Journal of Computational Physics 229 (2010) 6381–6391

Contents lists available at ScienceDirect

Journal of Computational Physics

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

Multigrid method and fourth-order compact difference discretization scheme with unequal meshsizes for 3D poisson equation

Yongbin Ge *

Institute of Applied Mathematics and Mechanics, Ningxia University, Yinchuan 750021, China

ARTICLE INFO

Article history:
Received 12 November 2009
Received in revised form 5 March 2010
Accepted 30 April 2010
Available online 9 May 2010

Keywords:
3D Poisson equation
Compact difference scheme
Unequal meshsizes
Multigrid method
Partial semi-coarsening

ABSTRACT

A fourth-order compact difference discretization scheme with unequal meshsizes in different coordinate directions is employed to solve a three-dimensional (3D) Poisson equation on a cubic domain. Two multgrid methods are developed to solve the resulting sparse linear systems. One is to use the full-coarsening multigrid method with plane Gauss–Seidel relaxation, which uses line Gauss–Seidel relaxation to compute each planewise solution. The other is to construct a partial semi-coarsening multigrid method with the traditional point or plane Gauss–Seidel relaxations. Numerical experiments are conducted to test the computed accuracy of the fourth-order compact difference scheme and the computational efficiency of the multigrid methods with the fourth-order compact difference scheme.

1. Introduction

We consider the 3D Poisson equation with Dirichlet boundary conditions

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

$$u(x,y,z) = g(x,y,z), \quad (x,y,z) \in \partial\Omega. \tag{2}$$

where Ω is a bounded convex domain and $\partial\Omega$ is the boundary of Ω. The solution $u(x,y,z)$ and the forcing function $f(x,y,z)$ are assumed to be sufficiently smooth and to have the required continuous partial derivatives.

In the past two decades, a great deal of research work has been published on the development of numerical solution of Poisson equations [1–12]. On one hand, high-order compact (HOC) finite difference methods have been proposed for solving 2D and 3D Poisson equations [1–7]. On the other hand, recently there has been a renewed interest in combing HOC scheme with multigrid method to solve elliptic differential equations [8–17]. The efficiency and performance of this procedure have been verified in the literature. Gupta et al. [8] combined HOC difference approximation with multigrid V-cycle algorithm to solve the 2D Poisson equation, which has showed the dramatic improvement in the computed accuracy and the computational efficiency compared with the five-point second-order central difference scheme. Othman and Abdullah [9] presented an efficient technique which is named quarter sweeps multigrid method for solving 2D Poisson equation and in the technique the red–black Guass–Seidel smoothing scheme is shown to be the most superior. Zhang [10] employed a fourth-order compact finite difference scheme with multigrid algorithm to solve the 3D Poisson equation and compared the influence of different ordering of grid space and different grid transfer operators on the convergence and efficiency of high-order algorithm. Kouatchou and Zhang [11] proposed an optimal scaled injection operator for the multigrid algorithm for solving

* Tel.: +86 951 2061724; fax: +86 951 2087452.
E-mail address: gyb@nxu.edu.cn

0021-9991/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2010.04.048

the 3D Poisson equation with a fourth-order 19-point compact finite difference scheme. The goal of them is to seek high accuracy within the constraints imposed by limitations in computer time and storage.

It is noticed that performance of HOC schemes and some combined with multigrid method for solving elliptic differential equations using equal meshsize descritization in different directions have been extensively described in the literature [1–5,7–11,13–16]. However, in certain situations the physical quantity modeled may have uneven distribution in different directions, in which the use of unequal meshsizes in different coordinate directions is more cost-effective [6,12,17]. In [12], a fourth-order compact discretization scheme with unequal meshsizes and specialized multigrid methods using a partial semi-coarsening strategy and line Gauss–Seidel relaxation were designed to solve the 2D Poisson equation. The two special multigrid methods based on the fourth-order compact unequal-meshsize-discretization scheme were found to be very efficient for solving the 2D Poisson equation. However, the author pointed out in the concluding remarks that "the analogous generalizations of specialized multigrid methods to 3D are not straightforward; at least their implementations are nontrivial". In this paper, we will solve this problem and generalize two specialized muligrid methods for solving 2D Poisson equation in [12] to 3D case.

The organization of this paper is as follows. Section 2 introduces a fourth-order compact finite difference discretization scheme with unequal meshsizes in different coordinate directions for the 3D Poisson Eq. (1); In Section 3, a brief introduction to the general philosophy of the multigrid method is given and a plane relaxation full-coarsening multigrid method and a partial semi-coarsening multigrid method are designed; In Section 4, numerical experiments are conducted to show the efficiency and the effectiveness of the present method; Finally, Section 5 is the concluding remarks.

2. Fourth-order compact difference scheme

For convenience, we consider a cubic domain $\Omega = [0,L_x] \times [0,L_y] \times [0,L_z]$. We discretize (1) with unequal meshsizes h_x, h_y and h_z in the x, y and z coordinate directions, respectively, and $N_x = L_x/h_x$, $N_y = L_y/h_y$ and $N_z = L_z/h_z$ are the numbers of uniform intervals along the x, y and z directions. The grid points are (x_i,y_j,z_k), with $x_i = ih_x$, $y_j = jh_y$ and $z_k = kh_z$, $i = 0,1,\ldots,N_x$, $j = 0,1,\ldots,N_y$ and $k = 0,1,\ldots,N_z$.

If we use the central difference approximation to all the second partial derivative of u at the grid point (i,j,k) corresponding to (x_i,y_j,z_k), we can derive the familiar seven-point central difference scheme for (1)

$$\delta_x^2 u_{i,j,k} + \delta_y^2 u_{i,j,k} + \delta_z^2 u_{i,j,k} = f_{i,j,k} + \tau_{i,j,k}, \tag{3}$$

in which

$$\delta_x^2 u_{i,j,k} = \frac{u_{i+1,j,k} - 2u_{i,j,k} + u_{i-1,j,k}}{h_x^2}, \tag{4}$$

$$\delta_y^2 u_{i,j,k} = \frac{u_{i,j+1,k} - 2u_{i,j,k} + u_{i,j-1,k}}{h_y^2}, \tag{5}$$

$$\delta_z^2 u_{i,j,k} = \frac{u_{i,j,k+1} - 2u_{i,j,k} + u_{i,j,k-1}}{h_z^2}, \tag{6}$$

and $\tau_{i,j,k}$ is the truncation error term. Using Taylor series analysis, we can get

$$\tau_{i,j,k} = \frac{1}{12}\left(h_x^2 u_{xxxx} + h_y^2 u_{yyyy} + h_z^2 u_{zzzz}\right) + O(h_x^4 + h_y^4 + h_z^4). \tag{7}$$

Obviously, (3) is second-order accuracy; i.e., its truncation is $O(h_x^2 + h_y^2 + h_z^2)$. However, if u_{xxxx}, u_{yyyy} and u_{zzzz} in the leading term of $\tau_{i,j,k}$ in (7) can be approximated to order $O(h_x^2 + h_y^2 + h_z^2)$ compactly, then this approximation combined with the seven-point central difference can be led to a fourth-order compact scheme; i.e., its truncation is $O(h_x^4 + h_y^4 + h_z^4)$.

To achieve this goal, we first take the appropriate derivatives of (1) to write

$$u_{xxxx} = f_{xx} - u_{xxyy} - u_{xxzz}, \tag{8}$$

$$u_{yyyy} = f_{yy} - u_{xxyy} - u_{yyzz}, \tag{9}$$

$$u_{zzzz} = f_{zz} - u_{xxzz} - u_{yyzz}. \tag{10}$$

It is obvious that each of the expressions on the right hand in (8)–(10) has compact central difference approximations of order $O(h_x^2 + h_y^2 + h_z^2)$. Substituting (8)–(10) into (7), then (7) turns to

$$\tau_{i,j,k} = \frac{1}{12}\left(h_x^2 f_{xx} + h_y^2 f_{yy} + h_z^2 f_{zz}\right)_{i,j,k} - \frac{1}{12}\left[h_x^2(u_{xxyy} + u_{xxzz}) + h_y^2(u_{xxyy} + u_{yyzz}) + h_z^2(u_{xxzz} + u_{yyzz})\right]_{i,j,k} + O\left(h_x^4 + h_y^4 + h_z^4\right). \tag{11}$$

Simply substituting central difference expressions in (11) and including these in the central difference scheme (3), we get

$$(\delta_x^2 + \delta_y^2 + \delta_z^2)u_{i,j,k} + \frac{1}{12}[h_x^2(\delta_x^2\delta_y^2 + \delta_x^2\delta_z^2) + h_y^2(\delta_x^2\delta_y^2 + \delta_y^2\delta_z^2) + h_z^2(\delta_x^2\delta_z^2 + \delta_y^2\delta_z^2)]u_{i,j,k}$$
$$= f_{i,j,k} + \frac{1}{12}(h_x^2\delta_x^2 + h_y^2\delta_y^2 + h_z^2\delta_z^2)f_{i,j,k} + O(h_x^4 + h_y^4 + h_z^4). \tag{12}$$

Substituting (4)–(6) in both sides of (12), after some rearrangement and dropping the high-order term $O(h_x^4 + h_y^4 + h_z^4)$, we can get an unequal-meshsize-discretization fourth-order compact difference scheme for 3D Poisson equation, written out explicitly

$$\begin{aligned}
&-8\left(\frac{1}{h_x^2}+\frac{1}{h_y^2}+\frac{1}{h_z^2}\right)u_{i,j,k}+\left(\frac{4}{h_x^2}-\frac{1}{h_y^2}-\frac{1}{h_z^2}\right)(u_{i+1,j,k}+u_{i-1,j,k})+\left(\frac{4}{h_y^2}-\frac{1}{h_x^2}-\frac{1}{h_z^2}\right)(u_{i,j+1,k}+u_{i,j-1,k})\\
&\quad+\left(\frac{4}{h_z^2}-\frac{1}{h_x^2}-\frac{1}{h_y^2}\right)(u_{i,j,k+1}+u_{i,j,k-1})+\frac{1}{2}\left(\frac{1}{h_x^2}+\frac{1}{h_y^2}\right)(u_{i+1,j+1,k}+u_{i+1,j-1,k}+u_{i-1,j+1,k}+u_{i-1,j-1,k})\\
&\quad+\frac{1}{2}\left(\frac{1}{h_x^2}+\frac{1}{h_z^2}\right)(u_{i+1,j,k+1}+u_{i+1,j,k-1}+u_{i-1,j,k+1}+u_{i-1,j,k-1})+\frac{1}{2}\left(\frac{1}{h_y^2}+\frac{1}{h_z^2}\right)(u_{i,j+1,k+1}+u_{i,j+1,k-1}+u_{i,j-1,k+1}+u_{i,j-1,k-1})\\
&\quad=\frac{1}{2}(6f_{i,j,k}+f_{i+1,j,k}+f_{i,j+1,k}+f_{i-1,j,k}+f_{i,j-1,k}+f_{i,j,k+1}+f_{i,j,k-1}).
\end{aligned}\tag{13}$$

This scheme was originally derived by Wang et al. [6] by using a different way. If letting $h = h_x = h_y = h_z$ and multiplying h^2 on the both sides of (13), we can get the fourth-order compact difference scheme approximation of the 3D Poisson equation with equal meshsize disretization, which is the same as that proposed by Kwon and Stephenson [1]

$$\begin{aligned}
&-24u_{i,j,k}+2(u_{i+1,j,k}+u_{i-1,j,k}+u_{i,j+1,k}+u_{i,j-1,k}+u_{i,j,k+1}+u_{i,j,k-1})+u_{i+1,j+1,k}+u_{i+1,j-1,k}+u_{i-1,j+1,k}+u_{i-1,j-1,k}\\
&\quad+u_{i+1,j,k+1}+u_{i+1,j,k-1}+u_{i-1,j,k+1}+u_{i-1,j,k-1}+u_{i,j+1,k+1}+u_{i,j+1,k-1}+u_{i,j-1,k+1}+u_{i,j-1,k-1}\\
&\quad=\frac{h^2}{2}(6f_{i,j,k}+f_{i+1,j,k}+f_{i,j+1,k}+f_{i-1,j,k}+f_{i,j-1,k}+f_{i,j,k+1}+f_{i,j,k-1}).
\end{aligned}\tag{14}$$

3. Specialized multigrid method

Multigrid method has been known for many years and it is among the fastest and the most efficient iterative methods for solving a wide class of partial differential equations. This method offers convergence rate independent of the grid size and is very effective for solving large scale sparse linear systems which are derived by discretizing elliptic problems. The essential principle of mutigrid method is to approximate the smooth (long wavelength) part of the error on coarser grids. The non-smooth or rough part is reduced with a small number (independent of meshsize h) of iterations with a basic iterative method on the fine grid. For solving 2D and 3D Poisson equations discretized by the standard second-order central difference scheme and the fourth-order compact difference schemes (with equal meshsize), efficient multigrid methods are implemented in [8,10]. The multigrid methods employed standard grid coarsening strategy (the coarse grid meshsizes double that of the fine gird. See [8,10]). However, for a Poisson equation discretized with unequal meshsizes, i.e., for solving an anisotropic problem, a standard multigrid method does not work very well [12]. So, in [12], two specialized multigrid methods are proposed to solve 2D Poisson equation discretized with unequal meshsizes. The first is to use line Gauss–Seidel relaxation to replace point Gauss–Seidel relaxation because line Gauss–Seidel relaxation is shown to be very effective in removing high-frequency errors in the dominant direction with large coefficients. The second is to use partial semi-coarsening strategy; i.e., grid coarsening is only performed along the dominant direction while the meshsize along the other direction is not coarsened. Numerical results show that the specialized multigrid methods are more efficient than the traditional full-coarsening multigrid method and the fourth-order compact difference scheme with the multigrid methods is more cost-effective than the second-order central difference scheme with the multigrid methods.

For the particularity of 3D anisotropic problems, the implementations of the multigrid methods are more complicated than for 2D cases; i.e., the dominant direction is not always one direction, the x direction, the y direction or the z direction, but two directions, the x and y directions, the y and z directions, or the x and z directions, and one is the dominant direction and the other is the sub-dominant direction. So either a single-line relaxation or a semi-coarsening along one direction will not suffice for 3D case. For 3D problems, plane relaxation can be used to remove high-frequency error and each planewise solution still can be obtained by using line relaxation. It is easy to be conducted by performing one line Gauss–Seidel relaxation along the dominant direction followed by sweeping along the sub-dominant direction. If problems just show anisotropy in one direction, we just perform one line Gauss–Seidel relaxation along the dominant direction followed by sweeping along either of the two non-dominant directions. Numerical experiment results show that it is as effective as the line Gauss–Seidel relaxation for 2D problems. However, partial semi-coarsening is complicated since there are two directions in which the meshsizes may change. Under this condition, standard multigrid restriction and prolongation operators do not work well. So, new restriction and prolongation operators must be constructed. But we think this difficulty can be overcome. So, in this paper, we shall mostly put our attention on partial semi-coarsening multigrid strategy for 3D.

3.1. Partial semi-coarsening strategy

For convenience, we suppose that the domain Ω is a regular cube; i.e., $\Omega = [0,L] \times [0,L] \times [0,L]$, and the dominant directions are always in one direction or two directions, never in all three directions. To simplify our discussion without loss of

$$\begin{array}{c} 128\times32\times8 \\ \Downarrow \\ 64\times32\times8 \\ \Downarrow \\ 32\times32\times8 \\ \Downarrow \\ 16\times16\times8 \\ \Downarrow \\ 8\times8\times8 \\ \Downarrow \\ 4\times4\times4 \\ \Downarrow \\ 2\times2\times2 \end{array}$$

Fig. 1. Process of the grid partial semi-coarsening.

generality, we assume that the x direction is the dominant direction and the y direction is the sub-dominant direction, and the z direction is not the dominant direction. It means that the physical quantity has uneven distribution in different directions, which changes fastest in the x direction, faster in the y direction and slowest in the z direction, comparatively. To capture accurate solution of the problems, we need distribute most grids in the x direction, more grids in the y direction, and least grids in the z direction, comparatively. Consequently, we suppose that $h_x \leqslant h_y \leqslant h_z$. For efficient implementation of the multigrid method, we further suppose that $N_x = 2^{n_x}, N_y = 2^{n_y}$ and $N_z = 2^{n_z}$, and positive integers n_x, n_y and n_z are possibly grid level numbers in the x, y and z directions. By our assumptions above, it is easy to see that $n_x \geqslant n_y \geqslant n_z$.

In the implementation of partial semi-coarsening strategy the grid coarsening is only performed along the dominant direction(s). So, we start coarsening from the first plane in the x direction. Every other grid plane is alternately reduced, therefore we get the coarsen grid from fine grid. Because on each grid level the meshsize h_x is different, we have different difference equations. But with the process of grid coarsening, inevitably, there will be a coarse grid level on which $N_x = N_y$. On this coarse grid, the x and y directions are simultaneously becoming dominant. In order to get the successive coarse grid from this level, it needs to reduce grid planes from the first plane to every other plane along both the x and y directions. The difference equations are still different on each coarse grid level until to a level which satisfies $N_x = N_y = N_z$. On this coarse grid, the discrete equation is actually the standard Poisson equation with equal meshsize. Starting from this grid level, the following coarsening strategy will be standard full coarsening; i.e., every other grid plane in all directions is eliminated. Using the same name as in [12], we still call this specialized coarsening process partial semi-coarsening strategy for 3D case. As in standard multigrid method, the coarsest grid will have only one unknown. For example, if we simplify to write $N_x = 128$, $N_y = 32$, $N_z = 8$ as $128 \times 32 \times 8$, Fig. 1 gives the process of the grid partial semi-coarsening.

3.2. Restriction and prolongation operators

Since the grid number on the x, y and z directions may not the same under the discretization with unequal meshsizes, standard multigrid restriction and prolongation operators can not be used in the process of partial semi-coarsening. So, we need to construct specified restriction and prolongation operators. For residual restriction operator, it is designed as follows:

(i) If $N_x > N_y > N_z$, the grid coarsening is only performed along the x direction. So, we just use a one-direction weighting average operator. The residual at the coarse grid points is computed by averaging the residual at the corresponding fine grid points and its two neighboring grid points in the x direction. Letting $r_{i,j,k}$ be the residual at fine grid point (i,j,k) and $\bar{r}_{\bar{i},j,k}$ the corresponding residual at coarse grid $(\bar{i},j,k)$, it is easy to know $i = 2\bar{i}$ and we use

$$\bar{r}_{\bar{i},j,k} = \frac{1}{4}(r_{i-1,j,k} + 2r_{i,j,k} + r_{i+1,j,k}).$$

(ii) If $N_x = N_y > N_z$, the grid coarsening is performed along both the x and y directions. We use a two-direction weighting average operator. The residual at the coarse grid points is computed by averaging the residual at the corresponding fine grid points and its eight neighboring grid points in both the x and y directions. Letting $\bar{r}_{\bar{i},\bar{j},k}$ be the corresponding residual at coarse grid $(\bar{i},\bar{j},k)$, under this condition $i = 2\bar{i}, j = 2\bar{j}$, we use

$$\bar{r}_{\bar{i},\bar{j},k} = \frac{1}{16}[4r_{i,j,k} + 2(r_{i+1,j,k} + r_{i-1,j,k} + r_{i,j+1,k} + r_{i,j-1,k}) + r_{i+1,j+1,k} + r_{i+1,j-1,k} + r_{i-1,j+1,k} + r_{i-1,j-1,k}].$$

(iii) If $N_x = N_y = N_z$, the grid is reduced to have equal meshsize in three directions and the grid coarsening is performed along all three directions. Letting $\bar{r}_{\bar{i},\bar{j},\bar{k}}$ be the corresponding residual at coarse grid $(\bar{i},\bar{j},\bar{k})$, we have $i = 2\bar{i}, j = 2\bar{j}, k = 2\bar{k}$ and we use the full-weighting operator [10,18]

$$\begin{aligned}\bar r_{\bar i,\bar j,\bar k} = \frac{1}{64}[&8r_{i,j,k} + 4(r_{i-1,j,k} + r_{i+1,j,k} + r_{i,j-1,k} + r_{i,j+1,k} + r_{i,j,k-1} + r_{i,j,k+1}) + 2(r_{i+1,j+1,k} + r_{i-1,j+1,k} + r_{i+1,j-1,k} + r_{i-1,j-1,k} \\ &+ r_{i,j+1,k-1} + r_{i,j+1,k+1} + r_{i,j-1,k-1} + r_{i,j-1,k+1} + r_{i+1,j,k+1} + r_{i+1,j,k-1} + r_{i-1,j,k-1} + r_{i-1,j,k+1}) + r_{i+1,j+1,k-1} + r_{i-1,j+1,k-1} \\ &+ r_{i+1,j-1,k-1} + r_{i-1,j-1,k-1} + r_{i+1,j+1,k+1} + r_{i-1,j+1,k+1} + r_{i+1,j-1,k+1} + r_{i-1,j-1,k+1}].\end{aligned}$$

For prolongation operator, we use a similar strategy.

(i) If $N_x > N_y > N_z$, corrections for the approximate solution at fine grid points corresponding to the coarse grid points are transferred directly. Corrections for other fine grid points take the average of the neighboring two grid points in the x direction only

$$\begin{aligned}r_{i,j,k} &= \bar r_{\bar i,j,k}, \\ r_{i-1,j,k} &= \frac{1}{2}(\bar r_{\bar i-1,j,k} + \bar r_{\bar i,j,k}).\end{aligned}$$

(ii) If $N_x = N_y > N_z$, corrections for the approximate solution at fine grid points corresponding to the coarse grid points are transferred directly. Corrections for other fine grid points are interpolated in the x and y directions simultaneously

$$\begin{aligned}r_{i,j,k} &= \bar r_{\bar i,j,k}, \\ r_{i-1,j,k} &= \frac{1}{2}(\bar r_{\bar i-1,j,k} + \bar r_{\bar i,j,k}), \\ r_{i,j-1,k} &= \frac{1}{2}(\bar r_{i,\bar j-1,k} + \bar r_{i,\bar j,k}), \\ r_{i-1,j-1,k} &= \frac{1}{4}(\bar r_{\bar i-1,\bar j,k} + \bar r_{\bar i,\bar j,k} + \bar r_{\bar i-1,\bar j-1,k} + \bar r_{\bar i,\bar j-1,k}).\end{aligned}$$

(iii) If $N_x = N_y = N_z$, we use the standard tri-linear interpolation operators [10,18]

$$\begin{aligned}r_{i,j,k} &= \bar r_{\bar i,j,k}, \\ r_{i-1,j,k} &= \frac{1}{2}(\bar r_{\bar i-1,j,k} + \bar r_{\bar i,j,k}), \\ r_{i,j-1,k} &= \frac{1}{2}(\bar r_{i,\bar j-1,k} + \bar r_{i,\bar j,k}), \\ r_{i,j,k-1} &= \frac{1}{2}(\bar r_{i,j,\bar k-1} + \bar r_{i,j,\bar k}), \\ r_{i-1,j-1,k} &= \frac{1}{4}(\bar r_{\bar i-1,\bar j,k} + \bar r_{\bar i,\bar j,k} + \bar r_{\bar i-1,\bar j-1,k} + \bar r_{\bar i,\bar j-1,k}), \\ r_{i-1,j,k-1} &= \frac{1}{4}(\bar r_{\bar i-1,j,\bar k} + \bar r_{\bar i,j,\bar k} + \bar r_{\bar i-1,j,\bar k-1} + \bar r_{\bar i,j,\bar k-1}), \\ r_{i,j-1,k-1} &= \frac{1}{4}(\bar r_{i,\bar j,\bar k-1} + \bar r_{i,\bar j,\bar k} + \bar r_{i,\bar j-1,\bar k-1} + \bar r_{i,\bar j-1,\bar k}),\end{aligned}$$

$$r_{i-1,j-1,k-1} = \frac{1}{8}(\bar r_{\bar i-1,\bar j,\bar k} + \bar r_{\bar i-1,\bar j,\bar k-1} + \bar r_{\bar i,\bar j,\bar k} + \bar r_{\bar i,\bar j,\bar k-1} + \bar r_{\bar i,\bar j-1,\bar k} + \bar r_{\bar i,\bar j-1,\bar k-1} + \bar r_{\bar i-1,\bar j-1,\bar k} + \bar r_{\bar i-1,\bar j-1,\bar k-1}).$$

3.3. Relaxation operators

For relaxation operators (smoothers), we use the point Gauss–Seidel relaxation in lexicographical ordering (point GS), in red–black ordering (red–black GS), in four-coloring (four-color GS) and plane Gauss–Seidel (plane GS) relaxation. Since we have supposed that the x direction is dominant direction and y sub-dominant direction, we use xoy plane relaxation which performs one x direction line Gauss–Seidel relaxation followed by sweeping along the y direction. We point out that four-coloring with the fourth-order compact difference scheme can decouple the grid points completely (see [15]), so there is inherent parallelism from this implementation. And the red–black Gauss–Seidel relaxation with the fourth-order compact difference scheme, in [10] for 3D Poisson equation, is shown to have a better smoothing effect than the lexicographical Gauss–Seidel relaxation although red–black ordering with the fourth-order compact scheme does not decouple the grid points completely. In the next section, we will study smoothing effect of different relaxation operators by numerical experiments.

4. Numerical experiments

To test the accuracy and efficiency of the fourth-order compact difference scheme and the multigrid methods for solving 3D Poisson Eq. (1), we choose an experimental problem which has exact solution $u(x,y,z) = \sin(10\pi x)\sin(3\pi y)\sin(\pi z)$ on the unit cubic domain $[0,1] \times [0,1] \times [0,1]$. This function is designed so that it changes most rapidly in the x direction, more rapidly in the y direction, and slowly in the z direction. Consequently, we use 3D grids with $N_x \geqslant N_y \geqslant N_z$. If the solution changes more rapidly in the y and z directions or x and z directions, etc., we just need modify the multigrid methods for the changed dominant directions and the present multigrid methods should be equally effective.

Both the fourth-order compact difference scheme and the standard second-order central difference scheme are employed with the multigrid methods to solve this problem and the computed results in terms of solution accuracy, multigrid convergence rate, and computational cost (CPU time) between two schemes are compared. The multigrid methods use V (1,1) cycle algorithm; i.e., it performs one relaxation on each grid level before restricting the residual to the coarse grid space (pre-smoothing) and performs one relaxation after interpolating the solution back to the fine grid space (post-smoothing). The iterative procedure is started with zero initial data and is terminated when the Euclidean norm (2-norm) of the residual vector is reduced by 10^{-10}. The code is written in Fortran 77 programming language with double precision arithmetic and all computations are run on a private computer with an Intel 2.4 GHz CPU and 2 GB memory.

4.1. Comparison of second- and fourth-order difference schemes

The fourth-order compact difference scheme is first compared with the second-order central difference scheme using the partial semi-coarsening multigrid method with the four-coloring Gauss–Seidel relaxation. The number of multigrid V(1,1) cycle iterations (I), the corresponding CPU time (T) in seconds as well as the maximum absolute errors (Error) between the computed solution and the exact solution over the entire fine grid points are given in Table 1. It shows that the partial semi-coarsening multigrid method with the fourth-order compact difference scheme has apparently fast convergence rate against the second-order central difference scheme. And the solution computed from the fourth-order compact difference scheme is much more accurate than that computed from the second-order central difference scheme. Thus, the fourth-order compact difference scheme is more cost-effective than the second-order central difference scheme for computing an approximate solution with a given accuracy; e.g., the maximum absolute error computed from the fourth-order compact scheme with $N_x = 32$, $N_y = 16$ and $N_z = 8$ is 3.581×10^{-4} and cost CPU time 0.063 s which is more accurate than the result 4.613×10^{-3} from the second-order central scheme with $N_x = N_y = N_z = 128$ and the CPU cost is 30.422 s. The results in Table

Table 1
The number of iterations (I), CPU times (T) and maximum absolute errors (Error) of multigrid method with four-coloring Guass–Seidel relaxation for the fourth-order compact difference scheme and the second-order central difference scheme.

N_x	N_y	N_z	Second-order central scheme			Fourth-order compact scheme		
			I	T	Error	I	T	Error
128	128	128	17	30.422	4.613(−3)	12	58.750	1.131(−5)
128	128	64	17	10.750	4.614(−3)	11	30.531	1.061(−5)
128	128	32	16	7.921	4.620(−3)	11	15.359	7.806(−6)
128	128	16	14	3.344	4.642(−3)	10	6.501	3.357(−6)
128	128	8	12	1.344	4.730(−3)	9	2.703	4.737(−5)
128	64	64	16	10.750	4.726(−3)	11	18.641	3.987(−6)
128	64	32	15	4.813	4.732(−3)	10	8.704	1.104(−6)
128	64	16	13	1.985	4.754(−3)	9	3.579	1.039(−5)
128	64	8	11	0.781	4.842(−3)	9	1.610	5.572(−5)
128	32	32	14	2.407	5.178(−3)	10	4.328	2.389(−5)
128	32	16	12	0.984	5.200(−3)	9	1.897	3.670(−5)
128	32	8	10	0.375	5.288(−3)	10	0.922	8.731(−5)
128	16	16	10	0.407	6.970(−3)	9	0.875	1.125(−4)
128	16	8	8	0.156	7.059(−3)	9	0.391	1.843(−4)
64	64	64	17	3.579	1.860(−2)	11	5.953	1.825(−4)
64	64	32	15	1.734	1.860(−2)	11	3.297	1.713(−4)
64	64	16	14	0.797	1.863(−2)	10	1.453	1.263(−4)
64	64	8	12	0.312	1.872(−2)	8	0.547	5.280(−5)
64	32	32	14	1.000	1.906(−2)	10	1.859	6.522(−5)
64	32	16	13	0.453	1.908(−2)	9	0.812	1.857(−5)
64	32	8	11	0.187	1.918(−2)	8	0.313	1.674(−4)
64	16	16	11	0.204	2.090(−2)	9	0.390	3.828(−4)
64	16	8	8	0.062	2.099(−2)	8	0.156	5.964(−4)
32	32	32	15	0.391	7.678(−2)	10	0.672	3.024(−3)
32	32	16	13	0.188	7.681(−2)	9	0.312	2.842(−3)
32	32	8	11	0.078	7.691(−2)	8	0.125	2.114(−3)
32	16	16	11	0.094	7.884(−2)	8	0.156	1.139(−3)
32	16	8	9	0.031	7.894(−2)	8	0.063	3.581(−4)

1 also show that the fourth-order compact scheme demonstrates fourth-order convergence while the second-order central scheme just demonstrates second-order convergence; e.g., the error 1.857×10^{-5} with $N_x = 64$, $N_y = 32$ and $N_z = 16$ is around one 16 of that with $N_x = 32$, $N_y = 16$ and $N_z = 8$, which is 3.581×10^{-4} for the fourth-order compact scheme while the error 1.908×10^{-2} with $N_x = 64$, $N_y = 32$ and $N_z = 16$ is just one fourth of that with $N_x = 32$, $N_y = 16$ and $N_z = 8$, which is 7.894×10^{-2} for the second-order central scheme. Similar comparison can be made with other data to get same conclusions. On the other hand, we also can see that for the anisotropic problem defined as above, increasing N_y and N_z does not always lead to reasonable increase in accuracy in the computed solution; e.g., for the fourth-order compact difference scheme, the least error 1.104×10^{-6} is achieved with $N_x = 128$, $N_y = 64$ and $N_z = 32$ in 8.704 s. Using equal grid $N_x = N_y = N_z = 128$ does not produce more accurate solutions (the same phenomenon also occurs in 2D case, see Ref. [12]). The cost, however, is increased to 58.750 s. This also tells us that it is necessary to use unequal meshsize discretization to solve an anisotropic problem.

4.2. Comparison of different multigrid methods

For 2D Poisson equation, Ref. [12] compares the convergence histories of the full-coarsening multigrid method with the four-coloring Gauss–Seidel relaxation, the partial semi-coarsening multigrid method with the four-coloring Gauss–Seidel relaxation, and the full-coarsening multigrid method with the line Gauss–Seidel relaxation to find that the later two methods converge fast while the standard full-coarsening multigrid with the four-coloring Gauss–Seidel relaxation takes more iterations to converge. For 3D Poisson equation, we still compare the three multigrid methods in Figs. 2 and 3. It needs pointing out that the line Gauss–Seidel relaxation in 2D is replaced by the *xoy* plane Gauss–Seidel relaxation in 3D. In Fig. 2, we first choose $N_x = 128$, $N_y = 64$ and $N_z = 32$ so that the anisotropy is moderate. Then, we increase the anisotropy by choosing $N_x = 128$, $N_y = 32$ and $N_z = 16$, $N_x = 128$, $N_y = 16$ and $N_z = 8$. In Fig. 3, we first choose $N_x = 64$, $N_y = 32$ and $N_z = 16$, then $N_x = 64$, $N_y = 32$ and $N_z = 8$, finally $N_x = 64$, $N_y = 16$ and $N_z = 8$. We can see that both the partial semi-coarsening and the plane Gauss–Seidel relaxation multigrid methods still work very well and their convergence rates are nearly not affected by the change of meshsizes in the y and z directions. For all computed cases, it just needs no more than 10 iterations of the partial semi-coarsening multigrid method and 15 iterations of the plane relaxation multigrid method. In particular, in a few cases, when anisotropy is strong (Figs. 2(c) and 3(c)), the full-coarsening multigrid method with the *xoy* plane relaxation is the most efficient. However, the full-coarsening multigrid method with the four-coloring Gauss–Seidel relaxation does not work well for 3D case. It needs more than 40 iterations of the full-coarsening multigrid method to converge and the iteration number increases with the increase of the anisotropy. For $N_x = 128$, $N_y = 16$ and $N_z = 8$, the anisotropy is the strongest, the iteration number of the full -coarsening multigrid method with the four-coloring Gauss–Seidel relaxation is more than 100. These comparison shows that both multigrid methods with the plane Gauss–Seidel relaxation and partial semi-coarsening are efficient solvers for solving 3D anisotropic problems.

4.3. Comparison of different multigrid relaxation operators

For the fourth-order compact difference scheme, we compare multigrid methods with different Gauss–Seidel relaxation operators. In Table 2, the multigrid methods with the point Gauss–Seidel relaxation, the red–black Gauss–Seidel relaxation, the four-coloring Gauss–Seidel relaxation and the *xoy* plane Gauss–Seidel relaxation are compared in terms of the number of iterations and CPU time in seconds. All relaxations are implemented with semi-coarsening if $N_x \neq N_y$ or $N_y \neq N_z$. We can see that when the anisotropy is moderate, the four-coloring Gauss–Seidel relaxation and the red–black Gauss–Seidel relaxation

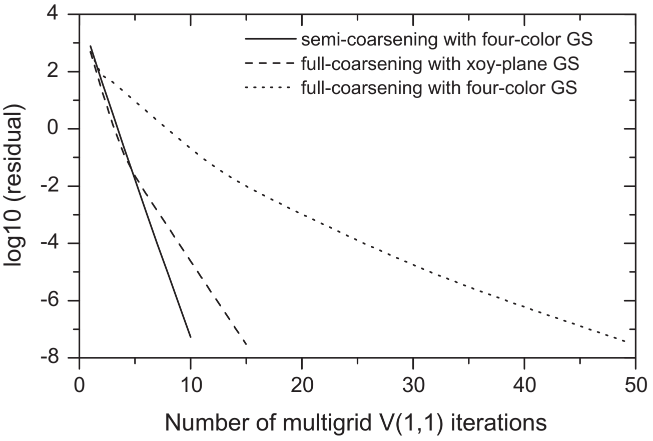

Fig. 2(a). Comparison of convergence rates of the partial semi-coarsening multigrid method with four-color GS relaxation, the full-coarsening multigrid method with *xoy* plane GS relaxation and the full-coarsening multigrid method with four-color GS relaxation. On the finest grid, $N_x = 128$, $N_y = 64$, $N_z = 32$.

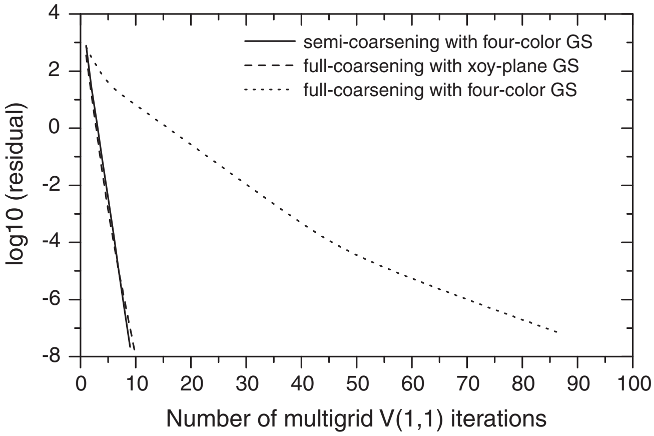

Fig. 2(b). Comparison of convergence rates of the partial semi-coarsening multigrid method with four-color GS relaxation, the full-coarsening multigrid method with *xoy* plane GS relaxation and the full-coarsening multigrid method with four-color GS relaxation. On the finest grid, $N_x = 128$, $N_y = 32$, $N_z = 16$.

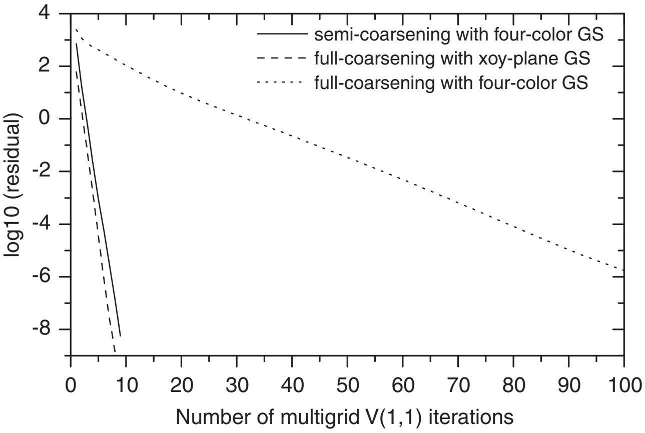

Fig. 2(c). Comparison of convergence rates of the partial semi-coarsening multigrid method with four-color GS relaxation, the full-coarsening multigrid method with *xoy* plane GS relaxation and the full-coarsening multigrid method with four-color GS relaxation. On the finest grid, $N_x = 128$, $N_y = 16$, $N_z = 8$.

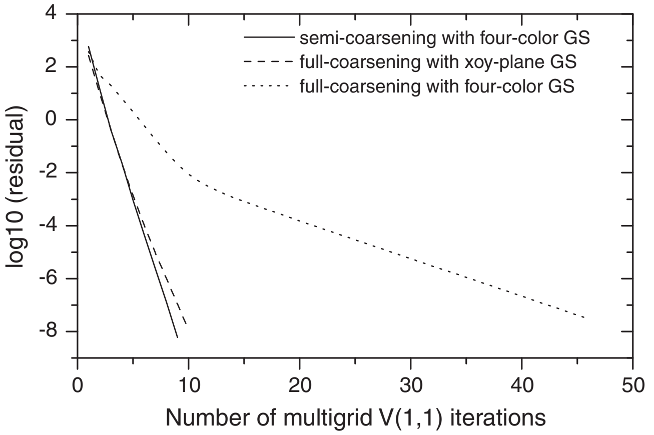

Fig. 3(a). Comparison of convergence rates of the partial semi-coarsening multigrid method with four-color GS relaxation, the full-coarsening multigrid method with *xoy* plane GS relaxation and the full-coarsening multigrid method with four-color GS relaxation. On the finest grid, $N_x = 64$, $N_y = 32$, $N_z = 16$.

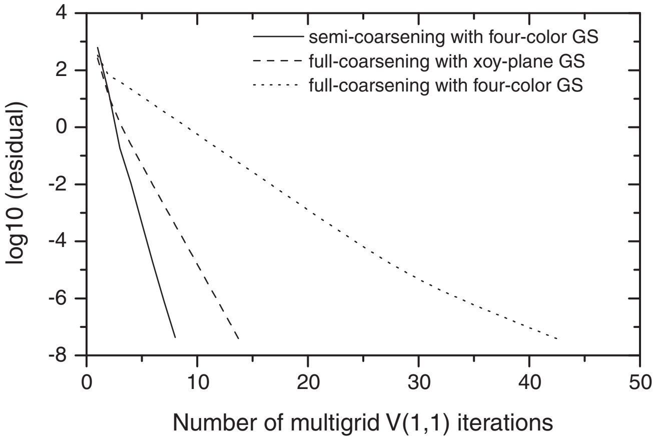

Fig. 3(b). Comparison of convergence rates of the partial semi-coarsening multigrid method with four-color GS relaxation, the full-coarsening multigrid method with *xoy* plane GS relaxation and the full-coarsening multigrid method with four-color GS relaxation. On the finest grid, $N_x = 64$, $N_y = 32$, $N_z = 8$.

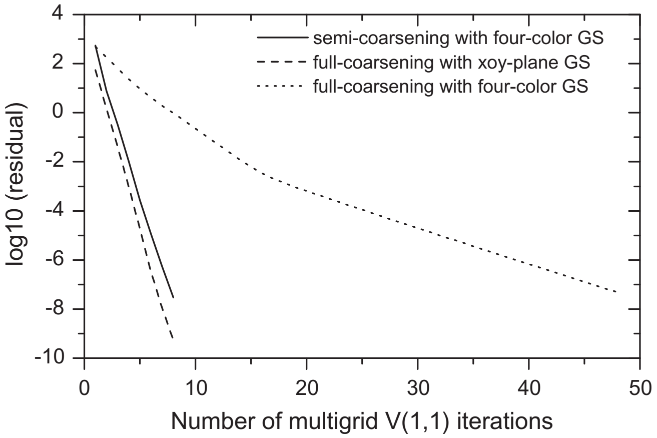

Fig. 3(c). Comparison of convergence rates of the partial semi-coarsening multigrid method with four-color GS relaxation, the full-coarsening multigrid method with *xoy* plane GS relaxation and the full-coarsening multigrid method with four-color GS relaxation. On the finest grid, $N_x = 64, N_y = 16$, $N_z = 8$.

are more robust than the point Gauss–Seidel relaxation and the *xoy* plane Gauss–Seidel relaxation. When anisotropy is strong, the *xoy* plane Gauss–Seidel relaxation shows the most superior. In a few cases, the *xoy* plane Gauss–Seidel relaxation achieves the fewest number of iterations among the four relaxation operators. However, as far as the CPU time is concerned, the four-coloring Gauss–Seidel relaxation, due to its inherent parallelism, takes the least CPU time and thus is the most cost-effective.

5. Concluding remarks

In this paper, a fourth-order compact difference discretization scheme with unequal meshsizes in different coordinate directions is employed to solve 3D Poisson equation. Then, the line Gauss–Seidel relaxation and the partial semi-coarsening multigrid methods proposed in [12] for solving 2D Poisson equation are generalized to solve 3D problems. Although Ref. [12] declares that the generalizations are not straightforward and their implementations are nontrivial, we fulfill it! It needs pointing out that the full-coarsening multigrid method with the plane relaxation, which uses the line Gauss–Seidel relaxation to compute each planewise solution, still works well as the line Gauss–Seidel relaxation does for solving 2D Poisson equation. In particular, in the partial semi-coarsening multigrid method, the four-coloring and red–black Gauss–Seidel relaxation are shown to be more robust when the anisotropy is moderate and the plane Gauss–Seidel relaxation is the most efficient when anisotropy is strong. As far as the computational cost is concerned, the four-coloring Gauss–Seidel relaxation takes the least CPU time and is the most cost-effective.

As we know, some fourth-order compact difference schemes have been developed to solve 2D incompressible Navier–Stokes equations [19–22], 3D convection diffusions [15,16,23] and the general 3D linear elliptic partial differential equations

Table 2
Comparison of the number of iterations (I) and CPU time (T) for different multigrid smoothers with the fourth-order compact difference scheme.

N_x	N_y	N_z	Point GS		Red–black GS		Four-color GS		Plane GS	
			I	T	I	T	I	T	I	T
128	128	128	13	59.859	12	59.079	12	58.750	13	63.141
128	128	64	13	34.281	12	33.250	11	30.531	12	33.594
128	128	32	12	16.063	11	15.500	11	15.359	13	18.140
128	128	16	12	7.422	10	6.641	10	6.501	13	8.421
128	128	8	11	3.141	9	2.750	9	2.703	13	3.875
128	64	64	14	23.297	11	18.781	11	18.641	11	19.625
128	64	32	15	12.656	10	8.782	10	8.704	10	9.157
128	64	16	15	5.750	9	3.609	9	3.579	10	4.156
128	64	8	14	2.422	9	1.625	9	1.610	9	1.719
128	32	32	16	7.016	11	4.938	10	4.328	9	4.484
128	32	16	16	3.234	11	2.281	9	1.897	8	1.921
128	32	8	15	1.343	10	0.938	10	0.922	8	0.970
128	16	16	17	1.657	10	1.106	9	0.875	8	0.906
128	16	8	15	0.656	9	0.406	9	0.391	8	0.395
64	64	64	13	6.984	11	6.031	11	5.953	13	7.359
64	64	32	12	3.609	11	3.329	11	3.297	12	3.797
64	64	16	12	1.735	10	1.468	10	1.453	12	1.813
64	64	8	11	0.734	9	0.609	8	0.547	12	0.843
64	32	32	13	2.391	10	1.875	10	1.859	10	1.984
64	32	16	14	1.204	9	0.812	9	0.812	9	0.860
64	32	8	14	0.515	9	0.344	8	0.313	8	0.344
64	16	16	16	0.672	9	0.391	9	0.390	8	0.391
64	16	8	15	0.281	9	0.187	8	0.156	7	0.156
32	32	32	12	0.782	10	0.687	10	0.672	12	0.891
32	32	16	11	0.375	10	0.360	9	0.312	10	0.375
32	32	8	10	0.141	8	0.125	8	0.125	10	0.156
32	16	16	12	0.235	8	0.156	8	0.156	8	0.172
32	16	8	12	0.109	8	0.094	8	0.063	7	0.078

[24] and some multigrid methods are developed [25–27]. Most high-order compact finite difference schemes and multigrid strategies are implemented on equal-meshsize-discretized grids [15,16,19–27]. So, developing high-order compact difference discretization schemes with unequal meshsizes and their specialized multigrid methods for these equations is meaningful. On the other hand, recently, Wang and Zhang [28] developed a sixth-order finite difference strategy which is based on Richardson extrapolation, an operator interpolation scheme and a multiscale multigrid method to solve 2D Poisson equation. The strategy can also be generalized to 3D differential equations mentioned above with unequal meshsize discretization.

Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grant 10502026 and the Natural Science Foundation of Ningxia under Grant NZ0937. The author also give his thanks to Dr. Jun Zhang, who opens the program code named mg3dcv.f on his personal website http://www.cs.uky.edu/~jzhang. The author conveniently developed the specialized multigrid restriction and prolongation operators from his original subroutine program code for the standard 3D restriction and prolongation operators.

References

[1] Y. Kwon, J.W. Stephenson, Single cell finite difference approximations for Poisson equation in three variables, Appl. Math. Notes 2 (1982) 13–17.
[2] M.M. Gupta, A fourth-order Poisson solver, J. Comput. Phys. 55 (1985) 166–172.
[3] J.C. Strikwerda, Finite difference schemes and partial differential equations, Chapman & Hall, 1989.
[4] W.F. Spotz, G.F. Carey, A high-order compact formulation for the 3D poisson equation, Numer. Methods Partial Differ. Eqn. 12 (1996) 235–243.
[5] M.M. Gupta, J. Kouatchou, Symbolic derivation of finite difference approximations for the three-dimensional Poisson equation, Numer. Methods Partial Differ. Eqn. 14 (1998) 593–606.
[6] J. Wang, W. Zhong, J. Zhang, A general meshsize fourth-order compact difference discretization scheme for 3D Poisson equation, Appl. Math. Comput. 183 (2006) 804–812.
[7] G. Sutmann, B. Steffen, High-order compact solvers for the three-dimensional Poisson equation, J. Comput. Appl. Math. 187 (2006) 142–170.
[8] M.M. Gupta, J. Kouatchou, J. Zhang, Comparison of second-order and fourth-order discretization for multigrid Poisson solvers, J. Comput. Phys. 132 (1997) 226–232.
[9] M. Othman, A.R. Abdullah, An efficient multigrid Poisson solver, Int. J. Comput. Math. 71 (1999) 541–553.
[10] J. Zhang, Fast and high accuracy multigrid solution of the three dimensional Poisson equation, J. Comput. Phys. 143 (1998). 449-161.
[11] J. Kouatchou, J. Zhang, Optimal injection operator and high order schemes for multigrid solution of 3D Poisson equation, Int. J. Comput. Math. 76 (2000) 173–190.
[12] J. Zhang, Multigrid method and fourth-order compact scheme for 2D Poisson equation with unequal mesh-size discretization, J. Comput. Phys. 179 (2002) 170–179.
[13] M.M. Gupta, J. Kouatchou, J. Zhang, A compact multigrid solver for convection–diffusion equations, J. Comput. Phys. 132 (1997) 123–129.

[14] J. Zhang, Accelerated multigrid high accuracy solution of the convection–diffusion equation with high Reynolds number, Numer. Methods Partial Differ. Eqn. 13 (1997) 77–92.

[15] M.M. Gupta, J. Zhang, High accuracy multigrid solution of the 3D convection–diffusion equation, Appl. Math. Comput. 113 (2000) 249–274.

[16] 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. Simulat. 54 (2000) 65–80.

[17] 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 (2002) 4661–4674.

[18] P. Wesseling, An Introduction to Multigrid Methods, Wiley, Chichester, 1992.

[19] 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 30 (1995) 1137–1151.

[20] 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.

[21] Z.F. Tian, Y.B. Ge, A fourth-order compact finite difference scheme for the steady stream function vorticity formulation of the Navier–Stokes/Boussinesq equations, Int. J. Numer. Methods Fluids 41 (2003) 495–518.

[22] E. Erturk, C. Gokcol, Fourth-order compact formulation of Navier–Stokes equations and driven cavity flow at high Reynolds numbers, Int. J. Numer. Methods Fluids 50 (2006) 421–436.

[23] J. Zhang, An explicit fourth-order compact finite difference scheme for three dimensional convection–diffusion equation, Commun. Numer. Methods Eng. 14 (1998) 263–280.

[24] L. Ge, J. Zhang, Symbolic computation of high order compact difference schemes for three dimensional linear elliptic partial differential equations with variable coefficients, J. Comput. Appl. Math. 143 (2002) 9–27.

[25] M.M. Gupta, High accuracy solutions of incompressible Navier–Stokes equations, J. Comput. Phys. 93 (1991) 343–359.

[26] Y.B. Ge, Z.F. Tian, W.Q. Wu, Simulation of time-dependent viscous flows in an cavity with high accuracy multigrid method, in: Proceedings of the 4th International Conference on Nonlinear Mechanics, Shanghai, 2002, pp. 980–986.

[27] J. Zhang, Numerical simulation of 2D square driven cavity using fourth-order compact finite difference schemes, Comput. Math. Appl. 45 (2003) 43–52.

[28] Y. Wang, J. Zhang, Sixth order compact scheme combined with multigrid method and extrapolation technique for 2D Poisson equation, J. Comput. Phys. 228 (2009) 137–146.