JOURNAL OF COMPUTATIONAL PHYSICS **143,** 449–461 (1998)
ARTICLE NO. CP985982

Fast and High Accuracy Multigrid Solution of the Three Dimensional Poisson Equation

Jun Zhang

Department of Computer Science and Engineering, University of Minnesota, 4-192 EE/CS Building,
200 Union Street, S.E., Minneapolis, Minnesota 55455
E-mail: jzhang@cs.umn.edu

Received May 1, 1997; revised March 17, 1998

We employ a fourth-order compact finite difference scheme (FOS) with the multigrid algorithm to solve the three dimensional Poisson equation. We test the influence of different orderings of the grid space and different grid-transfer operators on the convergence and efficiency of our high accuracy algorithm. Fourier smoothing analysis is conducted to show that FOS has a smaller smoothing factor than the traditional second-order central difference scheme (CDS). A new method of Fourier smoothing analysis is proposed for the partially decoupled red-black Gauss–Seidel relaxation with FOS. Numerical results are given to compare the computed accuracy and the computational efficiency of FOS with multigrid against CDS with multigrid. © 1998 Academic Press

1. INTRODUCTION

Numerical simulation of three-dimensional (3D) problems tends to be computationally intensive and may be prohibitive on conventional computers due to the requirements on the memory and CPU time to obtain solution with desirable accuracy. Traditional numerical schemes have low accuracy and thus require fine discretizations. The size of the resulting linear systems is usually so large that even the state-of-the-art computers may not be able to handle them directly. There are at least two ways to alleviate these difficulties: one is to decompose the computational domain into smaller ones so that each subproblem on a subdomain can be solved at a time, or all subprobelms can be solved simultaneously on parallel computers. This is the basic idea of domain decomposition methods. Another approach is to use high-order or spectral methods, which usually yield comparable accuracy with much coarser discretizations, resulting in linear systems of smaller size.

In the two dimensional (2D) case, the high-order multigrid methods for the Poisson equation were studied by Schaffer [10] and a comparison of the second-order scheme and the fourth-order scheme with the multigrid method was recently investigated by Gupta,

0021-9991/98 $25.00

Kouatchou, and Zhang [6]. It was found that the fourth-order scheme is computationally more efficient than the second-order scheme. To obtain a computed solution of given accuracy, the fourth-order scheme may be hundreds of times faster and uses less memory than the second-order scheme.

In this paper, we consider the 3D Poisson equation

$$-\Delta u(x, y, z) = f(x, y, z), \qquad (x, y, z) \in \Omega, \tag{1}$$

where Ω is a continuous convex domain in 3D space with suitable boundary conditions prescribed on its boundary $\partial\Omega$.

The traditional central difference scheme (CDS) for Eq. (1) is a 7-point formula

$$6u_{i,j,k} - (u_{i+1,j,k} + u_{i-1,j,k} + u_{i,j+1,k} + u_{i,j-1,k} + u_{i,j,k+1} + u_{i,j,k-1}) = h^2 f_{i,j,k}, \tag{2}$$

where h is the uniform meshsize. The approximation (2) has a truncation error of $O(h^2)$. A fourth-order compact approximation scheme (FOS) for Eq. (1) was derived by Kwon and Stephenson [9] (see Fig. 1 for the stencil coefficient representation)

$$\begin{aligned} 24u_{i,j,k} &- 2(u_{i+1,j,k} + u_{i-1,j,k} + u_{i,j+1,k} + u_{i,j-1,k} + u_{i,j,k+1} + u_{i,j,k-1}) \\ &- (u_{i+1,j+1,k} + u_{i-1,j+1,k} + u_{i+1,j-1,k} + u_{i-1,j-1,k} + u_{i+1,j,k+1} + u_{i-1,j,k+1} \\ &+ u_{i,j+1,k+1} + u_{i,j-1,k+1} + u_{i+1,j,k-1} + u_{i-1,j,k-1} + u_{i,j+1,k-1} + u_{i,j-1,k-1}) \\ &= \frac{h^2}{2}(6f_{i,j,k} + f_{i+1,j,k} + f_{i-1,j,k} + f_{i,j+1,k} + f_{i,j-1,k} + f_{i,j,k+1} + f_{i,j,k-1}). \end{aligned} \tag{3}$$

This 19-point formula has a truncation error of $O(h^4)$. In fact, fourth-order finite difference methods for 3D general linear elliptic problems with variable coefficients were studied by Ananthakrishnaiah, Manohar, and Stephenson [1], but our attention in this paper is restricted to the Poisson equation.

A natural procedure after the development of high-order schemes is to test if these new schemes fit into some existing popular fast solution methods. In the 3D case, fast iterative methods seem to be necessary due to the large dimension of the problems. A particularly important iterative method for the Poisson equation is the multigrid method, which has been well studied in the 2D case [6] and in the 3D case with CDS [5, 11, 12].

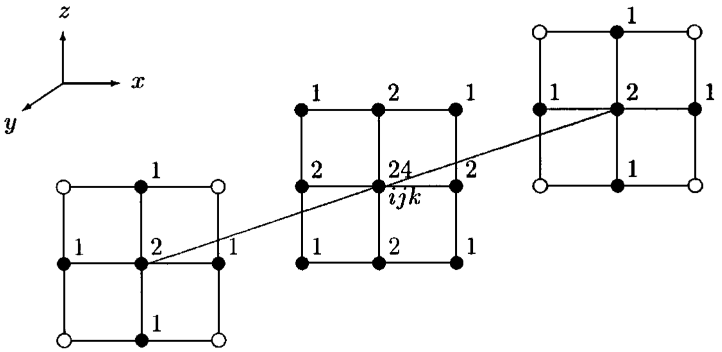

FIG. 1. Stencil coefficients of the 19-point scheme for the 3D Poisson equation.

We investigate the issue of using FOS with the multigrid method for solving Eq. (1) in this paper. We test the influence of the relaxation schemes, different ordering of the grid space, and different grid-transfer operators on the convergence and efficiency of our high accuracy algorithm. We conduct Fourier smoothing analysis to show that FOS has a better smoothing effect than CDS. We also compare our FOS-multigrid method with CDS-multigrid method. A new method of Fourier smoothing analysis is proposed to analyze the partially decoupled red-black Gauss–Seidel relaxation with FOS.

This paper is organized as follows. In Section 2, we briefly introduce the multigrid method and the individual operators that will be used in our numerical experiments. We conduct some Fourier smoothing analysis in Section 3. We give a brief cost analysis in Section 4. Section 5 contains results of our numerical experiments and related remarks. Conclusions are given in Section 6.

2. MULTIGRID METHOD

The multigrid method is one of the most efficient iterative methods to solve a linear system from discretized elliptic differential equations. It tries to solve the problems on a series of coarse grids and interpolates coarse grid correction back to the fine grids. Considerable computer time is saved by doing major computational work on the coarse grids.

One iteration of a simple multigrid V-cycle algorithm consists of smoothing the error using a relaxation technique (e.g., the Jacobi and Gauss–Seidel methods), 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 current 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 the high frequency errors followed by the projection of a residual equation on yet a coarser grid, and so on. Thus, the multigrid method requires a series of problems to be solved on a hierarchy of grids with different meshsizes. A multigrid V-cycle is the process that goes from the finest grid down to the coarsest grid and moves back from the coarsest up to the finest. A V(ν_1, ν_2)-cycle is a multigrid V-cycle algorithm that performs ν_1 relaxations on each level before projecting the residual to the coarse grid (pre-smoothing), and performs ν_2 relaxations after interpolating the coarse grid correction back to the fine grid (post-smoothing). For other multigrid cycling algorithms, we refer to the books of Briggs [4] and Wesseling [15].

The most important operator in the multigrid method is the relaxation operator (the smoother). Its role is not to remove the errors, but to dump the high frequency components of the errors on the current grid while leaving the low frequency components to be removed by the coarser grids. Thole and Trottenberg [11] found that the standard coarsening technique (the coarse grid meshsize doubles that of the fine grid) and (suitable) pointwise relaxation give good smoothing for the isotropic problems. Suitable point relaxations are the lexicographic or red-black Gauss–Seidel methods. The difference of these two methods is the order that the grid points are visited. In the lexicographic Gauss–Seidel (LGS), the grid points are visited in the natural order and the old values are updated as soon as the new values are available. The grid points may also be ordered in a checkerboard fashion in the alternating red and black order. For the 7-point scheme with the red-black Gauss–Seidel (RBGS), the updating of the red points only uses values at the black points, and vice versa. Hence the red and the black points are completely decoupled and the updating of both groups may

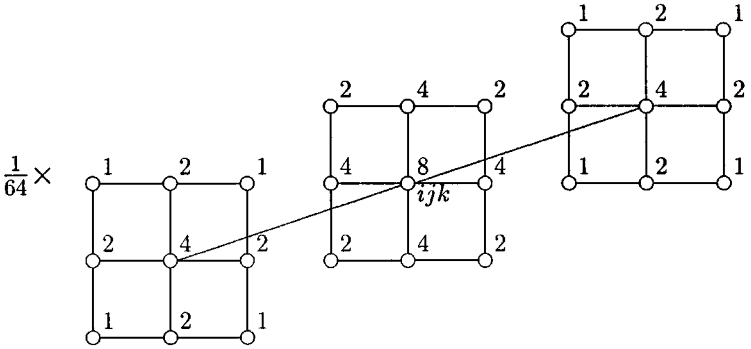

FIG. 2. Stencil for the 3D full-weighting operator.

be done in parallel. This is certainly beneficial on parallel computers. However, it has been found that the red-black ordering also has a better smoothing effect than the natural ordering [14]. When the 19-point scheme is used with Gauss–Seidel, the grid points are not completely decoupled by the red-black ordering. However, both analytical and numerical results [6, 14] suggested that at least in the 2D case, the partially decoupled red-black Gauss–Seidel relaxation with the high-order scheme offers a better smoothing effect than some of the fully decoupled relaxations (with more colors).

The prolongation operator is the tri-linear interpolation. Specifically, the values of the common points of the fine and coarse grids are directly transferred. Depending on the location, values of other fine grid points are obtained by averaging the values of the nearest two, four, or eight points on the coarse grid. For detailed formulas, see Wesseling [15, p. 67].

The residual restriction operator is the direct injection or the full-weighting. The injection operator directly transfers the residual values at the fine grid points to the corresponding coarse grid points. Thus we need only to evaluate the residuals at those grid points, which are about one-eighth of the total fine grid points. On the other hand, the full-weighting operator needs to evaluate residuals at all fine grid points and then projects a weighted average of the nearest 27 points in a cube centered at the reference point. For convenience, the stencil for the full-weighting operator is given in Fig. 2. We remark that the full-weighting operator in its matrix form is a transpose of the tri-linear interpolation operator up to the difference of a constant factor.

The full-weighting operator is more accurate than the injection operator. However, since the cost of the multigrid method using the full-weighting operator is approximately equivalent to one relaxation sweep more than that of the multigrid method using the injection operator, the efficiency of the overall scheme will be affected by the trade-off between more relaxation sweeps and more accurate restriction operators. In the case of the 7-point scheme and on vector computers, these trade-offs were studied by Gary, McCormick, and Sweet [5]. They preferred simpler (cheaper) grid-transfer operators with strong relaxation over the complicated (costly) inter-grid operators.

3. FOURIER SMOOTHING ANALYSIS

The convergence rate of the multigrid method may be predicted by Fourier smoothing analysis.

We assume a domain $\Omega = (0, 1)^3$, periodic boundary conditions, a uniform difference molecule, and row-wise upward point-numbering in each xy plane marching along the z direction. Our methodology and notations are similar to those used by Kettler [8]. For convenience, let

$$Ax = b \tag{4}$$

be the linear system under consideration.

Relaxation on a linear system of the form (4) is given by

$$u_{k+1} = u_k + K^{-1} r_k \qquad (r_k = f - Au_k \text{ is the residual}),$$

where $K = A + N$ is an approximation of A such that $K^{-1} r_k$ is easy to compute. Then, the error after $k + 1$ relaxations $e_{k+1} = u - u_{k+1}$ can be calculated from

$$e_{k+1} = K^{-1} N e_k.$$

We now expand the error in a Fourier series (of eigenfunctions of $K^{-1}N$) as

$$e_k(x, y, z) = \sum_{\theta=(\theta_1,\theta_2,\theta_3)} \varepsilon_\theta^k e^{i(\theta_1 x+\theta_2 y+\theta_3 z)},$$

where $-\pi \leq \theta_1, \theta_2, \theta_3 \leq \pi$ and $i = \sqrt{-1}$. We define the reduction factor of the $\theta = (\theta_1, \theta_2, \theta_3)$ Fourier component (which is only a function of θ and independent of k) by

$$\rho(\theta) = \frac{\varepsilon_\theta^{k+1}}{\varepsilon_\theta^k}.$$

For the problem considered in this paper, K and N are periodic Toeplitz matrices. We denote their coefficients by s_j and v_j, respectively ($j = (j_1, j_2, j_3) \in \mathcal{Z} \times \mathcal{Z} \times \mathcal{Z}$). $\rho(\theta)$ can then be calculated from

$$\rho(\theta) = \frac{\left|\sum_{j\in J_v} v_j e^{i(j\cdot\theta)}\right|}{\left|\sum_{j\in J_s} s_j e^{i(j\cdot\theta)}\right|}. \tag{5}$$

Typically, when the equations are stable, the periodic boundary analysis gives accurate convergence predictions for problems with the Dirichlet boundary conditions.

We define

$$\Theta_H = \left\{(\theta_1, \theta_2, \theta_3) \mid -\pi \leq \theta_1, \theta_2, \theta_3 \leq \pi, |\theta_1| \geq \frac{\pi}{2}, \text{ and/or } |\theta_2| \geq \frac{\pi}{2}, \text{ and/or } |\theta_3| \geq \frac{\pi}{2}\right\},$$

i.e., Θ_H is the region of Fourier components that have a high frequency relative to the meshsize h. The smoothing factor $\bar{\rho}$ is defined as

$$\bar{\rho} = \sup_{\theta\in\Theta_H} \rho(\theta),$$

which tells how well the relaxation method (smoother) damps the high frequency components. The low frequency components are removed on the coarse grids.

3.1. *Smoothing Factors for LGS Relaxation*

For the LGS relaxation, the coefficient matrix is split as $A = D - L - U$, where D is the diagonal of A, $-L$ is the strictly lower triangular, and $-U$ the strictly upper triangular parts of A. We have $K = D - L$ and $N = U$, so that the smoothing factor of the LGS relaxation with CDS is given by

$$\begin{aligned}\bar{\rho}_{\mathrm{CDS}}(\theta) &= \sup_{\theta\in\Theta_H} \frac{|e^{i\theta_1} + e^{i\theta_2} + e^{i\theta_3}|}{|6 - e^{-i\theta_1} - e^{-i\theta_2} - e^{-i\theta_3}|} \\ &= \sup_{\theta\in\Theta_H} \sqrt{\frac{3 + 2[\cos(\theta_1-\theta_2) + \cos(\theta_2-\theta_3) + \cos(\theta_3-\theta_1)]}{39 + 2[\cos(\theta_1-\theta_2) + \cos(\theta_2-\theta_3) + \cos(\theta_3-\theta_1)] - 12(\cos\theta_1 + \cos\theta_2 + \cos\theta_3)}}. \qquad (6)\end{aligned}$$

It is possible to find the exact value of $\bar{\rho}_{\mathrm{CDS}}(\theta)$ by solving the optimization problem (6) analytically. However, the work seems to be much involved, especially when we compute the smoothing factor involving FOS later. Hence, we numerically solved (6) and obtained $\bar{\rho}_{\mathrm{CDS}}(\theta) \approx 0.5669$. Note that the smoothing factor of LGS with CDS in the 2D case is 0.5 [3], hence the 3D multigrid Poisson solver with LGS-CDS converges more slowly than its 2D counterpart.

Similarly, the smoothing factor of the LGS relaxation with FOS is

$$\begin{aligned}\bar{\rho}_{\mathrm{FOS}}(\theta) &= \sup_{\theta\in\Theta_H} \frac{\left|2(e^{i\theta_1} + e^{i\theta_2} + e^{i\theta_3}) + e^{i(-\theta_1+\theta_2)} + e^{i(\theta_1+\theta_2)} + e^{i(-\theta_1+\theta_3)} + e^{i(-\theta_2+\theta_3)} + e^{i(\theta_1+\theta_3)} + e^{i(\theta_2+\theta_3)}\right|}{\left|\begin{array}{c} 24 - 2(e^{-i\theta_1} + e^{-i\theta_2} + e^{-i\theta_3}) \\ -\left(e^{i(-\theta_1-\theta_2)} + e^{i(\theta_1-\theta_2)} + e^{i(-\theta_1-\theta_3)} + e^{i(-\theta_2-\theta_3)} + e^{i(\theta_1-\theta_3)} + e^{i(\theta_2-\theta_3)}\right)\end{array}\right|} \\ &\approx 0.4793. \qquad (7)\end{aligned}$$

We see that $\bar{\rho}_{\mathrm{FOS}}(\theta) < \bar{\rho}_{\mathrm{CDS}}(\theta)$, which means that the multigrid method with FOS is supposed to converge faster than that with CDS.

3.2. *Smoothing Factor for RBGS Relaxation with FOS*

To the best of our knowledge, there has been no published report on how to perform a Fourier smoothing analysis for the 3D 19-point scheme with the partially decoupled RBGS relaxation. Even in the 2D case with a 9-point scheme, some smoothing factors were given by Stüben and Trottenberg [14] without mentioning how to obtain them. The difficulty involving the 19-point RBGS smoothing is that the grid points are only partially decoupled so that the ordinary (lexicographic) Gauss–Seidel and the fully decoupled Gauss–Seidel smoothing analyses are not applicable.

Let the grid space be colored by red and black in a checkerboard fashion as in Fig. 3.

The first half sweep is to update all the red points only. Within all the red points, the relaxation is performed lexicographically. Each red point is updated by using 6 old (un-updated) black points, 6 old (un-updated) red points, and 6 new (updated) red points. We use the idea of the lexicographic Gauss–Seidel relaxation (with the new and old values of the grid points in mind) and compute the reduction factor of the Fourier modes as

$$\rho_{\mathrm{red}}(\theta) = \frac{\left|\begin{array}{c} 2(e^{i\theta_1} + e^{i\theta_2} + e^{i\theta_3} + e^{-i\theta_1} + e^{-i\theta_2} + e^{-i\theta_3}) \\ + e^{i(-\theta_1+\theta_2)} + e^{i(\theta_1+\theta_2)} + e^{i(-\theta_1+\theta_3)} + e^{i(-\theta_2+\theta_3)} + e^{i(\theta_1+\theta_3)} + e^{i(\theta_2+\theta_3)}\end{array}\right|}{\left|24 - \left(e^{i(-\theta_1-\theta_2)} + e^{i(\theta_1-\theta_2)} + e^{i(-\theta_1-\theta_3)} + e^{i(-\theta_2-\theta_3)} + e^{i(\theta_1-\theta_3)} + e^{i(\theta_2-\theta_3)}\right)\right|}. \qquad (8)$$

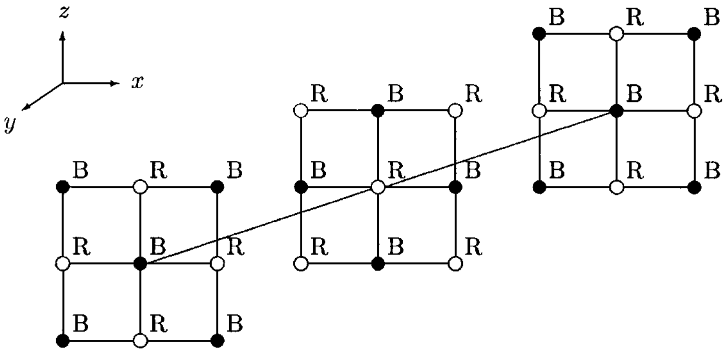

FIG. 3. Red-black ordering of the 3D grid points centered at a red point.

The second half sweep is performed on black points only. Each black point is (in lexicographic order) updated by using 6 new (updated) red points, 6 old (un-updated), and 6 new (updated) black points. (This can be viewed in Fig. 3 by interchanging the positions of the red and black points.) The reduction factor of the Fourier modes by the second half sweep is

$$\rho_{\text{black}}(\theta) = \frac{\left|e^{i(-\theta_1+\theta_2)} + e^{i(\theta_1+\theta_2)} + e^{i(-\theta_1+\theta_3)} + e^{i(-\theta_2+\theta_3)} + e^{i(\theta_1+\theta_3)} + e^{i(\theta_2+\theta_3)}\right|}{\left|\begin{array}{l} 24 - 2(e^{i\theta_1} + e^{i\theta_2} + e^{i\theta_3} + e^{-i\theta_1} + e^{-i\theta_2} + e^{-i\theta_3}) \\ -\left(e^{i(-\theta_1-\theta_2)} + e^{i(\theta_1-\theta_2)} + e^{i(-\theta_1-\theta_3)} + e^{i(-\theta_2-\theta_3)} + e^{i(\theta_1-\theta_3)} + e^{i(\theta_2-\theta_3)}\right)\end{array}\right|}. \quad (9)$$

The smoothing factor of the partially decoupled red-black Gauss–Seidel relaxation with the 19-point scheme is the average of those of the two half sweeps, i.e.,

$$\bar{\rho}_{\text{red-black}}(\theta) = \sup_{\theta\in\Theta_H} \sqrt{\rho_{\text{red}}(\theta)\rho_{\text{black}}(\theta)} \approx 0.3947. \quad (10)$$

Comparing (10) with (7), we conclude that the multigrid method using the 19-point scheme with the RBGS relaxation converges faster than that with the LGS relaxation, given that the other components of the multigrid method are idealized.

We note that the idea that we used to analyze the partially decoupled RBGS relaxation is somewhat analogous to that we used in the heuristic residual analysis to develop some scaled residual injection operator for the 2D RBGS relaxation with the standard and the high-order schemes, see [16, 17]. The essential idea behind these analyses is that both the geometry of the grid space and the relaxation pattern (order) are taken into consideration.

4. COST ANALYSIS

In the context of the multigrid method, the right-hand side of Eq. (3) is only evaluated once. This evaluation needs only to be done once on the finest grid when the initialization of data is performed; we may define the right-hand side of (3) as

$$F_{i,j,k} = \frac{1}{2}(6f_{i,j,k} + f_{i+1,j,k} + f_{i-1,j,k} + f_{i,j+1,k} + f_{i,j-1,k} + f_{i,j,k+1} + f_{i,j,k-1}).$$

The computation of $f_{i,j,k}$ for grid points close to the boundary requires the knowledge of $f(x, y, z)$ on $\partial\Omega$; we assume that $f(x, y, z)$ is extended naturally to $\partial\Omega$.

To access the computer's memory more efficiently, we use a long vector to store both $u_{i,j,k}$ and $f_{i,j,k}$ ($F_{i,j,k}$) for all grids. On the coarse grids, $u_{i,j,k}$ and $f_{i,j,k}$ are the coarse grid correction and residual, respectively. We note that there is no need to store the coefficient matrices for both schemes and thus the storage costs for both schemes are the same. On the other hand, the operation count for one relaxation with CDS is 7; it is 20 with FOS. Thus one relaxation sweep or residual evaluation with FOS is $20/7 \approx 3$ times as expensive as that with CDS.

5. NUMERICAL EXPERIMENTS

We conduct numerical experiments using the multigrid method with different relaxation schemes and residual restriction operators. Our two test problems are chosen the same as those used by Spotz and Carey [13].

For each discretization scheme, we test the LGS and RBGS relaxations as the smoothers. Both cases are tested with the full-weighting operator. We also test LGS with the injection operator. Their performances will be compared.

The problems are discretized on a unit cube $(0, 1)^3$ with different values of meshsize. The multigrid V-cycle algorithm is complete, i.e., the coarsest grid contains only one unknown and one relaxation sweep is used to obtain the solution there. All computations are done on a SUNsparc station using the Fortran 77 programming language in double precision. The computations are terminated when the discrete residual in L_2 norm defined as [18]$\|r\| = (h^3 \sum_{i,j,k} r_{i,j,k}^2)^{1/2}$ is reduced by a factor of 10^{10}. We record the number of iterations and the corresponding CPU time in seconds. The maximum errors are the absolute discrete errors in L_∞ norm over all grid points.

5.1. *Test Problem* 1

The first test problem has zero boundary values on the entire $\partial\Omega$ with a non-zero forcing function

$$u(x, y, z) = \sin(\pi x)\sin(\pi y)\sin(\pi z),$$
$$f(x, y, z) = 3\pi^2 \sin(\pi x)\sin(\pi y)\sin(\pi z).$$

Table 1 contains the test results for Test Problem 1 with one pre-smoothing and one post-

TABLE 1

Test Problem 1 with $\nu_1 = \nu_2 = 1$ Gauss–Seidel Relaxations

	Fourth-order compact scheme						Central difference scheme					
	Injection		Red-black		Weighting		Injection		Red-black		Weighting	
h	Iter.	CPU	Iter.	CPU	Iter.	CPU	Iter.	CPU	Iter.	CPU	Iter.	CPU
1/4	8	0.05	8	0.06	8	0.06	12	0.06	12	0.09	11	0.06
1/8	10	0.20	10	0.29	11	0.28	16	0.23	15	0.31	15	0.24
1/16	12	1.90	12	2.62	12	2.32	19	1.58	16	1.83	18	1.99
1/32	13	15.47	12	20.01	13	20.72	24	15.92	17	14.98	19	16.24
1/64	13	140.60	12	176.90	13	192.80	33	192.50	17	126.00	19	141.60

Note. The number of iterations and the CPU time in seconds.

TABLE 2
Test Problem 1 with $\nu_1 = 2$, $\nu_2 = 1$ Gauss–Seidel Relaxations

	Fourth-order compact scheme						Central difference scheme					
	Injection		Red-black		Weighting		Injection		Red-black		Weighting	
h	Iter.	CPU	Iter.	CPU	Iter.	CPU	Iter.	CPU	Iter.	CPU	Iter.	CPU
1/4	6	0.05	6	0.06	6	0.06	8	0.05	9	0.06	9	0.06
1/8	8	0.25	8	0.29	9	0.28	12	0.24	12	0.25	13	0.25
1/16	9	1.83	9	2.41	10	2.52	13	1.48	13	1.75	14	1.85
1/32	9	15.55	9	20.35	11	24.05	14	13.78	14	16.00	15	16.72
1/64	9	140.50	9	178.20	11	219.00	14	113.10	14	141.00	15	147.80

Note. The number of iterations and the CPU time in seconds.

smoothing sweep on each level. Hereinafter, in all tables, the column title *Injection* refers to the LGS relaxation with the injection operator; *Red-black* refers to RBGS with the full-weighting operator; *Weighting* refers LGS with the full-weighting operator.

We note that, with the same operators, FOS converges faster than CDS. FOS is more robust than CDS with respect to the variation of the component operators employed in the multigrid method. For example, LGS with the injection is satisfactory for FOS, but totally unsatisfactory for CDS. Due to the high cost of the full-weighting, we find that LGS with the injection, although it converges slightly slower, takes less CPU time and thus is more cost-effective than RBGS with the full-weighting. However, so far as the rate of convergence is concerned, RBGS with the full-weighting is the winner. These numerical results are in good agreement with the Fourier smoothing analysis results.

Tables 2 and 3 show that additional relaxation sweeps are beneficial for the injection operator, but seem not beneficial for the red-black ordering as the CPU times were usually increased. This can be explained as the increase of the smoothing sweeps helps smooth the residual and makes the weighting scheme unnecessary. In light of the high cost of the full-weighting, it is more efficient to use LGS with the injection operator.

Table 4 lists the maximum errors for FOS and CDS with different meshsizes. It is clear that FOS achieves the fourth-order convergence rate as the errors are decreased by a factor

TABLE 3
Test Problem 1 with $\nu_1 = \nu_2 = 2$ Gauss–Seidel Relaxations

	Fourth-order compact scheme						Central difference scheme					
	Injection		Red-black		Weighting		Injection		Red-black		Weighting	
h	Iter.	CPU	Iter.	CPU	Iter.	CPU	Iter.	CPU	Iter.	CPU	Iter.	CPU
1/4	5	0.05	5	0.06	5	0.06	7	0.04	7	0.05	7	0.05
1/8	6	0.20	8	0.34	8	0.35	9	0.19	10	0.23	11	0.24
1/16	7	1.85	8	2.67	9	2.79	10	1.46	11	1.86	12	1.93
1/32	7	15.60	8	22.13	9	23.81	10	12.11	11	15.74	12	16.49
1/64	7	143.60	8	195.70	9	220.90	10	104.20	11	140.10	13	158.70

Note. The number of iterations and the CPU time in seconds.

TABLE 4
Maximum Errors of the Computed Solution by the Fourth-Order Compact and the Central Difference Schemes for Test Problem 1

h	Fourth-order scheme	Central difference scheme
1/4	3.93(−3)	5.30(−2)
1/8	2.35(−4)	1.30(−2)
1/16	1.43(−5)	3.22(−3)
1/32	9.04(−7)	8.04(−4)
1/64	5.65(−8)	2.01(−4)

of 16 when the meshsize is halved. CDS achieves the second-order convergence rate as the errors are decreased by a factor of 4 when the meshsize is halved. We note that FOS yields a solution which is much more accurate than that by CDS. For example, to yield a solution with maximum error around 2.50(−4), CDS needs a meshsize $h = 1/64$; but FOS only needs a meshsize $h = 1/8$. If this fact is translated into CPU time, we see that FOS is hundreds of times faster than CDS to yield a computed solution of comparable accuracy.

5.2. *Test Problem 2*

For our second test problem, we consider Eq. (1) with $f = 0$ and the boundary conditions

$$\begin{aligned} u(x, y, z) &= \sin(\pi y)\sin(\pi z), && x = 0, \\ u(x, y, z) &= 2\sin(\pi y)\sin(\pi z), && x = 1, \\ u(x, y, z) &= 0, && y, z = \{0, 1\}. \end{aligned}$$

The exact solution is

$$u(x, y, z) = -\frac{\sin(\pi y)\sin(\pi z)}{\sinh(\pi\sqrt{2})}[2\sinh(\pi\sqrt{2}x) + \sinh(\pi\sqrt{2}(1 - x))].$$

Tables 5–8 contain similar information for Test Problem 2 corresponding to Tables 1–4 for Test Problem 1. Basically, LGS with an injection operator is again shown to be most

TABLE 5
Test Problem 2 with $\nu_1 = \nu_2 = 1$ Gauss–Seidel Relaxations

	Fourth-order compact scheme						Central difference scheme					
	Injection		Red-black		Weighting		Injection		Red-black		Weighting	
h	Iter.	CPU	Iter.	CPU	Iter.	CPU	Iter.	CPU	Iter.	CPU	Iter.	CPU
1/4	8	0.05	8	0.06	8	0.06	12	0.07	12	0.08	11	0.07
1/8	11	0.22	10	0.29	11	0.28	16	0.21	14	0.22	16	0.26
1/16	12	1.71	11	2.26	12	2.31	19	1.57	15	1.52	17	1.77
1/32	13	15.46	11	18.33	12	19.10	21	14.11	16	13.62	18	15.35
1/64	13	141.00	11	163.10	12	178.20	25	140.60	17	124.30	18	133.90

Note. The number of iterations and the CPU time in seconds.

TABLE 6
Test Problem 2 with $\nu_1 = 2, \nu_2 = 1$ Gauss–Seidel Relaxations

	Fourth-order compact scheme						Central difference scheme					
	Injection		Red-black		Weighting		Injection		Red-black		Weighting	
h	Iter.	CPU	Iter.	CPU	Iter.	CPU	Iter.	CPU	Iter.	CPU	Iter.	CPU
1/4	6	0.05	6	0.06	6	0.06	8	0.05	9	0.06	9	0.06
1/8	8	0.25	9	0.31	9	0.30	11	0.20	12	0.23	12	0.23
1/16	9	1.83	9	2.42	10	2.55	13	1.63	12	1.61	13	1.74
1/32	9	15.55	9	19.94	10	21.40	14	13.13	12	13.55	13	14.63
1/64	9	140.50	9	178.10	10	197.20	14	111.80	12	120.20	13	126.30

Note. The number of iterations and the CPU time in seconds.

efficient and the multigrid method with FOS is much faster than that with CDS for obtaining a solution of comparable accuracy.

6. CONCLUSIONS

We have shown that the fourth-order compact finite difference scheme (FOS), when combined with the multigrid techniques, yields a fast and high accuracy 3D Poisson solver. Fourier smoothing analysis unveiled that FOS is more efficient than CDS in damping the high frequency errors and is thus a more powerful smoother. We also proposed a new method of Fourier smoothing analysis to study the partially decoupled red-black Gauss–Seidel relaxation with the 19-point scheme. Our numerical results showed that FOS is much more cost-effective than the central difference scheme (CDS) to produce solution with comparable accuracy. The numerical results are in good agreement with the Fourier smoothing analysis results.

For the same meshsize, FOS uses the same storage space as CDS does. On the other hand, due to the perfect symmetry, FOS using a cheap residual injection operator yields satisfactory performance. In 3D computations, both the memory requirement and CPU time may become challenging for many computers; employment of high-order schemes is one of the promising ways to overcome these challenges.

TABLE 7
Test Problem 2 with $\nu_1 = \nu_2 = 2$ Gauss–Seidel Relaxations

	Fourth-order compact scheme						Central difference scheme					
	Injection		Red-black		Weighting		Injection		Red-black		Weighting	
h	Iter.	CPU	Iter.	CPU	Iter.	CPU	Iter.	CPU	Iter.	CPU	Iter.	CPU
1/4	5	0.05	5	0.06	4	0.05	6	0.04	7	0.05	7	0.05
1/8	6	0.20	7	0.35	8	0.31	9	0.19	10	0.23	11	0.25
1/16	7	1.95	8	2.70	8	2.49	10	1.43	10	1.65	11	1.79
1/32	7	15.50	8	22.71	8	21.17	10	12.09	10	14.23	11	15.43
1/64	7	144.60	8	204.50	8	196.30	10	104.50	10	129.80	11	133.40

Note. The number of iterations and the CPU time in seconds.

TABLE 8
Maximum Errors of the Computed Solution by the Fourth-Order Compact and the Central Difference Schemes for Test Problem 2

h	Fourth-order scheme	Central difference scheme
1/4	5.14(−3)	5.84(−2)
1/8	3.24(−4)	1.55(−2)
1/16	2.04(−5)	3.94(−3)
1/32	1.27(−6)	9.91(−4)
1/64	7.97(−8)	2.48(−4)

Our conclusion is that the red-black Gauss–Seidel relaxation with the full-weighting provides fast convergence, but the lexicographic Gauss–Seidel relaxation with the injection achieves best efficiency. It may be possible to find a scaled injection operator for the red-black Gauss–Seidel relaxation so that we can combine best convergence and efficiency, as we did in the 2D case [16, 17].

REFERENCES

1. U. Ananthakrishnaiah, R. Manohar, and J. W. Stephenson, Fourth-order finite difference methods for three-dimensional general linear elliptic problems with variable coefficients, *Numer. Methods Partial Differential Equations* **3**, 229 (1987).
2. A. Behie and P. A. Forsyth, Jr., Multi-grid solution of three dimensional problems with discontinuous coefficients, *Appl. Math. Comput.* **13**, 229 (1983).
3. A. Brandt, Multi-level adaptive solution to boundary-value problems, *Math. Comp.* **31**, 333 (1977).
4. W. L. Briggs, *A Multigrid Tutorial* (SIAM, Philadelphia, 1987).
5. J. Gary, S. McCormick, and R. Sweet, Successive overrelaxation, multigrid, and pre-conditioned conjugate gradients algorithms for solving a diffusion problem on a vector computer, *Appl. Math. Comput.* **13**, 285 (1983).
6. M. M. Gupta, J. Kouatchou, and J. Zhang, Comparison of second and fourth order discretizations for multigrid Poisson solver, *J. Comput. Phys.* **132**, 226 (1997).
7. W. H. Holter, A vectorized multigrid solver for the three-dimensional Poisson equation, in *Supercomputer Applications*, edited by A. H. L. Emmen (North-Holland, Amsterdam, 1985), p. 17.
8. R. Kettler, Analysis and comparison of relaxation scheme in robust multigrid and preconditioned conjugate gradient methods, in *Multigrid Methods*, edited by W. Hackbusch and U. Trottenberg, Lecture Notes in Math. (Springer-Verlag, Berlin, 1982), Vol. 960, p. 502.
9. Y. Kwon and J. W. Stephenson, Single cell finite difference approximations for Poisson's equation in three variables, *Appl. Math. Notes* **2**, 13 (1982).
10. S. Schaffer, High order multi-grid methods, *Math. Comp.* **43**, 89 (1984).
11. C.-A. Thole and U. Trottenberg, Basic smoothing procedures for the multigrid treatment of elliptic 3D-operator, in *Advances in Multi-Grid Methods*, edited by D. Braess, W. Hackbusch, and U. Trottenberg, Notes on Numerical Fluid Mechanics (Vieweg, Braunschweig, 1985), Vol. 11, p. 102.
12. C.-A. Thole and U. Trottenberg, A short note on standard parallel multigrid algorithms for 3D problems, *Appl. Math. Comput.* **27**, 101 (1988).
13. W. F. Spotz and G. F. Carey, A high-order compact formulation for the 3D Poisson equation, *Numer. Methods Partial Differential Equations* **12**, 235 (1996).
14. K. Stüben and U. Trottenberg, Multigrid methods: Fundamental algorithm, model problem analysis and applications, in *Multigrid Methods*, edited by W. Hackbusch and U. Trottenberg, Lecture Notes in Math. (Springer-Verlag, Berlin, 1981), Vol. 960, p. 1.

15. P. Wesseling, *An Introduction to Multigrid Methods* (Wiley, New York, 1992).

16. J. Zhang, A cost-effective multigrid projection operator, *J. Comput. Appl. Math.* **76**, 325 (1996).

17. J. Zhang, Residual scaling techniques in multigrid. II. Practical applications, *Appl. Math. Comput.* **90**, 229 (1998).

18. M. Zlámal, Superconvergence and reduced integration in the finite element method, *Math. Comp.* **32**, 663 (1978).