Integrated Fast and High Accuracy Computation of Convection Diffusion Equations Using Multiscale Multigrid Method

Yin Wang[1*] and Jun Zhang[2†]

Abstract

We present an explicit sixth order compact finite difference scheme for fast high accuracy numerical solutions of the two dimensional convection diffusion equation with variable coefficients. The sixth order scheme is based on the well-known fourth order compact scheme, the Richardson extrapolation technique, and an operator interpolation scheme. For a particular implementation, we use multiscale multigrid method to compute the fourth order solutions on both the coarse grid and the fine grid. Then an operator interpolation scheme combined with the Richardson extrapolation technique is used to compute a sixth order accurate fine grid solution. We compare the computed accuracy and the implementation cost of the new scheme with the standard nine-point fourth order compact scheme and Sun-Zhang's sixth order method. Two convection diffusion problems are solved numerically to validate our proposed sixth order scheme.

Keywords: Convection diffusion equation, Reynolds number, multigrid method, Richardson extrapolation.
Mathematics Subject Classification: 65N06, 65N55, 65F10.

1 Introduction

We consider the two dimensional (2D) convection diffusion equation with the Dirichlet boundary condition, which can be written as

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

where Ω is a convex domain in R^2 consisting of a union of rectangles and $\partial\Omega$ is the boundary of Ω. We assume that the convection coefficients $p(x,y)$ and $q(x,y)$ are sufficiently smooth on Ω.

*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 grant 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.

Eq. (1) can be discretized using some finite difference schemes to result in a system of linear equations

$$A^h u^h = f^h, \tag{2}$$

where h is the uniform grid spacing of the discretized domain Ω^h. The magnitudes of $p(x,y)$ and $q(x,y)$ determine the ratio of the convection to diffusion which can be characterized by the cell Reynolds number (Re) in the form of

$$Re = \max(\sup_{(x,y)\in\Omega} |p(x,y)|, \sup_{(x,y)\in\Omega} |q(x,y)|)h/2. \tag{3}$$

In many problems of practical interest, the convective terms dominate the diffusion terms, in which Re is larger than 1. Since the coefficient matrix A^h in Eq. (2) is usually nonsymmetric and indefinite for large Re [11], the numerical solutions of the Eq. (2) based on iterative methods become increasingly difficult as Re increases [7, 19].

Numerical simulation of the convection diffusion equation plays a very important role in many scientific and engineering modeling problems involving fluid flows and heat transfer. Generally, for the convection-dominated problems, traditional finite difference discretization schemes such as the five-point second order central difference scheme (CDS) and the upwind difference scheme (UDS) cannot yield satisfactory results [23]. CDS has a truncation error of order $O(h^2)$ but may produce nonphysical oscillations for large Re. UDS usually prevents oscillations but reduces the order of accuracy to $O(h)$ [14].

Higher order (more than two) accurate discretization schemes need more complex procedures than the lower order discretization schemes to generate the coefficient matrix for Eq. (2), but they may result in the linear systems of much smaller size [1, 9]. In the past two decades, there has been growing interest in using higher order compact schemes to solve elliptic partial differential equations. Gupta *et al*. proposed a nine-point fourth order compact (FOC) scheme to discretize the 2D convection diffusion equation with variable coefficients [8]. There are also some other similar fourth order compact schemes that have been developed for the 2D convection diffusion equations. Readers are referred to [12, 14, 15] and the references therein for more details.

For the sixth order schemes, Chu and Fan [5, 6] proposed a three point combined compact difference (CCD) scheme for solving two dimensional Stommel Ocean model, which is also a convection diffusion equation. Their scheme can achieve sixth order accuracy for the inner grid points and fifth order accuracy for the boundary grid points. CCD scheme is considered as an *implicit* scheme because it does not compute the solution of the variables of interest directly. It has a stability problem that, for certain problems, if a large meshsize is used, the computed solution may be oscillatory [24]. Numerical oscillations may be avoided by using finer meshsize. However, use of finer mesh discretization is in conflict with the motivation of using higher order compact schemes.

In contrary, the *explicit* compact schemes, e.g., the FOC scheme, compute the solution of the variables directly. From [14] we know that the explicit FOC schemes are stable and will not generate oscillatory solutions. But the higher order explicit compact schemes are more complicated to develop in higher dimensions, compared with the implicit 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. So, the multiscale grid method has been proposed to achieve the sixth order accuracy for the explicit compact formulations. Sun and Zhang [16] first proposed a sixth order explicit finite difference discretization strategy for solving 2D convection diffusion equations, but their solution method is not scalable with the meshsize.

Focusing on the 2D Poisson equation, we have developed an efficient and scalable sixth order explicit compact scheme by using multiscale multigrid method and operator based interpolation combined with an extrapolation technique [18]. In this paper, we derive a robust and similar sixth order explicit compact approximation scheme for 2D convection diffusion equation with variable coefficients. For the problems with high Reynolds number, we use X-Y line relaxation in the multiscale multigrid method to achieve the grid independent convergence. Since the order of the solution accuracy always decreases when Re increases, we modify the operator based interpolation scheme from [18] to use the computed order of solution accuracy from the FOC scheme to perform the Richardson extrapolation.

The ultimate goal of numerical computation is to compute accurate solutions using a minimum amount of computing time and other computer resources. Thus, our proposed sixth order multiscale multigrid method deals with the solution accuracy and fast computation simultaneously.

An outline of the paper is as follows. In Section 2, we illustrate our sixth order compact discretization scheme which is based on the fourth order scheme from Gupta *et al.* [8]. In Section 3, truncation error analysis is given for a 1D convection diffusion equation. In Section 4, we introduce our multiscale multigrid method. Numerical results are provided in Section 5 to demonstrate the high accuracy of the sixth order method, as well as the computational efficiency of our multiscale multigrid method. Section 6 contains the concluding remarks.

2 Sixth Order Compact Approximations

Our explicit sixth order compact scheme is based on the fourth order compact (FOC) discretization on the two scale grids. The FOC schemes have been proposed by several authors [8, 12, 14] in different ways. We believe that these schemes are mathematically equivalent, although they were derived with different approaches. In this paper, we use the FOC scheme by Gupta *et al.* [8].

2.1 The FOC finite difference scheme

We use u_0 to denote the approximate value of $u(x, y)$ at a mesh point (x, y). The approximate values at its eight immediate neighboring points are denoted by u_i, $i = 1, 2, ..., 8$. The nine-point compact grid points are labeled as follows

$$\begin{pmatrix} u_6 & u_2 & u_5 \\ u_3 & u_0 & u_1 \\ u_7 & u_4 & u_8 \end{pmatrix}.$$

We use p_i, q_i and f_i, $(i = 0, 1, ..., 4)$ to denote the function values at the corresponding grid points. The nine-point fourth order compact finite difference formula for the mesh point (x, y) can be written as

$$\sum_{j=0}^{8} \alpha_j u_j = \frac{h^2}{2}[8f_0 + f_1 + f_2 + f_3 + f_4] + \frac{h^3}{4}[p_0(f_1 - f_3) + q_0(f_2 - f_4)], \tag{4}$$

where h is the mesh spacing, $\alpha_i(i = 0, 1, ..., 8)$ are the coefficients as

$$
\begin{aligned}
\alpha_0 &= -[20 + h^2(p_0^2 + q_0^2) + h(p_1 - p_3) + h(q_2 - q_4)],\\
\alpha_1 &= 4 + \frac{h}{4}[4p_0 + 3p_1 - p_3 + p_2 + p_4] + \frac{h^2}{8}[4p_0^2 + p_0(p_1 - p_3) + q_0(p_2 - p_4)],\\
\alpha_2 &= 4 + \frac{h}{4}[4q_0 + 3q_2 - q_4 + q_1 + q_3] + \frac{h^2}{8}[4q_0^2 + p_0(q_1 - q_3) + q_0(q_2 - q_4)],\\
\alpha_3 &= 4 - \frac{h}{4}[4p_0 - p_1 + 3p_3 + p_2 + p_4] + \frac{h^2}{8}[4p_0^2 - p_0(p_1 - p_3) - q_0(p_2 - p_4)],\\
\alpha_4 &= 4 - \frac{h}{4}[4q_0 - q_2 + 3q_4 + q_1 + q_3] + \frac{h^2}{8}[4q_0^2 - p_0(q_1 - q_3) - q_0(q_2 - q_4)],\\
\alpha_5 &= 1 + \frac{h}{2}(p_0 + q_0) + \frac{h}{8}(q_1 - q_3 + p_2 - p_4) + \frac{h^2}{4}p_0q_0,\\
\alpha_6 &= 1 - \frac{h}{2}(p_0 - q_0) - \frac{h}{8}(q_1 - q_3 + p_2 - p_4) - \frac{h^2}{4}p_0q_0,\\
\alpha_7 &= 1 - \frac{h}{2}(p_0 + q_0) + \frac{h}{8}(q_1 - q_3 + p_2 - p_4) + \frac{h^2}{4}p_0q_0,\\
\alpha_8 &= 1 + \frac{h}{2}(p_0 - q_0) - \frac{h}{8}(q_1 - q_3 + p_2 - p_4) - \frac{h^2}{4}p_0q_0.
\end{aligned}
$$

When $p(x, y)$ and $q(x, y)$ are set to be zero, Eq. (1) is reduced to the 2D Poisson equation.

2.2 Extrapolation and operator based interpolation

In our previous work [18], we proposed an operator based interpolation scheme combined with the Richardson extrapolation technique to achieve sixth order accurate solution on the fine grid. The numerical results show that our method is very efficient and accurate for the 2D Poisson equation. We point out that Poisson equation is a special case of the convection diffusion equation with $Re = 0$, which means that the order of accuracy of the computed solution for the Poisson equation will not be affected by the Reynolds number. In practice, when Re is very small, we just assume that we obtain the exact fourth order accurate solutions from the FOC scheme and apply the fourth order Richardson extrapolation. For the convection diffusion equation with high Reynolds numbers, we need to consider the effect of the Reynolds number on the order of accuracy of the computed solutions. The detail will be given in Section 3.

The basic idea of our strategy is to apply the Richardson extrapolation technique to obtain the sixth order accurate solution for the $(even, even)$ indexed grid points on the fine grid first, as in Fig. 1. Then a mesh-refinement interpolation scheme [10] will be used to achieve the sixth order accurate solution for other grid points.

The detail of our strategy is illustrated in Algorithm 1. We assume that we have the converged solutions $u^{2h}_{i,j}$ and $u^{h}_{i,j}$ on the Ω_{2h} grid and the Ω_h grid respectively by solving the linear systems arising from the FOC schemes. Then we apply the Richardson extrapolation technique to increase the order of accuracy of the computed solution for the coarse grid solution $\tilde{u}^{2h}_{i,j}$ on Ω_{2h}. Since the exact order of solution accuracy is related to the Reynolds number (less than 4 with large Re), we assume the order of solution accuracy is m. The

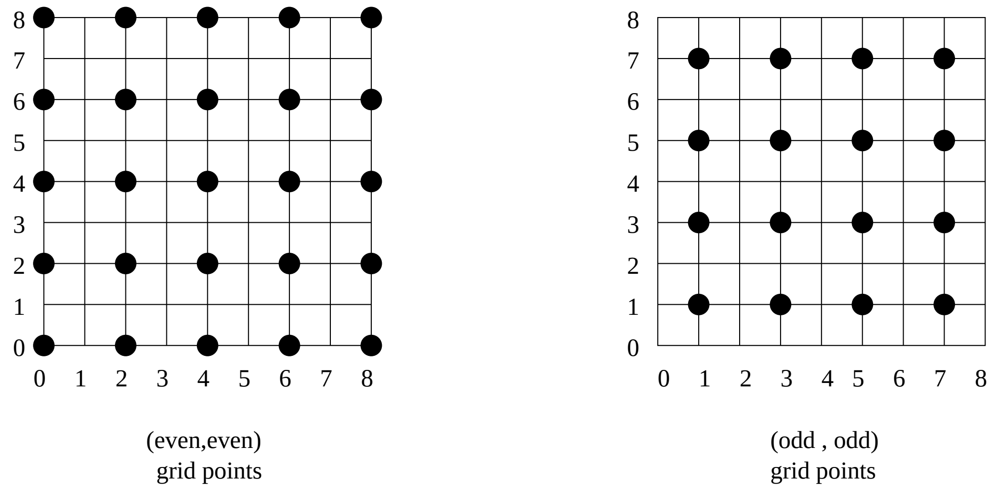

Figure 1: Illustration of the $(even, even)$ and (odd, odd) grid points for a 9×9 fine grid.

Richardson extrapolation formula that we use can be written in the form of [2, 13]

$$\tilde{u}_{i,j}^{2h} = \frac{(2^m u_{2i,2j}^h - u_{i,j}^{2h})}{2^{m-1}}. \tag{5}$$

In Algorithm 1, $A_{i,j}(l), l = 0, 1, ..., 8$, are the pre-computed coefficients for grid point (x_i, y_j). Ω_h^m and Ω_{2h}^m denote the m_{th} order accurate solution space from the FOC schemes, $\Omega_h^{\tilde{m}}$ and $\Omega_{2h}^{\tilde{m}}$ are the improved $\tilde{m}_{th}$ order accurate solution space $(m < \tilde{m} \leq 6)$. $\tilde{u}^{h,k}$ 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.

3 Truncation Error Analysis

In this section, we will give analysis to show how the high Reynolds number affects our operator based interpolation scheme. For better understanding, we consider the one dimensional (1D) convection diffusion type equation.

We consider the following 1D model convection diffusion equation

$$u_{xx} + bu_x = 0, \qquad 0 \leq x \leq l, \tag{6}$$

where b is the constant convection coefficient.

We denote h to be the meshsize, $x_j = jh$ and $u_j = u(x_j)$. The standard first and second order central difference operators are

$$\delta_x^h u_j = \frac{u_{j+1} - u_{j-1}}{2h}, \qquad \delta_{xx}^h u_j = \frac{u_{j+1} + u_{j-1} - 2u_j}{h^2}, \qquad j = 1, 2, ..., n.$$

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

1: Let $u^h_{old} = \tilde{u}^{h,k}$.

2: Update every $(even, even)$ grid point on Ω_h.
From $\tilde{u}^{2h,k}_{i,j} \in \Omega^m_{2h}$ and $\tilde{u}^{h,k}_{2i,2j} \in \Omega^m_h$, we first compute $\tilde{u}^{2h,k+1}_{i,j} \in \Omega^{\tilde{m}}_{2h}$ by Eq. (5), then use direct interpolation to obtain $\tilde{u}^{h,k+1}_{2i,2j} \in \Omega^{\tilde{m}}_h$.

3: Update every (odd, odd) grid point on Ω_h.
From Eq. (4), for each (odd, odd) point (i, j), the updated solution is

$$\begin{aligned}\tilde{u}^{h,k+1}_{i,j} = [&F_{i,j} - A_{i,j}(1)\tilde{u}^{h,k}_{i+1,j} - A_{i,j}(2)\tilde{u}^{h,k}_{i,j+1} - A_{i,j}(3)\tilde{u}^{h,k}_{i-1,j} - A_{i,j}(4)\tilde{u}^{h,k}_{i,j-1} \\ &- A_{i,j}(5)\tilde{u}^{h,k+1}_{i+1,j+1} - A_{i,j}(8)\tilde{u}^{h,k+1}_{i+1,j-1} - A_{i,j}(6)\tilde{u}^{h,k+1}_{i-1,j+1} - A_{i,j}(7)\tilde{u}^{h,k+1}_{i-1,j-1}]/A_{i,j}(0).\end{aligned}$$

Here, $F_{i,j}$ represents the right-hand side part of Eq. (4).

4: Update every $(odd, even)$ grid point on Ω_h.
From Eq. (4), the idea is similar to updating the (odd, odd) grid points.

5: Update every $(even, odd)$ grid point on Ω_h.
From Eq. (4), the idea is similar to updating the $(odd, even)$ grid points.

6: Compute the 2-norm $R = ||\tilde{u}^{h,k+1} - u^h_{old}||_2$. If not converged, go back to Step 1.

By using Taylor series, we have

$$\delta^h_{xx} u_j = u_{xx} + \frac{h^2}{12} u_{x^4} + \frac{h^4}{360} u_{x^6} + \frac{h^6}{20160} u_{x^8} + O(h^8), \tag{7}$$

and

$$\delta^h_x u_j = u_x + \frac{h^2}{6} u_{x^3} + \frac{h^4}{120} u_{x^5} + \frac{h^6}{5040} u_{x^7} + O(h^8). \tag{8}$$

From Eqs. (7) and (8) we can discretize Eq. (6) at the grid point x_j as

$$\begin{aligned}\delta^h_{xx} u_j + b\delta^h_x u_j =& \frac{bh^2}{6} u_{x^3} + \frac{h^2}{12} u_{x^4} + h^4 (\frac{b}{120} u_{x^5} + \frac{1}{360} u_{x^6}) \\ &+ h^6 (\frac{b}{5040} u_{x^7} + \frac{1}{20160} u_{x^8}) + O(h^8).\end{aligned} \tag{9}$$

By taking derivatives on both sides of Eq. (6), we have

$$u_{x^3} = -bu_{x^2}, \qquad u_{x^4} = b^2 u_{x^2}. \tag{10}$$

So, at the grid point x_j, we have

$$(u_{x^3})_j = -b \left[\delta^h_{xx} u_j - \frac{h^2}{12} (u_{x^4})_j - \frac{h^4}{360} (u_{x^6})_j - O(h^6) \right], \tag{11}$$

and

$$(u_{x^4})_j = b^2 \left[\delta^h_{xx} u_j - \frac{h^2}{12} (u_{x^4})_j - \frac{h^4}{360} (u_{x^6})_j - O(h^6) \right]. \tag{12}$$

Then we use Eqs. (11) and (12) to replace the u_{x^3} and u_{x^4} terms in Eq. (9) as

$$\delta_{xx}^h u_j + b\delta_x^h u_j = -\frac{h^2b^2}{12}\left[\delta_{xx}^h u_j - \frac{h^2}{12}(u_{x^4})_j - \frac{h^4}{360}(u_{x^6})_j\right] + h^4(\frac{1}{360}u_{x^6} + \frac{b}{120}u_{x^5}) + h^6(\frac{1}{20160}u_{x^8} + \frac{b}{5040}u_{x^7}) + O(h^8). \quad (13)$$

Eq. (13) can be rewritten in the form of

$$\left[(1 + \frac{h^2b^2}{12})\delta_{xx}^h + b\delta_x^h\right] u_j = (\tau_1)_j + (\tau_2)_j + O(h^8), \quad (14)$$

where $(\tau_1)_j$ is in the form of

$$(\tau_1)_j = \left[\frac{b^2}{144}(u_{x^4})_j + \frac{b}{120}(u_{x^5})_j + \frac{1}{360}(u_{x^6})_j\right] h^4,$$

and $(\tau_2)_j$ is in the form of

$$(\tau_2)_j = \left[\frac{b^2}{4320}(u_{x^6})_j + \frac{b}{5040}(u_{x^7})_j + \frac{1}{20160}(u_{x^8})_j\right] h^6.$$

If we drop $(\tau_2)_j$ and $O(h^8)$ terms in Eq. (14), we get the fourth order truncation error for the FOC scheme as $(\tau_1)_j$. After using our operator based interpolation scheme combined with the Richardson extrapolation technique, the fourth order error term $(\tau_1)_j$ will be canceled. So, we will get a sixth order truncation error in proportion to $(\tau_2)_j$.

The order of accuracy of the computed solution is formally defined when the meshsize h approaches zero. Generally, we consider the order of the truncation error for $(\tau_1)_j$ to be 4, but the actual order of solution accuracy of the FOC scheme depends on the value of the Reynolds number. From [8, 19], we know that the order of solution accuracy will decrease when Re increases.

By using Eq. (3), we have $Re = bh/2$. So, the truncation error from the FOC scheme $(\tau_1)_j$ can be rewritten as

$$(\tau_1)_j = \frac{Re^2}{36}(u_{x^4})_j h^2 + \frac{Re}{60}(u_{x^5})_j h^3 + \frac{1}{360}(u_{x^8})_j h^4. \quad (15)$$

When $b \to +\infty$ and $h \to 0$, Re can be considered as some constant. So, the order of accuracy of the FOC scheme will be degraded to second order when the convection coefficient approaches infinity. In practice, the computed order of accuracy of the FOC scheme varies from 4 to 2 as the Reynolds number increases.

Since the high Reynolds number will affect the accuracy of the computed solution, we need to compute the exact order of accuracy from the FOC scheme before we run the extrapolation procedure. In practice, we denote $E(h)$ and $E(H)$ to be the solution errors with the meshsizes h and H, respectively. The order of solution accuracy m is calculated from the following formula

$$
\begin{aligned}
&\frac{E(h)}{E(H)} = \frac{h^m}{H^m} \\
&\Longrightarrow m = \log_{(h/H)}(E(h)/E(H)).
\end{aligned}
$$

For the sixth order truncation error $(\tau_2)_j$, it is also affected by the Reynolds number. We denote λ to be the ratio of $(\tau_1)_j$ to $(\tau_2)_j$, so λ can be used to represent the improvement of the extrapolation procedure. Generally, a larger λ means the extrapolation procedure improves the solution more accurately and a smaller λ means the extrapolation procedure cannot improve the accuracy much.

The ratio λ is related to the value of b. When $b \to +\infty$, λ can be computed as

$$
\begin{aligned}
\lambda &= \lim_{b\to+\infty} \frac{(\frac{b^2}{144}(u_{x^4})_j + \frac{b}{120}(u_{x^5})_j + \frac{1}{360}(u_{x^6})_j)h^4}{(\frac{b^2}{4320}(u_{x^6})_j + \frac{b}{5040}(u_{x^7})_j + \frac{1}{20160}(u_{x^8})_j)h^6} \\
&= \lim_{b\to+\infty} \frac{(\frac{b^2}{144}(u_{x^4})_j}{\frac{b^2}{4320}(u_{x^6})_j h^2} \\
&= \frac{30(u_{x^4})_j}{(u_{x^6})_j h^2}.
\end{aligned}
$$

So, the accuracy improvement from the FOC scheme using the extrapolation scheme will be leveled off at the second order when *Re* (b) is beyond some threshold. Supporting numerical results will be shown in Section 5.

4 Multiscale Multigrid Method

After the 2D convection diffusion equation is discretized into a system of linear equations (2), it is important that the resulting linear system be solved efficiently. Existing solution methods fall into two categories: direct methods and iterative methods. Direct methods based on Gaussian elimination are not widely used for solving large sparse linear systems because they scale poorly when the memory space and the CPU cost become an issue. Basic iterative methods like Jacobi and Gauss-Seidel methods are easy to implement, but they are not robust for many problems of practical interest. Some iterative methods do not converge when the convective terms dominate and the cell Reynolds number is greater than a certain value.

We want to develop an efficient and robust iterative solver to solve the linear systems arising from the 2D convection diffusion equation. The multigrid method is among the fastest and most efficient iterative methods. The convergence rate of the multigrid method for idealized elliptic problems is independent of the grid size [3, 4, 17].

Using the point Gauss-Seidel relaxation in a standard multigrid method is efficient for solving elliptic problems like the Poisson equation and the convection diffusion equation with small *Re* [3], but the convergence will be slow with high *Re*. Several acceleration schemes have been developed to speed up the multigrid method, like the minimal residual smoothing method (MRS). Unfortunately, MRS combined with point relaxation in multigrid method still cannot achieve the grid independence for some high Reynolds number problems and it still needs more than 1000 iterations to converge when the magnitude of the Reynolds number is higher than 10^5 [19].

In order to achieve better efficiency and robustness, we will use alternating (X-Y) line Gauss-Seidel relaxation in our multigrid method. The X-Y line Gauss-Seidel relaxation in lexicographic order performs one sweep of line Gauss-Seidel relaxation along the x-coordinate direction first, then another sweep of the line Gauss-Seidel relaxation along the y-coordinate direction. However, it was shown in [21, 22] that merely using the X-Y line Gauss-Seidel relaxation in a standard multigrid method does not provide fast convergence for convection dominated problems with high Reynolds number. The reason is that the coarse grid solution may not provide a meaningful correction to the fine grid computed solution with a small amount of artificial viscosity. One simple and efficient approach to fix this problem is to properly scale the residual before it is projected to the coarse grid [20].

The residual scaling procedure at a grid point (x_i, y_j) can be written in the form of

$$\tilde{r}(x_i, y_j) = \beta r(x_i, y_j), \tag{16}$$

where β is the scaling factor, $r(x_i, y_j)$ is the residual from the fine grid and $\tilde{r}(x_i, y_j)$ is the residual after scaling. The actual scaling factor is determined by the absolute values of the convection coefficients at the reference grid point. The detail of how to choose an optimal residual scaling factor with high Reynolds number can be found in [21, 22]. In our experiment, we tested several scaling factors and only list the numerical results from the best one in Section 5.

We proposed a multiscale multigrid method in [18] to solve the 2D Poisson equation, which computes the fourth order solutions on both the fine and coarse grids. This method can also be used to solve the 2D convection diffusion equation. In practice, we choose to pre-compute the coefficients for every grid points. This strategy requires more memory space but it does reduce the overall CPU times. There is always a trade-off between the required storage space and the computational efficiency.

Our multiscale multigrid method is based on the standard multigrid V-Cycle. It is similar to the full multigrid method, but we do not start from the coarsest grid level. We describe it as below:

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 the m_{th} 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 the m_{th} order solution u_h.

5 Numerical Results

We tested our sixth order compact scheme (SOC) and compared the results with the standard fourth order compact difference scheme (FOC) and Sun-Zhang's sixth order method (REC) [16]. The codes were 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 2GB of local memory and runs at 2.0GHZ.

We used standard V(1,1)-Cycle to build our multiscale multigrid method. The initial guess for the V-Cycle on $4h$ grid was the zero vector. The stopping criteria for the operator based interpolation and the V-Cycle on $2h$ and h grid 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.

For all test problems, we first compared the computed solution accuracy and the CPU cost for different strategies with some fixed convection coefficients on different meshsizes. Then, we tested the effect of Re on the computed solution accuracy and the CPU cost for different strategies with a fixed meshsize. The graphic comparison of the numerical data will also be provided for each test case.

5.1 Test Problem 1

We first considered a case in which the convection coefficients $p(x,y)$ and $q(x,y)$ are polynomials of both x and y variables and therefore vary somewhat rapidly in the domain Ω. The test case is

$$\begin{cases} u(x,y) &= x^2y^2(1-x)(1-y), \\ p(x,y) &= Px(1-y), \\ q(x,y) &= Py(1-x). \end{cases}$$

Table 1 shows the number of iterations, maximum errors and the order of solution accuracy for different solution strategies with different meshsizes. For $P = 10$, the cell Reynolds number is small, we can see that the order of solution accuracy for the FOC scheme is nearly 4 as we expected. The numerical results illustrate that the SOC scheme solved the problem with better accuracy than the FOC scheme did, and the order of solution accuracy was close to 6. The results also indicate that the computed solutions from the SOC scheme and the Sun-Zhang's REC method were comparable, but the SOC scheme took much less CPU time. The number of iterations for the REC method increased very quickly when the mesh became finer. In addition, when $n > 64$, it did not converge within the maximum number of iterations (5000) we set. On the other hand, by using the X-Y line relaxation in the multigrid method combined with the residual scaling technique, the convergence rate of the FOC scheme and the SOC scheme was independent of the grid size. We can also see that, by using our new sixth order compact scheme, the number of V-Cycles for Ω_h and Ω_{2h} were reduced, compared to the standard FOC scheme.

When the magnitude of the convection coefficients increased, i.e., when $P = 1000$, we found that the order of solution accuracy from the FOC scheme was reduced as we expected, especially for $n = 16$. It is clear that the SOC scheme still increased the solution accuracy when Re increased. Since Re is a function of the meshsize h, the convergence was improved when h was refined. We can see that the number of V-Cycles for both the FOC scheme and the SOC scheme decreased when n increased and the REC method converged for all the meshsize we tested. For better understanding, we list the graphic comparison of the CPU cost and the maximum errors with different meshsize in Fig. 2.

Table 2 contains the results with various Re (P) with a fixed meshsize $h = 1/64$. We compared the CPU cost and the maximum errors for the FOC scheme and the SOC scheme.

Table 1: Test Problem 1: Comparison of CPU cost and solution accuracy with different meshsizes and fixed P values.

		P =10				P=1000			
n	strategy	# it	CPU	error	order	# it	CPU	error	order
16	REC	132	0.011	2.41e-6	5.66	58	0.005	4.20e-3	3.17
	FOC	6	0.003	3.45e-5	4.06	16	0.004	6.17e-3	2.85
	SOC	(6,6),18	0.004	2.41e-6	5.66	(13,14),96	0.004	4.19e-3	3.17
32	REC	518	0.173	4.36e-8	5.79	47	0.016	1.36e-4	4.95
	FOC	7	0.008	2.13e-6	4.02	19	0.013	4.15e-4	3.89
	SOC	(6,6),17	0.007	4.36e-8	5.79	(14,17),64	0.017	1.36e-4	4.95
64	REC	2064	2.845	7.47e-10	5.86	44	0.081	2.97e-6	5.51
	FOC	7	0.025	1.33e-7	4.00	19	0.055	2.57e-5	4.01
	SOC	(6,7),15	0.032	7.36e-10	5.89	(17,17),35	0.076	2.97e-6	5.51
128	REC	not converge	–	–	–	132	1.150	5.71e-8	5.70
	FOC	7	0.105	8.29e-9	4.00	17	0.358	1.60e-6	4.01
	SOC	(7,7),13	0.124	1.25e-11	5.87	(17,14),21	0.457	5.71e-8	5.70
256	REC	not converge	–	–	–	490	20.591	9.98e-10	5.83
	FOC	7	0.612	5.18e-10	4.00	15	1.146	1.00e-7	4.00
	SOC	(7,7),15	1.190	1.97e-13	5.98	(14,13),18	1.409	9.95e-10	5.84

We note that the magnitude of the Reynolds number affected the solution accuracy and the convergence rate for the FOC scheme and the SOC scheme inversely. When Re was small ($Re \leq 1$), these methods converged rapidly and yielded reasonably accurate solutions. When Re increased, the solution accuracy and the convergence rate severely deteriorated.

We note that there was only a little change for the accuracy and the number of iterations for the FOC and SOC schemes when $P > 10^5$. However, the SOC scheme still yielded better accuracy than the FOC scheme, but the improvement was degraded when Re increased as we expected. The ratio of the fourth order error to the sixth order error with different P is shown in Fig. 3. We can see that the ratio is close to a certain constant when $P \to +\infty$.

5.2 Test Problem 2

We chose the coefficients as multiples of the trigonometric functions

$$\begin{cases} u(x,y) &= \cos(4x+6y), \\ p(x,y) &= P\sin(\pi x), \\ q(x,y) &= P\cos(\pi y). \end{cases}$$

The test conditions were set to be the same as for the test Problem 1.

The numerical data with comparison are shown in Table 3, Table 4, Fig. 4 and Fig. 5. For $P = 10$, the SOC scheme achieved the sixth order solution accuracy and the convergence rate of our multiscale multigrid method was independent of the grid size. Once again, our sixth order method solved the problem with the same accuracy as Sun-Zhang's REC method did, but costed much less CPU time. In addition, the REC method still could not converge when $n > 64$.

For $P = 1000$, it seems that the magnitude of Re affected the order of accuracy more than the Problem 1 when n was smaller than 64. When n increased, the number of iterations for the FOC scheme and the REC method decreased, which once again showed that the

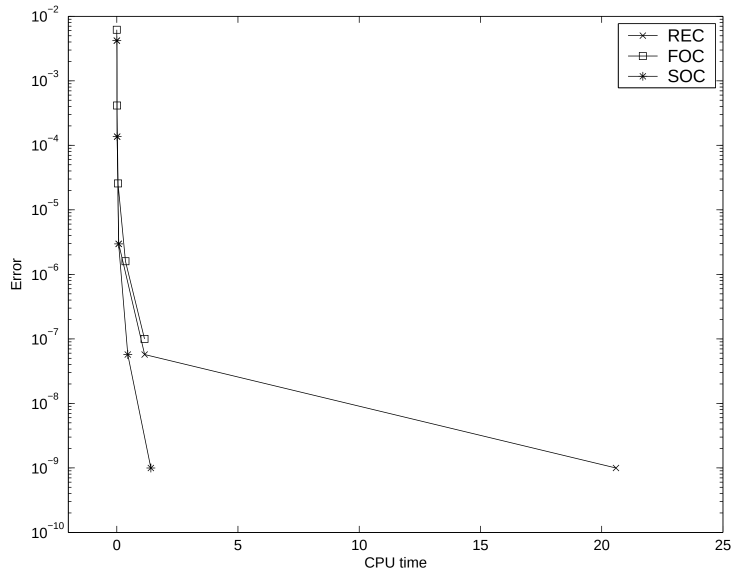

Figure 2: Comparison of the maximum errors and the CPU costs for the Problem 1 ($P = 1000$). Each symbol with increasing CPU time corresponds to an increasing fine grid: 16, 32, 64, 128, and 256 intervals.

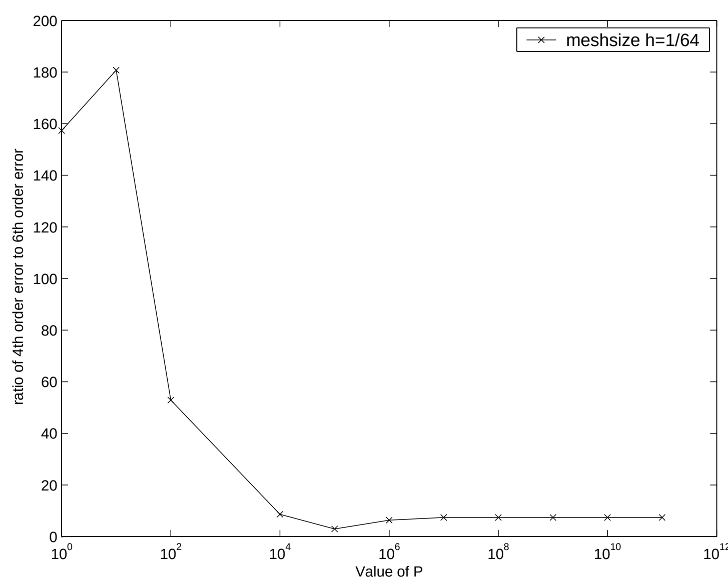

Figure 3: Ratio of the fourth order error to the sixth order error for the Problem 1.

Table 2: Test Problem 1: Comparison of CPU cost and solution accuracy with different P values for a fixed meshsize.

n=64		FOC			SOC		
P	Re	# it	CPU	error	# it	CPU	error
0	0.0	7	0.027	4.58e-12	(6,7),2	0.028	1.02e-13
1	7.8125e-3	7	0.027	1.01e-8	(6,7),12	0.031	6.42e-11
10	7.8125e-2	7	0.025	1.33e-7	(6,7),15	0.032	7.36e-10
10^2	7.8125e-1	9	0.038	2.11e-6	(8,8),19	0.041	3.99e-8
10^3	7.8125e0	19	0.055	2.57e-5	(57,56),35	0.055	2.97e-6
10^4	7.8125e1	73	0.186	2.63e-4	(32,40),526	0.293	8.87e-5
10^5	7.8125e2	44	0.115	1.03e-3	(32,41),1709	0.572	1.62e-4
10^6	7.8125e3	45	0.119	1.13e-3	(32,41),1803	0.595	1.52e-4
10^7	7.8125e4	44	0.123	1.13e-3	(32,41),1810	0.592	1.52e-4
10^8	7.8125e5	44	0.121	1.13e-3	(32,41),1811	0.603	1.52e-4
10^9	7.8125e6	44	0.119	1.13e-3	(32,41),1811	0.587	1.52e-4
10^{10}	7.8125e7	44	0.124	1.13e-3	(32,41),1811	0.598	1.52e-4

magnitude of the Reynolds number affected the convergence rate inversely.

Similar to the Problem 1, Table 4 lists the comparison results for different Re with a fixed meshsize. We note that the SOC scheme yielded more accurate solutions compared to the FOC scheme and it also costed less CPU time than the FOC scheme did when $P \geq 10^5$. And it once again showed the convergence rate and the computed accuracy approached some limits and did not deteriorate any more when Re (P) was beyond some threshold.

Table 3: Test Problem 2: Comparison of CPU cost and solution accuracy with different meshsizes and fixed P values.

		P=10				P=1000			
n	strategy	# it	CPU	error	order	# it	CPU	error	order
16	REC	94	0.007	2.30e-5	5.44	331	0.025	5.69e-3	2.02
	FOC	9	0.005	1.36e-4	4.19	24	0.008	6.25e-3	2.01
	SOC	(8,8),24	0.003	2.30e-5	5.44	(32,23),29	0.006	5.69e-3	2.02
32	REC	364	0.121	4.21e-7	5.77	294	0.098	5.38e-4	3.40
	FOC	9	0.012	7.97e-6	4.09	60	0.041	7.65e-4	3.03
	SOC	(8,8),24	0.008	4.21e-7	5.77	(23,51),31	0.036	5.38e-4	3.40
64	REC	1445	2.079	6.95e-9	5.92	224	0.305	1.90e-5	4.82
	FOC	9	0.041	4.82e-7	4.05	70	0.256	5.63e-5	3.76
	SOC	(8,8),23	0.043	6.95e-9	5.92	(51,63),30	0.197	1.90e-5	4.82
128	REC	not converge	–	–	–	130	1.532	3.36e-7	5.82
	FOC	9	0.261	2.96e-8	4.03	67	1.491	3.57e-6	3.97
	SOC	(8,8),21	0.241	1.11e-10	5.96	(63,63),28	1.587	3.36e-7	5.82
256	REC	not converge	–	–	–	419	17.708	6.54e-9	5.68
	FOC	9	1.209	1.84e-9	4.01	64	6.821	2.22e-7	4.01
	SOC	(8,9),19	1.594	1.74e-12	6.00	(64,52),26	4.774	6.54e-9	5.68

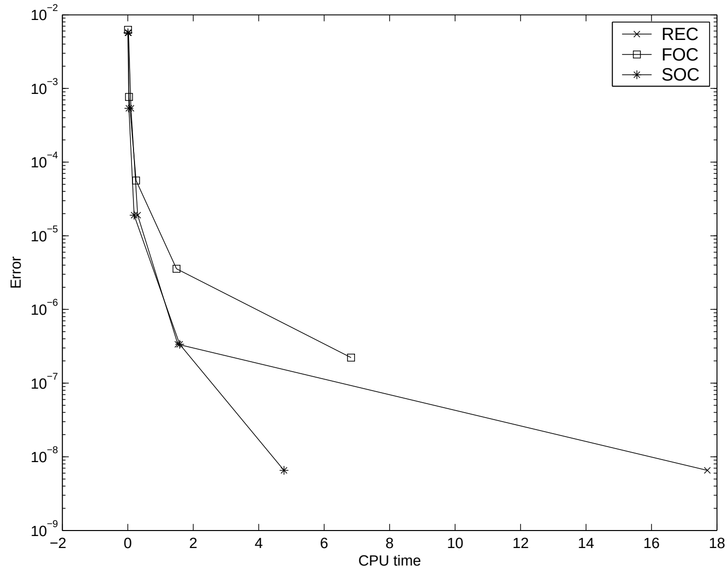

Figure 4: Comparison of the maximum errors and the CPU costs for the Problem 2 ($P = 1000$). Each symbol with increasing CPU time corresponds to an increasing fine grid: 16, 32, 64, 128, and 256 intervals.

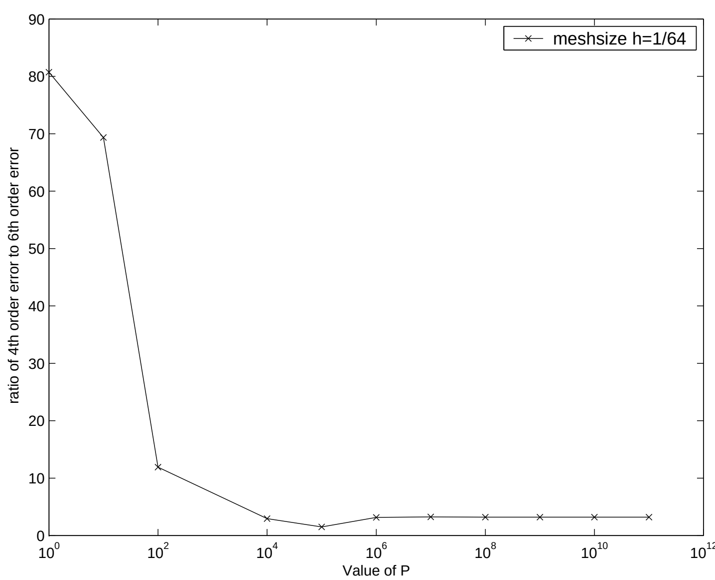

Figure 5: Ratio of the fourth order error to the sixth order error for the Problem 2.

Table 4: Test Problem 2: Comparison of CPU cost and solution accuracy with different P values for a fixed meshsize.

n=64		FOC			SOC		
P	Re	# it	CPU	error	# it	CPU	error
0	0.0	9	0.040	2.93e-9	(8,8),16	0.039	1.59e-10
1	7.8125e-3	9	0.041	3.81e-8	(8,8),19	0.040	4.72e-10
10	7.8125e-2	9	0.041	4.82e-7	(8,8),23	0.043	6.95e-9
10^2	7.8125e-1	14	0.055	5.17e-6	(15,14),25	0.065	4.34e-7
10^3	7.8125e0	70	0.256	5.63e-5	(51,63),30	0.197	1.90e-5
10^4	7.8125e1	108	0.337	3.45e-4	(76,93),33	0.358	2.03e-4
10^5	7.8125e2	109	0.367	5.01e-4	(75,94),34	0.363	1.58e-4
10^6	7.8125e3	113	0.381	4.81e-4	(75,92),34	0.352	1.48e-4
10^7	7.8125e4	113	0.372	4.79e-4	(75,92),34	0.347	1.49e-4
10^8	7.8125e5	113	0.369	4.79e-4	(75,92),34	0.356	1.49e-4
10^9	7.8125e6	113	0.376	4.79e-4	(75,92),34	0.349	1.49e-4
10^{10}	7.8125e7	113	0.378	4.79e-4	(75,92),34	0.361	1.49e-4

6 Concluding Remarks

We extended our idea from [18], which was to solve the 2D Poisson equation, to solve the 2D convection diffusion equation. In order to compute highly accurate solutions for the problems with high Reynolds numbers, we modified the operator based interpolation scheme to use the correct order of accuracy from the FOC schemes to perform the Richardson extrapolation. We also used the X-Y line relaxation combined with residual scaling technique in our multiscale multigrid method to achieve the grid independent convergence.

Our test results showed that our sixth order compact scheme is efficient, robust and accurate. It computed more accurate solutions than the FOC scheme, and the CPU cost was comparable. For some high Reynolds number cases, the SOC scheme even took less computation time than the FOC scheme did.

References

[1] Y. Adam, Highly accurate compact implicit methods and boundary conditions. *J. Comput. Phys.*, 24(1):10-22, 1977.

[2] C. Brezinski and M. R. Zaglia. *Extrapolation Method. Theory and Practice*, North-Holland, Berlin, 1991.

[3] A. Brandt. Multi-level adaptive solutions to boundary-value problems. *Math. Comput.*, 31(138):333-390, 1977.

[4] W. L. Briggs, V. E. Henson, and S. F. McCormick. *A Multigrid Tutorial.* SIAM, Philadelphia, PA, 2nd edition, 2000.

[5] P. C. Chu and C. Fan. A three-point combined compact difference scheme. *J. Comput. Phys.*, 140:370-399, 1998.

[6] P. C. Chu and C. Fan. A three-point six-order nonuniform combined compact difference scheme. *J. Comput. Phys.*, 148:663-674, 1999.

[7] L. Ge and J. Zhang. High accuracy iterative solution of convection diffusion equation with boundary layers on nonuniform grids. *J. Comput. Phys.*, 171(2):560-578, 2001.

[8] M. M. Gupta, R. P. Manohar, and J. W. Stephenson. A single cell high order scheme for the convection-diffusion equation with variable coefficients. *Int. J. Numer. Methods Fluids*, 4:641-651, 1984.

[9] M. M. Gupta, J. Kouatchou, and J. Zhang. Comparison of second and fourth order discretizations for multigrid Poisson solver. *J. Comput. Phys.*, 132:226-232, 1997.

[10] J. Hyman. Mesh refinement and local inversion of elliptic partial differential equations. *J. Comput. Phys.*, 23:124-134, 1977.

[11] S. Karaa and J. Zhang. On convergence and performance of iterative methods for solving variable coefficient convection-diffusion equation with a fourth-order compact difference scheme. *Comput. Math. Appli.*, 44:457-479, 2002.

[12] M. Li, T. Tang, and B. Fornberg. A compact fourth-order finite difference scheme for the steady incompressible Navier-Stokes equations. *Int. J. Numer. Methods Fluids*, 20:1137-1151,1995.

[13] 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.

[14] W. F. Spotz. *High-Order Compact Finite Difference Schemes for Computational Mechanics*. PhD thesis, University of Texas at Austin, Austin, TX, 1995.

[15] W. F. Spotz and G. F. Carey. High-order compact scheme for the steady streamfunction vorticity equations. *Int. J. Numer. Methods Engrg.*, 38:3497-3512, 1995.

[16] H. Sun and J. Zhang. A high order finite difference discretization strategy based on extrapolation for convection diffusion equations. *Numer. Methods Partial Differential Eq.*, 20(1):18-32, 2004.

[17] P. Wesseling. *An Introduction to Multigrid Methods*. Wiley, Chichester, England, 1992.

[18] 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.

[19] 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.

[20] J. Zhang. Residual scaling techniques in multigrid, I: equivalence proof. *Appl. Math. Comp.*, 86:283-303, 1997.

[21] J. Zhang. On convergence and performance of iterative methods with fourth-order compact schemes. *Numer. Methods Partial Differential Eq.*, 14:263-280, 1998.

[22] 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.

[23] J. Zhang. Preconditioned iterative methods and finite difference schemes for convection-diffusion. *Appl. Math. Comput.*, 109:11-30, 2000.

[24] J. Zhang and J. J. Zhao. Truncation error and oscillation property of the combined compact difference scheme. *Appl. Math. Comput.*, 161(1):241-251, 2005.