JOURNAL OF COMPUTATIONAL PHYSICS **132,** 123–129 (1997)
ARTICLE NO. CP965627

A Compact Multigrid Solver for Convection-Diffusion Equations

Murli M. Gupta,* Jules Kouatchou, and Jun Zhang

Department of Mathematics, The George Washington University, Washington, DC 20052
E-mail: *mmg@math.gwu.edu

Received April 29, 1996; revised October 30, 1996

We introduce a multigrid algorithm to solve the convection-diffusion equations using a nine-point compact difference scheme. We test the efficiency of the algorithm with various smoothers and intergrid transfer operators. The algorithm displays a grid-independent convergence rate and produces solutions with high accuracy. Numerical results are presented to validate the conclusions. © 1997 Academic Press

1. INTRODUCTION

Consider the two-dimensional convection-diffusion equation

$$\begin{aligned} Lu \equiv u_{xx} + u_{yy} + \sigma u_x + \tau u_y = f(x, y), \qquad (x, y) \in \Omega, \\ u(x, y) = g(x, y), \qquad (x, y) \in \partial\Omega, \end{aligned} \tag{1}$$

which often appears in the description of transport phenomena. The magnitudes of σ and τ determine ratios of convection to diffusion. In many problems of practical interest the convective terms dominate the diffusion and the values of σ and τ are large. Many numerical simulations of (1) become increasingly difficult (converge slowly or even diverge) as the ratio of convection to diffusion increases.

For convection-diffusion equations with constant coefficients (1), Gupta *et al.* [6] proposed a compact fourth-order finite difference scheme which was shown to be both accurate and cost-effective; it is also stable for all values of σ and τ. In [7], this compact nine-point formula ($\mathcal{NPF}$) was extended to solve convection-diffusion equations with variable coefficients. The new scheme also has a truncation error of order h^4 and the resulting systems of linear equations could be solved by classical iterative methods for large values of σ and τ.

In this paper, we present a method that combines multigrid techniques with $\mathcal{NPF}$ to solve the convection-diffusion equation (1). For a wide range of σ and τ, we compare the effectiveness of a number of smoothers for solving the linear systems resulting from the use of $\mathcal{NPF}$. We also compare the choices of the intergrid transfer operators.

In Section 2 we present the $\mathcal{NPF}$ difference scheme to solve the convection-diffusion equation (1), describe its implementation with multigrid, and carry out a Fourier smoothing analysis of the Gauss–Seidel operator. In Section 3 we present numerical experiments that demonstrate the effectiveness and accuracy of the multigrid algorithm. The paper ends with discussion and conclusions.

2. THE MULTIGRID IMPLEMENTATION

2.1. *The Compact Nine-Point Stencil*

Let $h(=1/N)$ be the uniform mesh-size. The finite difference approximation for convection-diffusion equation (1) at a grid point (x, y) which is denoted by x_0 (Fig. 1) involves eight neighboring mesh points at $(x \pm h, y)$, $(x, y \pm h)$, $(x \pm h, y \pm h)$. These points are denoted by x_i, $i = 1, 2, ..., 8$, and the values of a function u at the point i are denoted by u_i. A compact fourth order approximation ($\mathcal{NPF}$) of the convection-diffusion equation (1) is given by [6]

$$\begin{aligned} \sum_{i=0}^{8} \alpha_i u_i = \frac{h^2}{2} [(f_4 + f_3 + f_2 + f_1 + 8f_0) \\ + \gamma(f_1 - f_3) + \delta(f_2 - f_4)], \end{aligned} \tag{2}$$

where u_i and α_i are depicted by the stencil in Fig. 1 and $\gamma = \sigma h/2$ and $\delta = \tau h/2$ are the cell Reynolds numbers.

When the cell Reynolds numbers δ and γ are greater than 1.0, the coefficient matrix is no longer an M-matrix (see [13] for the definition of M-matrix). The numerical experiments conducted in [6] showed that this scheme converges for any values of δ and γ when classical iterative methods are used.

2.2. *The Multigrid Method*

The multigrid method is among the fastest and most efficient iterative algorithms for solving linear systems arising from discretizing elliptic differential equations (see [3, 12]). This method offers convergence rates independent of the grid size and is very effective for solving large scale computation-intensive problems. Structurally, the algo-

0021-9991/97 $25.00

$$\begin{pmatrix} u_6 & u_2 & u_5 \\ u_3 & u_0 & u_1 \\ u_7 & u_4 & u_8 \end{pmatrix} \qquad \begin{pmatrix} 1-\gamma+\delta-\gamma\delta & 4+4\delta+2\delta^2 & 1+\gamma+\delta+\gamma\delta \\ 4-4\gamma+2\gamma^2 & -(20+4\gamma^2+4\delta^2) & 4+4\gamma+2\gamma^2 \\ 1-\gamma-\delta+\gamma\delta & 4-4\delta+2\delta^2 & 1+\gamma-\delta-\gamma\delta \end{pmatrix}$$

FIG. 1. Labeling of grid points and compact nine-point stencil.

rithm iterates on a hierarchy of consecutively coarser and coarser grids until convergence is reached; considerable computational time is saved by doing the major amount of computations on the coarse grids. For more details on the motivation, philosophy, and processes of the multigrid method, one is referred to [3, 5, 11, 14] and the references therein.

We design our nine-point multigrid ($\mathcal{NPF}$-MG) solver as follows:

(1) Start from the fine grid with an initial guess and perform ν_1 $\mathcal{NPF}$ relaxation sweeps.

(2) Calculate the residual on the fine level and project it onto the coarse level.

(3) Perform 2 multigrid cycles on this grid.

(4) Interpolate the coarse grid correction to the fine grid by bi-linear interpolation.

(5) Perform ν_2 $\mathcal{NPF}$ relaxation sweeps on the fine grid.

This corresponds to a *W*-cycle. We use either full-weighting or full-injection for fine to coarse grid transfer operation in Step (2). For relaxation in Steps (1) and (5) we consider the following smoothers: the lexicographical Gauss–Seidel (**L**), red-black Gauss–Seidel (**R**), red-black horizontal line Gauss–Seidel (**Z**), symmetric (forward followed by backward) horizontal line Gauss–Seidel (**S**), alternating (horizontal followed by vertical) zebra (**AZ**), and four-direction (forward and backward horizontal and vertical) Gauss–Seidel (**F**) (see [14, Chap. 4.3] for a description of these smoothers).

2.3. *Special Treatments*

In the context of multigrid, the right-hand side as it appears in (2) is only evaluated once on the finest grid when the initialization of data is performed. We may define

$$F_0 = \tfrac{1}{2}[f_4 + f_3 + f_2 + f_1 + 8f_0 + \gamma(f_1 - f_3) + \delta(f_2 - f_4)], \quad (3)$$

and (2) becomes

$$\sum_{i=0}^{8} \alpha_i u_i = h^2 F_0. \quad (4)$$

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

To efficiently utilize the computational space, practical multigrid solvers use a single long vector to store the discretized values of u and f (F here) for all grid levels [5]. On the coarse grids, u and f locations contain coarse grid corrections and residuals, respectively. The values of F_0 as defined by (3) are only evaluated once on the finest grid when the data are initialized.

For multigrid, as with the classical iterative methods, the computational considerations correspond to finding a relaxation procedure that properly damps the high frequencies. In particular, even if δ and γ are fairly modest on the finest grid, it may still be difficult to find an appropriate smoother on the coarse grid levels due to the fact that the effective values of the cell Reynolds numbers δ and γ become large on coarser grids.

2.4. *Fourier Smoothing Analysis*

For the multigrid implementation of the convection-diffusion equations with constant coefficients, it is natural to use the lexicographical Gauss–Seidel (**L**) smoother. We carry out a simple Fourier analysis to predict the effectiveness of this smoother. The general idea behind smoothing analysis is to determine how well the relaxation procedure damps the high frequency errors. If the coarse grid correction satisfactorily annihilates the low frequencies, then the smoothing analysis will accurately predict the asymptotic multigrid convergence rate [3, 14].

The smoothing number $\mu(\mathcal{NPF})$ of $\mathcal{NPF}$ is given by

$$\mu(\mathcal{NPF}) = \max_{|\theta| \geq \pi/2} \left| \frac{(1-\gamma+\delta-\gamma\delta)e^{i(-\theta_1+\theta_2)} + (4+4\delta+2\delta^2)e^{i\theta_2} + (1+\gamma+\delta+\gamma\delta)e^{i(\theta_1+\theta_2)} + (4-4\gamma+2\gamma^2)e^{-i\theta_1}}{(1+\gamma-\delta-\gamma\delta)e^{i(\theta_1-\theta_2)} + (1-\gamma-\delta+\gamma\delta)e^{-i(\theta_1+\theta_2)} + (4-4\delta+2\delta^2)e^{-i\theta_2} + (4+4\gamma+2\gamma^2)e^{i\theta_1} - (20+4\gamma^2+4\delta^2)} \right|,$$

where $|\theta_1|, |\theta_2| \leq \pi$ and $|\theta| = \max(|\theta_1|, |\theta_2|)$.

We plot the values of $\mu(\mathcal{NPF})$ as a function of δ and γ in Figs. 2, 3 for two cases: (1) when $\delta = 0$ and the convection is parallel to x-axis (this case is called anisotropic) and (2) when $\gamma = \delta$ and the direction of convection

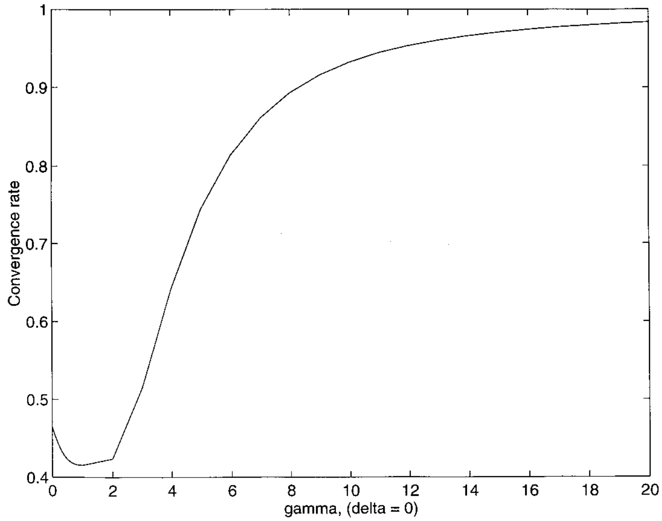

FIG. 2. Smoothing numbers for $\mathcal{NPF}$ lexicographical Gauss–Seidel when $\delta = 0$.

is along the line $y = x$. Though the graphs are plotted for γ values in the interval [0, 20], similar behavior is apparent for larger γ (>1000).

We observe that $\mathcal{NPF}$ yields convergent smoothing rates (i.e., $\mu < 1$) for all values of γ and δ, even when the operator is highly convective. However, the smoothing rate deteriorates (i.e., $\mu \approx 1$) for the first case ($\delta = 0$) for large γ and we expect the multigrid method to converge very slowly in this case. In the second case when γ and δ are equal and large, the value of μ tends to a constant ($\approx$0.73) and we expect the multigrid method to perform well for this case.

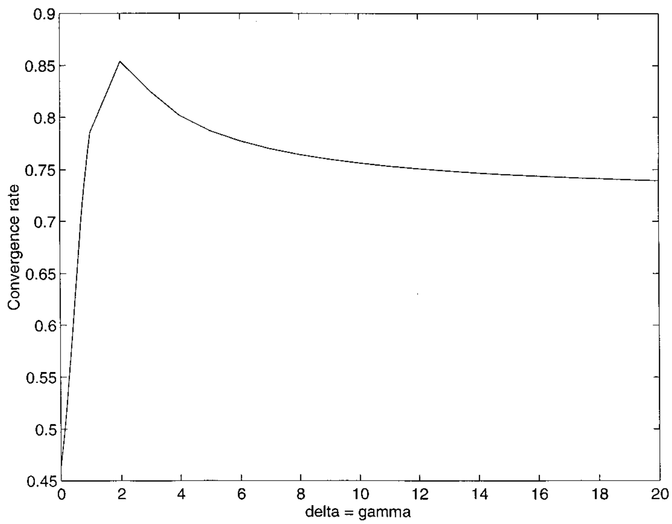

FIG. 3. Smoothing numbers for $\mathcal{NPF}$ lexicographical Gauss–Seidel when $\gamma = \delta$.

TABLE I

Number of Iterations

σ	$N = 16$	$N = 32$	$N = 64$
10000	52	164	350
100000	53	171	533
1000000	53	171	537
10000000	53	171	537

Note. Test Problem 1 with lexicographical Gauss–Seidel and full-weighting, $\tau = 0$.

The Fourier smoothing analysis indicates, as expected, that the lexicographical Gauss–Seidel (**L**) is not effective for the case $\delta = 0$ and some other smoothers must be considered. In the next section we compare the computational performances of various smoothers.

3. COMPUTATIONAL EFFICIENCY

We give convergence data for two problems using $\mathcal{NPF}$ with multigrid ($\mathcal{NPF}$-MG). The test problems were solved on a unit square $[0, 1] \times [0, 1]$ using a uniform meshsize h ($=1/N$). The multigrid W-cycle algorithm was applied and the calculations were done on a Silicon Graphic Indy workstation using FORTRAN 77 programming language in 64-bit precision. We chose the initial guess as $u(x, y) = \sin(x + y)$. Unless otherwise specified, the computation is terminated when the discrete euclidean norm of the initial residual is reduced by $tol = 10^5$. We employ the standard coarsening technique (the mesh-size of the coarse grid doubles that of the fine grid) and the coarsest grid contains only one unknown.

3.1. *Comparison of Smoothers and Intergrid Transfer Operators*

We first compare the performance of six smoothers frequently used to solve partial differential equations.

As *Test Problem 1* we take the boundary value problem (1) with

TABLE II

Number of Iterations

σ	$N = 16$	$N = 32$	$N = 64$
10000	29	55	94
100000	29	51	92
1000000	29	51	88
10000000	29	51	87

Note. Test Problem 1 with lexicographical Gauss–Seidel and full-weighting, $\sigma = \tau$.

TABLE III

Number of Iterations

σ	$N = 16$ Z	F	S	AZ	$N = 32$ Z	F	S	AZ	$N = 64$ Z	F	S	AZ
10000	29	17	16	34	93	51	51	108	201	110	109	224
100000	29	17	16	34	96	53	53	112	307	171	169	356
1000000	29	17	16	34	96	53	53	112	308	172	171	358
10000000	29	17	16	34	96	53	53	112	308	172	171	358

Note. Test Problem 1 with red-black horizontal line (**Z**), four-direction (**F**), symmetric horizontal line Gauss–Seidel (**S**), alternating zebra (**AZ**), and full-weighting, $\tau = 0$.

$$\begin{aligned} f(x, y) &= 2(2 + \sigma(2x - 1))(\cos 2\pi y - 1) \\ &\quad - 4\pi x(x - 1)(2\pi \cos 2\pi y + \tau \sin 2\pi y), \qquad (5) \\ g(x, y) &= 2x(x - 1)(\cos 2\pi y - 1). \end{aligned}$$

This problem was used by Gupta *et al.* [6] to test $\mathcal{NPF}$ in the context of direct methods and classical iterative methods such as the SOR (successive over-relaxation) method.

We solve this problem with $N = 16, 32, 64$ and for large values of σ for the two cases when $\tau = 0$ and $\tau = \sigma$. In Table I and II we present the number of iterations when full-weighting and lexicographical Gauss–Seidel smoothers are employed and observe that $\mathcal{NPF}$-MG converges for any N and any σ. As σ increases, the number of iterations tends to a constant for a given N. The highly convective behavior of the equation does not seem to affect the convergence of $\mathcal{NPF}$. However, as N increases, the number of iterations dramatically increases. Moreover as predicted by the Fourier smoothing analysis, the performance of the lexicographical Gauss-Seidel (**L**) method is better with $\sigma = \tau$ (Table II) than with $\tau = 0$ (Table I).

We also solved this problem with red-black Gauss–Seidel (**R**) and found this method (**R**) to be worse than (**L**) when full-weighting is utilized. For problems with large values of σ and τ, the red-black Gauss–Seidel (**R**) is not robust as discovered by Wesseling [14] for the five-point star formula with dissipation terms. However, as we see at the end of this section, this smoother is very efficient when we change the restriction operator in the $\mathcal{NPF}$-MG algorithm.

For problems with grid aligned convection ($\tau = 0$), robust smoothers are symmetric horizontal line Gauss–Seidel (**S**), alternating zebra (**AZ**), red-black Gauss–Seidel horizontal line (**Z**), and four-direction Gauss–Seidel (**F**) [14]. We now examine how these smoothers affect the convergence of $\mathcal{NPF}$-MG.

In Tables III, IV we give the convergence performance data for these four smoothers for Test Problem 1 using full-weighting and note that all four smoothers give a better convergence than the classical Gauss–Seidel (as seen in Tables I and II). The best results are obtained with the four-direction Gauss–Seidel (**F**) and the symmetric horizontal line Gauss–Seidel (**S**) relaxations.

We performed similar computations for other convection directions; e.g., for $\tau/\sigma = \alpha$ with $\alpha = 5, 1/5$ and discovered that the results are similar to the case when $\tau/\sigma = 1$: the four-direction Gauss–Seidel and the symmetric horizontal line Gauss–Seidel methods are the most efficient smoothers.

TABLE IV

Number of Iterations

σ	$N = 16$ Z	F	S	AZ	$N = 32$ Z	F	S	AZ	$N = 64$ Z	F	S	AZ
10000	22	14	16	21	42	28	31	41	72	46	51	70
100000	22	14	16	21	40	28	30	39	72	51	56	72
1000000	22	14	16	21	39	27	30	39	69	50	54	69
10000000	22	14	16	21	39	27	30	39	69	50	54	69

Note. Test Problem 1 with red-black horizontal line (**Z**), four-direction (**F**), symmetric horizontal line Gauss–Seidel (**S**), alternating zebra (**AZ**), and full-weighting, $\sigma = \tau$.

TABLE V

Computational Cost for the Different Smoothers

L	R	F	Z	S	AZ
1 WS	1 WS	4 WS	2 WS	4 WS	2 WS

Note. Lexicographical Gauss–Seidel (**L**), red-black Gauss–Seidel (**R**), red-black horizontal line (**Z**), symmetric horizontal line Gauss–Seidel (**S**), Alternating Zebra (**AZ**), four-direction Gauss–Seidel (**F**).

We now examine the computational cost of these smoothers. To perform the red-black horizontal line (**Z**) relaxation, the grid points of each line are visited twice before going to the next line. This is in fact a pseudo-line relaxation. A similar technique is used for the symmetric horizontal line (**S**) relaxation. If one work sweep (WS) is defined to be the cost of performing one lexicographical Gauss–Seidel (**L**) sweep, the computational cost for all six smoothers may be summarized in Table V. The four-direction (**F**) and the symmetric horizontal line (**S**) smoothers cost 4 WS; the red-black horizontal (**Z**) and the alternating zebra (**AZ**) cost 2 WS. In Tables VI and VII we give the CPU times in seconds for five smoothers for $N = 64$. We note that the symmetric horizontal line Gauss–Seidel (**S**) gives the best performance for $\tau = 0$, whereas the classical Gauss–Seidel (**L**) has the smallest overall CPU time for $\tau = \sigma$.

As expected, the smoothers with larger WS require fewer number of iterations to converge. However, this does not necessarily translate into smaller overall CPU time.

The performance of multigrid methods depends not only on the smoothing technique but also on the coarse-grid approximations to the fine-grid problem. A proper choice of projection and prolongation operators can often improve the convergence. A good convergence rate with a particular smoother is characterized by the h-independence; i.e., the number of iterations should be constant for any values of h (or N).

TABLE VI

Time in Seconds for $n = 64$

σ	L	Z	F	S	AZ
1000	1.13	1.72	1.46	1.44	1.83
10000	21.9	19.4	17.9	17.4	29.3
100000	34.0	29.7	28.0	27.3	34.9
1000000	33.5	29.9	31.0	27.5	34.8
10000000	33.4	29.9	31.0	27.5	34.8

Note. Test Problem 1 with lexicographical Gauss–Seidel (**L**), red-black horizontal line (**Z**), four direction (**F**), symmetric horizontal line Gauss–Seidel (**S**), alternating Zebra relaxation (**AZ**), and full-weighting, $\tau = 0$.

TABLE VII

Time for $n = 64$

σ	L	Z	F	S	AZ
1000	2.29	2.61	2.13	2.25	2.50
10000	5.84	6.96	7.58	8.20	6.80
100000	5.72	6.93	8.32	9.12	7.08
1000000	5.47	6.70	8.14	8.70	6.67
10000000	5.55	6.60	8.14	8.70	6.67

Note. Test Problem 1 with lexicographical Gauss–Seidel (**L**), red-black horizontal line (**Z**), four direction (**F**), symmetric horizontal line Gauss–Seidel (**S**), alternating Zebra (**AZ**), and full-weighting, $\sigma = \tau$.

The previous computations were done using full-weighting as a projection operator. We also utilized full-injection with injection factor $\beta = 2.0$ (direct injection of the fine grid residuals to the corresponding coarse grid point weighted by the constant β) for the case $\tau/\sigma = 1$ and give the convergence data in Table VIII. It is noted that the use of full-injection significantly reduces (up to 70% for $N = 64$) the number of iterations when the classical Gauss–Seidel (**L**) is used (compare Table VIII with II). Similar reduction is also observed for the four-direction Gauss–Seidel (**F**) (compare Table VIII with IV). The most interesting result is obtained with red-black Gauss–Seidel (**R**): its convergence is faster than that of the four-direction (**F**) smoother even though this smoother requires 4 WS whereas the red-black Gauss-Seidel costs 1 WS. In addition, with red-black Gauss–Seidel, the rate of convergence seems to exhibit h-independence.

For the case $\tau = 0$, the use of full-weighting and full-injection give approximatively the same rate of convergence for all three smoothers. But if we include the CPU time, the full-injection is more cost effective.

When the coefficients σ and τ are small the full-weighting performs better than the full-injection operator (as seen in our other works [9, 10]). The efficiency of full-weighting for diffusion dominated and full-injection for convection

TABLE VIII

Number of Iterations

	$N = 16$			$N = 32$			$N = 64$		
σ	L	F	R	L	F	R	L	F	R
10000	13	9	8	22	15	11	31	21	13
100000	13	9	8	19	13	10	26	20	9
1000000	13	9	8	19	13	9	25	19	9
10000000	13	9	8	19	13	9	25	19	9

Note. Test Problem 1 with lexicographic Gauss–Seidel (**L**), red-black Gauss–Seidel (**R**), four-direction (**F**), and full-injection, $\sigma = \tau$.

TABLE IX

Number of Iterations

	$\alpha = \pi/3$			$\alpha = -\pi/3$			$\alpha = \pi/10$			$\alpha = 4\pi/10$		
$\varepsilon \Downarrow$ $N \Rightarrow$	64	128	256	64	128	256	64	128	256	64	128	256
10^{-6}	9	9	8	10	9	8	20	21	20	20	22	21
10^{-7}	9	9	8	10	9	8	20	21	20	20	22	20
10^{-8}	9	9	8	10	9	8	20	21	20	20	22	20

Note. Test Problem 2 with red-black Gauss–Seidel and full-injection projection.

dominated problems is consistent with the following relationship between the orders of inter-grid operators and the order of the original equation [11],

$$m_P + m_R > 2m,$$

where m_P is the order of the prolongation operator (2 for the bi-linear operator), m_R the order of the restriction (2 for full-weighting and 0 for full-injection), and $2m$ the order of the differential equation (2 when the problem is diffusion dominated and 1 when convection dominated).

3.2. *Dependence on the Convection Direction*

We now consider *Test Problem 2*

$$\varepsilon(u_{xx} + u_{yy}) + \cos \alpha u_x + \sin \alpha u_y = 0, \tag{6}$$

where ε is positive, α is any real number, and u is prescribed to be 0 on the boundary: the exact solution is $u = 0$. Equation (6) reduces to Eq. (1) with $\sigma = \cos \alpha/\varepsilon$ and $\tau = \sin \alpha/\varepsilon$.

In the previous section, we observed that for large values of cell Reynolds numbers γ and δ, the red-black Gauss–Seidel smoother performs well if it is used in conjunction with full-injection. This observation was made when the "angle" of convection was $\pi/4$ ($\sigma = \tau$). We solve Test Problem 2 with small values of ε and different convection directions (choices of α) for $N = 64, 128, 256$. The results summarized in Table IX show that for all cases, $\mathcal{NPF}$-MG converges very rapidly and for a given α, the rate of convergence seems to be independent of N and ε.

3.3. *Computed Accuracy*

We observe that the accuracy of the computed solutions is not affected by the choice of smoothers. In Table X, we give sample accuracy data of $\mathcal{NPF}$ solutions for Test Problem 1 with $\sigma = \tau = 16$. We present the maximum error, the CPU times, and the number of iterations for different values of the mesh size. The errors are seen to decay in accordance with $O(h^4)$ truncation error: the errors decay by a factor of 16 when the mesh size is halved. Similar results are obtained with other test problems and larger values of σ and τ [10].

4. DISCUSSION

We discovered that the red-black Gauss–Seidel method is a good smoother for $\mathcal{NPF}$ with the injection as restriction operator. We took the injection scaling factor β to be 2. A natural choice often is $\beta = 1$ (corresponding to full-injection) and the red-black Gauss–Seidel (**R**) still gives a constant grid-independent rate of convergence though with a higher number of iterations. With the other smoothers (**L**, **F**, **Z**, **AZ**) considered here, the use of full-injection and full-weighting gives either the same results or the results sometimes deteriorate with the injection operator. For red-black Gauss–Seidel, $\beta = 1$ works well when σ and τ are relatively small; $\beta = 2$ is more efficient when σ and τ are large.

One may consider the use of a four-color ordering instead of two-color ordering (red-black Gauss–Seidel). However, our experiments showed that the four-color ordering does not even perform as well as the lexicographical Gauss–Seidel (**L**). In [9] we note that for the Poisson equation the more colors we use, the closer we come to the Jacobi relaxations.

Our numerical experiments show that $\mathcal{NPF}$-MG is stable and achieves high accuracy. Other approaches are

TABLE X

N	IT	CPU (s)	Error
8	9	0.014	5.80(−3)
16	11	0.044	3.65(−4)
32	9	0.136	2.27(−5)
64	7	0.44	1.42(−6)
128	6	1.62	8.91(−8)
256	6	6.91	5.57(−9)
512	6	29.5	3.48(−10)

Note. Number of iterations (IT), CPU time, and accuracy for Test Problem 1 with RB Gauss–Seidel, full-weighting projection, $\sigma = \tau = 16$, $tol = 10^{10}$.

known for solving the convection-diffusion equation (1): the defect-correction method, the introduction of dissipation, and matrix-dependent restriction and prolongation operators.

The dominance of the convection terms in Eq. (1) often causes deterioration in the rate of convergence. In [16] de Zeeuw proposed matrix-dependent restriction and prolongation operators to overcome this difficulty. So far we have not considered the combination of the defect-correction technique [2] or the artificial viscosity method [15] with $\mathscr{NPF}$ in the multigrid process. However, Altas and Burrage [1] have reported successful use of defect-correction with the nine-point compact approximation (2). For the solution of the general convection-diffusion equation our recent work suggests that the convergence of $\mathscr{NPF}$ can be noticeably improved by just choosing an appropriate injection scaling factor for the projection operator (as Brandt and Yavneh did in [4]) or by introducing the minimal residual smoothing method. We are currently working on a dynamic injection (the injection factor computed at each grid point) operator for the solution of the convection-diffusion equations with $\mathscr{NPF}$.

5. CONCLUSION

We have studied the performance of a multigrid method with the compact nine-point finite difference approximation of the convection-diffusion equations with constant coefficients. Numerical experiments were performed for different choices of smoothers and projection operators. The results show that the nine-point scheme displays a constant convergent rate when the cell Reynolds numbers are large. We also discovered that the red-black Gauss–Seidel smoother is efficient and cost effective when it is used in conjunction with full-injection.

REFERENCES

1. I. Altas and K. Burrage, A high accuracy defect-correction multigrid method for the steady incompressible Navier-Stokes equations, *J. Comput. Phys.* **114,** 227 (1994).
2. W. Auzinger and H. J. Stetter, Defect correction and multigrid iterations, in *Lecture Notes in Math.,* Vol. 960 (Springer-Verlag, Berlin/New York, 1982), pp. 327–351.
3. A. Brandt, Multi-level adaptive solution to boundary-value problems, *Math. Comp.* **31,** 333 (1977).
4. A. Brandt and I. Yavneh, Accelerated multigrid convergence and high-Reynolds recirculating flows, *SIAM J. Sci. Comput.* **14,** 607 (1993).
5. W. Briggs, *A Multigrid Tutorial* (SIAM, Philadelphia, 1984).
6. M. M. Gupta, R. Manohar, and J. W. Stephenson, A fourth order, cost effective and stable finite difference scheme for the convection-diffusion equation, in *Numerical Properties & Methodologies in Heat Transfer, Proc. Second National Symposium* (Hemisphere, Washington, DC, 1983), pp. 201–209.
7. 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 Fluids* **4,** 641 (1984).
8. M. M. Gupta, A fourth-order Poisson solver, *J. Comput. Phys.* **55,** 166 (1984).
9. M. M. Gupta, J. Kouatchou, and J. Zhang, Comparison of 2nd and 4th order discretizations for multigrid Poisson solvers, *J. Comput. Phys.,* in press.
10. M. M. Gupta, J. Kouatchou, and J. Zhang, Preconditioning free multigrid method for convection-diffusion equations with variables coefficients, Department of Mathematics, The George Washington University, Washington, DC, 1995 preprint available through anonymous `ftp` at `casper.cs.yale.edu` under the directory `mgnet/papers/Gupta-Kouatchou-Zhang`).
11. W. Hackbusch, *Multigrid Methods and Applications* (Springer-Verlag, Berlin, 1985).
12. K. Stüben and U. Trottenberg, Multigrid methods: Fundamental algorithms, model problem analysis and applications, in *GMD-Studien,* Vol. 96 (Gesellsch. Math. Datenverarbeitung, St. Augustin, 1984).
13. R. S. Varga, *Matrix Iterative Analysis* (Prentice Hall, Englewood Cliffs, NJ, 1962).
14. P. Wesseling, *An Introduction to Multigrid Methods* (Wiley, Chichester, 1992).
15. P. M. de Zeeuw and E. J. van Asselt, The convergence rate of multigrid algorithms applied to the convection-diffusion equation, *SIAM J. Sci. Statist. Comput.* **6,** 492 (1985).
16. P.M. de Zeeuw, Matrix-dependent prolongations and restrictions in a blackbox multigrid solver, *J. Comput. Appl. Math.* **33,** 1 (1990).