

1. Start from the fine grid with some initial guess and perform ν_1 RBGS $\mathcal{NPF}$ relaxation sweeps.
2. Calculate the residuals corresponding to the coarse grid points, multiply the residuals by a scaling factor α and inject the residuals to the coarse grid.
3. Perform μ multigrid cycles on this grid.
4. Interpolate the coarse grid correction to the fine grid by bi-linear interpolation.
5. Perform ν_2 RBGS $\mathcal{NPF}$ relaxation sweeps on the fine grid.

If $\mu = 1$, the multigrid cycle is called the V-cycle. If $\mu = 2$, it is the W-cycle. For convection-dominated problems, it has been shown by Brandt and Yavneh [9] that the W-cycle algorithm is usually better than the V-cycle algorithm. ν_1 and ν_2 are the numbers of pre-smoothing and post-smoothing sweeps.

For all Re, choosing $\alpha = 0.5424$ with residual injection guarantees convergence. We note that the scaling parameter α can be fine tuned to accelerate the convergence. Nevertheless, in all our numerical experiments followed, we take $\alpha = 0.5424$ for all the Reynolds number Re and no divergence has been found.

The multigrid cycle usually goes down to the coarsest grid with only one unknown. On the coarse grids, the effective cell Reynolds number is large and the relaxation becomes more difficult. For high Reynolds number problems, it is sometimes advantageous to stop the multigrid process at some coarse grid level before reaching the coarsest grid.

6.5 Numerical Experiments

Numerical results for three test problems are obtained using the $\mathcal{NPF}$-MG solver.

The test problems given here are solved using a uniform mesh-size h on a square domain $\Omega = (-0.5, 0.5) \times (-0.5, 0.5)$.

The boundary values of the solution are assumed to be known. The number of the multigrid W(1,1) cycles (MW), the discrete error in L_∞ norm (Error), and the CPU time in seconds, are reported. All computations are done on an SGI (Silicon Graphic

Indy) workstation using FORTRAN 77 programming language in double precision. The computation is terminated when the initial residual (in L_2 norm) on the finest grid is reduced by a factor of 10^{10}.

For the $\mathcal{NPF}$-MG solver, standard coarsening technique (the mesh-size of the coarse grid doubles that of the fine grid) is used and the coarsest grid contains only one unknown (nine points in total including boundary points). The only exception is the coarse-grid restriction option, where we stop coarsening when $h = 1/16$ (in this case, there are 225 unknowns on the coarsest grid).

For the first test problem, $\mathcal{NPF}$-MG is first applied for different values of Re with fixed mesh-size $h = 1/128$ to test the convergence of $\mathcal{NPF}$-MG as a function of the Reynolds number. Then we vary the mesh-size h to test the 4th order accuracy promised by $\mathcal{NPF}$-MG. The second test problem is used to further test the Reynolds number effect. For the third test problem, Re is fixed and we vary the mesh-size to test the accuracy and the cost-effectiveness of the injection operator.

6.5.1 Test Problem 6.5.1

Consider the boundary value problem (6.1) with

Test Problem 6.5.1

$$\begin{cases} p(x,y) &= Px, \\ q(x,y) &= -Py, \\ u(x,y) &= xy(1-x)(1-y)\exp(x+y). \end{cases}$$

This problem was used by Gupta et al. [26] to test the high order formula (6.3) with classical iterative methods (SOR) using $h \leq 1/32$ and it was shown that $\mathcal{NPF}$ with SOR is convergent for any P, while the central difference scheme (CDS) is divergent with $P \geq 1000$. A similar problem was used by Shapira et al. [55] to test their automatic multigrid method (with a five-point stencil $\mathcal{FPF}$) for $h = 1/64$. When $P = 150$, both the standard multigrid method and the automatic multigrid method converge. However, when $P = 300$, both methods diverge with the RBGS smoothing. Note that the presence of a stagnation point at $(0, 0)$.

Tables 6.1 to 6.3 contain test results for fixed mesh-size $h = 1/128$. Computations are reported with full-weighting, residual-injection and coarse-grid restriction techniques. We note that $\mathcal{NPF}$-MG with the full-weighting residual projection operator provides accurate solutions for the values of P ranging from 0 to 1200. However the iterations diverge when Re is large. Many multigrid solvers based on $\mathcal{FPF}$ diverge even when Re is less than 1 on the finest grid (see an example in [55]) and the refinement of mesh-size on the finest grid does not bring convergence. This suggests that the reason for divergence is on the coarse grid. For example, when $P = 1500$, Re $= 2.9279$ on the finest grid with $h = 1/128$, Re $= 187.5$ on the coarsest grid with $h = 1/2$.

One remedy to this divergence is to restrict the mesh-size of the coarsest grid and to avoid the worst Reynolds number. We were able to obtain convergent solution for values of P ranging from 0 to 12000. However, this approach reduces the number of

P	Re	Full-Weighting	Residual-Injection	Grid-Restriction
0	0.0000	8	10	38
10	0.0195	8	10	36
50	0.0977	8	10	19
100	0.1953	9	10	12
500	0.9767	17	19	17
800	1.5625	24	27	24
1000	1.9531	29	31	29
1200	2.3438	32	35	32
1500	2.9297	diverge	38	35
2000	3.9062	diverge	46	43
5000	9.7656	diverge	72	67
8000	15.625	diverge	101	94
10000	19.531	diverge	136	127
12000	23.438	diverge	172	163
40000	78.125	diverge	713	diverge

Table 6.1: Test Problem 6.5.1 with $h = 1/128$: Comparison of iteration (the W(1,1)-cycle) number for $\mathcal{NPF}$-MG with options.

P	Re	Full-Weighting	Residual-Injection	Grid-Restriction
0	0.0000	7.45	6.85	32.03
10	0.0195	7.48	6.85	31.25
50	0.0977	7.48	7.01	16.02
100	0.1953	8.33	6.84	10.17
500	0.9767	15.80	12.90	14.35
800	1.5625	22.00	18.29	20.34
1000	1.9531	26.63	21.20	24.70
1200	2.3438	29.39	23.69	27.07
1500	2.9297	diverge	26.03	29.52
2000	3.9062	diverge	31.29	36.28
5000	9.7656	diverge	49.22	56.42
8000	15.625	diverge	68.87	79.78
10000	19.531	diverge	92.98	107.96
12000	23.438	diverge	117.45	138.11
40000	78.125	diverge	483.31	diverge

Table 6.2: Test Problem 6.5.1 with $h = 1/128$: Comparison of CPU time in seconds for $\mathcal{NPF}$-MG with options.

P	Re	Full-Weighting	Residual-Injection	Grid-Restriction
0	0.0000	2.4(-11)	2.2(-11)	1.8(-09)
10	0.0195	2.9(-10)	2.9(-10)	1.6(-09)
50	0.0977	4.9(-09)	4.9(-09)	5.0(-08)
100	0.1953	1.3(-08)	1.3(-08)	1.3(-08)
500	0.9767	8.7(-08)	8.7(-08)	8.7(-08)
800	1.5625	1.4(-07)	1.4(-07)	1.4(-07)
1000	1.9531	1.8(-07)	1.8(-07)	1.8(-07)
1200	2.3438	2.2(-07)	2.2(-07)	2.2(-07)
1500	2.9297	diverge	2.7(-07)	2.7(-07)
2000	3.9062	diverge	3.5(-07)	3.5(-07)
5000	9.7656	diverge	8.2(-07)	8.2(-07)
8000	15.625	diverge	1.2(-06)	1.2(-06)
10000	19.531	diverge	1.5(-06)	1.5(-06)
12000	23.438	diverge	1.7(-06)	1.7(-06)
40000	78.125	diverge	4.1(-06)	diverge

Table 6.3: Test Problem 6.5.1 with $h = 1/128$: Comparison of maximum errors for $\mathcal{NPF}$-MG with options.

multigrid levels and reduces the multigrid efficiency for the diffusion-dominated problems where P is small. Coarse-grid restriction computations take many more multigrid cycles than full-weighting computations when P is small. For very large P such as $P = 40000$, Re is large even on the finest grid and $\mathcal{NPF}$-MG diverges with this option. This indicates that more coarse grids may need to be removed to ensure convergence.

On the other hand, $\mathcal{NPF}$-MG with the residual-injection operator converged for all values of P ranging from 0 to 40000. It is also more cost-effective than both the full-weighting operator and the coarse-grid restriction option.

The residual-injection operator guarantees the convergence at a lower computational cost. In [30], we noticed that the injection operator (with a factor of 1/2 or 1) is more cost-effective than the full-weighting operator when the cell Reynolds number Re is greater than 1 though it is not as good as the full-weighting operator when Re ≤ 1. However, with the optimal residual injection factor developed here, we are able to gain cost-effectiveness even for the diffusion-dominated cases.

For the convection-dominated cases, the computed accuracy is the same for different residual transferring operators and coarse-grid restriction option.

Next, we test the convergence and the improvement in the computed accuracy as the mesh-size is refined. For $\mathcal{NPF}$-MG with full-weighting operator, the results are given in Tables 6.4 and 6.5. It can be observed that, in most cases, when the mesh-size is halved, the maximum error is decreased by a factor of 16. This is the performance of 4th order convergence. On the other hand, Table 6.4 shows that the convergence of

P	$h = 1/32$	$h = 1/64$	$h = 1/128$	$h = 1/256$	$h = 1/512$
0	8	8	8	8	8
1	8	8	8	8	8
10	8	8	8	8	8
50	10	9	8	8	8
100	13	11	9	8	8
500	24	24	17	12	10
1000	27	33	29	19	13

Table 6.4: Test Problem 6.5.1: Iteration (W(1,1)-cycle) numbers of $\mathcal{NPF}$-MG with full-weighting and different meshsizes.

P	$h = 1/32$	$h = 1/64$	$h = 1/128$	$h = 1/256$	$h = 1/512$
0	6.1(-9)	3.8(-10)	2.4(-11)	1.5(-12)	2.6(-13)
1	6.7(-9)	4.2(-10)	2.6(-11)	1.6(-12)	2.6(-13)
10	7.4(-8)	4.6(-9)	2.9(-10)	1.8(-11)	1.1(-12)
50	1.3(-6)	7.8(-8)	4.9(-9)	3.1(-10)	1.9(-11)
100	3.4(-6)	2.1(-7)	1.3(-8)	8.3(-10)	5.2(-11)
500	1.9(-5)	1.4(-6)	8.7(-8)	5.5(-9)	3.4(-10)
1000	3.3(-5)	2.6(-6)	1.8(-7)	1.1(-8)	7.1(-10)

Table 6.5: Test Problem 6.5.1: Maximum errors of $\mathcal{NPF}$-MG with full-weighting and different meshsizes.

$\mathcal{NPF}$-MG is h-independent for diffusion-dominated problems, but is affected by the meshsize when Re is large.

Tables 6.6 and 6.7 contain data for the test for convergence and computed accuracy when $\mathcal{NPF}$-MG is used with the residual-injection operator. The 4th order accuracy is still maintained. The interesting information given by Table 6.6 is that the convergence of $\mathcal{NPF}$-MG with residual injection is less affected by the magnitude of Re than that with full-weighting.

6.5.2 Test Problem 6.5.2

In Test Problem 6.5.1, the coefficients $p(x,y)$ and $q(x,y)$ were linear functions. Next, we consider the nonlinear coefficient problem

Test Problem 6.5.2

$$\begin{cases} p(x,y) &= P\exp(x+y), \\ q(x,y) &= -P\exp(-x-y), \\ u(x,y) &= xy(1-x)(1-y)\exp(x+y). \end{cases}$$

P	$h = 1/32$	$h = 1/64$	$h = 1/128$	$h = 1/256$	$h = 1/512$
0	10	10	10	10	10
1	10	10	10	10	10
10	10	10	10	10	10
50	11	10	10	10	10
100	13	11	10	10	10
500	25	25	19	14	11
1000	27	35	31	22	14
1500	40	43	46	37	23
2000	53	43	46	37	23
10000	139	202	136	118	148

Table 6.6: Test Problem 6.5.1: Iteration (W(1,1)-cycle) numbers of $\mathcal{NPF}$-MG with residual injection operator and different meshsizes.

P	$h = 1/32$	$h = 1/64$	$h = 1/128$	$h = 1/256$	$h = 1/512$
0	6.1(-9)	3.8(-10)	2.2(-11)	4.5(-12)	5.2(-12)
1	6.7(-9)	4.2(-10)	2.5(-11)	4.6(-12)	5.2(-12)
10	7.4(-8)	4.6(-9)	2.9(-10)	1.7(-11)	4.8(-12)
50	1.3(-6)	7.8(-8)	4.9(-9)	3.1(-10)	1.9(-11)
100	3.4(-5)	2.1(-7)	1.3(-8)	8.3(-10)	5.2(-11)
500	1.9(-5)	1.4(-6)	8.7(-8)	5.5(-9)	3.4(-10)
1000	3.3(-5)	2.6(-6)	1.8(-7)	1.1(-8)	7.1(-10)
1500	4.4(-5)	4.9(-6)	3.5(-7)	2.3(-8)	1.5(-9)
2000	5.2(-5)	4.9(-6)	3.5(-7)	2.3(-8)	1.5(-9)
10000	8.4(-5)	1.5(-5)	1.5(-5)	1.1(-7)	7.2(-9)

Table 6.7: Test Problem 6.5.1: Maximum errors of $\mathcal{NPF}$-MG with residual injection operator and different meshsizes.

P	Re	Full-Weighting	Residual-Injection	Grid-Restriction
0	0.0000	8	10	38
10	0.0531	8	10	48
50	0.2655	9	10	27
100	0.5309	11	12	16
500	2.6546	22	23	22
800	4.2473	diverge	28	28
1000	5.3091	diverge	32	32
1200	6.3710	diverge	34	34
1500	7.9637	diverge	38	38
2000	10.618	diverge	51	47
5000	26.546	diverge	100	102
8000	42.473	diverge	147	145
10000	53.091	diverge	192	187
12000	63.710	diverge	236	231
40000	212.37	diverge	710	diverge

Table 6.8: Test Problem 6.5.2 with $h = 1/128$: Iteration (W(1,1)-cycle) numbers of $\mathcal{NPF}$-MG with options.

P	Re	Full-Weighting	Residual-Injection	Grid-Restriction
0	0.0000	7.46	6.83	32.43
10	0.0531	7.45	6.89	40.34
50	0.2655	8.34	6.92	23.03
100	0.5309	10.54	8.18	14.05
500	2.6546	20.28	15.75	18.47
800	4.2473	diverge	19.03	23.86
1000	5.3091	diverge	21.61	27.44
1200	6.3710	diverge	23.28	28.74
1500	7.9637	diverge	26.09	32.37
2000	10.618	diverge	34.55	39.66
5000	26.546	diverge	67.83	87.58
8000	42.473	diverge	99.37	122.15
10000	53.091	diverge	137.02	156.81
12000	63.710	diverge	161.48	193.72
40000	212.37	diverge	486.04	diverge

Table 6.9: Test Problem 6.5.2 with $h = 1/128$: The CPU time in seconds for $\mathcal{NPF}$-MG with options.

P	Re	Full-Weighting	Residual-Injection	Grid-Restriction
0	0.0000	2.4(-11)	2.2(-11)	1.8(-09)
10	0.0531	1.0(-09)	1.0(-09)	1.6(-09)
50	0.2655	1.3(-08)	1.3(-08)	1.3(-08)
100	0.5309	3.3(-08)	3.3(-08)	3.3(-08)
500	2.6546	2.2(-07)	2.2(-07)	2.2(-07)
800	4.2473	diverge	3.6(-07)	3.6(-07)
1000	5.3091	diverge	4.5(-07)	4.5(-07)
1200	6.3710	diverge	5.4(-07)	5.4(-07)
1500	7.9637	diverge	6.7(-07)	6.7(-07)
2000	10.618	diverge	8.9(-07)	8.9(-07)
5000	26.546	diverge	2.0(-06)	2.0(-06)
8000	42.473	diverge	2.9(-06)	2.9(-06)
10000	53.091	diverge	3.4(-06)	3.4(-06)
12000	63.710	diverge	3.8(-06)	3.8(-06)
40000	212.37	diverge	7.1(-06)	diverge

Table 6.10: Test Problem 6.5.2 with $h = 1/128$: Maximum errors of $\mathcal{NPF}$-MG with options.

Information contained in Tables 6.8 to 6.10 supports the remarks made on the Test Problem 6.5.1. Once again, the convergence and the accuracy worsen with increasing Re. The $\mathcal{NPF}$-MG solver with the injection-operator is more cost-effective than that with both the full-weighting operator and the coarse-grid restriction option.

It may be noticed that the convergence of $\mathcal{NPF}$-MG with the injection-operator is quite slow for large values of P, such as $P = 40000$. This is because we use the residual injection factor $\alpha = 0.5424$ for all Re. Faster convergence may be obtained by using different α for different Re. Although the issue of optimal values of α is still not fully resolved, we will give some test results in Section 6.5.4 to show that much faster convergence is obtained with large α.

We observed in our numerical computations that the convergence of $\mathcal{NPF}$-MG for large Re is fairly rapid for the first few cycles. Since the computed accuracy deteriorates for large Re, there may be no need to insist on reducing the initial residual by 10^{10} to reach the final solution. In practical applications, we may be satisfied with lower accuracy for high Reynolds number problems. Here we emphasize that the stability of $\mathcal{NPF}$-MG guarantees convergence for all Re. Again, test results for very large Re are given in Section 6.5.4.

Again, we note that there is no difference in the accuracy of the computed solutions for different residual transferring operators and coarse-grid restriction option when the problem is convection-dominated.

	Full-Weighting			Residual-Injection		
h	MW	CPU	Error	MW	CPU	Error
1/4	8	9.305(-3)	9.030(-05)	8	8.584(-3)	9.030(-05)
1/8	8	2.333(-2)	5.734(-06)	9	1.987(-2)	5.737(-06)
1/16	9	1.013(-1)	3.601(-07)	9	7.465(-2)	3.601(-07)
1/32	9	4.448(-1)	2.260(-08)	10	3.577(-1)	2.260(-08)
1/64	10	2.191(+0)	1.413(-09)	10	1.593(+0)	1.413(-09)
1/128	10	9.215(+0)	8.831(-11)	10	6.846(+0)	8.892(-11)
1/256	10	3.982(+1)	5.506(-12)	10	2.969(+1)	1.102(-11)
1/512	10	1.769(+2)	3.596(-13)	10	1.325(+2)	1.345(-11)

Table 6.11: Test Problem 6.5.3, $\mathcal{NPF}$-MG with the full-weighting and the residual-injection operators are tested on different mesh-sizes. The W(1,1)-cycle number (MW), the CPU time in seconds and the computed accuracy (Error) are reported.

6.5.3 Test Problem 6.5.3

The previous two test problems were primarily designed to test the robustness of $\mathcal{NPF}$-MG with respect to the cell Reynolds numbers, especially for large Reynolds numbers. Test Problem 6.5.3 is designed to test the accuracy of the solution computed using $\mathcal{NPF}$-MG with the full-weighting and the residual injection operators. Since this problem has a Reynolds number Re < 1 on all grids, the coarse-grid restriction technique is not efficient as also demonstrated in Test Problems 6.5.1 and 6.5.2. In this case the convection-diffusion equation (6.1) is defined by

Test Problem 6.5.3

$$\begin{cases} p(x,y) &= \sin(2x), \\ q(x,y) &= -\cos(2y), \\ u(x,y) &= x^2 + y^2. \end{cases}$$

Test Problem 6.5.3 is computed for different values of the mesh-size h. The multigrid W(1,1)-cycle number, the CPU time in seconds and the computed accuracy are listed in Table 6.11.

The data in Table 6.11 indicate clearly that $\mathcal{NPF}$-MG is a fourth-order algorithm. $\mathcal{NPF}$-MG with the full-weighting residual projection operator maintains this rate of convergence for all mesh-sizes tested. $\mathcal{NPF}$-MG with the residual injection operator maintains this property except for the very fine mesh-size ($N \geq 256$). This does not severely restrict the applications of the residual injection operator since the high-order method is able to solve problems on relatively coarse discretizations, but provides much higher accuracy than the other methods.

Probably more important is the fact that $\mathcal{NPF}$-MG is more cost-effective with the residual injection operator than with the full-weighting operator for all mesh-sizes.

h	Test Problem 6.5.1		Test Problem 6.5.2	
	$\alpha = 0.5424$	$\alpha = 1$	$\alpha = 0.5424$	$\alpha = 1$
1/8	9	8	11	11
1/16	21	18	26	19
1/32	38	34	52	36
1/64	61	31	62	41
1/128	99	32	100	52
1/256	157	32	158	66
1/512	240	32	243	82

Table 6.12: Number of $\mathcal{NPF}$-MG W(1,1)-cycles for Test Problems 6.5.1 and 6.5.2 with two scaling factors $\alpha = 0.5424$ and $\alpha = 1$. $P = 10^{10}$ and computations were terminated after residual norm is reduced by 10^5.

The residual injection operator reduces the residual transferring cost significantly, but no serious deterioration in convergence is observed.

6.5.4 Tests for Very Large Reynolds Numbers

The convergence of test Problems 6.5.1 and 6.5.2 may seem pessimistic for large values of P, say $P = 40000$. This is due to the fact that we use the same scaling factor α for the injection operator in all test conditions. As we discussed in Section 6.3.2, for convection-dominated problems, it may be advantageous to use a larger scaling factor. Moreover, there is no reason why we should not terminate the computations before the residual norm is reduced by a factor of 10^{10}, since we know that the achievable accuracy is affected inversely by the magnitude of P.

We experimented Test Problems 6.5.1 and 6.5.2 again for very large values of P and terminated the computations when the residual norm was reduced by a factor of 10^5. For each problem, we test two different scaling factors $\alpha = 0.5424$ and $\alpha = 1$ to assess the effect of the injection scaling factor on the convergence. For a particular problem and with a particular mesh-size h, we find that the convergence is no longer affected by the magnitude of P when $P \geq 10^5$. Table 6.12 contains the convergence histories of Test Problems 6.5.1 and 6.5.2 with $\alpha = 0.5424$ or $\alpha = 1$ and $P = 10^{10}$.

Table 6.12 shows that the magnitude of the scaling factor α makes a substantial impact on the rate of convergence of $\mathcal{NPF}$-MG. A large α accelerates the convergence of the high-Reynolds number problems (but causes divergence for diffusion-dominated problems). The rate of acceleration is more attractive when h is smaller.

Table 6.12 clearly indicates a very satisfactory convergence rate for large Reynolds number problems when we choose the scaling factor $\alpha = 1$. Although the data are presented for $P = 10^{10}$, similar results would be obtained for all $P \geq 10^5$ (see [82]). Note that for Test Problem 6.5.1, we have h-independent convergence. Since the convergence rate of $\mathcal{NPF}$-MG with residual injection is Re-independent for large Re, we have a very favorable convergence property of both h- and Re-independence. The results for Test Problem 6.5.1 are slightly influenced by h, but we are sure that this less than perfect results could be fixed by using slightly different residual scaling factor α.

6.6 Conclusions and Remarks

A nine-point compact discretization formula is used in conjunction with the multigrid technique to develop a high-order multigrid solver ($\mathcal{NPF}$-MG) to solve the general convection-diffusion equation with variable coefficients. A residual injection operator with a suitably chosen scaling factor is introduced to accelerate the convergence and to increase the cost-effectiveness of our $\mathcal{NPF}$-MG solver. Several test problems have been solved to demonstrate the efficiency and computed accuracy of our solver. From the numerical experiments, it is clear that $\mathcal{NPF}$-MG gives good results. The implementation of $\mathcal{NPF}$-MG is simple since it employs same $\mathcal{NPF}$ discretization scheme for all grids. The beauty of $\mathcal{NPF}$-MG is that it requires neither preconditioner nor added dissipation terms for high-Reynolds problems. The computed accuracy of $\mathcal{NPF}$-MG is usually much better than $\mathcal{FPF}$-MG (see [30]).

We have found that the full-weighting operator may cause divergence for some high-Reynolds number problems and the residual-injection operator may be used to regain the convergence. We have demonstrated that a larger scaling factor may be beneficial for the convergence of large Reynolds number problems.

The convergence of the convection-dominated problems may further be improved by using more powerful relaxation schemes such as the alternating line Gauss-Seidel or by introducing the minimal residual smoothing acceleration techniques, see Chapter 4 and [80, 81].

Chapter 7

High Accuracy Solution of the Navier-Stokes Equations

7.1 Introduction

The Navier-Stokes equations that represent the conservation of mass, momentum, and energy are used to model fluid dynamics phenomena describing two- and three-dimensional flows of an incompressible viscous fluid. These equations are highly nonlinear and are very difficult to solve, especially when the approximate solutions are required to have a high accuracy. Sometimes the nonlinear Navier-Stokes equations are linearized in different forms. One linearization approach is to consider a Navier-Stokes equation in a stream-function and vorticity formulation, which results in a Poisson equation coupled with a convection-diffusion equation. Hence, a problem closely related to the numerical solution of the Navier-Stokes equations is that of obtaining highly accurate solution of the convection-diffusion equation, especially when convection is the dominating phenomena.

The general convection-diffusion equation satisfying Dirichlet boundary conditions is of the form (6.1) and some high accuracy multigrid solution has been obtained in Chapter 6.

Suppose that Eq. (6.1) is discretized by some finite difference scheme and results in a linear system of the form (1.1). The linear system (1.1) is usually of very large dimension. For such large systems, direct methods usually can not handle and iterative methods become attractive for their low storage requirements as long as convergence is guaranteed. The performance of classical iterative methods such as Jacobi and SOR is sensitive to the number of equations to be solved, the type of boundary conditions and other factors. Furthermore, the matrix A^h in (1.1) is nonsymmetric and not positive definite if the magnitudes of the convection coefficients are large, and this property adds further difficulty for the classic iterative methods.

Since classical iterative methods for solving the system of linear equations resulting from the central difference (CDS) do not converge when the convective terms dominate and the cell Reynolds number (Re) is greater than a certain constant, the upwind difference approximation has been used for many years despite it is only first-order accurate. Hence, in this research area, the so-called high-order methods were usually of second order accuracy. To distinguish our following scheme from the traditional high-order methods, we refer to our scheme explicitly as the fourth-order method.

Recently, there has been some interest in developing fourth-order compact schemes for solving Eq. (6.1) and the incompressible Navier-Stokes equations with large Reynolds numbers, see [16, 26, 28, 38, 41]. These schemes are somewhat similar and the numerical results reported by these investigators have no substantial difference. There are at least three advantages shared by these schemes:

1.) Unconditional stability: Although the coefficient matrices are no longer diagonally dominant for large Reynolds numbers, the schemes have been shown numerically stable for any Reynolds numbers [31];

2.) High accuracy: It has been shown that these schemes do produce numerical solution of fourth-order accuracy;

3.) Easy boundary treatment: Since the computational stencil involves only the nearest nine grid points, the schemes are of compact type and no special formula is needed for computing grid points near the boundaries.

However, until recently, the computational advantages of these fourth-order compact schemes have not been fully investigated. For example, it is not known if these schemes can be used to solve the incompressible Navier-Stokes equations of very large Reynolds numbers because of the limitations of the available computer power and the difficulty with the traditional SOR-type iterative methods. Only in a recent paper [41], one of these schemes were able to solve the lid-driven cavity problem with Re $= 7500$ and it was claimed in [41] that their scheme does not converge for Re ≥ 9000 with SOR iteration.

To fully investigate the properties of the the compact schemes, non-traditional iterative methods are necessary for large Re. One promising technique is the multigrid method which has been successfully used with the first and second discretization schemes for solving problems in the computational fluid dynamics (including the driven cavity problem) (see, e.g., [8, 9, 21, 60, 70]). A preliminary investigation on combination of the fourth-order compact scheme of the type which we will discuss in this chapter with the multigrid techniques was made by Altas and Burrage recently [2], but their multigrid method was only shown to converge for small Re (≤ 100) where the behavior of the problems is relatively nice.

Our aim in this chapter is to investigate the possibility of combining multigrid technique with the fourth-order compact schemes to solve the steady-state incompressible Navier-Stokes equations for large Re. This follows the work of Chapter 6, where we used the fourth-order compact scheme to develop an efficient multigrid method for the convection-diffusion equations with variable coefficients and the resulting multigrid solver was shown to yield high accuracy solution. and convergent for very large values of $p(x, y)$ and $q(x, y)$.

In this chapter, we present the discretization schemes for the stream-function and vorticity formulation of the incompressible Navier-Stokes equations in Section 7.2. In

Section 7.3 we discuss issues related to the multigrid algorithms. The $\mathcal{NPF}$ multigrid solvers for the convection-diffusion (and the Poisson) equations and for the incompressible Navier-Stokes equations are formally designed in Section 7.4. In Section 7.5, we solve the driven cavity model problem and compare our numerical results with those obtained by other investigators and by other methods. Concluding remarks are given in Section 7.6.

7.2 Fourth-Order Finite Difference Schemes

The nine-point fourth-order compact discretization scheme for the convection-diffusion equations (6.1) is given in Chapter 6 (6.3). Similar fourth-order compact schemes were reported in [16, 38, 41]. There have been no convincing evidence that any of these schemes is better than others.

When Re $\equiv 0$, Eq. (6.1) reduces to the Poisson equation, and Eq. (6.3) reduces to the well-known (simpler) Mehrstellen formula [27]:

$$4[U_1+U_2+U_3+U_4]+U_5+U_6+U_7+U_8-20U_0=\frac{1}{2}h^2[8f_0+f_1+f_2+f_3+f_4]. \quad (7.1)$$

Multigrid applications of the Mehrstellen formula have been investigated by Schaffer [52], Gupta, Kouatchou and Zhang [29].

For small Reynolds numbers (Re $= 100$), Altas and Burrage [2] used (6.3) as the defect-correction procedure in a multigrid approach to solve the steady incompressible Navier-Stokes equations. In their approach, Eq. (6.3) was only used to evaluate the residuals on the finest grid. Their numerical results showed that the target accuracy of the Poisson equation is of fourth-order, but it was not clear e if the computed accuracy of the convection-diffusion equation (Re $\neq 0$) (and the Navier-Stokes equation) is of fourth-order with the defect correction techniques.

The Navier-Stokes equations representing the two-dimensional steady flow of an incompressible viscous fluid are given in stream-function and vorticity formulation as follows [21, 28, 68]:

$$\frac{\partial^2\Psi(x,y)}{\partial x^2}+\frac{\partial^2\Psi(x,y)}{\partial y^2}=-\Phi(x,y), \quad (7.2)$$

$$\frac{\partial^2\Phi(x,y)}{\partial x^2}+\frac{\partial^2\Phi(x,y)}{\partial y^2}-\mathrm{Re}\left[u(x,y)\frac{\partial\Phi(x,y)}{\partial x}+v(x,y)\frac{\partial\Phi(x,y)}{\partial y}\right]=0, \quad (7.3)$$

$$u(x,y)=\frac{\partial\Psi(x,y)}{\partial y}, \qquad v(x,y)=-\frac{\partial\Psi(x,y)}{\partial x}. \quad (7.4)$$

Here Ψ is the stream-function, Φ the vorticity; u and v are the velocities in y and x directions respectively; Re is the non-dimensional Reynolds number.

The stream-function (7.2) is a Poisson equation and the fourth-order approximation is given by the Mehrstellen formula (7.1) and by putting $U=\Psi$ and $f=-\Phi$. The vorticity equation (7.3) is a special case of the convection-diffusion equation (6.1) and the fourth-order approximation in this case may be obtained by putting $u=\Phi$, $f=0$ and $p(x,y)=-\mathrm{Re}u(x,y)$, $q(x,y)=-\mathrm{Re}v(x,y)$ in Eq. (6.3).

The velocities u, v at a grid point (x, y) are calculated from the discrete approximation of Eq. (7.4). It has been shown [28] that it is beneficial for both convergence and accuracy to use the fourth-order approximations for the velocities. In particular, Gupta [27] derived some high accuracy compact approximations for the gradients of the solution of the Poisson equations. As the stream-function equation (7.2) is a Poisson equation in Ψ, high accuracy approximations for the gradient Ψ_x and Ψ_y can be obtained from [27], and the corresponding fourth-order compact approximations for the velocities are given as (also see [28]):

$$\left.\begin{array}{l} u_0 = (\Psi_2 - \Psi_4)/3h + (\Psi_5 - \Psi_6 - \Psi_7 - \Psi_8)/12h + h(\Phi_2 - \Phi_4)/12, \\ v_0 = (\Psi_3 - \Psi_1)/3h - (\Psi_5 - \Psi_6 - \Psi_7 + \Psi_8)/12h + h(\Phi_3 - \Phi_1)/12. \end{array}\right\} \tag{7.5}$$

7.3 Multigrid Method

The multigrid method with the fourth-order nine-point compact schemes for the convection-diffusion equation have been discussed in Chapter 6 and in [29]. The smoother which we will use for solving the stream-function (Poisson) equation (7.2) is the red-black Gauss-Seidel relaxation method without an acceleration parameter. For solving the vorticity (convection-diffusion) equation (7.3) we use the red-black Gauss-Seidel relaxation method with an acceleration parameter ω_1 (which is the SOR method). In both cases, the smoothers will be referred to as the RBGS smoother.

The bi-linear interpolation will be used in all our algorithms to interpolate the coarse grid correction to the fine grid. Specifically, The values at the common mesh points will be directly transferred, while the values at the new grid points will be obtained by averaging either two or four nearest mesh points. The residual restriction operator will be a scaled injection operator. The residuals on the fine grid points which are common to the coarse grids are calculated and multiplied by a scaling factor α. The properly scaled residuals are then injected to the coarse grid to form the coarse grid subproblem. The scaling factor for the Poisson equation and for the convection-diffusion equation with small Re (say, the cell Reynolds number is smaller than 2) is chosen to be 0.5424, that for the convection-diffusion equation with large Re is chosen to be 1. The advantages of using residual injection and the reason of choosing these residual scaling factors are discussed in Chapter 6. For solving the Poisson equation and the convection-diffusion equation when the diffusion terms dominate, although the current combination of the restriction and interpolation operators does not satisfy the rule given by Brandt and Hackbusch [69] governing the orders of the grid transfer operators and the order of the differential equation, we have shown in Chapter 6 (and [31]) that the resulting multigrid solvers are more cost-effective than the standard multigrid method using the full-weighting scheme as the residual transfer operators. For the convection-diffusion equations with stagnation point and large Reynolds numbers we have shown in Chapter 6 (and [31]) that the injection operator is necessary for convergence. In this case, the orders of the transfer operators satisfy the Brandt and Hackbusch rule because the convection-diffusion equation approximates a first-order differential equation when the convection is very strong.

For solving nonlinear equations, it is advisable to use full approximation scheme (FAS). However, since we linearized our Navier-Stokes equation by using the stream-function equation and the vorticity equation, we use the linear multigrid method (the correction cycle as described in Chapter 6) to solve the two linear equations. Since FAS and the correction cycle are mathematically equivalent for solving linear equations and FAS is computationally more expensive, our implementation poses no difficulty and is more cost-effective.

In the context of linear multigrid the residual equations are solved on the coarse grids, the right-hand side as it appears in Eq. (7.1) is only evaluated once on the finest grid when the initialization of data (boundary conditions) is performed. With f being replaced by Φ we may define F_0 by

$$F_0 = -\frac{h^2}{2}[8\Phi_0 + \Phi_1 + \Phi_2 + \Phi_3 + \Phi_4]. \tag{7.6}$$

Now Eq. (7.1) (with U being replaced by Ψ) becomes

$$4[\Psi_1 + \Psi_2 + \Psi_3 + \Psi_4] + \Psi_5 + \Psi_6 + \Psi_7 + \Psi_8 - 20\Psi_0 = F_0. \tag{7.7}$$

There is an option of pre-computing all values of the coefficient matrix A^h for the vorticity equation (7.3), but this requires four and a half times more storage space than usually required for storing the coefficients of the nine-point multigrid solver. There is a trade-off between the storage and the computational efficiency. If the problem can be solved in a few multigrid cycles to the required accuracy, as it is the case in our current application, computing the coefficient matrix A^h in the iteration process may be more cost-effective.

We close this section with reference to some existing implementations of multigrid method to obtain accurate solution of Eq. (6.1) (or equivalently the system (1.1)) and the Navier-Stokes equation (7.2) – (7.4). Since the central difference scheme results in a matrix A^h which is not diagonally dominant for large Re, classical iterative methods such as the damped Jacobi and SOR methods diverge when they are employed to solve the resulting linear system (1.1). Although the traditional upwind discretization is convergent for any Re, it is only of first-order accuracy. (Higher order upwind schemes are usually complicated and not easy to implement.) Hence, efforts have been made by many investigators to combine these two schemes to guarantee convergence and accuracy at the same time. In the context of multigrid method, the popular trends seem to use the defect-correction techniques of various kinds on the finest grid. The main idea behind the defect-correction techniques is to use the upwind scheme for relaxation (stability) and the central difference scheme for residual evaluation (accuracy) [3, 33]. It was demonstrated by those and other investigators that if the basic discretization is of first-order and the target discretization is of second-order, then the resulting solution is of second-order.

Several defect-correction techniques have been developed and used with some success by many authors to obtain stable second-order accuracy solutions of the convection-diffusion equations and of the Navier-Stokes equations [8, 9, 70]. However, since most reported methods were published with numerical results on convergence rate only, it is not clear if all these methods achieved the second-order accuracy in practice. On the other hand, we have demonstrated in Chapter 6 that $\mathcal{NPF}$ with multigrid techniques does produce solution of fourth-order accuracy.

Since Gupta et al. [26] have shown that $\mathcal{NPF}$ is stable for all Re and is of fourth-order accuracy, it is not necessary to use any defect-correction technique for the sake of combining stability and accuracy. We also showed in [29] that, to compute the solution of the Poisson equation to a given accuracy, the $\mathcal{NPF}$ multigrid is much more efficient than the $\mathcal{FPF}$ multigrid. Hence, in our implementation, we use $\mathcal{NPF}$ for both relaxation and residual evaluation, this will guarantee (theoretically) that our solution is of fourth-order accuracy.

Hereinafter we refer to the algorithms using the multigrid cycling techniques and the $\mathcal{NPF}$ smoother as $\mathcal{NPF}$-MG; those use $\mathcal{FPF}$ (CDS) smoothers are referred to as $\mathcal{FPF}$-MG.

7.4 Design of $\mathcal{NPF}$-MG Solver

A pseudo code of the $\mathcal{NPF}$-MG μ-cycle algorithm is as follows.

Algorithm 7.4.1 *$\mathcal{NPF}$-MG μ-cycle algorithm.*

```
    $u^h \leftarrow \mathcal{NPF}\text{-}MG(u^h, f^h)$

Given any initial guess $u^h_0$.
For $k = 0, 1, 2, \ldots,$ do:
    If $\Omega^h$ = the coarsest grid, then
        Solve      $u^h_k = (A^h)^{-1} f^h$.
    Else
        Relax $\nu_1$ times on $A^h u^h_k = f^h$ with the given initial guess $u^h_k$.
        Compute    $r^h_k = f^h - A^h u^h_k$ corresponding to the coarse grid.
        Set        $f^{2h} = \alpha r^{2h}_k$.
        Set        $u^{2h}_k = 0$.
        Do         $u^{2h}_k \leftarrow \mathcal{NPF}\text{-}MG(u^{2h}_k, f^{2h})$ $\mu$ times.
        Correct    $u^h_{k+1} = u^h_k + P u^{2h}_k$.
        Relax $\nu_2$ times on $A^h u^h_{k+1} = f^h$ with the initial guess $u^h_{k+1}$.
    End if.
```

For the stream-function equation (7.2), a V(1,1)-cycle algorithm is sufficient to obtain accurate solution with acceptable convergence and is more cost-effective than a W(1,1)-cycle algorithm. As stated above the smoother for the stream-function is the RBGS without a parameter. For the vorticity equation (7.3), we use a W(1,1)-cycle algorithm. The smoother is the RBGS with a damping parameter $\omega_1 \in (0, 1)$.

The Navier-Stokes equation (7.2) – (7.4) may be solved by nested inner-outer iteration procedure (see [28]) with different multigrid cycling algorithms being applied to the stream-function equation (7.2) and the vorticity equation (7.3). While it is somewhat advantageous to solve Eqs. (7.2) – (7.4) simultaneously to maintain the physical coupling between them [21, 68], the relaxation methods (such as the coupled strongly implicitly (CSI) procedure [21] and incomplete LU decomposition [68]) used to accomplish this coupling are usually very expensive comparing with the point Gauss-Seidel relaxation. We are not sure which of these relaxation schemes is the best for the fourth-order compact scheme with the multigrid and decide to leave this problem as future research direction. Our primary concern of this chapter is to show that the fourth-order compact scheme with multigrid can accelerate the usual SOR iteration significantly and to show the computed results for large Reynolds numbers previously claimed impossible for SOR method.

7.5 Application to Model Problem

The steady flow of an incompressible viscous fluid in a square cavity $\Omega = [0,1] \times [0,1]$ has been used for a long time as the model problem by many investigators to test their new numerical schemes and solution methods [2, 21, 28, 60, 68], although there are singularities at two of its corners. Highly accurate benchmark solutions are available in the literature. In particular, Ghia et al. [21] used the multigrid technique and grid points of 257×257 to compute numerical solutions for $100 \leq \mathrm{Re} \leq 10000$. Their solutions have been considered to be accurate because of the small grid spacing employed.

The flow is induced by the sliding motion of the top wall ($y = 1$) from right to left and is described by the Navier-Stokes equation (7.2)–(7.4). The boundary conditions are those of no slip: on the stationary walls $u = 0$ and $v = 0$; on the sliding wall $u = -1$ and $v = 0$ (see Figure 7.1).

In order to solve the driven cavity problem, we replace the Navier-Stokes equation (7.2)–(7.3) by the finite difference approximations given in Eqs. (6.3) and (7.1) respectively. The velocities, defined in Eq. (7.4), are calculated by using the fourth-order approximations (7.5). The unit square is covered by a grid of uniform mesh-size h $(= 1/(N-1))$. The discrete approximations (6.3), (7.1) are written at each of the $(N-2)^2$ interior grid points. Zero values are prescribed for Ψ on the boundary. The usual approximations for vorticity Φ on the boundary are the Jensen formulas (see [28, 49]), which have a local truncation error of second order. Fourth-order approximations could also be defined for obtaining boundary values of Φ. In particular, some fourth-order approximations analogous to the Jensen formulas were obtained by Altas and Burrage [2]:

$$\left.\begin{array}{rcl} x=0: \quad \Phi_3 &=& (-6h^2\Phi_0 + h^2\Phi_1 - 2\Psi_2 - 2\Psi_4 - 20\Psi_0)/7h^2; \\ x=1: \quad \Phi_1 &=& (-6h^2\Phi_0 + h^2\Phi_3 - 2\Psi_2 - 2\Psi_4 - 20\Psi_0)/7h^2; \\ y=0: \quad \Phi_4 &=& (-6h^2\Phi_0 + h^2\Phi_2 - 2\Psi_1 - 2\Psi_3 - 20\Psi_0)/7h^2; \\ y=1: \quad \Phi_2 &=& (24h - 6h^2\Phi_0 + h^2\Phi_4 - 2\Psi_1 - 2\Psi_3 - 20\Psi_0)/7h^2. \end{array}\right\} \tag{7.8}$$

An inner-outer iteration procedure is employed to obtain numerical solutions (see [28]).

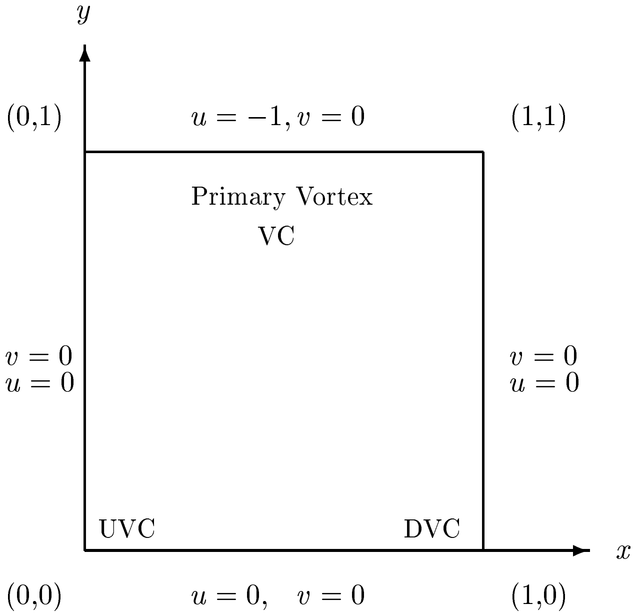

Figure 7.1: Driven cavity problem.

At each outer iteration, the linear system from the discrete stream-function equation (7.2) is solved by a multigrid V(1,1)-cycle algorithm with the RBGS smoother (using (7.7) without a damping factor). We then compute the velocities by using the fourth-order formulas (7.5); and evaluate the boundary conditions by using the fourth-order formulas (7.8). After that, we solve the discrete vorticity equation (7.3) using a multigrid W-cycle algorithm using RBGS smoother (6.3) with a relaxation parameter $\omega_{\mathrm{i}} \in (0, 1)$. The outer iteration process for the stream-function equation and the vorticity equation are also damped after each iteration using different damping factors $\omega_{\mathrm{S}} \in (0, 2)$ and $\omega_{\mathrm{V}} \in (0, 2)$ to give the new iterates.

At each inner iteration of the stream-function equation, one or two multigrid V(1,1)-cycles are applied; at each inner iteration of the vorticity equation, one multigrid W(1,1)-cycle is applied. Since the stream-function equation converges very fast and the nine-point Mehrstellen formula (7.7) is cheaper than the nine-point formula for the convection-diffusion equation (6.3), the major cost of each iteration step is in solving the vorticity equation. We have found that, at each outer iteration step, there is no need to solve each inner iteration to a higher accuracy. One or two multigrid cycles are enough and cost-effective.

The inner-outer iteration process for our multigrid solver may be described as following:

- Set initial guess as 0 for all values except the boundary values of the known velocities.

- For $k = 0, 1, 2, \cdots$, do

- Step 1: Compute the right-hand side of the stream-function using (7.6);
- Step 2: Solve approximately the stream-function equation (7.2) by performing one or two $\mathcal{NPF}$-MG V(1,1)-cycles using (7.7);
- Step 3: Compute the difference between the current and the previous values of the stream-function;
- Step 4: Damp the values of stream-function using a damping parameter $\omega_{\mathrm{S}} \in (0,2)$;
- Step 6: Compute the vorticity boundary values by using the fourth-order boundary approximations (7.8);
- Step 7: Solve approximately the vorticity equation (7.3) by performing one $\mathcal{NPF}$-MG W(1,1)-cycle and by using a relaxation parameter $\omega_{\mathrm{i}} \in (0,1)$ and using the fourth-order approximations (6.3);
- Step 8: Compute the difference between the current and the previous approximate values of the vorticity;
- Step 9: Damp the approximate values of the vorticity by using a parameter $\omega_{\mathrm{V}} \in (0,2)$;
- Step 10: Check the convergence, if both differences of the current and previous approximate values of the stream-function and vorticity computed from Steps 3 and 8 are less than a prescribed tolerance, then stop; otherwise go to Step 1 and begin next outer iteration.

Unlike the multigrid algorithms designed by other investigators [2, 21] who had to restrict the meshsize of the coarsest grid in order to insure convergence, our multigrid cycles are complete, i.e., the coarsest grid contains only one unknown. We solved the driven cavity problem (7.2) to (7.4) for $100 \leq \mathrm{Re} \leq 10000$. For each Re, we give the values and location coordinates of the maximum of the stream-function (strength of the main vortex) and the corresponding vorticity values. The problem with the same Re was solved several times using different discretization mesh-sizes on the finest grid to investigate what is the coarsest mesh-size producing acceptable solution for a given Re. The iterations were terminated when the maximum difference between successive approximations of both Ψ and Φ were smaller than 10^{-6} for $\mathrm{Re} \leq 5000$ and 10^{-4} for $7500 < \mathrm{Re} \leq 10,000$. Since each subsequent inner iteration uses the solution of the previous iteration as the starting values, we did not use nested multigrid iterations (the full multigrid method) for each inner iteration. The computations were carried out on an SGI (Silicon Graphics Indy) workstation using the Fortran 77 programming language in double precision.

N	$\Psi_{\max}$	Φ	Location
17	0.10290118	3.30727162	(0.375000, 0.750000)
33	0.10335397	3.28239537	(0.375000, 0.750000)
65	0.10350108	3.10864335	(0.390625, 0.734375)
129	0.10351141	3.16874534	(0.382813, 0.734375)

Table 7.1: Values and locations of the primary vortex for Re = 100 using different discretizations.

7.5.1 Comparison with Benchmark Solution

Unless otherwise indicated explicitly, we compare our results with those obtained by Ghia et al. [21] as the benchmark solutions. Our problem was set up slightly different from that of Ghia et al. [21] and the u velocity at the top wall ($y = 1$) is different due to the fact that the flow in our problem is induced by the sliding motion of the top wall from right to left (it was induced from left to right in the model problem solved by Ghia et al. in [21]). The computed values at a grid point (x, y) listed in our tables should be compared with those at the point $(1 - x, y)$ in tables of [21].

We first solve the model problem for Re = 100 with $N = 17, 33, 65, 129$ and compare our results with those of Ghia et al. with $N = 129$. The numerical values are given in Tables 7.1 and 7.2. It can be seen that, if a 5% departure from the benchmark solution is acceptable as the engineering accuracy, our method with $N = 17$ can produce acceptable results, while Ghia et al. used much finer discretization. Note that when the mesh is refined, the accuracy of our solution is increased rapidly.

When Re increases to 1000, The results in Tables 7.3 and 7.4 show that finer mesh is needed to produce accurate solution. However, we can see that with only a quarter of the number of equations used by Ghia et al., our method produced high accuracy solutions. With the same mesh-size, the solution given by our method, which we believe, is actually more accurate than that of Ghia et al.

When Re $\geq$ 2000, Gupta reported slow convergence when the compact scheme was used with the SOR iteration [28]. It is shown in Tables 7.5 and 7.6 that our multigrid accelerated SOR method can compute very accurate solution for Re = 3200 with $N = 129$.

With Re = 5000, Ghia et al. used $N = 257$ to compute accurate solution, we found (see Tables 7.7 and 7.8) that our fourth-order multigrid method can compute solution to comparable accuracy using $N = 129$.

For Re = 7500, with $N = 129$, Tables 7.9 and 7.10 show our method was still able to yield solution accurate enough (the error is less than 5%) to compare with benchmark solution of Ghia et al. [21] using $N = 257$.

For Re = 10000, no numerical result has been reported with the fourth-order compact scheme. (The largest Re reported was 7500 in [41].) We have been able to have our $\mathcal{NPF}$-MG converge with Re = 10000. Tables 7.11 and 7.12 show that the

N	Error in $\Psi_{\max}$	Error in Φ
17	0.59%	4.37%
33	0.15%	3.59%
65	0.01%	1.90%
129	0.00%	0.00%

Table 7.2: Errors in values of the primary vortex for Re = 100 using different discretizations, comparing with Ghia et al.'s solution.

N	$\Psi_{\max}$	Φ	Location
17	0.09120785	3.07333247	(0.437500, 0.750000)
33	0.10672291	2.02672391	(0.468750, 0.593750)
65	0.11738557	2.05660796	(0.468750, 0.562500)
129	0.11880609	2.06677691	(0.468750, 0.562500)

Table 7.3: Values and locations of the primary vortex for Re = 1000 using different discretizations.

N	Error in $\Psi_{\max}$	Error in Φ
17	23.36%	48.57%
33	13.32%	2.02%
65	1.36%	0.58%
129	0.17%	0.09%

Table 7.4: Errors in values of the primary vortex for Re = 1000 using different discretizations, comparing with Ghia et al.'s solution.

N	$\Psi_{\max}$	Φ	Location
65	0.10909425	1.89353365	(0.484375, 0.578125)
129	0.12015739	1.94893416	(0.484375, 0.539063)
257	0.12165982	1.95948705	(0.484375, 0.539063)

Table 7.5: Values and locations of the primary vortex for Re = 3200 using different discretizations.

N	Error in $\Psi_{\max}$	Error in Φ
65	9.95%	2.93%
129	0.18%	4.01%
257	1.07%	1.46%

Table 7.6: Errors in values of the primary vortex for Re = 3200 using different discretizations, comparing with Ghia et al.'s solution.

N	$\Psi_{\max}$	Φ	Location
129	0.11812146	1.90621386	(0.484375, 0.539063)

Table 7.7: Values and locations of the primary vortex for Re = 5000 using $N = 129$.

N	Error in $\Psi_{\max}$	Error in Φ
129	0.71%	2.48%

Table 7.8: Errors in values of the primary vortex for Re = 5000 using $N = 129$, comparing with Ghia et al.'s solution.

N	$\Psi_{\max}$	Φ	Location
129	0.11433776	1.87589672	(0.492188, 0.554688)

Table 7.9: Values and locations of the primary vortex for Re = 7500 using $N = 129$.

N	Error in $\Psi_{\max}$	Error in Φ
129	4.28%	0.21%

Table 7.10: Errors in values of the primary vortex for Re = 7500 using $N = 129$, comparing with Ghia et al.'s solution.

N	$\Psi_{\max}$	Φ	Location
129	0.10237164	1.62562304	(0.500000, 0.5546875)
257	0.10529822	2.12947152	(0.464835, 0.6484375)

Table 7.11: Values and locations of the primary vortex for Re = 10000 using different discretizations.

N	Error in $\Psi_{\max}$	Error in Φ
129	14.50%	13.57%
257	12.05%	13.22%

Table 7.12: Errors in values of the primary vortex for Re = 10000 using different discretizations, comparing with Ghia et al.'s solution.

Re	ω_{S}	ω_{V}	ω_{i}	Iteration
100	0.92	0.87	0.32	47
400	1.00	0.75	0.25	113
1000	1.24	0.77	0.15	269
2000	0.68	0.68	0.10	1088

Table 7.13: Convergence with different Re. $N = 64$ and the damping parameters. Only one multigrid cycle was used for each inner iteration.

solution with $N = 129$ and $N = 257$ computed from our method differ significantly from Ghia et al.'s solution. However, we did obtain converged solutions in both cases. Since there is no exact solution available for the driven cavity problem and published results for high Reynolds numbers are still open to discussion, we are not sure which of these solutions is more accurate.

In Table 7.13, we list the number of outer iterations of our multigrid solver and the damping factors used for solving the cavity problem for $h = 1/64$ with some Re's tested in [28, 41] (using $h = 1/40$). These results compare well with those listed in [28, 41] using the SOR method. Since we only use one multigrid V(1,1)- and W(1,1)-cycles in each inner iteration the cost of each inner iterations is about 4 to 6 SOR iterations on the finest grid. Had we used more multigrid cycles in each inner iteration the number of the outer iterations would be smaller, but the CPU timings could be larger and the algorithm might be less cost-effective.

7.5.2 Comparison of High Accuracy Solutions

In the last subsection, the errors for the vorticity values are usually larger than the errors of the stream-function values for large Re, as indicated by data in Tables 7.5 and 7.6 for Re = 3200 and in Tables 7.7 and 7.8 for Re = 5000. This was caused by the error of the benchmark solution for large Re. As indicated in [44], a higher order of accuracy in space is necessary for the (time-dependent) high Reynolds number simulations.

In Table 7.14, we give some recently available higher order results for Re = 3200 and compare them with our scheme with $N = 129$ and $N = 257$.

It can be seen from Table 7.14 that Ghia et al.'s solution has some difference from the high-order, high-accuracy solutions recently available in the literature, especially

Authors	$\Psi_{\max}$	Φ
Current	0.12015739	1.94893416
Current	0.12165982	1.95948705
Li, Tang & Fornberg [41]	0.120529	1.94286
Nishida & Satofuka [44]	0.121154	1.95078
Ghia, Ghia & Shin [21]	0.120377	1.98860

Table 7.14: Comparison of recent high accuracy results for Re = 3200.

Authors	Discretization	Accuracy Order
Current	129×129	4th-order
Current	257×257	4th-order
Li, Tang & Fornberg [41]	129×129	4th-order
Nishida & Satofuka [44]	129×129	6th-order
Ghia, Ghia & Shin [21]	129×129	2nd-order

Table 7.15: Comparison of recent high accuracy solution methods for Re = 3200.

the vorticity value which is shown to have a relative difference of 1.49% with respect to our very accurate solution computed by using the 4th-order algorithm and $N = 257$. In contrast, our solution with $N = 129$ has only a relative difference of 0.54% with respect to our very accurate solution, but has a relative error of 4.01% with respect to Ghia et al.'s solution. This comparison again supports the claim made by Nishida and Satofuka [44] that higher-order algorithm is necessary (at least beneficial) for the high Reynolds number computation.

7.5.3 Solution Contours

The streamlines and the vorticity contours for Re $= 3200, 5000, 7500, 10000$ for $N = 129$ are presented by Figures 7.2 to 7.5. These figures compare well with well-known figures obtained by Ghia et al. [21], taking into account the difference in definition. Note that although our $\Psi_{\max}$ value for Re $= 10000$ differs from the benchmark solution by more than 10%, the streamline and vorticity curves are qualitatively correct.

7.6 Concluding Remarks

Fourth-order compact discretization formulas have been used in conjunction with the multigrid technique to develop a high accuracy multigrid solver ($\mathcal{NPF}$-MG) for the steady-state incompressible Navier-Stokes equations. A driven cavity model problem with small to large Reynolds numbers has been solved by using our method. The computed solutions compare well with the benchmark solutions obtained by other investigators using finer discretizations.

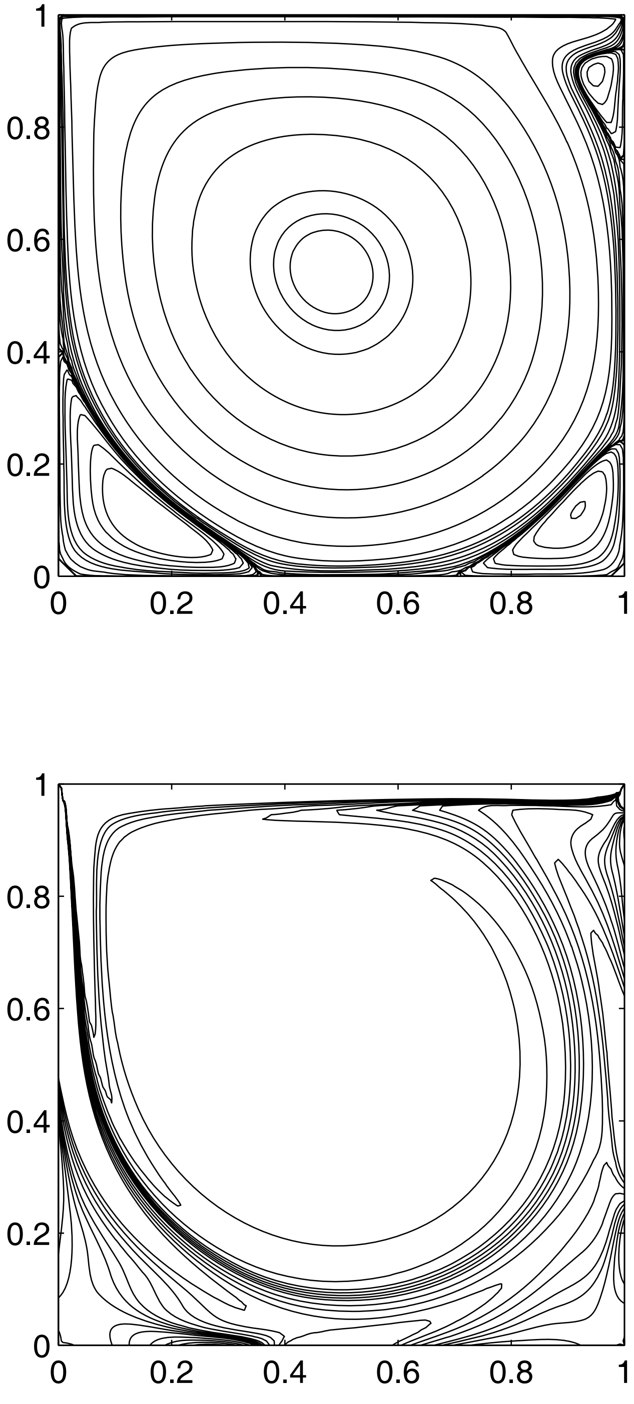

Figure 7.2: Streamline and equivorticity curves for Re $= 3200$ with $N = 129 \times 129$.

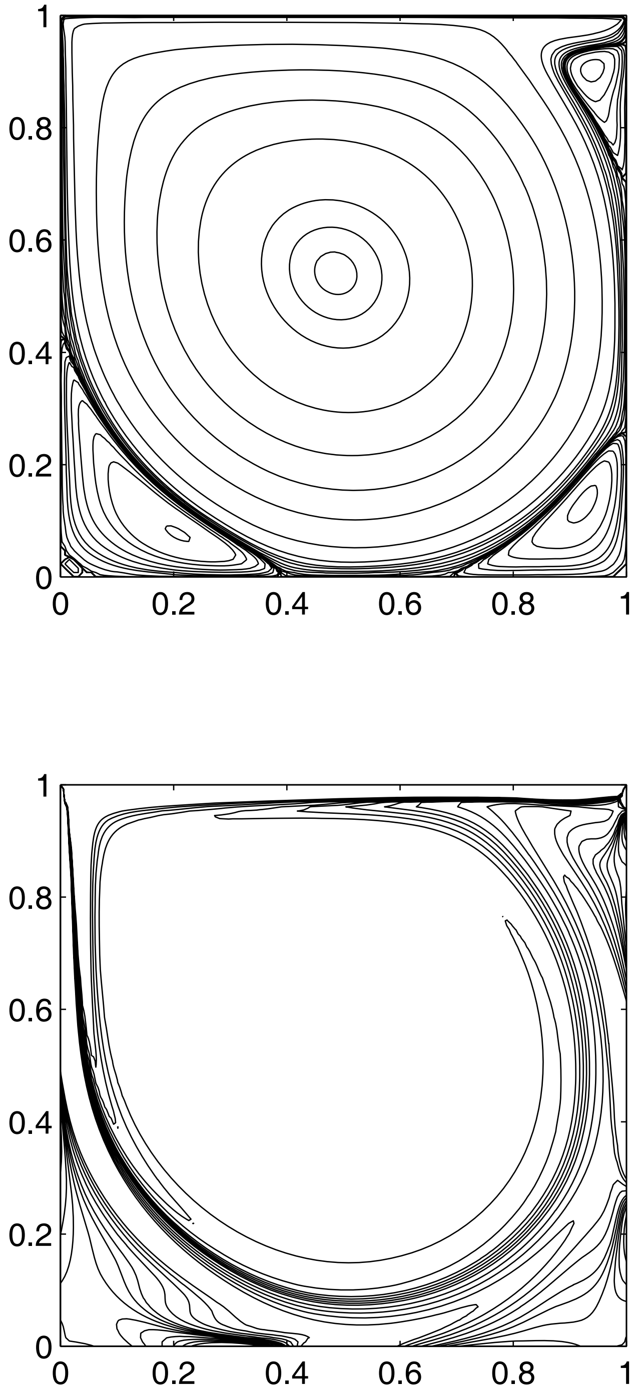

Figure 7.3: Streamline and equivorticity curves for Re $= 5000$ with $N = 129 \times 129$.

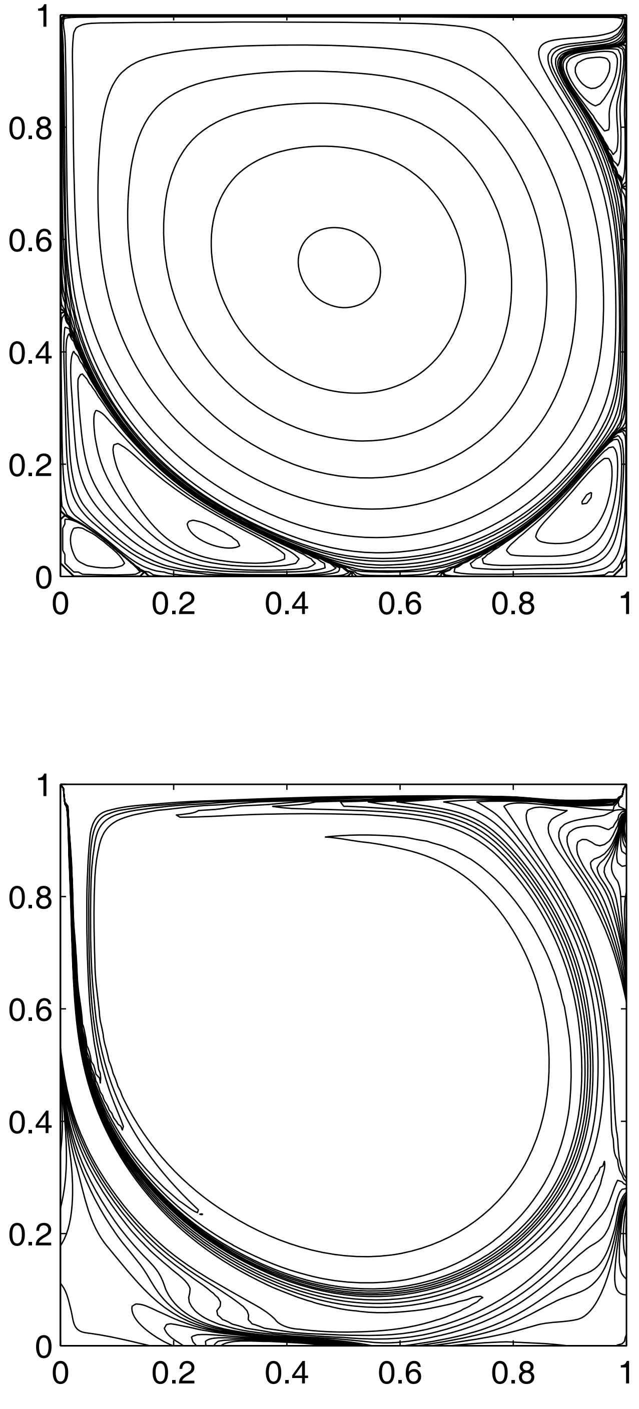

Figure 7.4: Streamline and equivorticity curves for Re $= 7500$ with $N = 129 \times 129$.

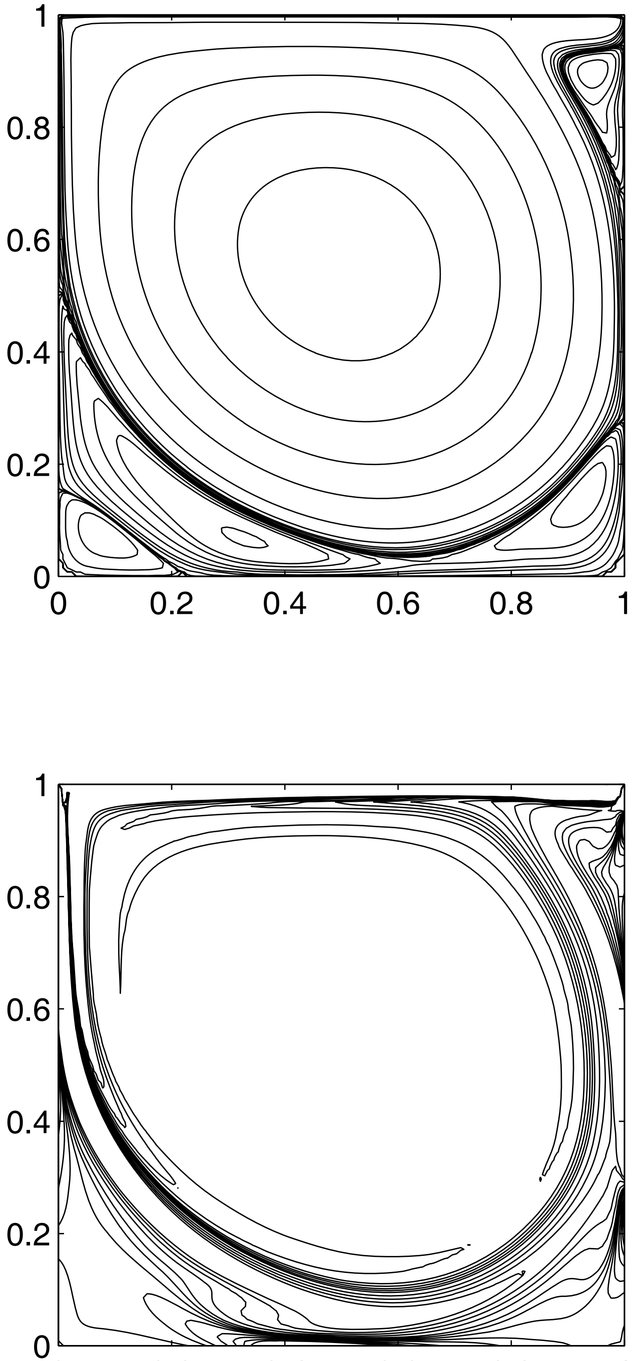

Figure 7.5: Streamline and equivorticity curves for Re = 10000 with $N = 129 \times 129$.

Chapter 8

Conclusions and Outlook

We have developed unified approach to the multigrid acceleration techniques and designed a class of efficient multigrid acceleration schemes. From our analysis and numerical experiments, it is clear that these acceleration schemes are useful and cost-effective in accelerating standard multigrid method, especially when the standard multigrid method converges slowly. Convergence of the standard multigrid method deteriorates when it is coupled with higher order finite difference schemes for solving the convection-diffusion equations with large Reynolds numbers. In addition to efficiency, the acceleration schemes presented in this dissertation are easy to implement.

We have designed some fourth-order multigrid methods for the convection-diffusion equations and the incompressible Navier-Stokes equations with large Reynolds numbers. The fourth-order multigrid methods have been shown to be stable, converge fast and produce high accuracy numerical solution.

8.1 Accomplishments of Chapter 2

- We developed efficient ways to analyze and design acceleration schemes for the isotropic operators. We proposed the idea that acceleration should be considered for smoothing both the low and high frequency components of the errors. We explained the reason why traditional SOR method is not cost-effective comparing with Gauss-Seidel relaxation in multigrid. The analysis that we employed in Chapter 2 unveils that different acceleration should be used for different part of the multigrid cycle (pre-smoothing and post-smoothing sweeps). The estimates of the under-relaxation parameter for the pre-smoothing sweep and the over-relaxation parameter for the post-smoothing have been verified by the numerical experiments.

- Perhaps the most important result of Chapter 2 is that it corrects a long-standing misunderstanding in multigrid community that standard multigrid with SOR for solving the Poisson equation does not pay. This research result clears the way for searching for effective acceleration schemes for the Poisson equation and other isotropic operators (or slightly anisotropic operators). Some more generally applicable acceleration schemes have been developed in Chapter 3 and 4.

- Although using two parameters to accelerate iterative methods have been investigated by several researchers including Golub and de Phill [22], the issue of how to estimate these parameters has not been fully resolved. The idea that we proposed in Chapter 2 is the first time (at least in the multigrid context) that an analytical means is used to accurately estimate the optimal relaxation parameters.

- In all, Chapter 2 provides a new way of thinking and opens a new direction for searching for efficient acceleration schemes in multigrid.

8.2 Accomplishments of Chapter 3

- There have been various acceleration schemes proposed to accelerate the multigrid methods in different situations. Some of them are theoretically justified. Typical examples are the so-called steplength optimization of Reusken and the over-correction scheme of Vaněk and Mika [42, 62]. The cost of these acceleration schemes is prohibitively high and the validity of these schemes are based on the assumption that the coefficient matrix is symmetric and positive definite. The symmetry and positive definiteness of the coefficient matrix are usually violated in many interesting applications, such as the linear system arising from discretizing the convection-diffusion equations with large convection coefficients. Hence, these acceleration schemes are of limited usefulness.

- Another class of acceleration schemes is to modify the residuals before they are projected to the coarse grid. Typical examples in this group are the over-weighted residual technique of Brandt and Yavneh [9], and the the scaled residual injection operators of Zhang [80, 81]. These techniques are primarily based on some heuristic residual analysis and do not have a very rigorous theoretical justification. However, the cost of these acceleration schemes are usually zero or negligible. There is no assumption on the coefficient matrix. These acceleration techniques are potentially useful in practical applications.

- The essential results of Chapter 3 are the unification of all the coarse-grid acceleration schemes, including the two categories mentioned above. We proposed the concept of *residual scaling techniques* to include all these techniques, because they all essentially modify the scale of the residual equation. We have proved the equivalence of the pre-scaling [9, 79, 80, 81] and post-scaling [42, 46, 62] techniques. The proof is essentially saying that all these (and possibly others) coarse-grid acceleration schemes are mathematically equivalent.

- Because of the equivalence that we proved and the residual scaling concept that we proposed in Chapter 3, we can research for alternative methods to estimate the residual scaling parameters and to avoid the assumption of the symmetry and positive definiteness of the coefficient matrix. The efficiency and the cost-effectiveness of the pre- and post-scaling techniques are unified.

- We also set a foundation for developing practical methods to estimate the residual scaling parameters. The method is the so-called *heuristic residual analysis* method which is based on the geometry of the grid points and the particular relaxation pattern employed. The heuristic residual analysis method is successfully used to estimate the residual scaling parameters for a standard multigrid Poisson solver in Appendix A and a high-order multigrid method for solving the Poisson equation and the convection-diffusion equations with small convection coefficients in Chapter 6.
- The research results obtained in Chapter 3 are of both theoretical and practical importance. Theory is initially proposed, rigorously proved and numerically verified in later chapters.

8.3 Accomplishments of Chapter 4

- Chapter 4 introduces the *minimal residual smoothing* techniques into the multigrid context. The minimal residual smoothing techniques have been primarily used in the conjugate gradient type methods, but rarely elsewhere. Chapter 4 develops several algorithms that may utilize the minimal residual smoothing technique in multigrid. Since this is a new application area, we gave several algorithms so that interested readers may find a suitable one for their particular applications. But we gave our preferred version of the algorithm and the reasons of our preference.
- The minimal residual smoothing technique may be considered as an (indirect) residual scaling technique. In particular, the minimal residual smoothing technique we used in the multigrid algorithm is a pre-scaling technique, which may compare with the post-scaling techniques of Reusken, Vaněk and Mika. Unlike other pre-scaling techniques, the minimal residual scaling has some theoretical basis for optimization (the norm of the residual). The effect of the minimal residual smoothing is supposed to compare well with the steplength optimization technique of Reusken and the over-correction techniques of Vaněk and Mika. The most important observation we would like to point out is that the minimal residual smoothing technique does not require that the coefficient matrix be symmetric and positive definite. In fact, it is independent of the coefficient matrix and of the particular relaxation method employed. This implies that the minimal residual smoothing technique may be used in a wide range of applications.

8.4 Accomplishments of Chapter 5

- Although heuristic justification may be given, there has been no rigorous theoretical convergence theory for applying the minimal residual smoothing techniques in multigrid. These techniques have been used in the Krylov subspace methods for many years without theoretical justification. Only in recent years, there have been efforts to justify these techniques. But their results are based on the Krylov subspace methods which are essentially based on the residual orthogonality assumption and therefore are not suitable for general analysis, such as in the multigrid methods.

- In analyzing the minimal residual smoothing technique accelerated multigrid method, we have focused on how to explain the acceleration effect. We do not want to impose limitation on the coefficient matrix because our original intention was to design an acceleration scheme that is independent of the coefficient matrix. So we imposed some conditions on the decomposition of the initial residual. It seems that our assumptions were well suitable for the analysis purpose.

- The analysis and theorems proved in Chapter 5 are very important in explaining how minimal residual smoothing technique accelerates and stabilizes the multigrid convergence. Some of the theoretical results are applicable to analyzing other iterative methods accelerated by the minimal residual smoothing techniques. Our results are obtained without the assumption of the orthogonality of the residuals.

8.5 Accomplishments of Chapter 6

- Fourth-order compact finite difference discretization methods are used with the multigrid techniques to solve the convection-diffusion equations with large Reynolds numbers. The methods are efficient, accurate and stable.

- Residual injection operator is used and the optimal residual scaling parameter is obtained through the *heuristic residual analysis* technique for the diffusion-dominated problems.

- We have found that for solving the convection-dominated problems with stagnation point and large Reynolds numbers, the residual injection is more robust than the full-weighting.

8.6 Accomplishments of Chapter 7

- Fourth-order compact finite difference discretization methods are used with the multigrid techniques to solve the steady-state incompressible Navier-Stokes equations. We have shown that the fourth-order multigrid do give high accuracy numerical solution for the model test problem of driven-cavity with relatively coarse grid.

- Fourth-order velocity and boundary approximations are used in conjunction with the fourth-order relaxation method.

- This is the first time that genuine fourth-order compact finite difference schemes are seriously considered with the multigrid techniques (without defect correction techniques) to solve the incompressible Navier-Stokes equations. Our results are very encouraging and comparing well with benchmark solutions.

8.7 Future Research Outlook

We think the efficiency of the multigrid method is in the correct scale of the residual that is projected to the coarse grid to form the coarse grid subproblem. If the residual is not in the correct scale, the multigrid method will not be in an optimal shape and the convergence will be seriously deteriorated. The acceleration schemes developed in this dissertation are essentially to modify the residual to the correct scale. This idea is compatible with the unified concept of the *residual scaling technique*.

The *minimal residual smoothing* technique can scale the residual dynamically and applicable to almost any problems. Other pre-scaling and post-scaling techniques use fixed scaling parameters and thus are very cheap in implementation, but the optimal scaling parameter is usually problem-dependent.

There exists a link between the successive over relaxation parameter ω and the residual scaling factor α. In fact, if a typical SOR step is represented as

$$u_k = u_{k-1} + \omega(\bar{u}_k - u_{k-1}),$$

then we have

$$r_k = r_{k-1} + \omega(\bar{r}_k - r_{k-1}).$$

Hence the change of ω affects α, but not vice versa. Also, the interval in which we may vary α seems much larger than the interval contains the useful ω. In our numerical experiments, we observed that if the scale of the residual is optimized, there is no need to use any relaxation parameter. On the other hand, these two parameters can be adjusted with respect to each other. Hence, it seems that there is no need to use relaxation parameter so long as the relaxation method does not diverge. We can adjust the residual scaling factor to achieve the same efficiency with lower cost. There may be a quantitative relation between these two parameters.

Our numerical experiments also suggested that the minimal residual smoothing technique be more efficient than the steplength length optimization or over-correction techniques. This seems contradicting to the conventional wisdom because the over-correction technique optimizes correction scale at each step while the minimal residual smoothing minimizes residual norm before the coarse-grid-correction step. However, the superiority of the minimal residual smoothing technique may be explained as it uses information from the previous step and we may expect that the effect is better than a single step optimization. Detailed comparison are given in Zhang [85].

Appendix A

An Optimal Residual Projection Operator

A.1 Introduction

Individual multigrid operators, including relaxation (smoother), projection (restriction) and interpolation (prolongation) operators, should be optimally combined to achieve true multigrid efficiency. There exist some options for each operator, some of them are much more expensive than others. In practical applications, sacrifice in convergence sometimes is made to favor the computational cost-effectiveness. If the discretization is the five-point 2nd-order central difference scheme and the grid space is ordered in a red-black fashion (see Figure A.1), the five-point red-black Gauss-Seidel (RBGS) relaxation, together with half-injection and bi-linear interpolation, is probably the most cost-effective two dimensional Poisson solver in existence. This combination is considered almost perfect. For example, Yavneh's recent work on multigrid acceleration is only applicable to the anisotropic operators (in two dimensional cases) [72, 73].

Nevertheless, we have made some progress in designing SOR-type acceleration schemes to accelerate the convergence of the RBGS smoothing in multigrid for the two dimensional isotropic operators (see Chapter 2 and [77]). Our work in Chapter 2 indicates that acceleration parameters may be used to accelerate RBGS in multigrid with negligible cost. The results corrected a long-time misunderstanding in multigrid that such an acceleration would not pay for the cost (see, e.g., [59, 73]). Our research work demonstrates that projection and interpolation processes should be treated (accelerated) separately, possibly by using different parameters. The results of Chapter 2 are indeed near-optimal in the sense of computational cost-effectiveness, as we shall claim here.

Other acceleration schemes which are restricted to the positive definite coefficient matrices are proposed by Reusken [46] and Vaněk [62]. These post-optimization acceleration schemes optimize the computed correction and the acceleration rates are optimal in the sense of per cycle convergence. In practice, however, these schemes are too costly (and restricted) to be efficient. A similar pre-optimization acceleration scheme (i.e., the minimal residual smoothing) which is applicable to any coefficient matrices and which is cheaper than the post-optimization schemes has been proposed in Chapter 4 and analyzed in Chapter 5. Most of these existing acceleration schemes have aimed at accelerating the convergence rate only. However, in this appendix, we introduce a different acceleration scheme derived from a novel heuristic residual analysis technique which is based on the geometry of the grid points and a particular relaxation pattern. The philosophy of developing residual injection operator is to achieve optimal computational efficiency as well as optimal convergence.

From a theoretical point of view, employment of residual injection has some disadvantages, as noted by Stüben and Trottenberg. The spectral and energy norms of the corresponding local two-grid operators are not bounded [59, p. 127]. In practice, convergence may deteriorate as the meshsize tends to zero. Hence, full-weighting is regarded as more robust. For RBGS, the injection operator has its special attraction. Since the residuals at the black points are zero, the injection operator is equivalent to the five-point half-weighting (optimal-weighting) operator (see [59]).

In this appendix, we optimize the residual injection operator by choosing an optimal residual injection factor (residual scaling parameter). The optimal injection operator maintains the low cost of half-injection, but provides convergence faster than full-weighting. The numerical results obtained by using this scaled residual injection operator are slightly better in average than the results obtained by using the two-way acceleration scheme introduced in Chapter 2, but not overwhelmingly. One advantage of the current approach is that it incurs virtually no extra cost. The two-way acceleration scheme in Chapter 2 requires about 4% additional cost for each V-cycle.

We restrict our attention to the two dimensional Poisson equation discretized by the five-point 2nd-order central difference scheme. (A similar residual injection operator for a high-order multigrid is considered in Chapter 6.) The RBGS relaxation and bi-linear interpolation are employed. We only optimize the projection operator.

A.2 A Heuristic Residual Analysis

The RBGS relaxation is probably the most efficient smoother in multigrid for Poisson-like equations [59, p. 85]. The bi-linear interpolation is customarily employed for a V-cycle algorithm. In practice, the half-injection projection operator is used in connection with RBGS. The residuals are directly injected (transferred) to the corresponding coarse grid points weighted by 1/2. The factor of 1/2 is motivated by the fact that the residuals are zero at black points on the fine grid, hence the other residuals should be multiplied by 1/2 to represent the correct average [4, p. 219].

The multigrid method solves the residual equations on the coarse grids. Since the half-injection operator does not take this difference into account and the linear system is not solved accurately on the finest grid, the residuals injected from the finest grid to the coarse grid using half-injection is not accurate.

To find the optimal residual injection operator with the optimal scaling parameter,

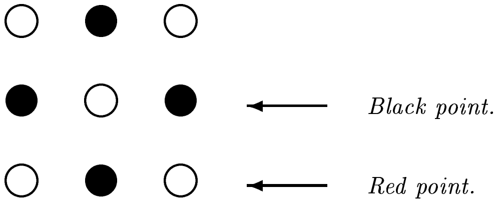

Figure A.1: A red-black ordered nine-point stencil for full-weighting scheme.

we consider the full-weighting scheme (see Figure A.1):

$$\begin{aligned}\bar{r}_{i/2,j/2} &= \frac{1}{16}[4r_{i,j} + 2(r_{i+1,j} + r_{i-1,j} + r_{i,j+1} + r_{i,j-1}) \\ &\quad +(r_{i+1,j+1} + r_{i+1,j-1} + r_{i-1,j+1} + r_{i-1,j-1})]. \qquad \text{(A.1)}\end{aligned}$$

Here $r_{i,j}$ is the residual on the finest grid at the point (i, j), i and j are even numbers (the center point in Figure A.1). ((i, j) is a red point.) $\bar{r}_{i/2,j/2}$ is the quantity to be transferred to the corresponding coarse grid point $(i/2, j/2)$. The weight assigned to the residual at each grid point is determined by the involvement of that point in the number of coarse grid point residual computations. For example, $r_{i+1,j+1}$ is weighted into the residual calculation of four coarse grid points at $(i/2, j/2), (i/2, j/2 + 1), (i/2 + 1, j/2)$ and $(i/2+1, j/2+1)$, respectively. The weights in formula (A.1) correctly reflect these algebraic relations. But they do not reflect the geometric relations of the reference point (i, j) and its immediate four neighboring red points involved in the computation of formula (A.1).

To take their relative geometric positions into consideration, we use the following simple heuristic residual analysis. Since RBGS is used, the residuals at the black points are zero as noted above, i.e.

$$r_{i+1,j} = r_{i-1,j} = r_{i,j+1} = r_{i,j-1} = 0. \qquad \text{(A.2)}$$

Formula (A.1) is reduced to

$$\bar{r}_{i/2,j/2} = \frac{1}{16}\left[4r_{i,j} + (r_{i+1,j+1} + r_{i+1,j-1} + r_{i-1,j+1} + r_{i-1,j-1})\right]. \qquad \text{(A.3)}$$

We look for an optimal scaling factor α such that $\alpha r_{i,j}$ approximates $\bar{r}_{i/2,i/2}$ as accurately as possible. After substituting $\bar{r}_{i/2,j/2} = \alpha r_{i,j}$ into equation (A.3) we have

$$r_{i,j} = \frac{1}{4(4\alpha - 1)}\left[r_{i+1,j+1} + r_{i+1,j-1} + r_{i-1,j+1} + r_{i-1,j-1}\right]. \qquad \text{(A.4)}$$

According to the multigrid philosophy the residuals should be sufficiently smoothed by relaxation before they are projected to the coarse grid, we may assume that the residual at the the grid point (i, j) is locally equal to the residuals of its immediate four neighboring red points involved in the weighting scheme (A.1) (or equivalently (A.4)), then

$$r_{i+1,j+1} = r_{i+1,j-1} = r_{i-1,j+1} = r_{i-1,j-1} = r_{i,j}. \tag{A.5}$$

If $r_{i,j} = 0$, any scaling parameter α is optimal with respect to the current reference point (i, j). We can neglect this point and choose another red point as the reference point. If the residuals at all red points are zero, we have reached convergence. Without loss of generality, we assume that $r_{i,j} \neq 0$.

If we neglect the relative geometric positions of the red points in formula (A.4) and substitute equation (A.5) into equation (A.4), we obtain the idealized half-injection factor $\alpha = 1/2$, which would be an upper bound of the injection factor, so we denote $\alpha_{upper} = 1/2$.

However, the real positions of these red points are rotated by 45° from the positions of the nearest (black) grid points. Their distance from the reference center point (i, j) is increased from 1 to $\sqrt{2}$. Therefore, their weights in formula (A.4) should be scaled by a factor of $1/\sqrt{2}$. Hence, we set

$$r_{i+1,j+1} = r_{i+1,j-1} = r_{i-1,j+1} = r_{i-1,j-1} = \frac{1}{\sqrt{2}} r_{i,j}. \tag{A.6}$$

Substituting (A.6) into (A.4) and cancel $r_{i,j} (\neq 0)$, we have

$$\alpha = \frac{2+\sqrt{2}}{8} \approx 0.4268.$$

This gives the lower bound of the factor α, we denote $\alpha_{lower} = 0.4268$.

The optimal scaling factor $\alpha_{optimal}$ lies between α_{upper} and α_{lower}. There exists some $\xi \in (0, 1)$ such that

$$\alpha_{optimal} = \frac{\xi \alpha_{lower} + (1-\xi)\alpha_{upper}}{2} = \frac{4 + (\sqrt{2} - 2)\xi}{8}. \tag{A.7}$$

In absence of further information to justify any preferred choice of ξ, we take $\xi = 1/2$ and equation (A.7) yields

$$\alpha_{optimal} = \frac{6+\sqrt{2}}{16} \approx 0.4634.$$

Since $\alpha_{optimal}$ is smaller than the traditional half-injection factor, we refer to the residual injection operator with this scaling factor as *under-injection*.

The scaling factor $\alpha_{optimal}$ is used for injecting the residuals from the finest grid to the coarse grid. Subsequent residual injection from coarse grid to coarser grid, however, uses $\alpha = 0.5$.

A.3 Computational Cost Analysis

For full-weighting, we must compute the residuals at all fine grid points and weight the residuals (according to formula (A.1)) at the red points which correspond to some

N	under-injection	full-weighting	half-injection	two-way accel.
16	**9**	11	9	9
32	**9**	11	10	10
64	**9**	11	11	10
128	**9**	11	11	10
256	**9**	11	11	10
512	**9**	11	12	10

Table A.1: Comparison of the number of V(1,1)-cycles for Test Problem 2.5.1.

coarse grid points. On the other hand, residual injection needs only to compute the fine grid residuals at the red points which correspond to some coarse grid points. The computation of residuals on a given grid is roughly equivalent to one full relaxation on that grid. Hence, the cost of residual injection is about a quarter of the cost of full-weighting. If we take into consideration the cost of the weighting scheme (A.1), the cost of residual injection is about one-fifth of the cost of full-weighting. If a V(1,1)-cycle algorithm is employed and both full-weighting and residual injection have the same convergence rate, using residual injection may save up to 30% computer's time.

A.4 Numerical Experiments

Our numerical experiments are conducted with the model Poisson equation (2.1) and the Test Problems 2.5.1, 2.5.2 and 2.5.3 in Chapter 2.

The Poisson equation is discretized by the usual five-point 2nd-order central difference scheme. The RBGS relaxation, bi-linear interpolation and full-weighting or injection of some kind are employed in the multigrid V(1,1)-cycle algorithm. All experiments are done on a SUN SPARCstation 1+ using Fortran 77 in double precision. Initial guess is $u(x,y) = 0$. $(N+1)^2$ is the number of points on the finest grid and the coarsest grid contains 9 points (one unknown). The program terminates when the residual on the finest grid in L_2 norm is less than 10^{-9}. (Note that this stopping criteria is the absolute reduction in residual norm, not the relative reduction in residual norm.)

For different N, we solve the three test problems of Chapter 2 using multigrid method with different residual projection operators, i.e., under-injection, full-weighting and half-injection. We also test the two-way acceleration scheme introduced in [77]. The numbers of V(1,1)-cycles (convergence rate) are tabulated in Tables A.1 to A.3.

From Tables A.1 to A.3, we note that in all cases, under-injection achieves convergence rate better than full-weighting and half-injection. For Test Problem 2.5.1, under-injection is better than the two-way acceleration scheme. For Test Problem 2.5.3, their convergence rates are similar. For Test Problem 2.5.2, with $N = 128, 512$, the two-way acceleration achieves better convergence.

These numerical tests show that under-injection and the two-way acceleration

N	under-injection	full-weighting	half-injection	two-way accel.
16	**11**	13	11	11
32	**11**	13	13	12
64	**12**	13	14	12
128	**13**	13	15	12
256	**13**	14	15	13
512	**14**	14	16	13

Table A.2: Comparison of the number of V(1,1)-cycles for Test Problem 2.5.2.

N	under-injection	full-weighting	half-injection	two-way accel.
16	**10**	12	11	10
32	**11**	13	13	11
64	**12**	13	14	12
128	**12**	13	15	12
256	**13**	14	15	13
512	**13**	14	16	13

Table A.3: Comparison of the number of V(1,1)-cycles for Test Problem 2.5.3.

scheme are effective ways of accelerating the convergence of standard multigrid method. Under-injection is more attractive because it incurs virtually no additional cost over half-injection. The two-way acceleration scheme incurs about 4% extra cost per V-cycle, although this additional cost is negligible.

The acceleration rates in convergence achieved by the under-injection operator are in the range of 10 – 20% with respect to the full-weighting operator. However, the comparison of convergence rate in Tables A.1 to A.3 does not take into consideration the fact that the cost of the residual injection operators is only about a quarter to one-fifth of the cost of the full-weighting operator (see discussion in Section A.3). Table A.4 gives the CPU time in seconds for the under-injection, full-weighting and half-injection operators. The efficiency rate in the fifth column of Table A.4 represents the reduction rate in CPU time for the under-injection operator with respect to the full-weighting operator. We note that the efficiency rates are almost 40%. This is more than what we estimated in Section A.3 because under-injection achieves faster convergence. These efficiency rates are very attractive.

One may tend to combine the two-way (SOR-type) acceleration scheme of Chapter 2 with the under-injection operator. However, since we have optimized the residual injection operator, we expect that the SOR acceleration will have little effect on the projection process. We have done some numerical experiments which showed that this is true and $\omega_1 = 1$ (no acceleration) is indeed optimal for the projection process. On

N	under-injection	full-weighting	half-injection	efficiency rate
16	**0.19**	0.27	0.19	29.63%
32	**0.61**	0.99	0.68	38.38%
64	**2.47**	3.91	3.03	36.83%
128	**10.37**	16.26	12.65	36.22%
256	**41.91**	66.05	51.57	36.55%
512	**177.28**	276.31	235.86	38.84%

Table A.4: Comparison of CPU time in seconds for Test Problem 2.5.1.

the other hand, interpolation process may be accelerated. The optimal parameter is about $\omega_2 = 1.16$ for acceleration on the second finest level of the interpolation process for Test Problems 2.5.2 and 2.5.3. Unfortunately, this acceleration option deteriorates the convergence rate of Test Problem 2.5.1. Since there is an extra 2% cost for this acceleration. The average cost-effectiveness is not as competitive as the optimal residual injection without additional SOR acceleration. This implies that both two original schemes are near-optimal in the sense of computational cost-effectiveness.

A.5 Conclusions and Remarks

We have obtained a near-optimal under-injection factor through a novel heuristic residual analysis. The under-injection operator has been tested to show near-optimal convergence rate in the sense of computational cost-effectiveness. In fact, these test results have been posted in the electronic multigrid newsletter [79] for open discussion. The reduction rates in CPU time resulted from using the under-injection operator are almost 40% with respect to the full-weighting operator. Although the overall CPU cost for solving a Poisson equation using multigrid method with any residual projection operator discussed above is trivial on modern computers, if a Poisson solver is repeatedly called as a subroutine in solving a complicated problem, such as the incompressible Navier-Stokes equation, using the under-injection operator is obviously advantageous. In addition, there is no coding complexity for the under-injection operator.

The main idea of the heuristic residual analysis technique is to consider the geometric locations of the grid points and the relaxation pattern. This technique may be extended to derive optimal residual injection operator for other multigrid applications, not necessarily limited to the Poisson equation. It has been shown in Chapter 6 and in [31] that using a residual injection operator is necessary for convergence when a high-order multigrid method is used to solve the convection-diffusion equations. A heuristic residual analysis technique similar to that used in this paper has been employed to develop some optimal residual injection operator for the high-order multigrid (see Chapter 6 and [31]).

Appendix B

On Convergence of Iterative Methods for a Fourth-Order Discretization Scheme

B.1 Introduction

We consider the two-dimensional constant coefficient convection-diffusion equation

$$\begin{aligned} u_{xx} + u_{yy} + pu_x + qu_y &= -f(x,y), \qquad (x,y) \in \Omega, \\ u(x,y) &= g(x,y), \qquad (x,y) \in \partial\Omega, \end{aligned} \tag{B.1}$$

where Ω is a smooth convex domain in $\mathbf{R}^2$.

Recently, there has been growing interest in developing fourth-order finite difference schemes for the convection-diffusion equation (and the Navier-Stokes equations) which give high accuracy approximations, see [16, 25, 26, 41, 57] and the references therein. In particular, Gupta et al. [25] proposed a fourth-order compact finite difference scheme for solving (B.1) and showed *numerically* that the scheme is both highly accurate and computationally efficient. Classical iterative methods with this scheme have been shown *numerically* to converge for all values of p and q [25]. In [26], this compact scheme was extended to solve the convection-diffusion equation with variable coefficients. The new scheme has also been shown *numerically* to have a truncation error of order h^4 and good numerical stability for large values of $p(x,y)$ and $q(x,y)$.

However, we are not aware of any *analytical* result to prove that any of the classical iterative methods converge with these fourth-order compact schemes. A rigorous justification is always desirable in spite of the fact that numerical experiments have been successfully conducted,

In this appendix, we give some conditional convergence results for some classical iterative methods using the fourth-order compact scheme developed by Gupta et al. [25, 26]. Although our results are limited, they are a first step towards the convergence analysis of such iterative methods for the fourth-order approximation schemes.

B.2 Fourth-Order Compact Scheme

The nine-point compact fourth-order discretization scheme results in a linear system (1.1) (for details, see [25]), where $A = (a_{i,j})_{n^2\times n^2}$ is a square matrix, which is usually nonsymmetric and non-positive definite. Each equation of (1.1) is of the form:

$$\sum_{i=0}^{8} \alpha_i u_i = \frac{h^2}{2}[(f_4 + f_3 + f_2 + f_1 + 8f_0) + \gamma(f_1 - f_3) + \delta(f_2 - f_4)], \tag{B.2}$$

where the coefficients $\alpha_i, i = 0, \ldots, 8$, are described by the computational stencil:

$$\begin{pmatrix} \alpha_6 & \alpha_2 & \alpha_5 \\ \alpha_3 & \alpha_0 & \alpha_1 \\ \alpha_7 & \alpha_4 & \alpha_8 \end{pmatrix} = \begin{pmatrix} -(1-\gamma)(1+\delta) & -2(1+\delta)^2 - 2 & -(1+\gamma)(1+\delta) \\ -2(1-\gamma)^2 - 2 & 20 + 4\gamma^2 + 4\delta^2 & -2(1+\gamma)^2 - 2 \\ -(1-\gamma)(1-\delta) & -2(1-\delta)^2 - 2 & -(1+\gamma)(1-\delta) \end{pmatrix}. \tag{B.3}$$

Here $\gamma = ph/2$ and $\delta = qh/2$ are referred to as the *cell Reynolds numbers* [17]. When $\max\{|\gamma|, |\delta|\} \le 1$, we say that the linear system (1.1) (and the discretized boundary value problem (B.1)) is diffusion-dominated, otherwise it is convection-dominated. The *numerical experiments* conducted in [25] showed that classical iterative methods with this scheme converge for any values of p and q. We also showed numerically in [80] that the multigrid method with this scheme converges for all values of p and q even when they are functions of x and y.

B.3 Convergence for Diffusion-Dominated Case

Lemma B.3.1 *The coefficients of the nine-point stencil (B.3) satisfy*

$$\alpha_i \le -2, \qquad i = 1, 2, 3, 4, \tag{B.4}$$

for all values of γ and δ.

Proof. Direct verification. □

Lemma B.3.2 *The coefficients of the nine-point stencil (B.3) satisfy $\alpha_5\alpha_7 > 0$ and $\alpha_6\alpha_8 > 0$ if one of the following conditions hold:*

$$|\gamma| < 1, |\delta| < 1, \tag{B.5}$$

or

$$|\gamma| > 1, |\delta| > 1. \tag{B.6}$$

Proof. $\alpha_5\alpha_7 > 0$ if

$$(1 - \gamma^2)(1 - \delta^2) > 0. \tag{B.7}$$

It is easy to see that (B.7) holds if either (B.5) holds or (B.6) holds.

The conditions for $\alpha_6\alpha_8 > 0$ can be verified similarly. □

Lemma B.3.3 *The matrix A is irreducible.*

Proof. It is readily verified that the directed graph of A is strongly connected. □

Lemma B.3.4 *The matrix A is irreducibly diagonally dominant if $|\gamma| \leq 1$ and $|\delta| \leq 1$.*

Proof. A is diagonally dominant if

$$|a_{i,i}| \geq \sum_{j=1, j\neq i}^{n^2} |a_{i,j}|, \quad \text{for} \quad i = 1, \ldots, n^2. \tag{B.8}$$

(B.2) and (B.8) imply

$$|\alpha_0| \geq \sum_{j=1}^{8} |\alpha_j|. \tag{B.9}$$

Substituting (B.3) in (B.9) after simplification, we have

$$(|1+\gamma| + |1-\gamma|)(|1+\delta| + |1-\delta|) \leq 4. \tag{B.10}$$

Since $(|1+\gamma| + |1-\gamma|) = 2$ if $|\gamma| \leq 1$, and $(|1+\delta| + |1-\delta|) = 2$ if $|\delta| \leq 1$, it follows that (B.10) holds if $|\gamma| \leq 1$ and $|\delta| \leq 1$ both hold.

A is irreducible by Lemma B.3.3. Since the strict inequality in (B.8) holds for at least the first row of A for $|\gamma| \leq 1$ and $|\delta| \leq 1$, A is irreducibly diagonally dominant (see pp. 23 of Varga [61]). □

From Lemmas B.3.3 and B.3.4, we have the following theorem:

Theorem B.3.5 *The point Jacobi and the point Gauss-Seidel methods associated with A for $|\gamma| \leq 1$ and $|\delta| \leq 1$ are convergent for any initial guess.*

Proof. A is irreducibly diagonally dominant by Lemma B.3.4. The result follows from Theorem 3.4, pp. 73 of Varga [61]. □

B.4 Symmetrization of Coefficient Matrix

Theorem B.3.5 establishes the convergence property of the point Jacobi and point Gauss-Seidel methods with the fourth-order compact scheme when the problem is diffusion-dominated. To analytically show the convergence of classical iterative methods with this scheme for larger cell Reynolds number, following Elman and Golub [17], we first show that, under certain circumstances, A is symmetrizable by a real diagonal similarity transformation.

Theorem B.4.1 *The coefficient matrix A can be symmetrized with a real diagonal similarity transformation if and only if one of the following conditions hold:*

$$\gamma = \delta = 0, \tag{B.11}$$

or

$$|\gamma| = |\delta| = \sqrt{2}. \tag{B.12}$$

Proof. The unknowns can be ordered so that the matrix A has the block tridiagonal form

$$A = \text{tri}[A_{j,j-1}, A_{j,j}, A_{j,j+1}],$$

where $A_{j,j-1} = \text{tri}[\alpha_7, \alpha_4, \alpha_8], A_{j,j} = \text{tri}[\alpha_3, \alpha_0, \alpha_1], A_{j,j+1} = \text{tri}[\alpha_6, \alpha_2, \alpha_5]$.

We look for a matrix $Q = \text{diag}(Q_1, Q_2, \ldots, Q_n)$, where Q_j is a real diagonal matrix of the same order as $A_{j,j}$, such that $Q^{-1}AQ$ is symmetric. Let

$$Q_j = \text{diag}(q_1^{(j)}, q_2^{(j)}, \ldots, q_n^{(j)}), \qquad j = 1, 2, \ldots, n.$$

We first consider the diagonal block: $Q_j^{-1}A_{j,j}Q_j$ is symmetric if and only if

$$\frac{q_i^{(j)}}{q_{i+1}^{(j)}}\alpha_3 = \frac{q_{i+1}^{(j)}}{q_i^{(j)}}\alpha_1, \qquad 1 \le i \le n-1, \qquad 1 \le j \le n, \tag{B.13}$$

where $q_1^{(j)}$ may be arbitrary. Thus, the diagonal blocks can be symmetrized provided

$$q_{i+1}^{(j)} = \sqrt{\frac{\alpha_3}{\alpha_1}}q_i^{(j)}, \tag{B.14}$$

and this recurrence is well defined if and only if α_3/α_1 is positive, which is true by Lemma B.3.1. The (equal) quantities (B.13) are the $(i, i-1)$ and $(i-1, i)$ entries of the jth diagonal block of the symmetrized matrix.

For the offdiagonal blocks, we require

$$Q_j^{-1}A_{j,j-1}Q_{j-1} = (Q_{j-1}^{-1}A_{j-1,j}Q_j)^T. \tag{B.15}$$

Relation (B.15) holds if and only the following three scalar relations hold:

$$\frac{q_i^{(j-1)}}{q_i^{(j)}}\alpha_4 = \frac{q_i^{(j)}}{q_i^{(j-1)}}\alpha_2, \qquad \text{or} \qquad q_i^{(j)} = \sqrt{\frac{\alpha_4}{\alpha_2}}q_i^{(j-1)}; \tag{B.16}$$

$$\frac{q_{i+1}^{(j-1)}}{q_i^{(j)}}\alpha_8 = \frac{q_i^{(j)}}{q_{i+1}^{(j-1)}}\alpha_6, \qquad \text{or} \qquad q_i^{(j)} = \sqrt{\frac{\alpha_8}{\alpha_6}}q_{i+1}^{(j-1)}; \tag{B.17}$$

$$\frac{q_i^{(j-1)}}{q_{i+1}^{(j)}}\alpha_7 = \frac{q_{i+1}^{(j)}}{q_i^{(j-1)}}\alpha_5, \qquad \text{or} \qquad q_{i+1}^{(j)} = \sqrt{\frac{\alpha_7}{\alpha_5}}q_i^{(j-1)}. \tag{B.18}$$

(B.16) is well defined if and only if $\alpha_2\alpha_4 > 0$ which is true for any γ and δ by Lemma B.3.1. (B.17) is well defined if and only if $\alpha_6\alpha_8 > 0$, which is true if either $|\gamma| < 1$ and $|\delta| < 1$ both hold, or $|\gamma| > 1$ and $|\delta| > 1$ both hold by Lemma B.3.2. The same conditions are required for (B.18) being well defined.

However, $Q^{-1}AQ$ is symmetric if and only if conditions (B.14), (B.16), (B.17) and (B.18) hold simultaneously.

From (B.14) and (B.16), we have

$$q_{i+1}^{(j)} = \sqrt{\frac{\alpha_3\alpha_4}{\alpha_1\alpha_2}}q_i^{(j-1)}, \tag{B.19}$$

which is well defined since $\alpha_1\alpha_2\alpha_3\alpha_4 > 0$ by Lemma B.3.1.

Substituting (B.19) into (B.18) and equating the coefficients of $q_i^{(j-1)}$, we have

$$\alpha_1\alpha_2\alpha_7 - \alpha_3\alpha_4\alpha_5 = 0. \tag{B.20}$$

Substituting (B.14) into (B.17), we obtain

$$q_{i+1}^{(j)} = \sqrt{\frac{\alpha_3\alpha_8}{\alpha_1\alpha_6}}q_{i+1}^{(j-1)}. \tag{B.21}$$

Substituting (B.16) with the subscript i being replaced by $i+1$ into (B.21) and comparing the coefficients of $q_{i+1}^{(j-1)}$, we have

$$\alpha_1\alpha_4\alpha_6 - \alpha_2\alpha_3\alpha_8 = 0. \tag{B.22}$$

Substituting the coefficient values of (B.3) into (B.21) and (B.22), after simplification, we have the system of two equations which must hold simultaneously

$$\begin{array}{rcl}(\gamma+\delta)(-2\gamma^2+2\gamma\delta-2\delta^2+\gamma^2\delta^2) &=& 0,\\ (-\gamma+\delta)(2\gamma^2+2\gamma\delta+2\delta^2-\gamma^2\delta^2) &=& 0.\end{array} \tag{B.23}$$

The solutions to system (B.23) are

$$\begin{array}{ccccc}\gamma &=& \delta &=& 0,\\ |\gamma| &=& |\delta| &=& 1,\\ |\gamma| &=& |\delta| &=& \sqrt{2}.\end{array}$$

Since we must also have $\alpha_6\alpha_8 > 0$ and $\alpha_5\alpha_7 > 0$ for (B.17) and (B.18) being well defined, solution $|\gamma| = |\delta| = 1$ is excluded by Lemma B.3.2. □

Remark B.4.2 *If (B.11) holds, (B.1) reduces to the Poisson equation and the matrix A is symmetric and positive definite by itself. The interesting case is when $|\gamma| = |\delta| = \sqrt{2}$, equation (B.1) is convection-dominated and the linear system (1.1) is nonsymmetric and non-diagonally dominant.*

The following corollaries can be verified directly:

Corollary B.4.3 *If $\gamma = \delta = \pm\sqrt{2}$, the coefficient matrix A with the following computational stencil*

$$\begin{pmatrix} 1 & -4(2\pm\sqrt{2}) & -(\sqrt{2}\pm 1)^2 \\ -4(2\mp\sqrt{2}) & 36 & -4(2\pm\sqrt{2}) \\ -(\sqrt{2}\mp 1)^2 & -4(2\mp\sqrt{2}) & 1 \end{pmatrix}$$

is symmetrizable by the real diagonal similarity transformation $Q = diag(Q_1, Q_2, \ldots, Q_n)$, where

$$Q_1 = diag[1, \sqrt{2}\mp 1, (\sqrt{2}\mp 1)^2, \ldots, (\sqrt{2}\mp 1)^{n-1}],$$
$$Q_j = (\sqrt{2}\mp 1)Q_{j-1}, \qquad j = 2, 3, \ldots, n.$$

The symmetrized coefficient matrix $\tilde{A}$ has the computational stencil

$$\begin{pmatrix} 1 & -4\sqrt{2} & -1 \\ -4\sqrt{2} & 36 & -4\sqrt{2} \\ -1 & -4\sqrt{2} & 1 \end{pmatrix}$$

Corollary B.4.4 *If* $\gamma = \sqrt{2}$*,* $\delta = -\sqrt{2}$ *or* $\gamma = -\sqrt{2}, \delta = \sqrt{2}$*, the coefficient matrix* A *with the following computational stencil*

$$\begin{pmatrix} -(\sqrt{2} \mp 1)^2 & -4(2 \mp \sqrt{2}) & 1 \\ -4(2 \mp \sqrt{2}) & 36 & -4(2 \pm \sqrt{2}) \\ 1 & -4(2 \pm \sqrt{2}) & -(\sqrt{2} \pm 1)^2 \end{pmatrix}$$

is symmetrizable by the real diagonal similarity transformation $Q = diag(Q_1, Q_2, \dots, Q_n)$*, where*

$$Q_1 = diag[1, \sqrt{2} \mp 1, (\sqrt{2} \mp 1)^2, (\sqrt{2} \mp 1)^3, \dots, (\sqrt{2} \mp 1)^{n-1}],$$

$$Q_j = (\sqrt{2} \pm 1)Q_{j-1}, \qquad j = 2, 3, \dots, n.$$

The symmetrized coefficient matrix $\tilde{A}$ *has the computational stencil*

$$\begin{pmatrix} -1 & -4\sqrt{2} & 1 \\ -4\sqrt{2} & 36 & -4\sqrt{2} \\ 1 & -4\sqrt{2} & -1 \end{pmatrix}$$

Remark B.4.5 *Although the original coefficient matrix* A *is not diagonally dominant when* $|\gamma| = |\delta| = \sqrt{2}$*, the symmetrized coefficient matrix* $\tilde{A}$ *is strictly diagonally dominant.*

B.4.1 A Bound for Line Jacobi Method

Let A be split by the line Jacobi iteration, i.e.

$$A = D - C, \tag{B.24}$$

where D is the diagonal block and $-C$ contains the upper and lower diagonal parts of A. Suppose that A can be symmetrized by a real diagonal similarity transformation Q and the symmetrized matrix is $\tilde{A} = Q^{-1}AQ$. Corresponding to the line Jacobi splitting (B.24), $\tilde{A}$ is split as

$$\tilde{A} = \tilde{D} - \tilde{C}. \tag{B.25}$$

We now derive a bound for the spectral radius of the iteration matrix $M = D^{-1}C$ based on the line Jacobi splitting of the coefficient matrix A, in the case where A is symmetrizable, i.e., when $|\gamma| = |\delta| = \sqrt{2}$. Note that

$$M = Q\tilde{D}^{-1}\tilde{C}Q^{-1},$$

i.e., M is similar to $\tilde{M} = \tilde{D}^{-1}\tilde{C}$ and they have the same eigenvalues. Hence, we can restrict our attention to $\tilde{M}$. The analysis is based on the result

$$\varrho(\tilde{D}^{-1}\tilde{C}) \le \|\tilde{D}^{-1}\tilde{C}\|_2 \le \|\tilde{D}^{-1}\|_2 \|\tilde{C}\|_2 = \frac{\varrho(\tilde{C})}{\lambda_{\min}(\tilde{D})}, \tag{B.26}$$

where the equality follows from the symmetry of $\tilde{D}$ and $\tilde{C}$. $\varrho(\tilde{C})$ is the spectral radius of $\tilde{C}$ and $\lambda_{\min}(\tilde{D})$ is the smallest eigenvalue in absolute value of $\tilde{D}$. For $|\gamma| = |\delta| = \sqrt{2}$, $\tilde{D}$ is symmetric positive definite and can be factored symmetrically as $\tilde{D} = LL^T$. Hence

$$L^T \tilde{D}^{-1} \tilde{C} L^{-T} = L^{-1} \tilde{C} L^{-T}.$$

That is, $\tilde{M}$, and therefore M, are similar to a symmetric matrix, and their eigenvalues are real. The M-matrix $\tilde{D}$ has block diagonal form $\mathrm{diag}(\bar{D}, \ldots, \bar{D})$, where each $\bar{D} = \mathrm{tri}(-4\sqrt{2}, 36, -4\sqrt{2})$. Hence, $\lambda_{\min}(\tilde{D}) = \lambda_{\min}(\bar{D})$.

By Lemma 2 of Elman and Golub [17], the eigenvalues of the tridiagonal matrix $\bar{D}$ of order n are $\{\lambda_j = 4(9 - 2\sqrt{2}\cos j\pi h), j = 1, 2, \ldots, n\}$. Hence,

$$\lambda_{\min}(\tilde{D}) = \lambda_{\min}(\bar{D}) = 4(9 - 2\sqrt{2}\cos \pi h). \tag{B.27}$$

The spectral radius of $\tilde{C}$ is bounded by Gerschgorin's theorem [61]:

$$\varrho(\tilde{C}) \leq 4(1 + 2\sqrt{2}). \tag{B.28}$$

Hence, from (B.26), (B.27) and (B.28) we have the following theorem:

Theorem B.4.6 *If $|\gamma| = |\delta| = \sqrt{2}$, the spectral radius of the line Jacobi iteration matrix for the linear system (1.1) is bounded by*

$$\begin{aligned} \varrho(M) &\leq \frac{1 + 2\sqrt{2}}{9 - 2\sqrt{2}\cos \pi h} \\ &\rightarrow \frac{17 + 20\sqrt{2}}{73} \approx 0.6023, \quad \textit{as} \quad h \rightarrow 0. \end{aligned}$$

B.5 Conclusions

We proved that the point Jacobi and point Gauss-Seidel methods converge for solving the linear system resulted from a fourth-order finite difference discretization of the convection-diffusion equation when the equation is diffusion-dominated. We also proved that the line Jacobi method converges when the coefficient matrix is symmetrizable.

Bibliography

[1] L. Adams and J. Ortega, A multi-color SOR method for parallel computation, in *Proceeding of the 1982 International Conference on Parallel Processing*, (K.E. Batcher, W.C. Meilander and J.L. Potter, Eds.), 1982. pp. 53–56.

[2] I. Altas and K. Burrage, A high accuracy defect-correction multigrid method for the steady incompressible Navier-Stokes equations, *J. Comput. Phys.* **114**, 227–233 (1994).

[3] W. Auzinger and H.J. Stetter, Defect corrections and multigrid iterations, in *Multigrid Methods*, Lecture Notes in Maths., Vol 960, Springer-Verlag, New York, 1981. pp. 327–351.

[4] D. Barkai and A. Brandt, Vectorized multigrid Poisson solver for the CDC CYBER 205, *Appl. Math. Comput.* **13** 215–227 (1983).

[5] A.M. Berger, J.M. Solomon, M. Ciment, S.H. Leventhal and B.C. Weinberg, Generalized OCI schemes for boundary layer problems, *Math. Comput.* **35**, 695–731 (1980).

[6] A. Brandt, Multi-level adaptive solution to boundary-value problems, *Math. Comput.* **31**, 333-396 (1977).

[7] A. Brandt, Multigrid techniques: 1984 guide with applications to fluid dynamics, monograph, GMD-Studien Nr. 85, Postfach 1240, Schloss Birlinghoven, D-5205 St. Augustin 1, Germany, 1984.

[8] A. Brandt and I. Yavneh, On multigrid solution of high-Reynolds incompressible entering flows, *J. Comput. Phys.* **101**, 151–164 (1992).

[9] A. Brandt and I. Yavneh, Accelerated multigrid convergence and high-Reynolds recirculating flows, *SIAM J. Sci. Comput.* **14**, 607–626 (1993).

[10] A. Brandt, Rigorous quantitative analysis of multigrid, I: constant coefficients two-level cycle with L_2-norm, *SIAM J. Numer. Anal.* **31**, 1695–1730 (1994).

[11] A. Brandt and V. Mikulinsky, On recombining iterants in multigrid algorithms and problems with small islands, *SIAM J. Sci. Comput.* **16**, 20–28 (1995).

[12] C. Brezinski and M. Redivo-Zaglia, Hybrid procedures for solving linear systems, *Numer. Math.* **67**, 1–19 (1994).

[13] W. Briggs, *A Multigrid Tutorial*, SIAM, Philadelphia, 1987.

[14] T.F. Chan, L. de Pillis and H. van der Vorst, A transpose-free squared Lanczos algorithm and application to solving non-symmetric linear systems, Technical Report, CAM 91–17, Department of Mathematics, UCLA, September 1991.

[15] T.F. Chan, E. Gallopoulos, V. Simoncini, T. Szeto and C.H. Tong, QMRCGSTAB: a quasi-minimal residual variant of the Bi-CGSTAB algorithm for nonsymmetric linear systems, Technical Report, CAM 92–26, Department of Mathematics, UCLA, June 1992.

[16] S.C.R. Dennis and J.D. Hudson, Compact h^4 finite-difference approximations to operators of Navier-Stokes type, *J. Comput. Phys.* **85**, 390–416 (1989).

[17] H. C. Elman and G. H. Golub, Iterative method for cyclically reduced non-self-adjoint linear systems, *Math. Comput.* **54**, 671–700 (1990).

[18] R. Fletcher, Conjugate gradient methods for indefinite systems, in *Numerical Analysis* (G.A. Watson, Ed.), Dundee 1975, Scotland, Lecture Notes in Math. Vol 506, Springer–Verlag, Berlin, 73–89 (1976).

[19] R.W. Freund and N.M. Nachtigal, QMR: a quasi-minimal residual method for non-Hermitian linear systems, *Numer. Math.* **60**, 315–339 (1991).

[20] R.W. Freund, A transpose-free quasi-minimal residual algorithm for non-Hermitian linear systems, *SIAM J. Sci. Comput.* **14**, 470–482 (1993).

[21] U. Ghia, K.N. Ghia and C.T. Shin, High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method, *J. Comput. Phys.* **48**, 387–411 (1982).

[22] G. Golub and J.E. de Phillis, Towards an effective two-parameter SOR method, in *Iterative Methods for Large Linear Systems,* (D.R. Kincard and L.J. Hayer, Eds.) Academic Press, San Diego, 107–119 (1990).

[23] G. H. Golub and R. S. Tuminaro, Cyclic Reduction/Multigrid, Tech. Rep. NA-92-14, Stanford University, Stanford, CA, 1992.

[24] G. Golub and J.M. Ortega, *Scientific Computing, An Introduction with Parallel Computing,* Academic Press, San Diego, 1993.

[25] M. M. Gupta, R.P. Manohar and J.W. Stephenson, A fourth order, cost effective and stable finite difference scheme for the convection-diffusion equation, in *Numer. Properties & Methodologies in Heat Transfer.* Proc. 2nd National Symp. Hemisphere Pub. Co., Washington, DC, 1983. pp. 201–209.

[26] M. M. Gupta, R.P. Manohar and J.W. Stephenson, A single cell high order scheme for the convection-diffusion equation with variable coefficients, *Int. J. Numer. Methods Fluid.* **4**, 641–651 (1984).

[27] M. M. Gupta, A fourth-order Poisson solver, *J. Comput. Phys.* **55**, 166–172 (1984).

[28] M.M Gupta, High accuracy solutions of incompressible Navier-Stokes equations, *J. Comput. Phys.* **93**, 343–359 (1991).

[29] M.M. Gupta, J. Kouatchou and J. Zhang, Comparison of 2nd and 4th order discretizations for multigrid Poisson solvers, *J. Comput. Phys.* (to appear).

[30] M.M. Gupta, J. Kouatchou and J. Zhang, An accurate and stable multigrid method for convection-diffusion equation, Department of Mathematics, The George Washington University, Washington, DC 20052, (1995) (preprint).

[31] M.M. Gupta, J. Kouatchou and J. Zhang, Preconditioning free multigrid method for convection-diffusion equation with variable coefficients, Department of Mathematics, The George Washington University, Washington, DC 20052, (1995) (preprint).

[32] M.H. Gutknecht, Variants of BICGSTAB for matrices with complex spectrum, Technical Report, 91–15, Interdisziplinäres Projektzentrum für Supercomputing, Eidgenössische Technische Hochschutz, Zürich, August 1991.

[33] W. Hackbusch, On multigrid iterations with defect correction, in *Multigrid Methods,* Lecture Notes in Maths., Vol 960, (W. Hackbusch, Ed.), Springer-Verlag, New York, 1981. pp. 461–473.

[34] W. Hackbusch, *Multi-grid Methods and Applications*, Springer-Verlag, Berlin, 1985.

[35] W. Hackbusch and A. Reusken, Analysis of a damped nonlinear multilevel method, *Numer. Math.* **55**, 225–246 (1989).

[36] W. Hackbusch, *Iterative Methods for Large Sparse Linear Systems of Equations,* Springer-Verlag, New York, 1994.

[37] L.A. Hageman and D.M. Young, *Applied Iterative Methods,* Academic Press, New York, 1981.

[38] T.Y. Hou and B.T.R. Wetton, Stable fourth order stream-function methods for incompressible flows with boundaries, (submitted).

[39] C-C.J. Kuo and B.C. Levy, Two-color Fourier analysis of the multigrid method with red-black Gauss-Seidel smoothing, *Appl. Math. Comput.* **29**, 69–87 (1989).

[40] C. Lanczos, Solution of systems of linear equations by minimized iterations, *J. Res. Nat. Bur. Stand.* **49**, 33–53 (1952).

[41] M. Li, T. Tang and B. Fornberg, A compact fourth-order finite difference scheme for the steady incompressible Navier-Stokes equations, *Int. J. Numer. Methods Fluids,* **20**, 1137–1151 (1995).

[42] S. Mika and P. Vaněk, A modification of the two-level algorithm with overcorrection, *Appl. of Math.* **37**, 13–28 (1992).

[43] S. Moussavi, More about two parameter SOR method, *Rend. Instit. Mat. Univ. Trieste.* **22**, 7–27 (1992).

[44] H. Nishida and N. Satofuka, Higher-order solutions of square driven cavity flow using a variable-order multi-grid method, *Int. J. Numer. Methods Eng.* **34**, 637–653 (1992).

[45] M. Prager, A two parameter iterative method for solving algebraic systems of domain decomposition type, *Appl. of Math.* **38**, 470–478 (1993).

[46] A. Reusken, Steplength optimization and linear multigrid methods, *Numer. Math.* **58**, 819–838 (1991).

[47] A. Reusken, Multigrid with matrix-dependent transfer operators for convection-diffusion problems, in *Multigrid Methods IV,* Proc. of 4th European Multigrid Conference, (P.W. Hemker and P. Wesseling, Eds.), Amsterdam, Birkhauser Verlag, Basel, 1994. pp. 269–280.

[48] A. Reusken, Fourier analysis of a robust multigrid method for convection-diffusion equations, *Numer. Math.* **71**, 365–397 (1995).

[49] P.J. Roache, *Computational Fluid Dynamics,* Hermosa, Albuquerque, NM, 1973.

[50] Y. Saad and M.H. Schultz, A generalized minimal residual method for solving nonsymmetric linear systems, *SIAM J. Sci. Stat. Comput.* **7**, 856–869 (1986).

[51] S. Schaffer, High order multi-grid methods to solve the Poisson equation, in *Proc. NASA-Ames Res. Center Symp. on Multigrid Methods,* Moffett Field, 1981.

[52] S. Schaffer, High order multi-grid methods, *Math. Comput.* **43**, 89–115 (1984).

[53] W. Schönauer, H. Müller and E. Schnepf, Pseudo residual type methods for the iterative solution of large linear systems on vector machines, in *Parallel Computing 85* (M. Feilmeier, J. Joubert and U. Schendel, Eds.) North–Holland, 1986. pp. 193–198.

[54] W. Schönauer, *Scientific Computing on Vector Computers*, North-Holland, Amsterdam, 1987.

[55] Y. Shapira, M. Israeli and A. Sidi, Towards automatic multigrid algorithms for SPD, nonsymmetric and indefinite problems, *SIAM J. Sci. Comput.* **17**, 439–453 (1996).

[56] P. Sonneveld, CGS, a fast Lanczos-type solver for nonsymmetric linear systems, *SIAM J. Sci. Stat. Comput.* **10**, 36–52 (1989).

[57] W.F. Spotz and G.F. Carey, High-order compact scheme for the steady streamfunction vorticity equations, *Int. J. Numer. Methods Eng.* **38**, 3497–3512 (1995).

[58] K. Stüben, Local mode analysis for the solution of elliptic problems by multigrid methods, Internal Report, GMD-IMA, St. Augustin 1, Germany, 1982.

[59] K. Stüben and U. Trottenberg, Multigrid methods: fundamental algorithms, model problem analysis and applications, monograph, GMD-Studien Nr. 96, Postfach 1240, Schloss Birlinghoven, D-5205 St. Augustin 1, Germany, 1984.

[60] M.C. Thompson and J.H. Ferziger, An adaptive multigrid technique for the incompressible Navier-Stokes equations, *J. Comput. Phys.* **82**, 94–121 (1989).

[61] R.S. Varga, *Matrix Iterative Analysis,* Prentice-Hall, New Jersey, 1962.

[62] P. Vaněk, Fast multigrid solver, *Appl. of Math.* **40**, 1–20 (1995).

[63] S.P. Vanka, Block-implicit multigrid solution of Navier-Stokes equations in primitive variables, *J. Comput. Phys.* **65**, 138 (1986).

[64] H.A. van der Vorst, BI-CGSTAB: a fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems, *SIAM J. Sci. Stat. Comput.* **13**, 631–644 (1992).

[65] H.F. Walker, Residual smoothing and peak/plateau behavior in Krylov subspace methods, *Appl. Numer. Math.* **19**, 279–286 (1995).

[66] R. Weiss, *Convergence Behavior of Generalized Conjugate Gradient Methods*, Ph.D. Thesis, University of Karlsrushe, Germany, 1990.

[67] R. Weiss, Relations between smoothing and QMR, Technical Report, Universität Karlsruhe Nerikforschung für Supercomputer, 1993.

[68] P. Wesseling, Multigrid solution of the Navier-Stokes equations in the vorticity-streamfunction formulation, in *Efficient Solutions of Elliptic Systems,* Notes on Numer. Fluid Mechs., Vol 10, (W. Hackbusch, Ed.), Viewege, Braunschweig, 1984. pp. 145–154.

[69] P. Wesseling, *An Introduction to Multigrid Methods,* John Wiley & Sons, Chichester, 1992.

[70] N.G. Wright and P.H. Gaskell, An efficient multigrid approach to solving highly recirculating flows, *Computers & Fluids,* **24**, 63–79 (1995).

[71] I. Yavneh, *Multigrid Techniques for Incompressible Flows,* PhD Thesis, The Weizmann Institute of Science, Rehovot, Israel, 1991.

[72] I. Yavneh, Multigrid smoothing factors for Red-Black Gauss-Seidel applied to a class of elliptic operators, *SIAM J. Numer. Anal.* **32**, 1126–1138 (1995).

[73] I. Yavneh, On Red-Black SOR smoothing in multigrid, *SIAM J. Sci. Comput.* **17**, 180–192 (1996).

[74] P.M. de Zeeuw and E.J. van Asselt, The convergence rate of multigrid algorithms applied to the convection-diffusion equation, *SIAM J. Sci. Stat. Comput.* **6**, 492–503 (1985).

[75] P.M. de Zeeuw, Matrix-dependent prolongations and restrictions in a blackbox multigrid solver, *J. Comput. Appl. Math.* **33**, 1–27 (1990).

[76] S. Zeng and P. Wesseling, Numerical study of a multigrid method with four smoothing methods for the incompressible Navier-Stokes equations in general coordinates, in *Contributions to Multigrid,* (P.W. Hemker and P. Wesseling, Eds.) Stichting Mathematisch Centrum, Amsterdam, 1994. pp. 203–218.

[77] J. Zhang, Acceleration of five-point Red-Black Gauss-Seidel in multigrid for two dimensional Poisson equation, *Appl. Math. Comput.* **80**, 73–93 (1996).

[78] J. Zhang, A cost-effective multigrid projection operator, *J. Comput. Appl. Math.* **76**, 325–333 (1996).

[79] J. Zhang, The best V-cycle algorithm for 2D Poisson equation, in electronic *Multigrid Newsletter,* (C. Douglas, Ed.), Vol. **5** (05) (1995).

[80] J. Zhang, Minimal residual smoothing in multi-level iterative method, *Appl. Math. Comput.* (to appear).

[81] J. Zhang, Accelerated multigrid high accuracy solution of the convection-diffusion equations with high Reynolds number, *Numer. Methods for PDEs.* **13**, 77–92 (1997).

[82] J. Zhang, Multigrid solution of convection-diffusion equations with high-Reynolds number, in *Preliminary Proceedings of the 1996 Copper Mountain Conference on Iterative Methods,* Copper Mountain, Colorado, April 9–13, 1996. pp. 1–9.

[83] J. Zhang, On convergence of iterative methods for a fourth-order discretization scheme, *Appl. Math. Letters* (to appear).

[84] J. Zhang, Residual scaling techniques in multigrid, I: equivalence proof, *Appl. Math. Comput.* (to appear).

[85] J. Zhang, Residual scaling techniques in multigrid, II: practical applications, *Appl. Math. Comput.* (to appear).

[86] L. Zhou and H.F. Walker, Residual smoothing techniques for iterative methods, *SIAM J. Sci. Comput.* **15**, 297–312 (1994).