

High Accuracy and Scalable Multiscale Multigrid Computation for 3D Convection Diffusion Equation

Yin Wang^{1*} and Jun Zhang^{2†}

Abstract

We present a sixth order explicit compact finite difference scheme to solve the three dimensional (3D) convection diffusion equation. We first use multiscale multigrid method to solve the linear systems arising from a 19-point fourth order discretization scheme to compute the fourth order solutions on both the coarse grid and the fine grid. Then an operator based interpolation scheme combined with an extrapolation technique is used to approximate the sixth order accurate solution on the fine grid. Since the multigrid method using a standard point relaxation smoother may fail to achieve the optimal grid independent convergence rate for solving convection diffusion equation with a high Reynolds number, we also implement the plane relaxation smoother in the multigrid solver to achieve better grid independency. Supporting numerical results are presented to demonstrate the efficiency and accuracy of the sixth order compact scheme (SOC), compared with the previously published fourth order compact scheme (FOC).

Keywords: convection diffusion equation, Reynolds number, multigrid method, Richardson extrapolation, sixth order compact scheme.

Mathematics Subject Classification: 65N06, 65N55, 65F10.

1 Introduction

In this paper, we consider the three dimensional (3D) convection diffusion equation as

$$u_{xx} + u_{yy} + u_{zz} + p(x, y, z)u_x + q(x, y, z)u_y + r(x, y, z)u_z = f(x, y, z), \quad (1)$$

^{*1} Laboratory for High Performance Scientific Computing and Computer Simulation, Department of Computer Science, University of Kentucky, Lexington, KY 40506-0046, USA. E-mail: ywangf@csr.uky.edu, URL: <http://www.csr.uky.edu/~ywangf>.

^{†2} Laboratory for High Performance Scientific Computing and Computer Simulation, Department of Computer Science, University of Kentucky, Lexington, KY 40506-0046, USA. E-mail: jzhang@cs.uky.edu, URL: <http://www.cs.uky.edu/~jzhang>. This author's research work was supported in part by NSF under grants CCF-0727600, in part by the Kentucky Science and Engineering Foundation under grant KSEF-148-502-06-186, and in part by NIH under grant 1 R01 HL086644-01.

for a specified forcing function f in a continuous domain Ω of a 3D space with suitable boundary conditions prescribed on $\partial\Omega$. Here Ω is assumed to be comprised of a union of rectangular solids. Functions p , q , r , f , and u are assumed to be continuously differentiable and have the required partial derivatives on Ω .

Eq. (1) is widely used to model the transport processes, including heat transfer and fluid flows [14, 18]. It describes the convection and diffusion of various physical quantities, e.g., momentum, heat, material concentrations, etc. The functions p , q , and r in Eq. (1) are convection coefficients, whose magnitude is usually referred to be the Reynolds number in literature. When the Reynolds number is bigger than one (large convection coefficients), Eq. (1) is considered as convection dominated and is difficult to solve numerically [26, 29].

Compared with lower dimensional problems, the numerical simulation of 3D problems tends to be computationally intensive because it requires much more memory space and CPU time to obtain solutions with desired accuracy. So the direct solution methods based on Gaussian elimination are not widely used because they scale poorly with the problem size when the memory space and computational cost become an issue [3]. Thus iterative solution methods are considered as the best choice in such situations. For the convection dominated problems, traditional iterative methods may fail to converge for solving the linear systems arising from the second order central difference scheme (CDS) and the CDS scheme may produce nonphysical oscillations for large Reynolds number. The upwind difference scheme is usually stable but it will reduce the order of accuracy to the first order.

One approach to reducing computational cost, keeping good numerical stability and yielding high accuracy approximations is to utilize the higher order discretization schemes. Although the higher order discretization schemes need more complex procedures to compute the coefficient matrix, they use relatively coarser mesh griddings to achieve approximate solutions of comparable accuracy, relative to the lower discretization schemes using finer mesh griddings. The higher order schemes are reported to be efficient [1, 7, 9] and some of them may have an additional advantage of suppressing nonphysical numerical oscillations [21]. For 3D convection diffusion equation and Poisson equation, there have been several reported papers on using higher order discretization schemes [2, 8, 19, 28].

However, sixth order compact approximation for the 3D convection diffusion equation is still an open question. Compared with implicit compact schemes, the high order explicit compact schemes are more complicated to develop in higher dimensions. As far as we know, there is no existing explicit compact scheme on a single scale grid that is higher than fourth order for the 3D convection diffusion equation. One possible approach is to use the multiscale grid computation combined with an extrapolation technique. For the two dimensional (2D) case, we have developed the sixth order accuracy compact scheme

to solve the Poisson equation and convection diffusion equation [22, 23]. It was found that the sixth order scheme is computationally more efficient than the fourth order scheme. To obtain a computed solution of given accuracy, the sixth order scheme uses less memory and computational cost than the fourth order scheme does. Since there has been no such sixth order compact scheme for the 3D convection diffusion equation, the aim of this paper is to fill this gap.

This paper is organized as follows. In Section 2, we introduce the standard explicit fourth order compact finite difference scheme and our operator based interpolation scheme to approximate the sixth order fine grid solution. Section 3 contains the solution strategies for the resulting linear systems, including the multiscale multigrid method, plane relaxation smoother, and the residual scaling technique. Supporting numerical results are provided in Section 4. Conclusions are given in Section 5.

2 Finite Difference Scheme

Our sixth order compact finite difference (SOC) scheme is based on the fourth order (FOC) discretization scheme. We first use multiscale multigrid method to compute the converged fourth order solutions on both the fine grid and the coarse grid. Then an operator based interpolation scheme combined with the Richardson extrapolation technique is used to approximate the sixth order fine grid solution. In this paper, we use the FOC scheme from Zhang's paper [28].

2.1 Fourth Order Discretization

We assume the discretization is done on an uniform grid with meshsize h . We use u_0 to denote the approximate value of $u(x, y, z)$ at an internal mesh point (i, j, k) . The approximate values of its immediate 18 neighboring points are denoted by u_l , $l = 1, 2, \dots, 18$, as in Fig. 1. The 8 corner points, which are the white colored points in Fig. 1, are not used in the finite difference scheme. The discrete values of p_l , q_l , r_l and f_l for $l = 0, 1, \dots, 6$, are defined similarly.

Zhang's explicit fourth order compact scheme for Eq. (1) is derived from the general implicit formula by Ananthakrishnaiah et al. [2], which yields a 19-point formula as

$$\sum_{l=0}^{18} \alpha_l u_l = F_0, \quad (2)$$

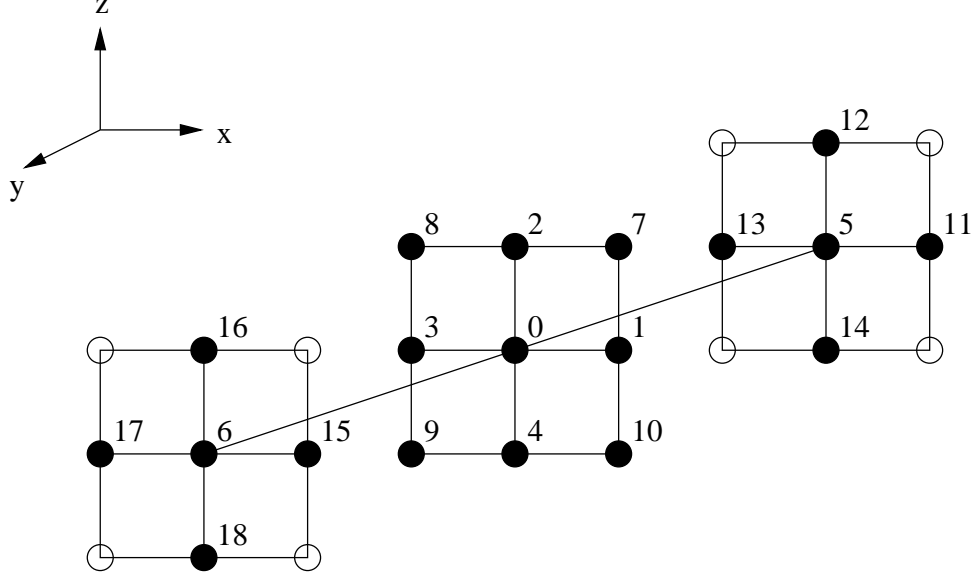


Figure 1: Labeling of the 3D grid points in a unit cube.

where the coefficients α_i and the right hand side F_0 are given by

$$\begin{aligned}
\alpha_0 &= -[24 + h^2(p_0^2 + q_0^2 + r_0^2) + h(p_1 - p_3 + q_2 - q_4 + r_5 - r_6)], \\
\alpha_1 &= 2 - \frac{h}{4}(2p_0 - 3p_1 - p_2 + p_3 - p_4 - p_5 - p_6) \\
&\quad + \frac{h^2}{8}[4p_0^2 + p_0(p_1 - p_3) + q_0(p_2 - p_4) + r_0(p_5 - p_6)], \\
\alpha_2 &= 2 - \frac{h}{4}(2q_0 - q_1 - 3q_2 - q_3 + q_4 - q_5 - q_6) \\
&\quad + \frac{h^2}{8}[4q_0^2 + p_0(q_1 - q_3) + q_0(q_2 - q_4) + r_0(q_5 - q_6)], \\
\alpha_3 &= 2 + \frac{h}{4}(2p_0 + p_1 - p_2 - 3p_3 - p_4 - p_5 - p_6) \\
&\quad + \frac{h^2}{8}[4p_0^2 - p_0(p_1 - p_3) - q_0(p_2 - p_4) - r_0(p_5 - p_6)], \\
\alpha_4 &= 2 + \frac{h}{4}(2q_0 - q_1 + q_2 - q_3 - 3q_4 - q_5 - q_6) \\
&\quad + \frac{h^2}{8}[4q_0^2 - p_0(q_1 - q_3) - q_0(q_2 - q_4) - r_0(q_5 - q_6)], \\
\alpha_5 &= 2 - \frac{h}{4}(2r_0 - r_1 - r_2 - r_3 - r_4 - 3r_5 + r_6) \\
&\quad + \frac{h^2}{8}[4r_0^2 + p_0(r_1 - r_3) + q_0(r_2 - r_4) + r_0(r_5 - r_6)], \\
\alpha_6 &= 2 + \frac{h}{4}(2r_0 - r_1 - r_2 - r_3 - r_4 + r_5 - 3r_6) \\
&\quad + \frac{h^2}{8}[4r_0^2 - p_0(r_1 - r_3) - q_0(r_2 - r_4) - r_0(r_5 - r_6)],
\end{aligned}$$

$$\begin{aligned}
\alpha_7 &= 1 + \frac{h}{2}(p_0 + q_0) + \frac{h}{8}(p_2 - p_4 + q_1 - q_3) + \frac{h^2}{4}p_0q_0, \\
\alpha_8 &= 1 - \frac{h}{2}(p_0 - q_0) - \frac{h}{8}(p_2 - p_4 + q_1 - q_3) - \frac{h^2}{4}p_0q_0, \\
\alpha_9 &= 1 - \frac{h}{2}(p_0 + q_0) + \frac{h}{8}(p_2 - p_4 + q_1 - q_3) + \frac{h^2}{4}p_0q_0, \\
\alpha_{10} &= 1 + \frac{h}{2}(p_0 - q_0) - \frac{h}{8}(p_2 - p_4 + q_1 - q_3) - \frac{h^2}{4}p_0q_0, \\
\alpha_{11} &= 1 + \frac{h}{2}(p_0 + r_0) + \frac{h}{8}(p_5 - p_6 + r_1 - r_3) + \frac{h^2}{4}p_0r_0, \\
\alpha_{12} &= 1 + \frac{h}{2}(q_0 + r_0) + \frac{h}{8}(q_5 - q_6 + r_2 - r_4) + \frac{h^2}{4}q_0r_0, \\
\alpha_{13} &= 1 - \frac{h}{2}(p_0 - r_0) - \frac{h}{8}(p_5 - p_6 + r_1 - r_3) - \frac{h^2}{4}p_0r_0, \\
\alpha_{14} &= 1 - \frac{h}{2}(q_0 - r_0) - \frac{h}{8}(q_5 - q_6 + r_2 - r_4) - \frac{h^2}{4}q_0r_0, \\
\alpha_{15} &= 1 + \frac{h}{2}(p_0 - r_0) - \frac{h}{8}(p_5 - p_6 + r_1 - r_3) - \frac{h^2}{4}p_0r_0, \\
\alpha_{16} &= 1 + \frac{h}{2}(q_0 - r_0) - \frac{h}{8}(q_5 - q_6 + r_2 - r_4) - \frac{h^2}{4}q_0r_0, \\
\alpha_{17} &= 1 - \frac{h}{2}(p_0 + r_0) + \frac{h}{8}(p_5 - p_6 + r_1 - r_3) + \frac{h^2}{4}p_0r_0, \\
\alpha_{18} &= 1 - \frac{h}{2}(q_0 + r_0) + \frac{h}{8}(q_5 - q_6 + r_2 - r_4) + \frac{h^2}{4}q_0r_0, \\
F_0 &= \frac{h^2}{2}(6f_0 + f_1 + f_2 + f_3 + f_4 + f_5 + f_6) \\
&\quad + \frac{h^3}{4}[p_0(f_1 - f_3) + q_0(f_2 - f_4) + r_0(f_5 - f_6)].
\end{aligned}$$

If we set the convection coefficients $p = q = r \equiv 0$, Eq. (1) will be reduced to 3D Poisson equation. The 19-point formula for the 3D Poisson equation can be reduced from the Zhang's fourth order scheme. It has been developed by several authors like Kwon et al. [12], Spatz and Carey [19].

We call the FOC scheme "compact" because it only involves the 18 neighboring grid points nearest to the reference grid point in a unit cube. Eq. (2) can be used for every grid point, no special formulas are needed for approximations at grid points near the boundaries.

By using Eq. (2), for every discrete grid point, we obtain a system of linear equations

$$Au = b. \tag{3}$$

Generally, the coefficient matrix A is very large and sparse. It is nonsymmetric and non-

positive definite if the convection coefficients are nonzero. When the convection diffusion equation is convection dominated, the matrix A loses its diagonal dominance. Even without the diagonal dominance, our previous work showed that there is no stability difficulty with the FOC scheme for both the 2D and 3D convection diffusion equations, and the multigrid method converges by using efficient smoothers and residual scaling techniques [7, 23, 25].

2.2 Operator Based Interpolation and Extrapolation

For 2D Poisson equation and convection diffusion equation, we proposed an operator based interpolation scheme combined with extrapolation technique to approximate the sixth order accurate fine grid solution [22, 23]. The numerical results show that our interpolation scheme is very efficient and accurate for the 2D problems.

The interpolation scheme for 2D problems is an iterative procedure combined with the Richardson extrapolation technique, which updates the solutions of grid points by groups in each iteration. For the 3D convection diffusion equation, the basic idea of the interpolation scheme is almost the same as for the 2D problems, but it needs more complicated grouping strategy. Like in Fig. 2, we divide the fine grid points into eight different groups by their *odd* or *even* indexing in the x , y and z -coordinate directions. Group a contains the $(\text{even}, \text{even}, \text{even})$ grid points on the Ω_h grid, which are the corresponding grid points on the Ω_{2h} coarse grid.

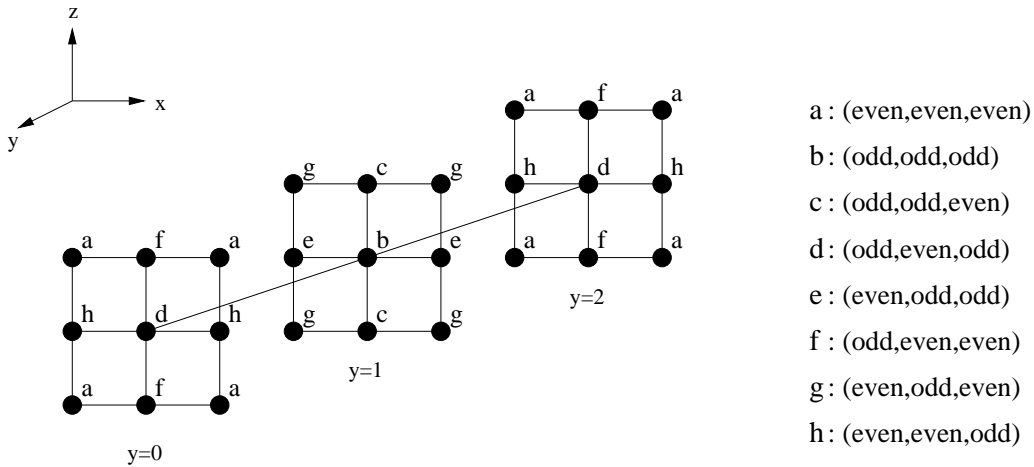


Figure 2: Group information of 3D grid points in a unit cube.

By solving the system of linear equations arising from the FOC scheme, we can get the fourth order solutions $u_{i,j,k}^h$ and $u_{i,j,k}^{2h}$ on Ω_h and Ω_{2h} , respectively. For the sixth order accurate solutions, it is easy to approximate the solution for the coarse grid Ω_{2h} by using

the Richardson extrapolation like [5, 15]

$$\tilde{u}_{i,j,k}^{2h} = \frac{2^m u_{2i,2j,2k}^h - u_{i,j,k}^{2h}}{2^m - 1}. \quad (4)$$

Here, m is the order of the computed solution accuracy from the FOC scheme. From [23], we know that the magnitude of the convection coefficients will affect the value of m . Generally, when the convection diffusion equation is diffusion dominated (small convection coefficients), we assume $m = 4$. If the convection diffusion equation is convection dominated (large convection coefficients), we need to compute the exact order of solution accuracy from the FOC scheme to use the Richardson extrapolation.

For the sixth order fine grid solution, we first directly interpolate the sixth order coarse grid solution $\tilde{u}_{i,j,k}^{2h}$ to the corresponding grid points in group a . Then we use an iterative mesh refinement interpolation technique to approximate sixth order solutions for the fine grid points in other groups. The detail of one iteration (from step n to step $n+1$) is outlined in Algorithm 2.2.

Algorithm 2.2: Sixth order operator based interpolation scheme for 3D convection diffusion equation.

- 1: Let $u_{old}^h = \tilde{u}^{h,n}$.
- 2: If $n = 0$, goto Step 3, else goto Step 4.
- 3: *Update every grid point in group a on Ω_h .*
 From $u_{i,j,k}^{2h} \in \Omega_{2h}^4$ and $u_{2i,2j,2k}^h \in \Omega_h^4$, we first compute $\tilde{u}_{i,j,k}^{2h,n+1} \in \Omega_{2h}^6$ by Eq. (4). Then we use direct interpolation to obtain $\tilde{u}_{2i,2j,2k}^{h,n+1} \in \Omega_h^6$.
- 4: *Update every grid point in groups c, d, and e on Ω_h .*
 For group c : (*odd, odd, even*) grid point (x_i, y_j, z_k) .
 The updated solution is approximated from Eq. (2) as

$$\begin{aligned} \tilde{u}_{i,j,k}^{h,n+1} = & [F_{i,j,k} - A_{i,j,k}(1)\tilde{u}_{i+1,j,k}^{h,n} - A_{i,j,k}(2)\tilde{u}_{i,j,k+1}^{h,n} - A_{i,j,k}(3)\tilde{u}_{i-1,j,k}^{h,n} \\ & - A_{i,j,k}(4)\tilde{u}_{i,j,k-1}^{h,n} - A_{i,j,k}(5)\tilde{u}_{i,j-1,k}^{h,n} - A_{i,j,k}(6)\tilde{u}_{i,j+1,k}^{h,n} - A_{i,j,k}(7)\tilde{u}_{i+1,j,k+1}^{h,n} \\ & - A_{i,j,k}(8)\tilde{u}_{i-1,j,k+1}^{h,n} - A_{i,j,k}(9)\tilde{u}_{i-1,j,k-1}^{h,n} - A_{i,j,k}(10)\tilde{u}_{i+1,j,k-1}^{h,n} \\ & - A_{i,j,k}(11)\tilde{u}_{i+1,j-1,k}^{h,n+1} - A_{i,j,k}(12)\tilde{u}_{i,j-1,k+1}^{h,n} - A_{i,j,k}(13)\tilde{u}_{i-1,j-1,k}^{h,n+1} \\ & - A_{i,j,k}(14)\tilde{u}_{i,j-1,k-1}^{h,n} - A_{i,j,k}(15)\tilde{u}_{i+1,j+1,k}^{h,n+1} - A_{i,j,k}(16)\tilde{u}_{i,j+1,k+1}^{h,n} \\ & - A_{i,j,k}(17)\tilde{u}_{i-1,j+1,k}^{h,n+1} - A_{i,j,k}(18)\tilde{u}_{i,j+1,k-1}^{h,n}] / A_{i,j,k}(0). \end{aligned}$$

Here, $F_{i,j,k}$ represents the right-hand side part of Eq. (2). The sixth order solutions for grid points in groups d and e are approximated similarly like those in group c . Each

grid point in these three groups has 4 neighboring fine grid points in group a .

5: *Update every grid point in groups f , g , and h on Ω_h .*

For group f : (*odd, even, even*) grid point (x_i, y_j, z_k) .

The updated solution is computed as

$$\begin{aligned} \tilde{u}_{i,j,k}^{h,n+1} = & [F_{i,j,k} - A_{i,j,k}(1)\tilde{u}_{i+1,j,k}^{h,n+1} - A_{i,j,k}(2)\tilde{u}_{i,j,k+1}^{h,n+1} - A_{i,j,k}(3)\tilde{u}_{i-1,j,k}^{h,n+1} \\ & - A_{i,j,k}(4)\tilde{u}_{i,j,k-1}^{h,n+1} - A_{i,j,k}(5)\tilde{u}_{i,j-1,k}^{h,n+1} - A_{i,j,k}(6)\tilde{u}_{i,j+1,k}^{h,n+1} - A_{i,j,k}(7)\tilde{u}_{i+1,j,k+1}^{h,n} \\ & - A_{i,j,k}(8)\tilde{u}_{i-1,j,k+1}^{h,n} - A_{i,j,k}(9)\tilde{u}_{i-1,j,k-1}^{h,n} - A_{i,j,k}(10)\tilde{u}_{i+1,j,k-1}^{h,n} \\ & - A_{i,j,k}(11)\tilde{u}_{i+1,j-1,k}^{h,n} - A_{i,j,k}(12)\tilde{u}_{i,j-1,k+1}^{h,n} - A_{i,j,k}(13)\tilde{u}_{i-1,j-1,k}^{h,n} \\ & - A_{i,j,k}(14)\tilde{u}_{i,j-1,k-1}^{h,n} - A_{i,j,k}(15)\tilde{u}_{i+1,j+1,k}^{h,n} - A_{i,j,k}(16)\tilde{u}_{i,j+1,k+1}^{h,n} \\ & - A_{i,j,k}(17)\tilde{u}_{i-1,j+1,k}^{h,n} - A_{i,j,k}(18)\tilde{u}_{i,j+1,k-1}^{h,n}] / A_{i,j,k}(0). \end{aligned}$$

The sixth order solutions for grid points in groups g and h are approximated similarly like those in group f . Each grid point in these three groups has 2 neighboring fine grid points in group a .

6: *Update every grid point in group b on Ω_h .*

For group b : (*odd, odd, odd*) grid point (x_i, y_j, z_k) .

The updated solution is computed as

$$\begin{aligned} \tilde{u}_{i,j,k}^{h,n+1} = & [F_{i,j,k} - A_{i,j,k}(1)\tilde{u}_{i+1,j,k}^{h,n+1} - A_{i,j,k}(2)\tilde{u}_{i,j,k+1}^{h,n+1} - A_{i,j,k}(3)\tilde{u}_{i-1,j,k}^{h,n+1} \\ & - A_{i,j,k}(4)\tilde{u}_{i,j,k-1}^{h,n+1} - A_{i,j,k}(5)\tilde{u}_{i,j-1,k}^{h,n+1} - A_{i,j,k}(6)\tilde{u}_{i,j+1,k}^{h,n+1} - A_{i,j,k}(7)\tilde{u}_{i+1,j,k+1}^{h,n+1} \\ & - A_{i,j,k}(8)\tilde{u}_{i-1,j,k+1}^{h,n+1} - A_{i,j,k}(9)\tilde{u}_{i-1,j,k-1}^{h,n+1} - A_{i,j,k}(10)\tilde{u}_{i+1,j,k-1}^{h,n+1} \\ & - A_{i,j,k}(11)\tilde{u}_{i+1,j-1,k}^{h,n+1} - A_{i,j,k}(12)\tilde{u}_{i,j-1,k+1}^{h,n+1} - A_{i,j,k}(13)\tilde{u}_{i-1,j-1,k}^{h,n+1} \\ & - A_{i,j,k}(14)\tilde{u}_{i,j-1,k-1}^{h,n+1} - A_{i,j,k}(15)\tilde{u}_{i+1,j+1,k}^{h,n+1} - A_{i,j,k}(16)\tilde{u}_{i,j+1,k+1}^{h,n+1} \\ & - A_{i,j,k}(17)\tilde{u}_{i-1,j+1,k}^{h,n+1} - A_{i,j,k}(18)\tilde{u}_{i,j+1,k-1}^{h,n+1}] / A_{i,j,k}(0). \end{aligned}$$

Each grid point in group b has no neighboring fine grid points in group a .

7: Compute the 2-norm $R = \|\tilde{u}^{h,n+1} - u_{old}^h\|_2$. If R is bigger than a certain tolerance (10^{-10} in this paper), go back to Step 1.

In Algorithm 2.2, $A_{i,j,k}(l)$, $l = 0, 1, \dots, 18$, are the pre-computed coefficients for grid point (x_i, y_j, z_k) . Ω_h^4 and Ω_{2h}^4 denote the fourth order accurate solution space from the FOC schemes, Ω_h^6 and Ω_{2h}^6 are the improved sixth order accurate solution space. $\tilde{u}^{h,n}$ is the approximate solution for the fine grid after the n th iterations. We update the fine grid points group by group based on the number of their neighboring grid points with sixth order solution (group a) from Step 2. Grid points in groups c , d and e have more qualified

neighbors than those in other groups, so we update these three groups first. The iteration will continue until the 2-norm R of the correction vector is reduced to below a certain tolerance.

3 Solution Strategies

For solving the system of linear equations arising from the FOC scheme, it is well known that classical iterative methods like the Jacobi, Gauss-Seidel and SOR converge slowly for large linear systems. For the 2D convection diffusion equation, we proposed a multiscale multigrid method which has been shown to be very efficient and stable [23]. This method can also be used to solve the 3D convection diffusion equation.

3.1 Multiscale Multigrid Method

The multigrid method is among the fastest and most efficient iterative methods for solving linear system arising from the discretized PDEs. The multigrid method iterates on a hierarchy of successively coarser grids until the convergence is reached. For model problems, such as the idealized elliptic problems, the convergence rate of the multigrid method is independent of the grid size [4, 6, 24]. Various multigrid implementation strategies with the FOC scheme to solve the 2D and 3D convection diffusion equations are discussed in [7, 17, 25].

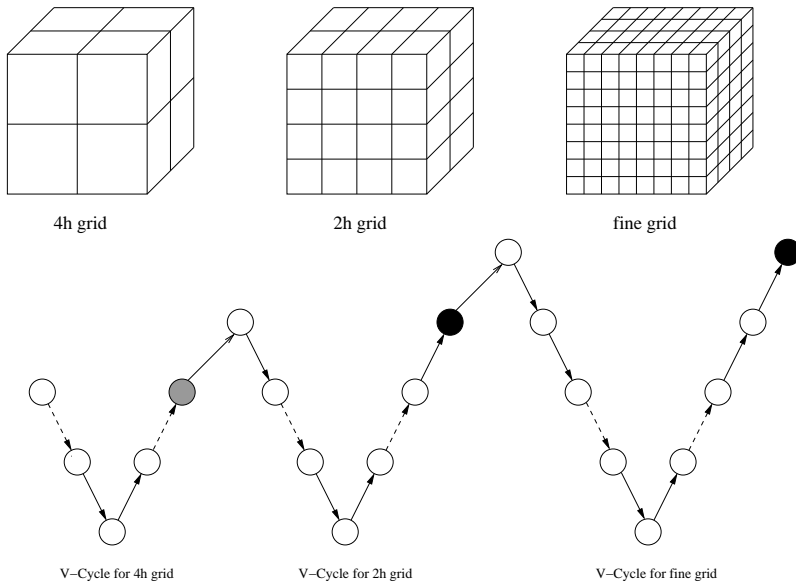


Figure 3: Representation of multiscale multigrid method for 3D convection diffusion equation.

Our multiscale multigrid method is based on the standard multigrid V-Cycle [16]. The V-Cycle is the computational process that goes from the fine grid down to the coarsest grid and then comes back from the coarsest grid up to the fine grid. From Fig. 3, we can see that the multiscale multigrid method is similar to the full multigrid method, but we do not start from the coarsest grid level. We describe it in Algorithm 3.1 as below:

Algorithm 3.1: Multiscale multigrid method.

- 1: Run the multigrid V-Cycle on $4h$ grid for one or two cycles to get an approximate solution u_{4h} .
- 2: Use high order interpolation scheme to interpolate u_{4h} to $2h$ grid as the initial guess.
- 3: Run the multigrid V-Cycle on $2h$ grid until it converges to get $4th$ order solution u_{2h} .
- 4: Use high order interpolation scheme to interpolate u_{2h} to h grid as the initial guess.
- 5: Run the multigrid V-Cycle on h grid until it converges to get $4th$ order solution u_h .

3.2 Relaxation Strategies in Multigrid Method

Standard multigrid method with point Gauss-Seidel relaxation is known to be highly efficient in solving systems of elliptic partial differential equations, but it fails to achieve optimal grid independent convergence rate for some convection diffusion equations like the convection dominated problems with high Reynolds number and the Poisson equation that has anisotropic discrete operators [7, 10, 11, 25]. There are at least two strategies to treat linear systems arising from these equations. The first strategy is to use semicoarsening, i.e., mesh coarsening is only performed along the dominated direction. The second strategy is to use alternating $(x - y)$ line Gauss-Seidel relaxation in the multiscale multigrid method for solving these 2D problems [22, 23] and use alternating plane relaxation smoother for these 3D problems. In this paper, we prefer to use the second one.

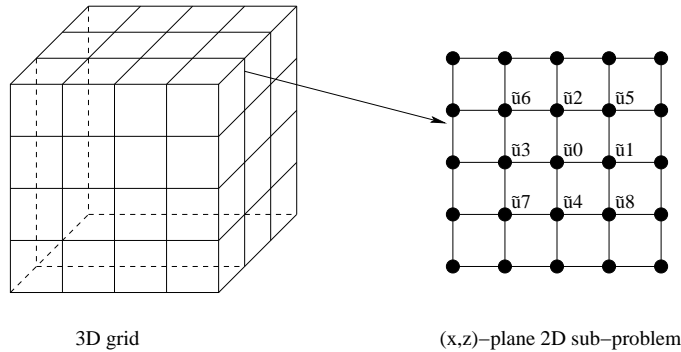


Figure 4: 2D sub-problem in plane relaxation.

The alternating $(x - y - z)$ plane Gauss-Seidel relaxation in lexicographic order performs one sweep of (y, z) -plane Gauss-Seidel relaxation along the x -coordinate direction first, followed by one sweep of (x, z) -plane Gauss-Seidel relaxation and (x, y) -plane Gauss-Seidel relaxation along the y -coordinate direction and z -coordinate direction, respectively. For each direction, we divide the 3D problem into N 2D sub-problems as in Fig. 4, where N is the number of intervals along that direction. Let us first consider the (x, z) -plane 2D sub-problem, its nine point computation stencil can be generated from the Eq. (2) as

$$\begin{aligned}
& A_0 \tilde{u}_0 + A_1 \tilde{u}_1 + A_2 \tilde{u}_2 + A_3 \tilde{u}_3 + A_4 \tilde{u}_4 + A_5 \tilde{u}_5 + A_6 \tilde{u}_6 + A_7 \tilde{u}_7 + A_8 \tilde{u}_8 \\
& = F_0 - \alpha_5 u_5 - \alpha_6 u_6 - \alpha_{11} u_{11} - \alpha_{12} u_{12} - \alpha_{13} u_{13} - \alpha_{14} u_{14} - \alpha_{15} u_{15} \\
& \quad - \alpha_{16} u_{16} - \alpha_{17} u_{17} - \alpha_{18} u_{18},
\end{aligned} \tag{5}$$

where the coefficients A_l and the 2D solution \tilde{u}_l ($l = 0, 1, \dots, 8$) are set as

$$A_l = \begin{cases} \alpha_l & 0 \leq l \leq 4, \\ \alpha_{l+2} & 5 \leq l \leq 8, \end{cases} \quad \tilde{u}_l = \begin{cases} u_l & 0 \leq l \leq 4, \\ u_{l+2} & 5 \leq l \leq 8. \end{cases}$$

The nine point computation stencil for the (x, y) and (y, z) -plane can be generated like Eq. (5) similarly, but the index of the 3D grid points for those corresponding 2D grid points are different.

The plane relaxation is considered in the multigrid literature to have poor numerical and parallel properties because it needs to solve a large number of 2D sub-problems. However, it is shown in [10] that an exact solution of the 2D sub-problems for planes is not needed and that one multigrid cycle is sufficient if we use multigrid method as the inner 2D solver. This behavior has been also reported by other researchers in [13, 20].

We would like to mention here that if we use the geometric multigrid method as the inner 2D solver to solve the 3D convection diffusion equation with variable coefficients, for some 2D planes, we are not able to compute the full coefficient matrix for their coarse grids. For example in Fig. 5, we consider a $5 \times 5 \times 5$ 3D grid. The 19 black color grid points are the coarse grid points that are needed for plane $y = 2$ to compute its 9 point computation stencil by Eq. (5). In comparison, the 19 gray color grid points are the coarse grid points needed for plane $y = 1$. We note that, plane $y = 1$ needs some coarse grid points from plane $y = -1$, which is not in the 3D cube, to generate its coarse grid coefficient matrix for solving the 2D sub-problem. Due to these drawbacks, we can only use the multigrid method as the inner 2D solver in plane relaxation for solving the 3D convection diffusion equation with constant coefficients. If we rewrite the Eq. (2) by using constant coefficients p , q , and r , the coefficients for each plane along the same direction are the same. For plane

$y = 1$ in Fig. 5, we do not need the additional information from plane $y = -1$, because every plane along the y direction has the same coarse grid coefficients.

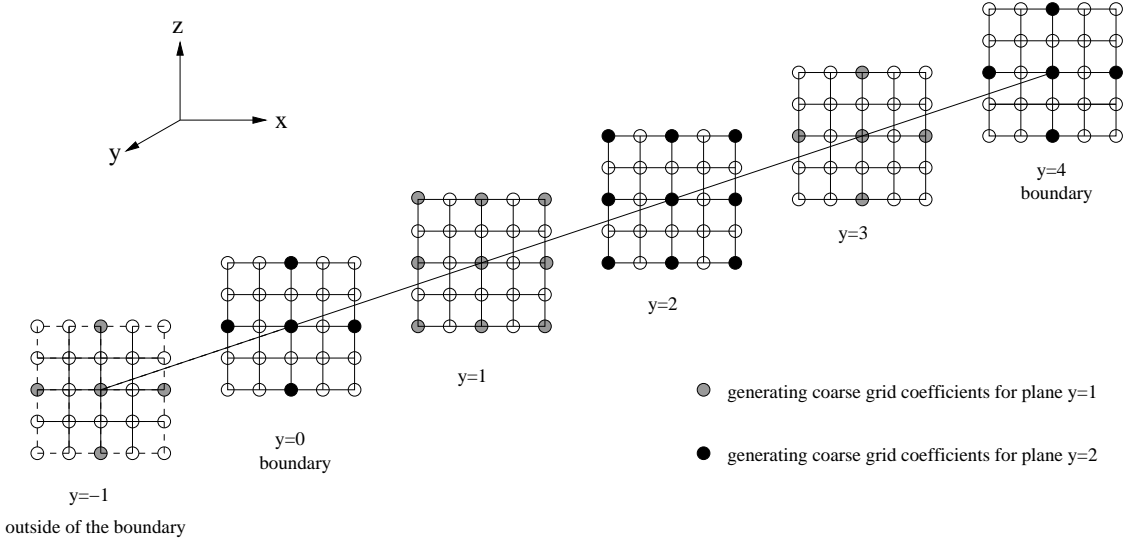


Figure 5: coarse grid of the 2D sub-problem in plane relaxation.

In order to keep the optimal convergence rate, for the inner 2D multigrid solver, we use alternating line relaxation smoother in the 2D V-Cycle because the 2D sub-problem may still have the high convection operator or the anisotropy.

3.3 Residual Scaling

From [25, 27, 30], we also know that merely using line relaxation in standard multigrid method still cannot give us the fast convergence for convection dominated 2D problems with relatively large Reynolds number. One simple approach is to properly scale the residual before it is projected to the coarse grid. We believe this approach combined with the plane relaxation can build a more efficient and scalable solver for the 3D convection diffusion equation with large Reynolds number.

The residual scaling procedure at the grid point (x_i, y_j, z_k) is like

$$\tilde{r}_{i,j,k} = \beta r_{i,j,k}.$$

Here, $r_{i,j,k}$ is the computed residual at the grid point (x_i, y_j, z_k) , $\tilde{r}_{i,j,k}$ is the residual after scaling, and β is a user input scaling factor to scale the residual. Readers are referred to [7, 30] for more details about how to choose the correct scaling factor. In this paper, we only list the optimal numerical results we got by testing different scaling factors.

4 Numerical Experiments

In our numerical experiments, we tested the sixth order scheme (SOC) and compared the results with the standard fourth order scheme (FOC). All computations were done on one processor of an IBM HS21 blade cluster at the University of Kentucky using Fortran 77 programming language. The processor has 2GB local memory and runs at 2.0GHZ.

The domain Ω for the two test problems we solved was chosen as the unit cube $(0, 1)^3$. We tested two problems with both constant and variable coefficients. We chose Problem 1 with variable coefficients, so we could not use the plane relaxation with inner multigrid 2D solver. We verified that it was sufficient to use the standard point Gauss-Seidel relaxation smoother to solve the 3D convection diffusion equation efficiently with relatively small Reynolds number and its efficiency was degraded when the Reynolds number increased. For Problem 2 with constant coefficients, we tested the plane Gauss-Seidel relaxation smoother with some large Reynolds number and compared the results with the point relaxation smoother.

We used standard V(1,1) cycle in the multiscale multigrid method. The initial guess for the V-Cycle on the $4h$ grid was the zero vector. The stopping criteria for the operator based interpolation and the V-Cycle on $2h$ and h grids were 10^{-10} . The errors reported were the maximum absolute errors over the discrete grid of the finest level. For the SOC 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.

4.1 Test Problem 1

The first test problem is

$$\begin{cases} u(x, y, z) = \cos(4x + 6y + 8z), \\ p(x, y, z) = Re \sin y \sin z \cos x, \\ q(x, y, z) = Re \sin x \sin z \cos y, \\ r(x, y, z) = Re \sin x \sin y \cos z. \end{cases}$$

This problem has variable coefficients and the constant Re represents the magnitude of the convection coefficients and simulates the Reynolds number in a flow simulation. The Dirichlet boundary conditions and the forcing term f are set to satisfy the exact solution.

We tested the first problem using point relaxation smoother for small to relatively large values of the Reynolds number ($Re \leq 10^5$). The numerical results are listed from Table 1 to Table 3.

Table 1 contains the numerical results for Problem 1 when $Re = 0$, which reduces it to

Table 1: Maximum errors, CPU seconds and the number of iterations of the FOC and SOC schemes for Problem 1 with $Re = 0$.

h	FOC Point				SOC Point			
	# iter	CPU	error	order	# iter	CPU	error	order
1/8	11	0.004	2.04e-3	4.2	(8,11),33	0.005	1.55e-3	5.0
1/16	12	0.031	1.09e-4	4.1	(11,12),42	0.051	4.90e-5	5.4
1/32	12	0.281	6.29e-6	4.0	(12,12),44	0.617	1.15e-6	5.8
1/64	12	2.412	3.76e-7	4.0	(12,11),43	6.234	2.14e-8	5.8

Table 2: Maximum errors, CPU seconds and the number of iterations of the FOC and SOC schemes for Problem 1 with $Re = 10$.

h	FOC Point				SOC Point			
	# iter	CPU	error	order	# iter	CPU	error	order
1/8	12	0.004	2.55e-3	4.0	(9,12),35	0.006	1.95e-3	5.0
1/16	13	0.032	1.41e-4	4.1	(12,13),46	0.056	6.13e-5	5.5
1/32	13	0.291	8.18e-6	4.1	(13,12),47	0.637	1.40e-6	5.8
1/64	12	2.397	4.90e-7	4.1	(12,12),44	6.531	2.56e-8	5.7

the 3D Poisson equation. Hereinafter, in all tables, the column titles # iter refers to the number of iterations; CPU refers to the CPU cost in seconds; error refers to the maximum absolute error; order refers to the order of accuracy for the computed solution.

We noted that, with the point relaxation smoother, both the SOC scheme and the FOC scheme solved the Problem 1 with optimal grid independent convergence rate. The order of the computed solution from the SOC scheme was close to 6 as we expected and was higher than that from the FOC scheme. In terms of computational cost with the same mesh size h , the FOC scheme was faster because it only ran a standard multigrid V-Cycle and the SOC scheme needed to run the multiscale multigrid method and the operator based interpolation scheme. For the computed solution accuracy, the SOC scheme was more accurate than the FOC scheme for every meshsize we tested, this can be seen from Table 1.

When we chose $Re = 10$, it was clear from Table 2 that our SOC scheme still yielded a sixth order solution accuracy for small Reynolds number though there was a slight increase in the number of iterations as Re was increased and our multiscale multigrid method still kept the grid independent convergence rate. Similar behavior was observed for the FOC scheme.

The numerical results in Table 1 and Table 2 indicated that using point relaxation smoother was sufficient to solve the 3D convection diffusion equation with relatively small Reynolds numbers. However, the data in Table 3 showed that when the magnitude of the Reynolds number was large enough ($Re = 10^5$), the convergence and the computed accuracy were severely degraded. The solution methods for both the FOC and the SOC schemes

Table 3: Maximum errors, CPU seconds and the number of iterations of the FOC and SOC schemes for Problem 1 with $Re = 10^5$.

h	FOC Point				SOC Point			
	# iter	CPU	error	order	# iter	CPU	error	order
1/8	51	0.009	8.40e-2	2.0	(18,53),67	0.014	7.24e-2	2.7
1/16	158	0.262	1.97e-2	2.1	(53,149),139	0.232	1.18e-2	2.9
1/32	521	7.847	4.77e-3	2.0	(150,505),276	7.463	1.57e-3	3.0
1/64	not converge	–	–	–	not converge	–	–	–

did not obtain the grid independent convergence. They took hundreds of multigrid cycles to converge when $h < 1/8$ and did not converge within the maximum number of iterations (1000) we set when $h = 1/64$.

4.2 Test Problem 2

For test problem 2, we chose the constant coefficients with large Reynolds number as

$$\begin{cases} u(x, y, z) = \cos(4x + 6y + 8z), \\ p(x, y, z) = q(x, y, z) = r(x, y, z) = Re. \end{cases}$$

We used both the point and the plane relaxation smoothers in the multiscale multigrid method to solve this problem.

We tested this problem with a very large Reynolds number ($Re = 10^7$) and the reported numerical results were listed in Table 4. We noted that the accuracy of the solutions computed by the multigrid method using plane relaxation smoother and point relaxation smoother was comparable. For the CPU cost with the same meshsize, the point relaxation method was faster than the plane relaxation method when $h \leq 1/16$, that was because the plane relaxation needed to run a large number of 2D V-Cycles. When the mesh became finer, the number of iterations of the point relaxation method increased very quickly and it did not converge when $h = 1/64$. On the other hand, the number of iterations of the plane relaxation method was almost stable with respect to the meshsize. For better understanding, we listed the graphic comparison of the number of iterations for different relaxation smoothers in Fig. 6. It was clear that the multigrid method using the plane relaxation smoother took much less iterations than that from point relaxation smoother and the grid independent convergence was kept when the mesh became finer.

Table 5 contained the test results with various Re and fixed meshsize $h = 1/32$. Computations were reported for the FOC and the SOC schemes using the plane relaxation smoother. We noted that the magnitude of the Reynolds number (Re) affected the convergence and the computed solution accuracy of both schemes inversely. The SOC scheme yielded better

Table 4: Maximum errors, CPU seconds and the number of iterations of Problem 2 using plane relaxation and point relaxation smoothers with $Re = 10^7$.

h	strategy	Point relaxation				Plane relaxation			
		# iter	CPU	error	order	# iter	CPU	error	order
1/8	FOC	52	0.009	6.10e-2	2.1	8	0.017	6.10e-2	2.1
	SOC	(16,51),55	0.010	5.18e-2	2.7	(4,8),55	0.019	5.18e-2	2.7
1/16	FOC	164	0.262	1.41e-2	2.1	17	0.320	1.41e-2	2.1
	SOC	(51,164),74	0.294	7.96e-3	3.0	(10,16),74	0.389	7.96e-3	3.0
1/32	FOC	500	7.621	3.36e-3	2.0	26	4.852	3.36e-3	2.0
	SOC	(164,472),79	7.808	1.01e-3	3.3	(16,26),79	3.247	1.01e-3	3.3
1/64	FOC	not converge	–	–	–	46	70.073	8.21e-4	2.0
	SOC	not converge	–	–	–	(26,45),81	80.787	1.06e-4	3.2

solution accuracy than the FOC scheme with every Re value we tested. When $Re > 10^4$, there was only a little change for the solution accuracy and the number of iterations for both the FOC and the SOC schemes. We believed that the convergence rate and the number of iterations approached some limits and did not deteriorate any more when Re was beyond certain large values.

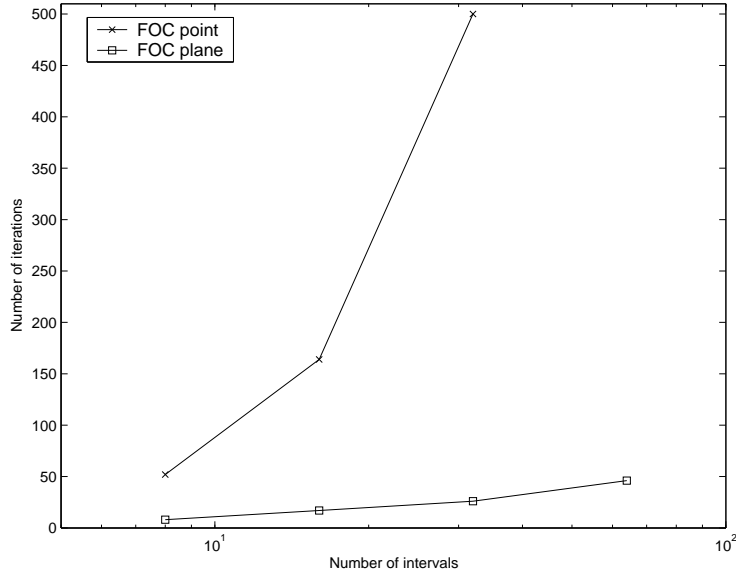


Figure 6: Comparison of the number of iterations of the FOC scheme for solving Problem 2 using plane relaxation and point relaxation($Re = 10^7$). Each symbol corresponds to an increasing fine grid: 8, 16, 32, and 64 intervals.

Table 5: Maximum errors, CPU seconds and the number of iterations of Problem 2 using plane smoother with different Re .

$Re(h = 1/32)$	FOC plane relaxation			SOC plane relaxation		
	# iter	CPU	error	# iter	CPU	error
0	6	1.062	6.29e-6	(6,6),45	1.482	1.15e-6
1	7	1.234	3.48e-5	(6,6),45	1.693	1.24e-6
10^2	12	2.109	4.84e-4	(10,12),48	2.925	2.87e-4
10^4	28	4.854	3.36e-3	(16,26),79	5.337	1.01e-3
10^5	26	4.518	3.36e-3	(16,26),79	5.349	1.01e-3
10^6	26	4.489	3.36e-3	(16,26),79	5.335	1.01e-3
10^7	26	4.455	3.36e-3	(16,26),79	5.367	1.01e-3
10^{10}	26	4.492	3.36e-3	(16,26),79	5.402	1.01e-3

5 Concluding Remarks

We derived a sixth order compact finite difference scheme for the 3D convection diffusion equation and solved the discrete problem by using multiscale multigrid method combined with an operator based interpolation scheme. Our numerical results indicated that the SOC scheme is more cost-effective than the FOC scheme to produce solution with comparable accuracy and the order of solution accuracy from the SOC scheme is higher than that from the FOC scheme.

The multiscale multigrid method using point relaxation smoother is sufficient to solve the convection diffusion equation with small Reynolds numbers. The plane relaxation smoother combined with the residual scaling technique is more scalable and efficient than the point relaxation smoother for solving convection diffusion equation with large Reynolds numbers.

References

- [1] Y. Adam. Highly accurate compact implicit methods and boundary conditions. *J. Comput. Phys.*, 24(1):10-22, 1977.
- [2] U. Ananthakrishnaiah, R. Monahar, and J. W. Stephenson. Fourth-order finite difference methods for three-dimensional general linear elliptic problems with variable coefficients. *Numer. Methods Partial Differential Equations*, 3:229-240, 1987.
- [3] O. Axelsson. *Iterative Solution Methods*. Cambridge Univ. Press, Cambridge, 1994.
- [4] A. Brandt. Multi-level adaptive solutions to boundary-value problems. *Math. Comput.*, 31(138):333-390, 1977.
- [5] C. Brezinski and M. R. Zaglia. *Extrapolation Method. Theory and Practice*, North-Holland, Berlin, 1991.

- [6] W. L. Briggs, V. E. Henson, and S. F. McCormick. *A Multigrid Tutorial*. SIAM, Philadelphia, PA, 2nd edition, 2000.
- [7] M. M. Gupta and J. Zhang. High accuracy multigrid solutions of the 3D convection-diffusion equation. *Appl. Math. Comput.*, 113(2-3):249-274, 2000.
- [8] M. M. Gupta, J. Kouatchou. Symbolic derivation of finite difference approximations for the three dimensional Poisson equation. *Numer. Methods Partial Differential Equations*, 18(5):593-606, 1998.
- [9] M. M. Gupta, J. Kouatchou, and J. Zhang. Comparison of second and fourth order discretization for multigrid Poisson solver. *J. Comput. Phys.*, 132:226-232, 1997.
- [10] I. M. Llorente and N. D. Melson. Behavior of plane relaxation methods as multigrid smoothers. *Electron. T. Numer. Ana.*, 10:92-114, 2000.
- [11] I. M. Llorente, M. Prieto, and B. Diskin. A parallel multigrid solver for 3D convection and convection-diffusion problems. *Parallel Comput.*, 27(13):1715-1741, 2001.
- [12] Y. Kwon, R. Manohar, and J. W. Stephenson. A single cell finite difference approximations for Poisson's equation in three variables. *Appl. Math. Notes*, 2:13-20, 1982.
- [13] C. W. Oosterlee. A GMRES-based plane smoother in multigrid to solve 3-D anisotropic fluid flow problems. *J. Comput. Phys.*, 130:41-53, 1997.
- [14] S. V. Pakantar. *Numerical Heat Transfer and Fluid Flow*. McGraw-Hill, New York, 1980.
- [15] L. F. Richardson. The approximate arithmetical solution by finite differences of physical problems including differential equations, with an application to the stresses in a masonry dam. *Philosophical Transactions of the Royal Society of London, Series A*, 210:307-357, 1910.
- [16] Y. Saad. *Iterative Methods for Sparse Linear Systems*, 2nd edition. SIAM, Philadelphia, PA, 2003.
- [17] S. Schaffer. High order multigrid method. *Math. Comput.*, 43(167): 89-115, 1984.
- [18] T. M. Shih. *Numerical Heat Transfer*. Hemisphere, Washington, DC, 1984.
- [19] W. F. Spitz and G. F. Carey. A high-order compact formulation for the 3D Poisson equation. *Number. Methods Partial Difference Equations*, 12:235-243, 1996.
- [20] C. Thole and U. Trottenberg. Basic smoothing procedures for the multigrid treatment of elliptic 3D operators. *Appl. Math. Comput.*, 19:333-345, 1986.

- [21] C. P. Tzanos. Higher-order difference method with a multigrid approach for the solution of the incompressible flow equations at high Reynolds numbers. *Numer. Heat Transfer: Part B*, 22:179-198, 1992.
- [22] Y. Wang and J. Zhang. Sixth order compact scheme combined with multigrid method and extrapolation technique for 2D Poisson equation. *J. Comput. Phys.*, 228:137-146, 2009.
- [23] Y. Wang and J. Zhang. Integrated Fast and High Accuracy Computation of Convection Diffusion Equations Using Multiscale Multigrid Method. *Technical Report CMIDA-HiPSCCS 015-09*, Department of Computer Science, University of Kentucky, Lexington, KY, 2009.
- [24] P. Wesseling. *An Introduction to Multigrid Methods*. Wiley, Chichester, England, 1992.
- [25] J. Zhang. Accelerated multigrid high accuracy solution of the convection-diffusion equation with high Reynolds number. *Numer. Methods Partial Differential Eq.*, 13:77-92, 1997.
- [26] 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.
- [27] J. Zhang. Residual scaling techniques in multigrid, I: equivalence proof. *Appl. Math. Comp.*, 86:283-303, 1997.
- [28] J. Zhang. An explicit fourth-order compact finite difference scheme for three dimensional convection-diffusion equation. *Commun. Numer. Methods Engrg.*, 14:209-218, 1998.
- [29] J. Zhang. Preconditioned iterative methods and finite difference schemes for convection-diffusion. *Appl. Math. Comput.*, 109:11-30, 2000.
- [30] J. Zhang. A note on accelerated high accuracy multigrid solution of convection-diffusion equation with high Reynolds number. *Numer. Methods Partial Differential Eq.*, 16(1):1-10, 2000.