Journal of Computational Physics 228 (2009) 137–146

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

Sixth order compact scheme combined with multigrid method and extrapolation technique for 2D poisson equation ☆

Yin Wang, Jun Zhang *

Laboratory for High Performance Scientific Computing and Computer Simulation, Department of Computer Science, University of Kentucky, Lexington, KY 40506-0046, USA

ARTICLE INFO

Article history:
Received 12 March 2008
Received in revised form 29 August 2008
Accepted 2 September 2008
Available online 12 September 2008

MSC:
65N06
65N55
65F10

Keywords:
Poisson equation
Compact difference scheme
Multigrid method
Richardson extrapolation

ABSTRACT

We develop a sixth order finite difference discretization strategy to solve the two dimensional Poisson equation, which is based on the fourth order compact discretization, multigrid method, Richardson extrapolation technique, and an operator based interpolation scheme. We use multigrid V-Cycle procedure to build our multiscale multigrid algorithm, which is similar to the full multigrid method (FMG). The multigrid computation yields fourth order accurate solution on both the fine grid and the coarse grid. A sixth order accurate coarse grid solution is computed by using the Richardson extrapolation technique. Then we apply our operator based interpolation scheme to compute sixth order accurate solution on the fine grid. Numerical experiments are conducted to show the solution accuracy and the computational efficiency of our new method, compared to Sun–Zhang's sixth order Richardson extrapolation compact (REC) discretization strategy using Alternating Direction Implicit (ADI) method and the standard fourth order compact difference (FOC) scheme using a multigrid method.

1. Introduction

Poisson equation is a partial differential equation (PDE) with broad applications in mechanical engineering, theoretical physics and other fields. The two dimensional (2D) Poisson equation can be written in the form of

$$u_{xx}(x,y) + u_{yy}(x,y) = f(x,y), \quad (x,y) \in \Omega, \tag{1}$$

where Ω is a rectangular domain, or a union of rectangular domains, with suitable boundary conditions defined on $\partial\Omega$. The solution $u(x,y)$ and the forcing function $f(x,y)$ are assumed to be sufficiently smooth and have the necessary continuous partial derivatives up to certain orders.

A second order accurate solution can be computed by applying the standard second order central difference operators to $u_{xx}(x,y)$ and $u_{yy}(x,y)$ in Eq. (1). Higher order (more than two) accurate discretization methods need more complex procedure than the second order accurate discretization method to compute the coefficient matrix, but they usually generate linear

☆ This author's research work was supported in part by NSF under Grants CCF-0727600 and CCF-0527967, in part by the Kentucky Science and Engineering Foundation under Grant KSEF-148-502-06-186, in part by the Alzheimer's Association under Grant NIGR-06-25460, and in part by NIH under Grant 1 R01 HL086644-01.

* Corresponding author. Tel.: +1 859 257 3892; fax: +1 859 323 1971.
E-mail addresses: ywangf@csr.uky.edu (Y. Wang), jzhang@cs.uky.edu (J. Zhang).
URLs: http://www.csr.uky.edu/~ywangf (Y. Wang), http://www.cs.uky.edu/~jzhang (J. Zhang).

0021-9991/$ - see front matter
doi:10.1016/j.jcp.2008.09.002

systems of much smaller size, compared with that from the lower order accurate discretization methods [1,7,10]. There has been growing interest in developing higher order accurate discretization methods, especially the high order compact difference schemes, to solve partial differential equations (PDEs) [11,15,18,23,25,26]. We call them "compact" because these schemes only use the minimum three grid points in one dimension in the discretization formulas.

Previously, Chu and Fan [5,6] proposed a three point combined compact difference (CCD) scheme for solving two dimensional Stommel Ocean model, which is a special two dimensional convection–diffusion equation. They used Hermitian polynomial approximation to achieve sixth order accuracy for the inner grid points and fifth order accuracy for the boundary grid points. The advantage of the CCD scheme is that it can be used to solve many types of PDEs without major modifications. And the Alternating Direction Implicit (ADI) [14] method can be used to reduce the higher dimensional problems to a series of lower dimensional problems. So, their scheme is referred to as the implicit high order compact scheme because they do not compute the solution of the dependent variables of the PDEs directly. Instead, the first derivative and the second derivative of the dependent variables are computed at the same time.

In contrary, the explicit fourth order compact schemes [9,10,12,13,18] compute the solution of the variables directly, no redundant computation is needed. Some accelerating iterative methods like multigrid method and preconditioned iterative method have been used to efficiently solve the resulting sparse linear systems arising from the high order compact finite difference discretizations [22,24,25]. But the higher order explicit compact schemes are more complicated to develop in higher dimensions [8,27], compared with the implicit compact schemes. As far as we know, there is no existing explicit compact scheme on a single scale grid that is higher than the fourth order accuracy.

Since a sixth order explicit compact scheme may be impossible to develop on a single scale grid, the multiscale grid method has been considered to achieve the sixth order accuracy for the explicit compact formulations. Sun and Zhang [20] first proposed a sixth order explicit finite difference discretization strategy for solving the 2D convection–diffusion equation. They used ADI method to compute the fourth order accurate solution on the fine and the coarse grids first, then apply the Richardson extrapolation technique and an operator based interpolation scheme in each ADI iteration to achieve the sixth order accurate solution on the fine grid. The major disadvantage of Sun–Zhang's method is that the ADI iteration is not scalable with respect to the meshsize. When the mesh becomes finer, the number of ADI iterations needed for convergence increases quickly.

By using the idea of two scale grid computation from Sun–Zhang's method, we intend to develop a new explicit sixth order compact computing strategy for the 2D Poisson equation, which can efficiently solve the resulting linear system and is scalable with respect to the problem size. We do not use the ADI method, instead, we develop a multigrid method that is similar to the full multigrid method as our convergence acceleration method. With point Gauss–Seidel relaxation method and line Gauss–Seidel relaxation method, we iteratively solve the resulting sparse linear system to get the fourth order accurate solutions on both the fine and the coarse grids. Then we apply the Richardson extrapolation technique combined with our new operator based interpolation scheme to compute the sixth order accurate solution on the fine grid.

In this paper, we present the sixth order compact difference discretization strategy for the 2D Poisson equation in Section 2. In Section 3, we develop our modified multigrid method to solve the fourth order accurate solution on the fine and the coarse grids. Section 4 contains the numerical experiments to demonstrate the high accuracy of the sixth order compact difference scheme, as well as the computational efficiency of our modified multigrid method. Concluding remarks are given in Section 5.

2. Sixth order compact approximations

Our explicit sixth order compact difference scheme is based on the fourth order compact discretization on the two scale grids. In this section, we first introduce the fourth order compact difference scheme for the 2D Poisson equation. The basic idea is from Zhang's previous papers [20,25,28]. More detailed discussions about the fourth order compact difference schemes can be found in [9,17].

In order to discretize Eq. (1), let us consider a rectangular domain $\Omega = [0, L_x] \times [0, L_y]$. We discretize Ω with uniform meshsizes $\Delta x = L_x/N_x$ and $\Delta y = L_y/N_y$ in the x and y coordinate directions, respectively. Here N_x and N_y are the number of uniform intervals in the x and y coordinate directions. The mesh points are (x_i, y_j) with $x_i = i\Delta x$ and $y_j = j\Delta y$, $0 \leqslant i \leqslant N_x$, $0 \leqslant j \leqslant N_y$.

We write the standard second order central difference operators as

$$\delta_x^2 u_{i,j} = \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{\Delta x^2}, \quad \delta_y^2 u_{i,j} = \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{\Delta y^2}.$$

Using Taylor series expansions, at the grid point (x_i, y_j), we have

$$\delta_x^2 u_{i,j} = u_{xx} + \frac{\Delta x^2}{12} u_x^4 + \frac{\Delta x^4}{360} u_x^6 + O(\Delta x^6), \tag{2}$$

and

$$\delta_y^2 u_{i,j} = u_{yy} + \frac{\Delta y^2}{12} u_y^4 + \frac{\Delta y^4}{360} u_y^6 + O(\Delta y^6). \tag{3}$$

From previous studies [20,25,28], we know that with Eqs. (2) and (3), we can apply the symbolic fourth order compact approximation operator to the second derivatives u_{xx} and u_{yy} in Eq. (1), respectively. The discrete 2D Poisson equation will be formulated symbolically as [19]

$$\left(1+\frac{\Delta x^2}{12}\delta_x^2\right)^{-1}\delta_x^2 u+\left(1+\frac{\Delta y^2}{12}\delta_y^2\right)^{-1}\delta_y^2 u=f+\tau_1\Delta x^4+\tau_2\Delta y^4+O(\Delta^6), \tag{4}$$

where τ_1 and τ_2 are used to denote some complex representations that will be canceled in the Richardson extrapolation procedure, Δ^6 denotes the truncated terms in the order of $O(\Delta x^6+\Delta y^6)$. By applying the symbolic operators, setting τ_1 and τ_2 both equal to zero and dropping the Δ^6 terms, Eq. (4) can be rewritten as

$$\begin{aligned}\left(1+\frac{\Delta y^2}{12}\delta_y^2\right)\delta_x^2 u+\left(1+\frac{\Delta x^2}{12}\delta_x^2\right)\delta_y^2 u&=\left(1+\frac{\Delta x^2}{12}\delta_x^2\right)\left(1+\frac{\Delta y^2}{12}\delta_y^2\right)f+O(\Delta^4)\\&=\left[1+\frac{1}{12}(\Delta x^2\delta_x^2+\Delta y^2\delta_y^2)\right]f+O(\Delta^4).\end{aligned} \tag{5}$$

If we set the mesh aspect ratio $\gamma=\Delta x/\Delta y$, we can rewrite Eq. (5) into the following form [25]

$$\begin{aligned}&au_{i,j}+b(u_{i+1,j}+u_{i-1,j})+c(u_{i,j+1}+u_{i,j-1})+d(u_{i+1,j+1}+u_{i+1,j-1}+u_{i-1,j+1}+u_{i-1,j-1})\\&\quad=\frac{\Delta x^2}{2}(8f_{i,j}+f_{i+1,j}+f_{i-1,j}+f_{i,j+1}+f_{i,j-1}),\end{aligned} \tag{6}$$

which has a nine point computational stencil. Here the coefficients in Eq. (6) are

$$a=-10(1+\gamma^2),\quad b=5-\gamma^2,\quad c=5\gamma^2-1,\quad d=(1+\gamma^2)/2.$$

Extrapolation and operator interpolation. With Eq. (6) we can first use the multigrid method to compute the fourth order accurate solutions $u_{i,j}^{2h}$ and $u_{i,j}^{h}$ on the Ω_{2h} grid and the Ω_h grid, respectively. Then we apply the Richardson extrapolation technique, to compute a sixth order accurate solution $\tilde{u}_{i,j}^{2h}$ on Ω_{2h}.

The general Richardson extrapolation can be written as [4]

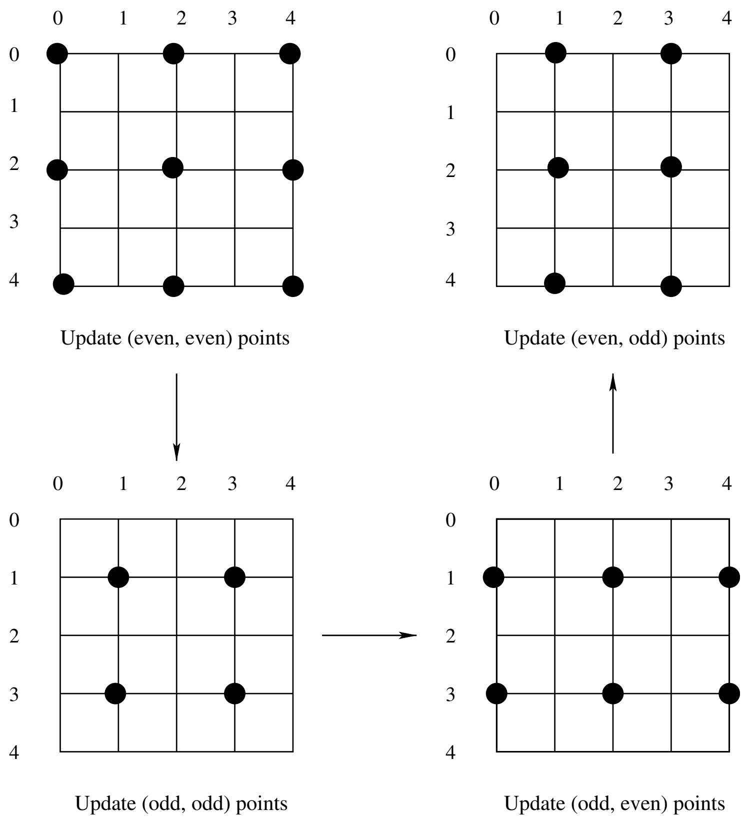

Fig. 1. Illustration of the operator based interpolation scheme for a 5×5 fine grid.

$$\tilde{u}_{i,j}^{2h} = \frac{(2^p u_{2i,2j}^h - u_{i,j}^{2h})}{2^{p-1}}, \tag{7}$$

where p is the order of accuracy before the extrapolation, and the order of accuracy will be increased to $p+2$ after the extrapolation. In this paper, the order p is equal to four, so the Richardson extrapolation formula that we use is

$$\tilde{u}_{i,j}^{2h} = \frac{(16u_{2i,2j}^h - u_{i,j}^{2h})}{15}. \tag{8}$$

By using Eq. (8) we can compute the sixth order accurate solution $\tilde{u}_{2h}$ on the coarse grid Ω_{2h}. Then we need some interpolation technique, as shown in Fig. 1, to interpolate the sixth order accurate solution from the coarse grid Ω_{2h} to the fine grid Ω_h. The $(even, even)$ indexed grid points on Ω_h can be directly interpolated as $\tilde{u}_{2i,2j}^h = \tilde{u}_{i,j}^{2h}$ and it keeps the sixth order accuracy. For other grid points, we need an operator based interpolation scheme to achieve the sixth order accuracy.

In Sun–Zhang's sixth order method [20] for the 2D convection–diffusion equation, they also apply an operator based interpolation scheme together with the Richardson extrapolation in each ADI iteration. Since the number of ADI iterations will increase quickly when the meshsize becomes finer, their extrapolation and interpolation parts will take a large amount of CPU cost. In order to avoid this, our new operator based interpolation procedure and the Richardson extrapolation are carried out only after we get the converged fine and coarse grid fourth order accurate solutions.

We assume that N_x and N_y are both even numbers, our operator based interpolation scheme is an iterative procedure. In each iteration, it will run the Richardson extrapolation first to get the sixth order solution on the coarse grid, then it will use a different interpolation strategy to interpolate the sixth order solution for different grid points on the fine grid. One interpolation iteration (from step k to step $k+1$) is outlined in Algorithm 1.

In Algorithm 1, Ω_h^4 and Ω_{2h}^4 denote the fourth order accurate solution space, Ω_h^6 and Ω_{2h}^6 mean the sixth order accurate solution space. $\tilde{u}^{h,k}$ is the approximate solution for the fine grid after k iterations. The operator based interpolation iteration will continue until the 2-norm R of the correction vector is reduced to below a certain tolerance.

Algorithm 1. Operator based interpolation iteration combined with the sixth order Richardson extrapolation technique

1: Let $u_{old}^h = \tilde{u}^{h,k}$.
2: Update every $(even, even)$ grid point on Ω_h.
From $\tilde{u}_{i,j}^{2h,k} \in \Omega_{2h}^4$ and $\tilde{u}_{2i,2j}^{h,k} \in \Omega_h^4$, we first compute $\tilde{u}_{i,j}^{2h,k+1} \in \Omega_{2h}^6$ by Eq. (8), then use direct interpolation to get $\tilde{u}_{2i,2j}^{h,k+1} \in \Omega_h^6$.
3: Update every (odd, odd) grid point on Ω_h.
From Eq. (6), for each (odd, odd) point (i,j), the updated solution is

$$\tilde{u}_{i,j}^{h,k+1} = \frac{1}{a}\left[F_{i,j} - b\left(\tilde{u}_{i+1,j}^{h,k} + \tilde{u}_{i-1,j}^{h,k}\right) - c\left(\tilde{u}_{i,j+1}^{h,k} + \tilde{u}_{i,j-1}^{h,k}\right) - d\left(\tilde{u}_{i+1,j+1}^{h,k+1} + \tilde{u}_{i+1,j-1}^{h,k+1} + \tilde{u}_{i-1,j+1}^{h,k+1} + \tilde{u}_{i-1,j-1}^{h,k+1}\right)\right]$$

Here, $F_{i,j}$ represents the right-hand side part of Eq. (6).
4: Update every $(odd, even)$ grid point on Ω_h.
From Eq. (6), the idea is similar to the (odd, odd) grid point.
5: Update every $(even, odd)$ grid point on Ω_h.
From Eq. (6), the idea is similar to the $(odd, even)$ grid point.
6: Compute the 2-norm $R = \|\tilde{u}^{h,k+1} - u_{old}^h\|_2$. If not converged, go back to Step 1.

3. Modified multiscale multigrid method

The convergence rate of the multigrid method is independent of the grid size [2,3,21]. It is a very efficient method to solve large sparse linear systems arising from PDEs. Various multigrid implementation strategies with the fourth order compact schemes to solve the 2D and 3D Poisson equations or other PDEs like convection–diffusion equations are discussed in [7,10,16]. In this paper, we use a geometric multiscale multigrid method [3,16], similar to the full multigrid method, to compute the fourth order accurate solution on both the fine and the coarse grids.

We use the notations u_{lh}, f_{lh} and L_{lh} to represent the approximate solution, the right-hand side vector and the finite difference operator for the grid Ω_{lh}, respectively. $I_{(l-1)h}^{lh}$ is the restriction operator from the grid $\Omega_{(l-1)h}$ to the grid Ω_{lh} and $I_{lh}^{(l-1)h}$ is the interpolation operator from the grid Ω_{lh} to the grid $\Omega_{(l-1)h}$. The procedure of our multiscale multigrid method is shown in Fig. 2. The gray color circle indicates the unconverged solution u_{4h} and the black color circles are the fourth order converged solutions u_{2h} and u_h.

Below we describe a multigrid V-Cycle based algorithm to solve the 2D Poisson equation in Algorithm 2.

Algorithm 2 is similar to the full multigrid method, but we do not start from the coarsest grid. Since we use the interpolated coarse grid solution as the initial guess for the fine grid V-Cycle, this algorithm will need fewer number of multigrid cycles than we run the V-Cycle on Ω_h and Ω_{2h} separately to get the converged fourth order accurate solutions u_h and u_{2h} [3,16].

Algorithm 2. Multiscale multigrid method

1: Run the multigrid V-Cycle algorithm $MG(u_{4h}, f_{4h})$ on the coarser grid Ω_{4h} as in Fig. 2 for one or two cycles to get an approximate solution u_{4h}.
2: Use some high order interpolation schemes, like the bicubic interpolation or operator based interpolation, to interpolate u_{4h} to the coarse grid Ω_{2h}, $u_{2h} = I_{4h}^{2h} u_{4h}$.
3: Relax ν_1 times on $L_{2h}u_{2h} = f_{2h}$.
4: Use u_{2h} from the previous step as the initial guess to run the multigrid V-Cycle algorithm $MG(u_{2h}, f_{2h})$ on the coarse grid Ω_{2h} until it converges. We can get the converged fourth order accurate solution u_{2h}.
5: Use a high order interpolation to interpolate u_{2h} to the fine grid Ω_h like $u_h = I_{2h}^{h} u_{2h}$.
6: Relax ν_1 times on $L_h u_h = f_h$.
7: Use u_h from the previous step as the initial guess to run the multigrid V-Cycle algorithm $MG(u_h, f_h)$ on the fine grid Ω_h until it converges. We can get the converged fourth order accurate solution u_h.

In our multiscale multigrid method, we use standard bilinear interpolation to transfer corrections from the coarse grid to the fine grid, full weighting scheme to project residual from the fine grid to the coarse grid, and bicubic interpolation to interpolate the initial guess in Steps 2 and 5. For our multigrid relaxation schemes (smoothers), we use point Gauss–Seidel relaxation and line Gauss–Seidel relaxation. The standard multigrid method with point Gauss–Seidel relaxation scheme is simple to implement and can solve the isotropic Poisson equation well. For the anisotropic Poisson equation, sometimes the point Gauss–Seidel relaxation may not work well [21,25]. For the consideration of robustness, we can use a line Gauss–Seidel relaxation scheme. The line relaxation schemes include X-Line scheme, Y-Line scheme and X–Y-Line scheme.

4. Numerical results

In this section, we compare our new sixth order multigrid method with Richardson extrapolation (MG-Six) strategy with Sun–Zhang's sixth order REC method (REC-ADI) [20] and with the standard fourth order compact difference scheme using multigrid (MG-FOC). The codes are written in Fortran 77 programming language and run on one processor of an IBM HS21 blade cluster at the University of Kentucky. The processor has 2 GB of local memory and runs at 2.0 GHZ.

The initial guess for the V-Cycle on Ω_{4h} is the zero vector. For Problem 1, the multigrid V-Cycle for the Ω_{2h} and Ω_h grids will stop when the 2-norm of the residual vector is reduced by 10^{-13}, the iterative interpolation procedure will stop when the 2-norm of the correction vector of the approximate solution is less than 10^{-13}. For Problem 2, both of the stopping criteria will be changed to 10^{-10}. The errors reported are the maximum absolute errors over the discrete grid of the finest level.

We would like to comment on the fact that we use different stopping criteria for these two test problems. Generally, 10^{-10} is our standard stopping criteria to check the 2-norm of the residual or the correction vector, but sometimes it may be changed depends on the test case itself. For Problem 1, if we look at the experimental results from Table 1, we will find that when $n = 256$ the maximum error of our FOC scheme has dropped around the 10^{-10}. If we still use 10^{-10} as our stopping tolerance, we may not get the enough accuracy when the iteration stops. In order to get sufficient accuracy we need, we choose 10^{-13} as the criteria for Problem 1.

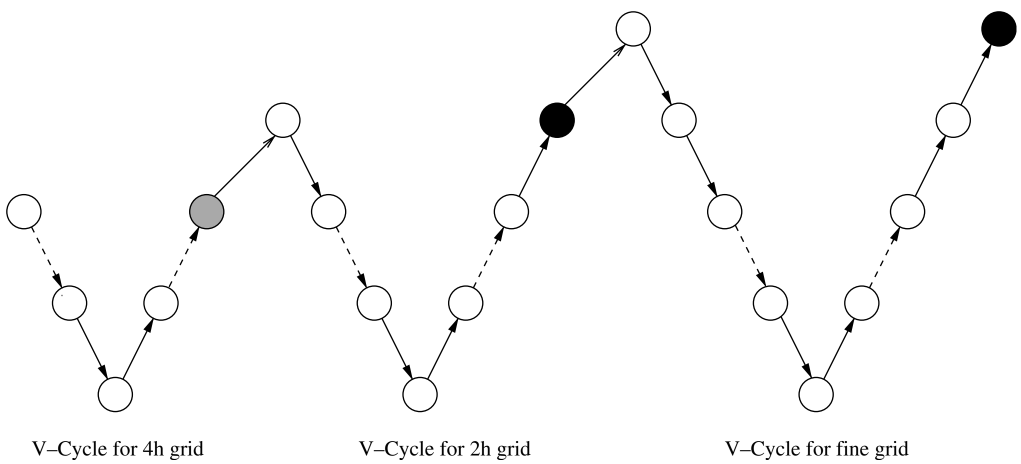

Fig. 2. Representation of our multiscale multigrid method.

Table 1
Numerical comparison results for Problem 1

n	Strategy	# Iteration	CPU	Error	Order
16	REC-ADI	237	0.020	1.32e−6	5.7
	MG-Six(point)	(11,11), **40**	0.001	1.32e−6	5.7
	MG-FOC(point)	14	0.002	1.63e−5	3.9
	MG-Six(line)	(5,6), **40**	0.001	1.32e−6	5.7
	MG-FOC(line)	7	0.001	1.63e−5	3.9
32	REC-ADI	901	0.302	2.27e−8	5.9
	MG-Six(point)	(11,12), **39**	0.007	2.27e−8	5.9
	MG-FOC(point)	14	0.004	1.02e−6	4.0
	MG-Six(line)	(6,7), **39**	0.008	2.27e−8	5.9
	MG-FOC(line)	8	0.006	1.02e−6	4.0
64	REC-ADI	3447	4.662	3.68e−10	5.9
	MG-Six(point)	(12,13), **37**	0.029	3.67e−10	6.0
	MG-FOC(point)	15	0.022	6.39e−8	4.0
	MG-Six(line)	(7,7), **36**	0.033	3.67e−10	6.0
	MG-FOC(line)	9	0.024	6.40e−8	4.0
128	REC-ADI	Not converged	–	–	–
	MG-Six(point)	(13,13), **33**	0.129	5.26e−12	6.1
	MG-FOC(point)	15	0.093	3.99e−9	4.0
	MG-Six(line)	(7,8), **34**	0.161	5.87e−12	6.0
	MG-FOC(line)	9	0.140	4.00e−9	4.0
256	REC-ADI	Not converged	–	–	–
	MG-Six(point)	(14,14), **30**	0.880	1.27e−13	5.4
	MG-FOC(point)	16	0.465	2.47e−10	4.0
	MG-Six(line)	(8,8), **30**	1.202	1.11e−13	5.7
	MG-FOC(line)	8	0.737	2.50e−10	4.0

For the line Gauss–Seidel relaxation schemes for these two test cases, we choose X–Y-Line relaxation scheme for Problem 1. For Problem 2, since x is its dominant direction, we only perform line relaxation along the x-direction, which is the X-Line relaxation scheme.

We also compute the estimated order of accuracy for every computing strategy in different grid size. Let us consider two meshsizes Δ^H and Δ^h on Ω^H and Ω^h, respectively. The maximum absolute errors of these two grids are denoted as $Error^H$ and $Error^h$. If we set the order of accuracy to be m, then we have the following form:

$$\frac{(\Delta^H)^m}{(\Delta^h)^m} = \frac{\mathrm{Error}^H}{\mathrm{Error}^h}.$$

So, the order of accuracy m can be computed as

$$m = \frac{\log \frac{\mathrm{Error}^H}{\mathrm{Error}^h}}{\log \frac{\Delta^H}{\Delta^h}}.$$

The order of accuracy is formally defined when the meshsize approaches zero. Therefore, when the meshsize is relatively large, the discretization scheme may not achieve its formal order of accuracy.

Problem 1. In order to compare with Sun–Zhang's sixth order method, we consider one of the test cases in Sun–Zhang's paper [20]. Sun and Zhang used a 2D convection–diffusion equation, we set the convection coefficients to be zero, then the equation becomes a 2D Poisson equation. The test Problem 1 can be written as

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = -\alpha \sin\left(\frac{\pi}{b} y\right), \quad (x, y) \in \Omega = [0, \lambda] \times [0, b], \tag{9}$$

where the boundary conditions are

$$u(0, y) = u(\lambda, y) = u(x, 0) = u(x, b) = 0.$$

In this equation, the parameter α is chosen as

$$\alpha = \frac{F\pi}{Rb}.$$

The analytic solution of Eq. (9) is

$$u = -\alpha \left(\frac{b}{\pi}\right)^2 \sin\left(\frac{\pi y}{b}\right)\left(e^{\frac{\pi x}{b}} - 1\right).$$

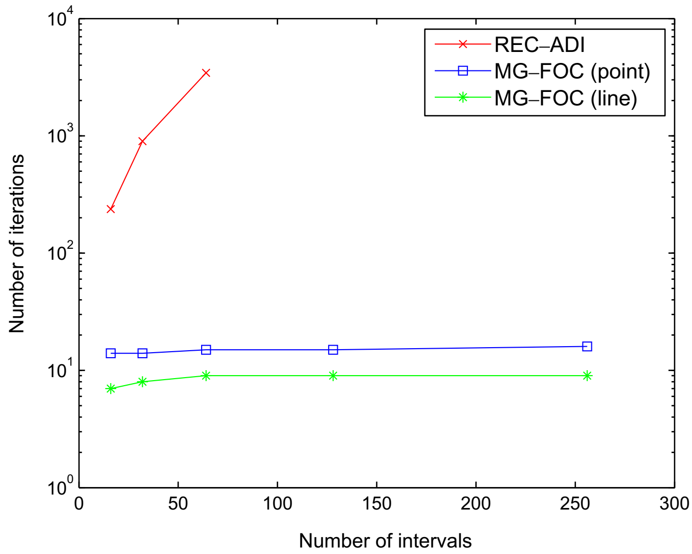

Fig. 3. Comparison of the number of iterations with REC-ADI, MG-FOC(point) and MG-FOC(line) methods for Problem 1. Each symbol with increasing number of iterations corresponds to an increasing fine grid: 16, 32, 64, 128, and 256 intervals.

The other parameters are chosen as

$$\lambda = 10^7 \text{ m}, \quad b = 2\pi \times 10^6 \text{ m}, \quad F = 0.3 \times 10^{-7} \text{ m}^2 \text{ s}^{-2}, \quad R = 0.6 \times 10^{-3} \text{ m s}^{-1}.$$

In the following, we define $N_x = N_y = n$. The meshsizes Δ_x and Δ_y are equal to λ/n and b/n, respectively. Table 1, Figs. 3 and 4 show the comparison results for Problem 1.

Table 1 shows the number of iterations and other information for different solution strategies that we compared. We can find that when the mesh becomes finer, the number of ADI iterations increases very quickly. When $n > 64$, the ADI iteration cannot converge within the maximum number of iterations we set, which is 5000. For the MG-Six method, the number of iterations contains three parts. They are the number of V-Cycles for Ω_{2h}, the number of V-Cycles for Ω_h, and the number of iterations for the iterative interpolation combined with the Richardson extrapolation. These three numbers are listed in the iteration columns for the MG-Six(point) and MG-Six(line) strategies in Table 1. We can see that, by using our new sixth order compact scheme, the number of V-Cycles for Ω_h and Ω_{2h} are reduced, compared to the traditional multigrid V-Cycle with the FOC scheme. We can also see that the REC-ADI method takes much more iterations and CPU cost than the MG-FOC strategies and the MG-Six strategies from Figs. 3 and 4.

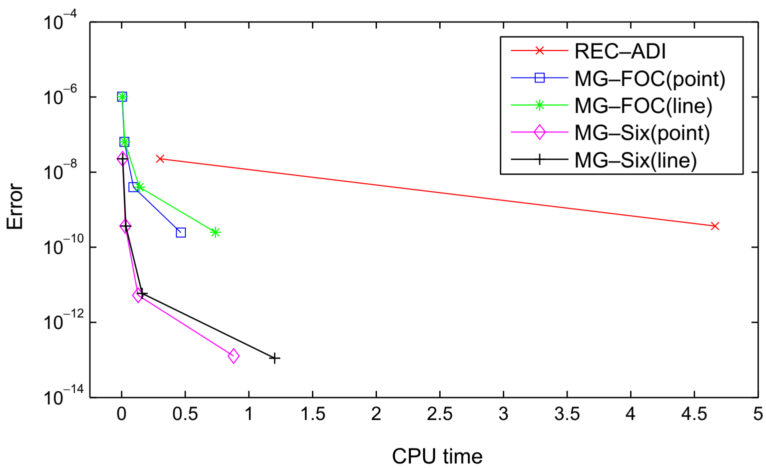

Fig. 4. Comparison of the maximum error and the CPU cost for the Problem 1. Each symbol with increasing CPU cost corresponds to an increasing fine grid: 16, 32, 64, 128, and 256 intervals.

The data in Table 1 and Fig. 4 also indicate that the accuracy of the approximate solutions computed by our new sixth order method and the Sun–Zhang's REC-ADI method is comparable. When $n > 64$, our multigrid method can still compute the highly accurate solution. The CPU time for MG-Six method is much less than that needed by the ADI iteration, and is better than running the MG-FOC separately twice to get u_h and u_{2h}.

Since the grid is almost isotropic for Problem 1, we can see that the point Gauss–Seidel relaxation scheme remains competitive compared with the line Gauss–Seidel relaxation. The point Gauss–Seidel relaxation scheme actually needs less CPU time than the line Gauss–Seidel relaxation scheme to compute numerical solution of comparable accuracy.

Problem 2. In order to better compare the line Gauss–Seidel relaxation scheme and the point Gauss–Seidel relaxation scheme. We consider an anisotropic Poisson equation to show the efficiency and scalability of the line relaxation scheme in solving 2D Poisson equation.

We choose the following equation:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = -2\pi^2 \sin(\pi x)\cos(\pi y), \quad (x, y) \in \Omega = [0, 4] \times [0, 1], \tag{10}$$

which has the Dirichlet boundary condition.

The analytic solution of Eq. (10) is

$$u(x, y) = \sin(\pi x)\cos(\pi y).$$

Since we use the same number of intervals in both the x and y directions, the solution changes more rapidly in the x-direction than in the y-direction. So, x is the dominant direction. We choose X-Line Gauss–Seidel relaxation scheme.

As for Problem 1, we also compare different solution strategies indexed by the number of multigrid cycles or iterations, CPU time, the maximum absolute errors and the estimated order of accuracy. The results are shown in Table 2, Figs. 5 and 6. We can see obviously that, even with the anisotropy, the convergence rates of our MG-Six(line) and MG-FOC(line) are barely affected. These two schemes can keep both the scalability and the efficiency when the number of intervals increases. For MG-Six(point) and MG-FOC(point), they need much more iterations and CPU cost than the line relaxation schemes. When $n < 64$, even the REC-ADI method can converge with less CPU time than the MG-Six(point) method. So, multigrid method with the line relaxation scheme is the most efficient way to solve the anisotropic 2D Poisson equation compared with the other methods we tested.

Again in this test case, our new sixth order accurate method can solve the problem with high order accuracy which is comparable with Sun and Zhang's method and keep the lower CPU costs. It is clear that the MG-Six method outperforms other methods.

Table 2
Numerical comparison results for Problem 2

n	Strategy	# Iteration	CPU	Error	Order
16	REC-ADI	14	0.002	1.12e−4	5.4
	MG-Six(point)	(14,43), **55**	0.005	1.12e−4	5.4
	MG-FOC(point)	39	0.003	2.45e−4	4.2
	MG-Six(line)	(1,7), **55**	0.001	1.12e−4	5.4
	MG-FOC(line)	6	0.001	2.45e−4	4.2
32	REC-ADI	42	0.014	2.50e−6	5.5
	MG-Six(point)	(43,60), **85**	0.025	2.50e−6	5.5
	MG-FOC(point)	58	0.016	1.43e−5	4.1
	MG-Six(line)	(7,9), **85**	0.011	2.50e−6	5.5
	MG-FOC(line)	10	0.005	1.43e−5	4.1
64	REC-ADI	155	0.271	4.58e−8	5.8
	MG-Six(point)	(60,73), **93**	0.122	4.58e−8	5.8
	MG-FOC(point)	72	0.095	8.70e−7	4.0
	MG-Six(line)	(9,9), **93**	0.044	4.58e−8	5.8
	MG-FOC(line)	11	0.021	8.70e−7	4.0
128	REC-ADI	607	6.849	7.66e−10	5.9
	MG-Six(point)	(73,79), **89**	0.571	7.66e−10	5.9
	MG-FOC(point)	78	0.423	5.37e−8	4.0
	MG-Six(line)	(9,9), **89**	0.188	7.66e−10	5.9
	MG-FOC(line)	12	0.091	5.37e−8	4.0
256	REC-ADI	2411	130.393	1.22e−11	6.0
	MG-Six(point)	(79,83), **80**	3.774	1.24e−11	6.0
	MG-FOC(point)	82	3.201	3.33e−9	4.0
	MG-Six(line)	(9,9), **80**	1.982	1.24e−11	6.0
	MG-FOC(line)	13	1.649	3.33e−9	4.0

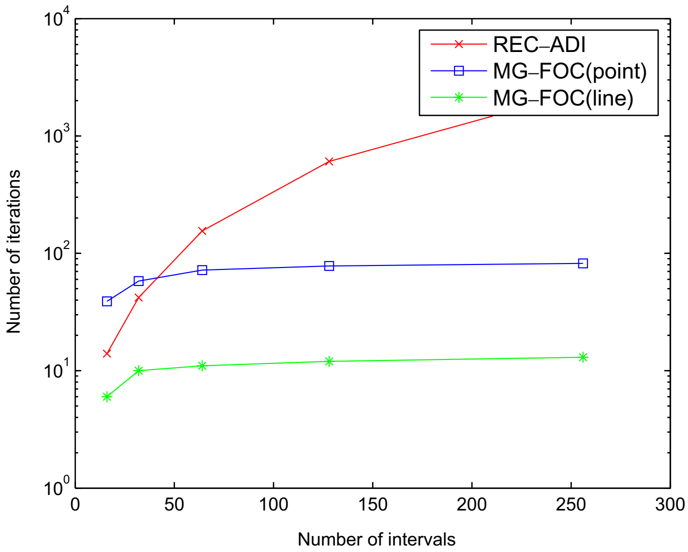

Fig. 5. Comparison of the number of iterations with REC-ADI, MG-FOC(point) and MG-FOC(line) methods for Problem 2. Each symbol with increasing number of iterations corresponds to an increasing fine grid: 16, 32, 64, 128, and 256 intervals.

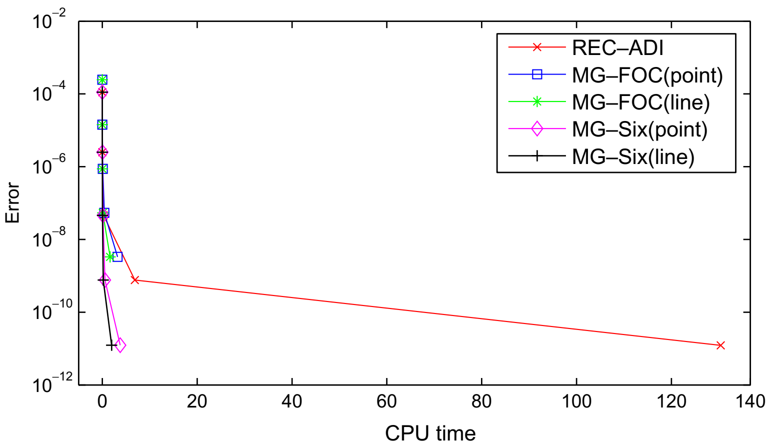

Fig. 6. Comparison of the maximum error and the CPU cost for the Problem 2. Each symbol with increasing CPU cost corresponds to an increasing fine grid: 16, 32, 64, 128, and 256 intervals.

5. Concluding remarks

We designed a new sixth order compact computation scheme with a multigrid method and Richardson extrapolation to solve the 2D Poisson equation. This new idea is based on designing a geometric multiscale multigrid method, similar to the full multigrid method, to compute the approximate solution using the fourth order compact scheme in both the fine and the coarse grids. We also present a new iterative interpolation scheme, which is combined with the Richardson extrapolation to achieve the sixth order accuracy on the fine grid.

Numerical results show that the new numerical solution method can solve the 2D Poisson equation with highly accurate solution compared with other sixth order compact schemes, and also keep the low CPU cost. This two scale grid idea can also be extended to solve other PDEs such as the 3D Poisson equation, 2D and 3D convection–diffusion equations. For the convection-dominated problems, multigrid methods with the line relaxation schemes will be expected to work well. For various multigrid algorithms with the high order compact schemes to solve convection–diffusion equations, we refer readers to [22].

References

[1] Y. Adam, Highly accurate compact implicit methods and boundary conditions, J. Comput. Phys. 24 (1) (1977) 10–22.
[2] A. Brandt, Multi-level adaptive solutions to boundary-value problems, Math. Comput. 31 (138) (1977) 333–390.
[3] W.L. Briggs, V.E. Henson, S.F. McCormick, A Multigrid Tutorial, second ed., SIAM, Philadelphia, PA, 2000.

[4] W. Cheney, D. Kincard, Numerical Mathematics and Computing, fourth ed., Brooks/Cole Publishing, Pacific Grove, CA, 1999.
[5] P.C. Chu, C. Fan, A three-point combined compact difference scheme, J. Comput. Phys. 140 (1998) 370–399.
[6] P.C. Chu, C. Fan, A three-point six-order nonuniform combined compact difference scheme, J. Comput. Phys. 148 (1999) 663–674.
[7] M.M. Gupta, J. Kouatchou, J. Zhang, Comparison of second and fourth order discretization for multigrid Poisson solver, J. Comput. Phys. 132 (1997) 226–232.
[8] M.M. Gupta, J. Kouatchou, Symbolic derivation of finite difference approximations for the three dimensional Poisson equation, Numer. Methods Partial Differential Equat. 18 (5) (1998) 593–606.
[9] M.M. Gupta, R.P. Manohar, J.W. Stephenson, A single cell high order scheme for the convection–diffusion equation with variable coefficients, Int. J. Numer. Methods Fluids 4 (1984) 641–651.
[10] M.M. Gupta, J. Zhang, High accuracy multigrid solution of the 3D convection–diffusion equation, Appl. Math. Comput. 113 (2–3) (2000) 249–274.
[11] S.K. Lele, Compact finite difference schemes with spectral-like resolution, J. Comput. Phys. 103 (1992) 16–42.
[12] M. Li, T. Tang, B. Fornberg, A compact fourth-order finite difference scheme for the steady incompressible Navier–Stokes equations, Int. J. Numer. Methods Fluids 20 (1995) 1137–1151.
[13] R.J. MacKinnon, R.W. Johnson, Differential-equation-based representation of truncation errors for accurate numerical simulation, Int. J. Numer. Methods Fluids 13 (1991) 739–757.
[14] D. Peaceman, H. Rachford, The numerical solution of elliptic and parabolic differential equations, J. SIAM 3 (1955) 28–41.
[15] K. Sakurai, T. Aoki, W.H. Lee, K. Kato, Poisson equations solver with fourth-order accuracy by using interpolated differential operator scheme, Comput. Math. Appl. 43 (2002) 621–630.
[16] S. Schaffer, Higher order multigrid method, Math. Comput. 43 (167) (1984) 89–115.
[17] W.F. Spotz, High-Order Compact Finite Difference Schemes for Computational Mechanics, Ph.D. Thesis, University of Texas at Austin, Austin, TX, 1995.
[18] W.F. Spotz, G.F. Carey, High-order compact scheme for the steady stream-function vorticity equations, Int. J. Numer. Methods Eng. 38 (1995) 3497–3512.
[19] J.C. Strikwerda, Finite Difference Schemes and Partial Differential Equations, Chapman & Hall, New York, NY, 1989.
[20] H. Sun, J. Zhang, A high order finite difference discretization strategy based on extrapolation for convection diffusion equations, Numer. Methods Partial Differential Equat. 20 (1) (2004) 18–32.
[21] P. Wesseling, An Introduction to Multigrid Methods, Wiley, Chichester, England, 1992.
[22] J. Zhang, Multigrid Acceleration Techniques and Applications to the Numerical Solution of Partial Differential Equations, Ph.D. Thesis, The George Washington University, Washington, DC, 1997.
[23] J. Zhang, An explicit fourth-order compact finite difference scheme for three dimensional convection–diffusion equation, Commun. Numer. Methods Eng. 14 (1998) 209–218.
[24] J. Zhang, Preconditioned iterative methods and finite difference schemes for convection diffusion, Appl. Math. Comput. 109 (1) (2000) 11–30.
[25] J. Zhang, Multigrid method and fourth order compact difference scheme for 2D Poisson equation with unequal meshsize discretization, J. Comput. Phys. 179 (2002) 170–179.
[26] J. Zhang, L. Ge, J. Kouatchou, A two colorable fourth-order compact difference scheme and parallel iterative solution of the 3D convection diffusion equation, Math. Comput. Simul. 54 (1–3) (2000) 65–80.
[27] J. Zhang, L. Ge, Symbolic computation of high order compact difference schemes for three dimensional linear elliptic partial differential equations with variable coefficients, J. Comput. Appl. Math. 143 (1) (2002) 9–27.
[28] J. Zhang, H. Sun, J.J. Zhao, High order compact scheme with multigrid local mesh refinement procedure for convection diffusion problems, Comput. Methods Appl. Mech. Eng. 191 (41–42) (2002) 4661–4674.