JOURNAL OF COMPUTATIONAL PHYSICS **114**, 227–233 (1994)

A High Accuracy Defect-Correction Multigrid Method for the Steady Incompressible Navier–Stokes Equations

IRFAN ALTAS

Centre for Image Analysis, School of Information Studies, Charles Sturt University-Riverina, Wagga Wagga, NSW 2650, Australia

AND

KEVIN BURRAGE

Department of Mathematics, University of Queensland, St. Lucia, Queensland 4072, Australia

Received August 2, 1993; revised February 15, 1994

The solution of large sets of equations is required when discrete methods are used to solve fluid flow and heat transfer problems. Although the cost of the solution is often a drawback when the number of equations in the set becomes large, higher order numerical methods can be employed in the discretization of differential equations to decrease the number of equations without losing accuracy. For example, using a fourth-order difference scheme instead of a second-order one would reduce the number of equations by approximately half while preserving the same accuracy. In a recent paper, Gupta has developed a fourth-order compact method for the numerical solution of Navier–Stokes equations. In this paper we propose a defect-correction form of the high order approximations using multigrid techniques. We also derive a fourth-order approximation to the boundary conditions to be consistent with the fourth-order discretization of the underlying differential equations. The convergence analysis will be discussed for the parameterized form of a general second-order correction difference scheme which includes a fourth-order scheme as a special case.

1. INTRODUCTION

Consider the two-dimensional convection–diffusion problem given by

$$\frac{\partial^2 z}{\partial x^2}+\frac{\partial^2 z}{\partial y^2}+p(x, y)\frac{\partial z}{\partial x}+q(x, y)\frac{\partial z}{\partial y}=f(x, y) \qquad (1)$$

wich we will use as the basis for the study of the two-dimensional Navier–Stokes equations for steady flow of an incompressible viscous fluid. The discretized form of these equations often requires the solution of a large system of equations. For such large problems, iterative solvers become attractive for their low storage requirements as long as convergence is guaranteed. The performance of iterative methods is sensitive to the number of equations to be solved, the type of boundary conditions applied and other factors. In particular, if the number of equations or the Reynolds number increases, the rate of convergence of an iterative procedure often deteriorates. Hence, an increase in the number of equations to be solved is associated with a higher cost per iteration, thereby limiting the practical size of the problem that can be solved. Applying a higher order method which decreases the number of equations while preserving a high accuracy can partially alleviate this problem. In particular, a fourth-order discretized difference scheme rather than a second-order scheme can reduce the number of equations by approximately half. Consequently in this paper we consider a fourth-order approach to the solution of the Navier–Stokes problem. Thus a fourth-order difference discretization of (1) may be written in the form (see [5])

$$\sum_{j=1}^{8} c_j z_j - c_0 z_0 = h^2[f_1+f_2+f_3+f_4+8f_0]/2 + h^3[p_0(f_1-f_3)+q_0(f_2-f_4)]/4, \quad (2)$$

where

$$\begin{aligned}
c_1 &= 4+h(4p_0+3p_1+p_2-p_3+p_4)/4 \\
&\quad + h^2(4p_0^2+p_0(p_1-p_3)+q_0(p_2-p_4))/8 \\
c_2 &= 4+h(4q_0+q_1+3q_2+q_3-q_4)/4 \\
&\quad + h^2(4q_0^2+p_0(q_1-q_3)+q_0(q_2-q_4))/8 \\
c_3 &= 4-h(4p_0-p_1+p_2+3p_3+p_4)/4 \\
&\quad + h^2(4p_0^2-p_0(p_1-p_3)-q_0(p_2-p_4))/8
\end{aligned}$$

0021-9991/94 $6.00

$$c_4 = 4 - h(4q_0 + q_1 - q_2 + q_3 + 3q_4)/4$$
$$+ h^2(4q_0^2 - p_0(q_1 - q_3) - q_0(q_2 - q_4))/8$$
$$c_5 = 1 + h(p_0 + q_0)/2 + h(q_1 + p_2 - q_3 - p_4)/8 + h^2 p_0 q_0/4$$
$$c_6 = 1 - h(p_0 - q_0)/2 - h(q_1 + p_2 - p_3 - p_4)/8 - h^2 p_0 q_0/4$$
$$c_7 = 1 - h(p_0 + q_0)/2 + h(q_1 + p_2 - q_3 - p_4)/8 + h^2 p_0 q_0/4$$
$$c_8 = 1 + h(p_0 - q_0)/2 - h(q_1 + p_2 - q_3 - p_4)/8 - h^2 p_0 q_0/4$$
$$c_0 = 20 + h^2(p_0^2 + q_0^2) + h(p_1 - p_3) + h(q_2 - q_4)$$

and where the subscripts are defined as in Fig. 1 and h denotes the discretization step size.

The Navier–Stokes equations for two-dimensional steady fl v of an incompressible viscous fluid are given as

$$\frac{\partial^2 \Psi}{\partial x^2} + \frac{\partial^2 \Psi}{\partial y^2} = -w \tag{3}$$

$$\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} - R\left(u\frac{\partial w}{\partial x} + v\frac{\partial w}{\partial y}\right) = 0 \tag{4}$$

$$u = \frac{\partial \Psi}{\partial y}, \qquad v = -\frac{\partial \Psi}{\partial x} \tag{5}$$

in the stream function vorticity form where Ψ and w represent the stream function and vorticity, respectively, u and v are the velocities in the x and y axes directions, and R is the Reynolds number.

For the Navier–Stokes equations the fourth-order discretization of (3) can be obtained (see [5]) by substituting $z = \Psi, f = -w, p(x, y) = 0$, and $q(x, y) = 0$ in (2) while the fourth-order discretization of (4) can be obtained by substituting $z = w, p(x, y) = -Ru, q(x, y) = -Rv$, and $f = 0$.

Note that u and v are given by (5), and a fourth-order discrete representation of them at the grid point 0 can be obtained from [6] as

$$u_0 = (\Psi_2 - \Psi_4)/3h + (\Psi_5 + \Psi_6 - \Psi_7 - \Psi_8)/12h$$
$$+ h(w_2 - w_4)/12 \tag{6}$$
$$v_0 = (\Psi_3 - \Psi_1)/3h - (\Psi_5 - \Psi_6 - \Psi_7 + \Psi_8)/12h$$
$$+ h(w_3 - w_1)/12. \tag{7}$$

Although a higher order scheme will reduce the size of the ensuing linear system, for two- and three-dimensional problems this size will still be large and it is well known that convergence properties of iterative solvers such as Jacobi or Gauss–Seidel deteriorate as the size of the system increases. One way of overcoming this difficulty is through multigrid techniques (see [3], for example), which rapidly accelerate the convergence of the underlying iterative scheme. However, in this paper we consider a defect-correction multigrid approach. This technique uses only the fourth-order

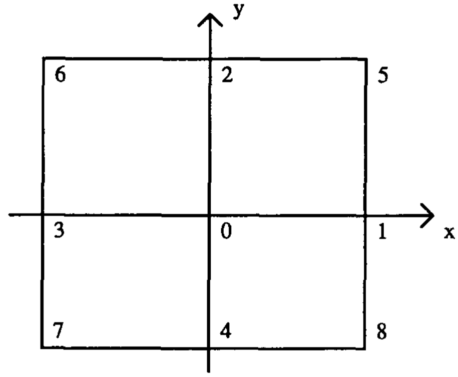

FIGURE 1

discretization for defect evaluation on the finest multigrid level. For the rest of the algorithm it employs a second-order discretization, further improving the overall efficiency of the algorithm. We explain this approach in the next section. This approach would also provide an efficient implementation in a parallel architecture environment since the defect can be evaluated separately from the multigrid process.

In Sections 3 and 4 some numerical comparisons are made between the defect-correction approach and a standard multigrid implementation with a fourth-order discretization. In all cases the defect-correction approach is shown to be more efficient for a given tolerance. In Section 4 a general family of second-order discretizations is suggested and the effect that this parameterized family has on the performance of the defect-correction multigrid technique is investigated numerically. In the Appendix an analysis is given of the convergence behaviour of this parameterized scheme when applied to the Poisson equation on a unit square with Dirichlet boundary conditions. The theoretical results are shown to agree closely with the numerical results of Section 4 for a more general problem.

2. A DEFECT-CORRECTION MULTIGRID APPROACH

Since the work of Hackbush (see [7], for example) in the late 1970s and early 1980s multigrid methods have been widely applied to the numerical solution of differential equations. The multigrid approach is used to accelerate the process of the underlying iterative process by a coarsening process, followed by a refining process which, respectively, halves or doubles the number of grid points in each dimension. The effect of the coarsening process is to turn the smooth eigenvalues of the underlying iterative process into oscillatory ones which are then rapidly damped (see Brand [3] and Briggs [4], for example). Defect-correction in the multigrid context has been studied by several authors [1, 2, 7]. It was demonstrated by these authors that if the basic discretization is of second-order and the target discretization is of fourth-order, then the resulting solutions are fourth-order.

In this paper we develop a defect-correction multigrid algorithm for the Navier–Stokes equations, in which the defect-correction procedure is similar to the one given in [2].

Let $L_2 z = f_2$ and $L_4 z = f_4$ denote the second- and the fourth-order discretizations of a differential equation, respectively, and let z^i be the initial approximation to the solution. Then, the defect-correction multigrid algorithm is as follows:

1. Smooth z^i by relaxation with respect to $L_2 z = f_2$ to obtain z'^i.
2. Compute $d^i = f_4 - L_4 z'^i$.
3. Perform one multigrid cycle for $L_2 z = d^i + L_2 z'^i$ starting with z'^i to obtain z^{i+1}.
4. If $|z^{i+1} - z^i| <$ tolerance, stop; otherwise set $z'^i = z^{i+1}$ and continue with step 2.

The above algorithm is applied to the Navier–Stokes equations given in (3)–(5) by first applying this algorithm to (3) and modifying step 4. Instead of continuing with step 2 from step 4 we evaluate the boundary conditions and the u and v values from (6) and (7). We then apply the algorithm to (4) and update the boundary conditions. This completes one step of an outer iteration procedure. This outer iteration procedure is completed when the tolerance criterion given in the fourth step of the algorithm is satisfied both for w and Ψ.

The order of the algorithm can be verified by applying this approach to the Poisson equation in the unit square,

$$\frac{\partial^2 z}{\partial x^2} + \frac{\partial^2 z}{\partial y^2} = -f(x, y), \tag{8}$$

where the solution z is given by $z = \sin(3x + y)$ and $f(x, y)$ is obtained by differentiation of z, and with boundary values obtained from the exact solution. The results are summarised in Table I, where the error represents the maximum error over all grid points. The ratio of errors is approximately 16 as is to be expected from a fourth-order method.

TABLE I

Mesh size	Error
8×8	3.3×10^{-5}
16×16	2.1×10^{-6}

3. APPLICATION TO NAVIER–STOKES EQUATIONS

As a model problem we select the steady flow of an incompressible viscous fluid in a unit square cavity (as in [5]). The differential equations for this fluid are given in (3)–(5). The boundary condition for (3) has $\Psi = 0$, while for (4) $\Psi_x = 0$ on the left and the right walls. On the top boundary $\Psi_y = -1$, while as on the bottom wall $\Psi_y = 0$. These derivatives can be approximated by the Jensen formula (see [9]). Thus on the sliding wall, $y = 1$,

$$w_2 = (-8\Psi_0 + \Psi_2)/2h^2 + 3/h,$$

while, for $x = 1$

$$w_1 = (-8\Psi_0 + \Psi_1)/2h^2.$$

Similar formulas to w_1 can be found for $y = 0$ and $x = 0$. We note that the subscripts are again written as in Fig. 1. Although the Jensen formulas for the vorticity function are widely used in the literature, including [5], the discretization order of these formulas are only two and may spoil the fourth-order discretization. Thus we derive a fourth-order discretization analogue to the Jensen formulas for the boundary values of (4).

Now along $x = 0$, we have $\Psi = 0$, $\Psi_x = 0$, and the solution to (3), and so we construct a fourth-order polynomial of the form

$$\Psi(x, y) = \sum_{j=0}^{4} \sum_{i=0}^{4-j} a_{ij} x^i y^j \tag{9}$$

satisfying these conditions.

Since both $\Psi = 0$ and $\Psi_x = 0$, necessarily $a_{0j} = a_{1j} = 0$, $j = 0, ..., 4$, and substituting $\Psi(x, y)$ into (3) gives

$$-w = 2a_{20} + 6a_{30}x + 2a_{21}y + 12a_{40}x^2 + 6a_{31}yx + 2a_{22}(x^2 + y^2).$$

By assuming the origin of local coordinates at 3 in Fig. 1 we find

$$-w_3 = 2a_{20}, \qquad -w_0 = 2a_{20} + 6a_{30}h + 12a_{40}h^2 + 2a_{22}h^2.$$

After similar expressions are obtained for the points 1, 4, and 2 and noting that $\Psi_2 + \Psi_4 - 2\Psi_0 = a_{22}$, systems for the unknown a_{20}, a_{30}, a_{21}, a_{40}, a_{31}, a_{22} can be solved to give

$$w_3 = (-6h^2 w_0 + h^2 w_1 - 2\Psi_2 - 2\Psi_4 - 20\Psi_0)/7h^2.$$

A similar analysis gives for the bottom boundary, right boundary, and top boundary, respectively,

$$w_4 = (-6h^2 w_0 + h^2 w_2 - 2\Psi_1 - 2\Psi_3 - 20\Psi_0)/7h^2$$
$$w_1 = (-6h^2 w_0 + h^2 w_3 - 2\Psi_2 - 2\Psi_4 - 20\Psi_0)/7h^2$$
$$w_2 = (24h - 6h^2 w_0 + h^2 w_4 - 2\Psi_1 - 2\Psi_3 - 20\Psi_0)/7h^2.$$

TABLE II

Mesh size	2nd order	4th order
33×33	0.019	0.014
65×65	0.011	0.007

In order to obtain some idea of the error behaviour of the Jensen formulae we solved the model problem described in this section by applying both the second-order Jensen formula and the fourth-order formula given above, on mesh sizes 33×33 and 65×65 with $R = 100$ and a tolerance 10^{-6}. We also computed approximations on a mesh of size 129×129 and used these results to obtain errors in the computed solution on the smaller meshes. The results given in Table II are the difference between the computed results on the 129×129 mesh and smaller meshes at (0.5, 1). As can be seen computation with the fourth-order formulae gives a reasonable improvement in terms of the accuracy of the procedure.

4. EFFICIENCY OF DEFECT-CORRECTION MULTIGRID

In order to investigate the efficiency of the defect-correction multigrid approach the model problem was solved with $R = 100$ on both 33×33 and 65×65 meshes using

(i) a multigrid defect-correction method with a second order central difference discretization in the multigrid steps.

(ii) a multigrid method with a fourth order discretization of the model equations.

In both cases a fourth-order approximation of the boundaries was used and for each method the problem was solved twice with an error convergence tolerance of 10^{-3} and 10^{-6} for both Ψ and w. The implementation was based on a V-cycle with a Gauss–Seidel smoother and for the boundary values of w a damping factor was employed. As an interpolation operator the values at the common mesh points were directly transferred, while the values at the new grid points were obtained by averaging either two or four nearest mesh points. Direct injection of the residual was used as the restriction operator from fine to coarse mesh. In the multigrid implementation v iterations of the smoothing process were performed at each level for the stream function Ψ and m iterations for the vorticity function w. The minimum level of the multigrid procedure as a 17×17 mesh. The elapsed CPU time in minutes on a VAX 6310 for both cases is given in Table III. A perusal of Table III shows that for both tolerances and mesh sizes the defect-correction approach is more efficient. Furthermore, there is a substantial improvement in efficiency to be made (by almost a factor of 2) if the number of smoothing operations at each level for the stream function is increased from one to three. After this observation it is worthwhile to check whether a full multigrid or W-cycle implementation of the multigrid could further improve the convergence rate of the method. The CPU time from the W-cycle and the full multigrid is given in Table IV to compare with the results in Table III. The results in Table IV were obtained with a 33×33 mesh, $v = 1$ and $m = 1$. For the full multigrid we started with a 17×17 mesh and 10^{-2} tolerance for coarser mesh. The minimum level for the full multigrid was a 9×9 mesh.

TABLE III

Mesh size	Tol	Method (i)	Method (ii)	(v, m)
33×33	10^{-3}	4.50	4.95	(1, 4)
65×65	10^{-3}	50.6	64.66	
33×33	10^{-6}	8.76	9.31	
65×65	10^{-6}	126.03	153.93	
33×33	10^{-3}	2.96	3.26	(2, 4)
65×65	10^{-3}	39.78	49.16	
33×33	10^{-6}	5.81	5.93	
65×65	10^{-6}	86.61	95.16	
33×33	10^{-3}	2.46	2.66	(3, 4)
65×65	10^{-3}	35.08	38.86	
33×33	10^{-6}	4.66	4.76	
65×65	10^{-6}	70.7	76.2	

In addition, the defect-correction algorithm given here could also possibly be improved further by searching the effect of employing different lower order discretization techniques in multigrid steps. As an illustration we use the following parameterised discretization of (3):

$$(1-4a)(\Psi_1+\Psi_2+\Psi_3+\Psi_4)-4(1-2a)(\Psi_0) + 2a(\Psi_5+\Psi_6+\Psi_7+\Psi_8) = -h^2[a(w_1+w_2+w_3+w_4)+(1-4a)\,w_0]. \quad (10)$$

In this case it is easy to show that the difference scheme is stable for $0 \leqslant a < \frac{1}{4}$ and gives the second-order discretization of (3) except if $a = \frac{1}{12}$, in which case a fourth-order discretization is obtained. The case $a = 0$ produces the central difference discretization of (3).

In order to test the significance that the parameter a may have in our implementation we implemented method (i) with the parameterized discretization given in (10) for various values of a with the mesh size 65×65, Gauss–Seidel smoothing, parameters ($v = 1$, $m = 1$), and tolerance 10^{-3}. Table V was obtained.

TABLE IV

	W-cycle		Full multigrid	
Tol	Method (i)	Method (ii)	Method (i)	Method (ii)
10^{-3}	2.38	3.50	5.77	5.82
10^{-6}	5.58	8.11	5.88	5.97

TABLE V

a	CPU minutes
0	49.76
0.0625	47.15
0.0833	45.41
0.125	43.48
0.2	38.43
0.22	36.46
0.24	34.85
0.249	33.96

It is difficult to determine without further investigation how much this variance of a with efficiency is problem dependent and what the dependence on v and m is. In this case, however, there seems to be approximately a 33% improvement in efficiency as a ranges from 0 to 0.25. Furthermore, the efficiency appears to be a monotonic decreasing function of a. In the Appendix at the end of this paper we present a convergence analysis of the iterative behaviour of both the Jacobi method and the Gauss–Seidel method for a discretization based on (10).

Another advantage that can be gained from our defect-correction approach arises from the fact that upwind differencing schemes introduce artificial viscosity/diffusion terms into the solution to improve the convergence behaviour of problems with large Reynolds numbers. In some cases these artificial terms in the numerical solution are not desirable, and our fourth-order defect-correction approach eliminates these undesirable cases even if upwind differencing schemes are employed in the defect phase of the defect-correction algorithm, because the overall results depend on the fourth-order scheme employed in the correction phase of the algorithm.

APPENDIX: CONVERGENCE OF A GENERAL SECOND-ORDER SCHEME

If the second-order discretization scheme introduction (10) is applied to the standard Poisson equation on a unit square with Dirichlet boundary conditions a linear system of the form

$$Ax = b \tag{11}$$

is obtained. Here A is a symmetric block tridiagonal matrix which can be represented as (B, T, B), where B and T are tridiagonal matrices of dimension m with

$$\begin{aligned} B &= (2a,\ 1-4a,\ 2a), \\ T &= (1-4a,\ -4(1-2a),\ 1-4a). \end{aligned} \tag{12}$$

In this representation the tridiagonal matrix with g on the diagonal, h on the upper subdiagonal, and f on the lower subdiagonal is written as (f, g, h). Before studying the convergence behaviour of the Jacobi and Gauss–Seidel schemes applied to (11) and (12) the following lemma is presented.

LEMMA 1. *If there exists a non-zero vector u such that $(f, g, h)\, u = 0$, then necessarily $g = -2\sqrt{fh}\cos(k\pi/(m+1))$, $k = 1, ..., m$.*

Proof. The result is a trivial consequence of the well-known fact (see Muir [8], for example) that the eigenvalues of (f, g, h) are given by

$$\lambda_k = g + 2\sqrt{fh}\cos\frac{k\pi}{m+1}, \qquad k = 1, ..., m.$$

We now present the main result of this appendix.

THEOREM 1. *If $\rho_J(a)$ and $\rho_G(a)$ represent the spectral norms of the amplification matrices of, respectively, the Jacobi and Gauss–Seidel methods when applied to* (11) *and* (12), *then*

$$\rho_G(a) - (\rho_J(a))^2 = O(h^4), \qquad h = \left(\frac{\pi}{m+1}\right).$$

Furthermore, the eigenvalues of R_J (the amplification matrix corresponding to the Jacobi method) satisfy

$$\lambda_{jk} = \frac{(4a-1)}{4(1-2a)}(\theta_j + \theta_k) + \frac{a}{2(1-2a)}\theta_j\theta_k,$$

$$j, k = 1, ..., m,$$

$$\theta_j = -2\cos\frac{j\pi}{m+1}, \qquad j = 1, ..., m$$

$$\rho_J(a) = 1 - \frac{1}{2(1-2a)}h^2 + O(h^4).$$

Proof. Consider first the case of the Jacobi method. If λ and $u = (u_1^T, ..., u_m^T)^T$ represent an eigenvalue and corresponding eigenvector of R_J then

$$Cu = 0, \qquad C = A + (\lambda - 1)\, D, \qquad D = \operatorname{diag}(A). \tag{13}$$

But C can be written in block triangular form (B, F, B) and if B is nonsingular and $G = B^{-1}F$, then (13) leads to the recurrence relation

$$\begin{aligned} u_2 &= -Gu_1 \\ u_{m-1} &= -Gu_m \\ u_{j+1} &= -Gu_j - u_{j-1}, \qquad j = 2, ..., m-1. \end{aligned} \tag{14}$$

Let $u_j = p_{j-1}(G)\, u_1$, where $p_j(x)$ is a polynomial of degree j defined by

$$p_1(x) = x, \qquad p_2(x) = x^2 - 1,$$
$$p_j(x) = -x p_{j-1}(x) - p_{j-2}(x), \qquad j = 3, ...,$$

then (14) leads to

$$p_m(G)\, u_1 = 0.$$

Suppose now that u_1 is an eigenvector of G and θ the corresponding eigenvalue then

$$u = (u_1^T, p_1(\theta)\, u_1^T, ..., p_{m-1}(\theta)\, u_1^T)^T, \qquad p_m(\theta) = 0. \quad (15)$$

Using Lemma 1 we see that θ can take on m values given by

$$\theta_j = -2\cos\frac{j\pi}{m+1}, \qquad j = 1, ..., m. \quad (16)$$

But $(F - \theta_j B)\, u_1 = 0$, where $F - \theta_j B$ is the tridiagonal matrix (f, g, f) with $f = 1 - 4a - 2\theta_j a$ and $g = -4\lambda(1-2a) - \theta_j(1-4a)$, and again applying Lemma 1 it is found that

$$-4\lambda(1-2a) - \theta_j(1-4a) = \theta_k(1 - 4a - 2\theta_j a),$$
$$j, k = 1, ..., m,$$

so that the eigenvalues of R_J can be written as

$$\lambda_{jk} = \frac{4a-1}{4(2a-1)}(\theta_j + \theta_k) + \frac{a}{2(1-2a)}\theta_j\theta_k,$$
$$j, k = 1, ..., m. \quad (17)$$

Suppose now that λ and u are, respectively, the eigenvalue and eigenvector of the corresponding iteration matrix R_G associated with the Gauss–Seidel scheme; then

$$Cu = 0, \qquad C = A + (\lambda - 1)(L + D), \qquad A = L + D + U.$$

In this case letting $G = B^{-1}F$ leads to the recurrence relation

$$u_2 = -Gu_1$$
$$u_{m-1} = -Gu_m/\lambda$$
$$u_{j+1} = -Gu_j - \lambda u_{j-1}, \qquad j = 2, ..., m-1.$$

If again $\bar{\theta}$ is an eigenvalue of G and u_1 is the corresponding eigenvector, then an application of Lemma 1 to the matrix $(\lambda, \bar{\theta}, 1)$ gives

$$\bar{\theta}_j = -2\sqrt{\lambda}\cos\frac{j\pi}{m+1}, \qquad j = 1, ..., m. \quad (18)$$

Again $(F - \bar{\theta}_j B)\, u_1 = 0$, where $F - \bar{\theta}_j B$ is the tridiagonal matrix (f, g, h), where $f = \lambda(1-4a) - 2\bar{\theta}_j a$, $g = -4\lambda(1-2a) - \bar{\theta}_j(1-4a)$, $h = 1 - 4a - 2\bar{\theta}_j a$, and an application of Lemma 1 gives

$$g = -2\sqrt{fh}\cos\frac{k\pi}{m+1}, \qquad k = 1, ..., m,$$

or

$$-4t(1-2a) - \theta_j(1-4a)$$
$$= \theta_k\left[\left(\frac{1-4a}{t} - 2\theta_j a\right)(t(1-4a) - 2\theta_j a)\right]^{1/2}, \quad (19)$$

where $t = \sqrt{\lambda}$. Hence t must satisfy the cubic polynomial

$$16t^3(1-2a)^2 + 2t^2\theta_j(1-4a)(4 - 8a + a\theta_k^2)$$
$$+ t((\theta_j^2 - \theta_k^2)(1-4a)^2 - 4a^2\theta_k^2\theta_j^2)$$
$$+ 2\theta_j\theta_k^2 a(1-4a) = 0.$$

Unfortunately, it is not possible to obtain a closed-form expression for the eigenvalues associated with the Gauss–Seidel iteration for all values of a, but writing

$$\theta_j = -2 + j^2h^2 + O(h^4), \qquad h = \frac{\pi}{m+1} \quad (20)$$

an $O(h^4)$ expansion for $\rho_J(a)$ and $\rho_G(a)$ can be found. Substituting (20) into (17) gives for the Jacobi case, after simplification,

$$\lambda_{jk} = 1 - \frac{1}{4(1-2a)}(j^2 + k^2)\, h^2 + O(h^4)$$

and, hence,

$$\rho_J(a) = 1 - \frac{1}{2(1-2a)}h^2 + O(h^4). \quad (21)$$

A similar substitution of (20) into (19) gives for the Gauss–Seidel case, after simplification,

$$\sqrt{\lambda_{jk}} = 1 - \frac{1}{4(1-2a)}(j^2 + k^2)\, h^2 + O(h^4)$$

and, hence,

$$(\rho_J(a))^2 - \rho_G(a)) = O(h^4).$$

As a consequence of Theorem 1 we see that

$$\rho_J(0) \approx 1 - \frac{h^2}{2}, \qquad \rho_J\left(\frac{1}{4}\right) \approx 1 - h^2$$

$$\rho_G(0) \approx 1 - h^2, \qquad \rho_G\left(\frac{1}{4}\right) \approx 1 - 2h^2.$$

Since the scheme (10) is stable for $a \in [0, \frac{1}{4})$ we see that both iteration schemes will take approximately half the number of iterations to obtain a specified tolerance as the parameter a moves from 0 to $\frac{1}{4}$. Although $a = \frac{1}{12}$ leads to a fourth-order discretization scheme this is not an optimal value in terms of convergence. These theoretical results are borne out by the computational results presented in the previous section for the Gauss–Seidel case.

References

1. W. Auzinger, *Numer. Math.* **51**, 199 (1987).
2. W. Auzinger and H. J. Stetter, *Lecture Notes in Mathematics*, Vol. 960 (Springer-Verlag, New York, 1981).
3. K. Brand, *Lecture Notes in Mathematics*, Vol. 960 (Springer-Verlag, New York, 1981).
4. W. L. Briggs, *A Multigrid Tutorial* (SIAM, Lancaster Press, PA, 1988).
5. M. M. Gupta, *J. Comput. Phys.* **93**, 343 (1991).
6. M. M. Gupta, R. Manohar, and J. W. Stephenson, *Int. J. Numer. Methods Fluids* **4**, 641 (1984).
7. W. Hackbush, *Lecture Notes in Mathematics*, Vol. 960 (Springer-Verlag, New York, 1981).
8. T. Muir, *A Treatise on the Theory of Determinants* (Dover, New York, 1928).
9. P. J. Roache, *Computational Fluid Dynamics* (Hermosa, Albuquerque, NM, 1973).