Comparison of Two Higher Order Compact Computation Strategies for Handling Boundary Layers

Ruxin Dai[1*], Yin Wang[2†], Jun Zhang[1‡] and Yongbin Ge[3§]

Abstract

Two fourth order compact (FOC) schemes with transformation method and non-transformation method to solve the two dimensional convection diffusion equation with boundary layers are compared. The domain is discretized on a stretched nonuniform grid. A grid transformation technique maps the nonuniform grid to a uniform one, on which the difference scheme is applied. A transformation-free scheme on nonuniform grids does not involve any transformation. A multigrid method is used to solve the resulting sparse linear systems from both methods. We compare the accuracy of the computed solutions and the robustness and efficiency of two schemes through comparison of maximum absolute errors, order of accuracy and CPU timings.

Keywords: convection diffusion equation, boundary layer, grid transformation, nonuniform grids, multigrid method.

1 Introduction

In this paper, we consider the two dimensional (2D) convection diffusion equation with the Dirichlet boundary condition, which can be written in the form of

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

where Ω is a convex domain in R^2 consisting of a union of rectangles and $\partial\Omega$ is the boundary of Ω. The convection coefficients $p(x,y)$ and $q(x,y)$ are assumed to be sufficiently smooth on Ω. We may refer to the magnitude of $p(x,y)$ and $q(x,y)$ as the Reynolds number (Re), which determines the ratio of the convection to diffusion.

In many problems of practical interest, numerical solutions of the Equation (1) based on iterative solution methods become increasingly hard (converge slowly or even diverge) when the Reynolds number increases [21].

[*1] Laboratory for High Performance Scientific Computing and Computer Simulation, Department of Computer Science, University of Kentucky, Lexington, KY 40506-0046, USA. E-mail: rda222@uky.edu

[†2], Department of Mathematics and Computer Science, Lawrence Technological University, Southfield, MI 48075-1058, USA. E-mail: ywang12@ltu.edu

[‡1] 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.

[§3], Institute of Applied Mathematics and Mechanics, Ningxia University, Yinchuan, 750021, China. E-mail: gyb@nxu.edu.cn

Numerical solutions of the convection diffusion equation plays a crucial role in many simulations and engineering modeling applications, such as fluid flows and heat transfer. Traditional finite difference discretization schemes such as the second order central difference scheme (CDS) and the upwind difference scheme (UDS) cannot provide satisfactory results [23]. The CDS scheme has a truncation error of order $O(h^2)$, but classical iterative methods for solving the resulting linear system do not converge when Re becomes large. The UDS scheme usually has computational stability, but it will decrease the order of accuracy to $O(h)$ [13].

In order to find improved schemes which offer both stability and highly accurate approximate solutions, there has been growing interest in developing higher order compact (HOC) discretization schemes. Gupta *et al*. proposed a fourth order nine-point compact (FOC) scheme to discretize the two dimensional (2D) convection-diffusion equation with variable coefficients [8]. There are also some other similar fourth order compact schemes that have been developed for the convection-diffusion equations. Readers are referred to [5, 13, 14] for more details. Recently, sixth order schemes have attracted considerable attentions and have been applied to solve one, two and three dimensional convection diffusion equations [6, 7, 16, 19, 20].

For many practical problems with boundary layers in the computational domain, the above referred HOC schemes based on uniform grids may not get expected accurate approximate solutions. On the one hand, to acquire a high accuracy solution to the boundary layer problems, a certain number of mesh points should be put into the areas of steep solution gradients using appropriate methods. On the other hand, to avoid too many grid points in the computational domain and to reduce the total computational cost, the smooth solution areas should include relatively few grid points. There are several approaches such as local mesh refinement strategies [17, 24], grid adaptive algorithms [1, 3] and coordinate transformation techniques [9, 15] for handling boundary layer problems.

As far as we know, there are two approaches to construct HOC schemes on nonuniform grids. One is to use coordinate transformation technique, the other does not use any transformation. Ge and Zhang [10, 11] used coordinate transformation technique on 2D convection diffusion equation to map the physical space with nonuniform grids onto a computational space with uniform grids. The HOC difference scheme can then be applied on a transformed set of equations on the transformed uniform grid. Finally, the solutions can be mapped back onto the physical space. Kalita *et al*. [12] proposed an transformation-free HOC difference scheme on rectangular nonuniform grids for the 2D convection diffusion equation. It was based on the Taylor series expansion of a continuous function at a particular point for two different step lengths and approximation of the derivatives appearing in the 2D convection diffusion equation on a nonuniform stencil. The original differential equation was then used again to replace the derivative terms appearing in the finite difference approximations, resulting in a fourth order scheme on a compact stencil of nine points. Transformation method is considered more expensive and complicated because of the appearance of new cross-derivative terms in the transformed equations [12], but it may solve problems in irregular domain by transforming irregular (nonorthogonal grids) physical space to regular (orthogonal grids) computational space. Though non-transformation method owns an easier scheme, it asks for rectangular non-uniform grids. At present, it is not clear which of these two approaches are better in solving convection diffusion equation with boundary layers. This paper is intended to compare these two schemes for accurate and efficient computation of the convection diffusion equations with boundary layers on nonuniform grids. To obtain accurate, efficient and robust computation of boundary layer problems, several strategies (discretization, grid stretching, iterative solution) are utilized.

An outline of the paper is as follows. In Section 2, we separately illustrate the HOC schemes for transformed and non-transformed two dimensional convection diffusion equation on nonuniform grids. Experimental methods and numerical results are presented and interpreted in Section 3. Concluding remarks are summarized in Section 4.

2 Higher order compact discretization schemes on nonuniform grids

2.1 HOC scheme for transformed equation

Consider a nondegenerate map $x = x(\xi, \eta)$, $y = y(\xi, \eta)$, which transforms Equation (1) from a graded mesh on $0 < x < 1$, $0 < y < 1$ to a uniform mesh on $0 < \xi < 1$, $0 < \eta < 1$. The transformed equation can be written as [11]

$$\alpha(\xi,\eta)u_{\xi\xi} + \beta(\xi,\eta)u_{\eta\eta} + c(\xi,\eta)u_{\xi\eta} + \lambda(\xi,\eta)u_{\xi} + \mu(\xi,\eta)u_{\eta} = f(\xi,\eta) \tag{2}$$

where the coefficients are given by

$$\begin{aligned}
\alpha(\xi,\eta) &= \xi_x^2 + \xi_y^2, \\
\beta(\xi,\eta) &= \eta_x^2 + \eta_y^2, \\
c(\xi,\eta) &= 2(\xi_x\eta_x + \xi_y\eta_y), \\
\lambda(\xi,\eta) &= p(\xi,\eta)\xi_x + q(\xi,\eta)\xi_y + \xi_{xx} + \xi_{yy}, \\
\mu(\xi,\eta) &= p(\xi,\eta)\eta_x + q(\xi,\eta)\eta_y + \eta_{xx} + \eta_{yy}.
\end{aligned}$$

The difference between the transformed Equation (2) and the original Equation (1) is the variable coefficients α, β, and c of the second order derivative terms appeared in the transformed equation, but not in the original equation. Because of the orthogonal grids used in the current study, the coefficient $c(\xi, \eta)$ is identically zero throughout Ω. Hence, Equation (2) is simplified as

$$\alpha(\xi,\eta)u_{\xi\xi} + \beta(\xi,\eta)u_{\eta\eta} + \lambda(\xi,\eta)u_{\xi} + \mu(\xi,\eta)u_{\eta} = f(\xi,\eta) \tag{3}$$

Ge and Zhang [11] obtained the fourth order compact difference scheme by substituting the Taylor series expansions into the Equation (3) and by obtaining the representation for u in the Equation (3) to obtain a finite difference formula of order 4. This is achieved by truncating the Taylor series up to order 4 (by setting all the Taylor series coefficients of u_{ij} to zero for $i + j > 4$).

The nine point compact finite difference scheme of Equation (3) yields at each internal grid point a linear equation of the form [9]

$$\sum_{j=0}^{8} \alpha_j u_j = 6h^2 f_{00} + h^4[f_{20} + f_{02} + T_1 f_{10} + T_2 f_{01}], \tag{4}$$

where the coefficients are given by

$$
\begin{aligned}
\alpha_0 &= -(2R_1 + 2R_2 + 4S_1), \\
\alpha_1 &= R_1 + R_3, \\
\alpha_2 &= R_2 + R_4, \\
\alpha_3 &= R_1 - R_3, \\
\alpha_4 &= R_2 - R_4, \\
\alpha_5 &= S_1 + S_2 + S_3 + S_4, \\
\alpha_6 &= S_1 + S_2 - S_3 - S_4, \\
\alpha_7 &= S_1 - S_2 + S_3 - S_4, \\
\alpha_8 &= S_1 - S_2 - S_3 + S_4, \\
T_1 &= (\lambda_{00} - 2\alpha_{10})/(2\alpha_{00}), \\
T_2 &= (\mu_{00} - 2\beta_{01})/(2\beta_{00}), \\
R_1 &= 5\alpha_{00} - \beta_{00} + T_1 h^2(\lambda_{00} + \alpha_{10}) + T_2 h^2 \alpha_{01} + h^2(\alpha_{20} + \alpha_{02} + \lambda_{10}), \\
R_2 &= 5\beta_{00} - \alpha_{00} + T_2 h^2(\mu_{00} + \beta_{01}) + T_1 h^2 \beta_{10} + h^2(\beta_{20} + \beta_{02} + \mu_{01}), \\
R_3 &= \frac{h}{2}(5\lambda_{00} - 2\beta_{10} - 2\beta_{00}T_1) + \frac{h^3}{2}(T_1\lambda_{10} + T_2\lambda_{01} + \lambda_{20} + \lambda_{02}), \\
R_4 &= \frac{h}{2}(5\mu_{00} - 2\alpha_{01} - 2\alpha_{00}T_2) + \frac{h^3}{2}(T_2\mu_{01} + T_1\mu_{10} + \mu_{20} + \mu_{02}), \\
S_1 &= \frac{1}{2}(\alpha_{00} + \beta_{00}), \\
S_2 &= \frac{h}{4}(\mu_{00} + 2\alpha_{01} + 2T_2\alpha_{00}), \\
S_3 &= \frac{h^2}{4}(\mu_{10} + \lambda_{01} + T_1\mu_{00} + T_2\lambda_{00}), \\
S_4 &= \frac{h}{4}(\lambda_{00} + 2\beta_{10} + 2T_1\beta_{00}).
\end{aligned}
$$

The truncation error of the scheme (4) is $O(h^4)$, see [9] for details.

The Equation (4) utilizes partial derivatives of the function $\alpha, \beta, \lambda, \mu$ and f. A double subscript "ij" on any of these functions denotes the (i + j)th partial derivative defined by

$$\alpha_{ij} = \frac{1}{i!j!}\frac{\partial^{i+j}\alpha}{\partial\xi^i \partial\eta^j}$$

In this formulation, it requires that the partial derivatives exist analytically. It is also possible to approximate these partial derivatives by finite difference formulas. Actually, We used CDS for the partial derivatives approximation in our experiment.

2.2 HOC scheme for non-transformed equation

Consider a rectangular domain $a_1 \le x \le a_2$, $b_1 \le y \le b_2$. We divide the interval $[a_1, a_2]$ and $[b_1, b_2]$ into sub-intervals by the points $a_1 = x_0, x_1, x_2, ..., x_{m-1}, x_m = a_2$, $b_1 = y_0, y_1, y_2, ..., y_{n-1}, y_n = b_2$. In the x-direction, the forward and backward step lengths are given by $x_f = x_{i+1} - x_i$ and $x_b = x_i - x_{i-1}$, respectively, and similarly, in the y-direction, we have $y_f = y_{j+1} - y_j$ and $y_b = y_j - y_{j-1}$, $1 \le i \le m-1$, $1 \le j \le n-1$. The FOC finite difference scheme for non-transformed equation (1) on nonuniform grids derived by Kalita *et al.* [12] can be written as

$$[-A_{ij}\delta_x^2 - B_{ij}\delta_y^2 + C_{ij}\delta_x + D_{ij}\delta_y + G_{ij}\delta_x\delta_y - H_{ij}\delta_x\delta_y^2 - K_{ij}\delta_x^2\delta_y - L_{ij}\delta_x^2\delta_y^2]\phi_{ij} = F_{ij} \quad (5)$$

where the coefficients A_{ij}, B_{ij}, C_{ij}, D_{ij}, G_{ij}, H_{ij}, K_{ij}, L_{ij} and F_{ij} are given by

$$\begin{aligned}
A_{ij} &= 1 - [(H_1 + H_2 p)p + 2H_2\{\delta_x p - 0.5(x_f - x_b)\delta_x^2 p\}] + 0.5(x_f - x_b)C_{ij},\\
B_{ij} &= 1 - [(K_1 + K_2 q)q + 2K_2\{\delta_y q - 0.5(y_f - y_b)\delta_y^2 q\}] + 0.5(y_f - y_b)D_{ij},\\
C_{ij} &= [1 + (H_1 + H_2 p)\delta_x + (K_1 + K_2 q)\delta_y + \{H_2 - 0.5(x_f - x_b)(H_1 + H_2 p)\}\delta_x^2\\
&\quad + \{K_2 - 0.5(y_f - y_b)(K_1 + K_2 q)\}\delta_y^2]p,\\
D_{ij} &= [1 + (H_1 + H_2 p)\delta_x + (K_1 + K_2 q)\delta_y + \{H_2 - 0.5(x_f - x_b)(H_1 + H_2 p)\}\delta_x^2\\
&\quad + \{K_2 - 0.5(y_f - y_b)(K_1 + K_2 q)\}\delta_y^2]q,\\
G_{ij} &= (H_1 + H_2 p)q + (K_1 + K_2 q)p + 2H_2\delta_x q + 2K_2\delta_y p\\
&\quad - \{H_2(x_f - x_b)\delta_x^2 q + K_2(y_f - y_b)\delta_y^2 p\},\\
H_{ij} &= H_1 + H_2 p - K_2 p,\\
K_{ij} &= K_1 + K_2 q - H_2 q,\\
L_{ij} &= H_2 + K_2,\\
F_{ij} &= [1 + (H_1 + H_2 p)\delta_x + (K_1 + K_2 q)\delta_y + \{H_2 - 0.5(x_f - x_b)(H_1 + H_2 p)\}\delta_x^2\\
&\quad + \{K_2 - 0.5(y_f - y_b)(K_1 + K_2 q)\}\delta_y^2]f_{ij}.
\end{aligned}$$

and

$$\begin{aligned}
H_1 &= \frac{1}{6}\{2(x_f - x_b) - p x_f x_b\}, \qquad H_2 = \frac{1}{24}\{2(x_f^2 + x_b^2 - x_f x_b) - p x_f x_b (x_f - x_b)\},\\
K_1 &= \frac{1}{6}\{2(y_f - y_b) - q y_f y_b\}, \qquad K_2 = \frac{1}{24}\{2(y_f^2 + y_b^2 - y_f y_b) - q y_f y_b (y_f - y_b)\}.
\end{aligned}$$

The details of the finite difference operators δ_x, δ_y, δ_x^2, δ_y^2, $\delta_x\delta_y$, $\delta_x^2\delta_y$, $\delta_x\delta_y^2$ and $\delta_x^2\delta_y^2$ are given in the Appendix.

3 Numerical Results

In this section, we compare HOC scheme for 2D transformed convection diffusion equation with HOC scheme for 2D non-transformed convection diffusion equation by using multigrid method. Since both Equation (3) and Equation (5) form large sparse linear systems and multigrid method (MG) has been considered as the most efficient algorithm for solving linear systems arising from PDEs [2, 4, 18], we used standard V(1,1)-cycle multigrid method with the alternating line Guass-Seidel relaxation which has been used by Ge and Zhang [11]. The alternating line Gauss-Seidel relaxation is shown to be a more robust smoother for all Reynolds numbers than the point Gauss-Seidel relaxation and the line Gauss-Seidel relaxation for the multi-level methods [22]. In addition, standard bilinear interpolation operator and full weighting restriction operator are used as the inter grid transfer operators [18]. The initial guess for the V-cycle was the zero vector. The convergence criterion for the iteration was chosen to be 10^{-10}. The errors reported were the maximum absolute errors over the discrete grid of the finest level.

The codes are written in Fortran 77 programming language and run on a computer with Intel Pentium 4 3.00GHz and 1GB memory.

The test case is a constant coefficient convection diffusion equation

$$-\epsilon(u_{xx}+u_{yy})+u_x=0,\ \ 0\le x\le 1, 0\le y\le 1 \tag{6}$$

with the boundary conditions given as

$$u(x,0)=u(x,1)=0; \qquad u(0,y)=\sin\pi y; \qquad u(1,y)=2\sin\pi y.$$

The exact solution is [9]:

$$u(x,y)=\exp(x/2\epsilon)\sin\pi y[2\exp(-1/2\epsilon)\sinh\sigma x+\sinh\sigma(1-x)]/\sinh\sigma$$

where $\sigma^2=\pi^2+0.25/\epsilon^2.(Re=1/\epsilon)$

This problem represents a convection dominated flow. It was used as a test problem by Gupta *et al.* [9] for $0.01\le\epsilon\le 1$ and by Ge and Zhang [11] for $0.001\le\epsilon\le 1$. The coefficient of the convective term is a constant.

The coordinate transformation we use is $x=[1-\exp(-Q\xi)]/[1-\exp(-Q)]$. This transformation maps the interval $0<x<1$ onto $0<\xi<1$; the y coordinate direction is not changed. Before transformation, the graded mesh in x is coarser near $x=0$ and finer near $x=1$. The parameter Q relates the coarsest mesh width D near $x=0$ with the finest mesh width d near $x=1$. Note that $Q=\ln(\gamma)/(1-\Delta\xi)$, where $\gamma=D/d$ is the mesh stretching ratio and $\Delta\xi=1/N$ (N is the total number of mesh intervals in the ξ coordinate direction). Using the proposed coordinate transformation, the equation (6) becomes

$$(\frac{z}{Q})^2u_{\xi\xi}+u_{yy}-\frac{z}{Q}(\frac{1}{\epsilon}-z)u_\xi=0 \tag{7}$$

where $z=\exp(Q\xi)[1-\exp(-Q)]$. The original equation (6) is transformed from a constant problem to a variable coefficient problem since the parameter z is now a function of the independent variable ξ.

To compare two different HOC schemes, we first computed the numerical solutions of the equation (7) on transformed uniform grids for several values of the perturbation parameter ϵ and the mesh stretching parameter γ. Sample results for $\epsilon\ge 0.001$ are given in Table 1. Then we computed the numerical solutions of equation (6) using HOC scheme for non-transformed equation on similar groups of perturbation parameter and mesh stretching parameter. Results are shown in Table 2. In both tables, we listed the CPU time of each computation and the order of accuracy. The order was computed by

$$order=\log_2(\frac{error_h=64}{error_h=128})$$

From both tables, we noticed that for the same ϵ and N, the larger the stretching ratio γ, the more accurate the computed solution. A larger stretching ratio means that it puts too many grid points along the boundary $x=1$ and too few grid points along the boundary $x=0$. However, too much stretching does not always have positive effect. In the Table 1, for $\epsilon=0.1$ and N=32, the use of the stretching ratio $\gamma=50$ yielded computed solution with a lower accuracy than that with a stretching ratio $\gamma=10$. Similar situation happened when $\epsilon=0.01$, $\gamma=100$ and N=128 (see Table 1, row 11 and column 7). The reason may be

Table 1: Maximum absolute error, CPU time in seconds and accuracy using HOC scheme for transformed equation with different perturbation ϵ, stretching ratio γ, and discretization parameter N.

		$N = 32$		$N = 64$		$N = 128$		
ϵ	γ	Error	CPU time	Error	CPU time	error	CPU time	Order
1	5	2.39(-6)	0.012	1.63(-7)	0.060	8.33(-9)	0.312	4.05
0.1	5	1.76(-5)	0.012	1.06(-7)	0.060	7.54(-9)	0.312	4.00
	10	3.29(-6)	0.016	2.02(-7)	0.084	7.79(-9)	0.400	4.12
	50	4.14(-5)	0.024	2.52(-6)	0.116	1.38(-7)	0.536	3.93
0.01	5	1.38(-4)	0.024	8.43(-6)	0.124	5.62(-7)	0.596	4.07
	10	5.47(-5)	0.032	3.31(-6)	0.156	2.23(-7)	0.828	4.09
	20	2.70(-5)	0.032	1.63(-6)	0.180	1.08(-7)	1.032	4.08
	50	1.66(-5)	0.040	1.61(-6)	0.224	6.10(-8)	1.432	3.99
	100	4.31(-5)	0.036	2.82(-6)	0.252	1.49(-7)	1.876	3.85
	200	8.36(-5)	0.064	5.44(-6)	0.408	2.98(-7)	3.368	3.88
0.001	50	7.88(-4)	0.020	4.67(-5)	0.180	3.62(-6)	1.440	4.18
	100	2.55(-4)	0.016	1.53(-5)	0.176	1.17(-6)	1.620	4.17
	200	1.12(-4)	0.016	7.44(-6)	0.160	5.55(-7)	1.640	4.05
	300	7.84(-5)	0.020	5.72(-6)	0.168	4.35(-7)	1.660	3.96
	400	6.57(-5)	0.024	5.38(-6)	0.168	3.89(-7)	1.644	3.88

that the boundary layer is not too steep. Also, as we expected, given the same stretching ratio, more accurate computed solution can be obtained when finer mesh is used.

Compared two tables, it can be seen that these two schemes attained fourth order accuracy for perturbation $\epsilon \geq 0.1$. However, when $\epsilon < 0.1(\epsilon = 0.01, 0.001)$, the transformation method can still achieve fourth order accuracy, but the non-transformation method did not converge. For smaller ϵ denotes larger Reynolds number ($Re = 1/\epsilon$), it shows that the non-transformation method is hard to converge when Re increases. In addition, the CPU time for transformation method is much less than non-transformation method.

4 Concluding Remarks

We studied two higher order compact difference schemes with transformation and non-transformation to solve the two dimensional convection diffusion equation with boundary layers on nonuniform grids. One method is using grid transformation technique to map the nonuniform grid to a uniform grid and then the higher order compact scheme is able to applied on it. The other method is deriving a higher order compact scheme for nonuniform grids directly instead of using transformation. The main contribution of this paper is to compare the accuracy, robustness and efficiency between these two methods. The numerical results shows that the transformation method is more robust and efficient than the non-transformation method to solve the equation on nonuniform grids. Therefore, the transformation method is better than the non-transformation method in solving boundary layers because of its accuracy, stability and efficiency. Also for the transformation method finally solves the problem on uniform grids, many research achievements (such as developing sixth order schemes by using extrapolation technique) about computation on uniform grids may be extended on to nonuniform grids in the future.

Table 2: Maximum absolute error, CPU time in seconds and accuracy using HOC scheme for non-transformed equation with different perturbation ϵ, stretching ratio γ, and discretization parameter N.

		$N=32$		$N=64$		$N=128$		
ϵ	γ	Error	CPU time	Error	CPU time	Error	CPU time	Order
1	5	7.16(-7)	0.016	4.37(-8)	0.080	2.68(-9)	0.352	4.02
0.1	5	9.01(-6)	0.012	5.44(-7)	0.076	3.34(-8)	0.340	4.03
	10	1.80(-5)	0.024	1.08(-6)	0.080	6.59(-8)	0.392	4.03
	50	8.45(-5)	0.060	4.89(-6)	0.152	2.96(-7)	1.310	4.05
0.01	5	1.80(-4)	0.012	3.32(-6)	0.080	1.43(-7)	0.344	4.53
	10	Diverge	N/A	Diverge	N/A	Diverge	N/A	N/A
	50	Diverge	N/A	Diverge	N/A	Diverge	N/A	N/A
	100	Diverge	N/A	Diverge	N/A	Diverge	N/A	N/A
0.001	50	Diverge	N/A	Diverge	N/A	Diverge	N/A	N/A
	100	Diverge	N/A	Diverge	N/A	Diverge	N/A	N/A
	200	Diverge	N/A	Diverge	N/A	Diverge	N/A	N/A

References

[1] A. S. Almgren, J. B. Bell, P. Colella, W. L. Welcome. A conservative adaptive projection method for the variable density imcompressible Navier-Stokes equations. *J. Comput. Phys.*, 142:1-46, 1998.

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

[3] M. J. Berger and P. Coella, Local adaptive mesh refinement for shock hydrodynamics. *J. Comput. Phys.*, 82:64-84. 1997

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

[5] 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.

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

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

[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, R. P. Manohar, and J. W. Stephenson. High-order difference schemes for two-dimensional elliptic equations. *Numer. Methods Partial Differential Eq.*, 1:71-78, 1985.

[10] L. Ge and J. Zhang. Accuracy, robustness, and efficiency comparison in iterative computation of convection diffusion equation with boundary layers. *Numer. Methods Partial Differential Eq.*, 16(4):379-394, 2000.

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

[12] J. C. Kalita, A. K. Dass, D. C. Dalal. A transformation-free HOC scheme for steady convection-diffusion on non-uniform grids. *Int. J. Numer. Methods Fluids*, 44:33-53, 2004.

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

[14] 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.

[15] W. F. Spotz and G. F. Carey. Formulation and experiments with high-order compact schemes for nonuniform grids. *Int. J. Numer. Methods Fluids*, 8(3):288-303, 1998.

[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] R. Teigland, I. K. Eliassen, A multiblock/multilevel mesh refinement procedure for CFD computations. *Int. J. Numer. Methods Fluids*, 36:519-538, 2001

[18] P. Wesseling. *An introduction to Multigrid Methods*. Wiley, Chichester, England, 1992.

[19] Y. Wang and J. Zhang. Integrated fast and high accuracy computation of convection diffusion equations using multiscale mutigrid method. to appear in *Numer. Methods Partial Differential Equation*.

[20] Y. Wang and J. Zhang. Fast and robust sixth-order multigrid computation for the three-dimensioinal convection-diffusion equation. *J. Comput. Appl. Math.*, 234:3496-3506, 2010.

[21] 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.

[22] J. Zhang. On convergence and Performance of iterative methods with fourth-order compact schemes. *Numer. Methods Partial Differential Eq.*, 14:262-283, 1998.

[23] J. Zhang, L. Ge, and J. Kouatchou. A two colorable fourth-order compact difference scheme and parallel iterative solution of the 3D convection diffusion equation. *Math. Comput. Simulation*, 54(1-3):65-80, 2000.

[24] J. Zhang, H. Sun, and J. J. Zhao. High order compact scheme with multigrid local mesh refinement procedure for convection diffusion problems. *Comput. Methods Appl. Mech. Engrg.*, 191(41-42):4661-4674, 2002.

Appendix

The expressions for the finite difference operators appearing in Equation (5) are as follows:

$$
\begin{aligned}
\delta_x \Phi_{ij} &= \frac{\Phi_{i+1,j} - \Phi_{i-1,j}}{2h} \\
\delta_y \Phi_{ij} &= \frac{\Phi_{i,j+1} - \Phi_{i,j-1}}{2k} \\
\delta_x^2 \Phi_{ij} &= \frac{1}{h}[\frac{\Phi_{i+1,j}}{x_f} - (\frac{1}{x_f} + \frac{1}{x_b})\Phi_{ij} + \frac{\Phi_{i-1,j}}{x_b}] \\
\delta_y^2 \Phi_{ij} &= \frac{1}{k}[\frac{\Phi_{i,j+1}}{y_f} - (\frac{1}{y_f} + \frac{1}{y_b})\Phi_{ij} + \frac{\Phi_{i,j-1}}{y_b}] \\
\delta_x \delta_y \Phi_{ij} &= \frac{1}{4hk}(\Phi_{i+1,j+1} - \Phi_{i+1,j-1} - \Phi_{i-1,j+1} + \Phi_{i-1,j-1}) \\
\delta_x^2 \delta_y \Phi_{ij} &= \frac{1}{2hk}\{\frac{1}{x_f}(\Phi_{i+1,j+1} - \Phi_{i+1,j-1}) - (\frac{1}{x_f} + \frac{1}{x_b})(\Phi_{i,j+1} - \Phi_{i,j-1}) \\
&\quad + \frac{1}{x_b}(\Phi_{i-1,j+1} - \Phi_{i-1,j-1})\} \\
\delta_x \delta_y^2 \Phi_{ij} &= \frac{1}{2hk}\{\frac{1}{y_f}(\Phi_{i+1,j+1} - \Phi_{i-1,j+1}) - (\frac{1}{y_f} + \frac{1}{y_b})(\Phi_{i+1,j} - \Phi_{i-1,j}) \\
&\quad + \frac{1}{y_b}(\Phi_{i+1,j-1} - \Phi_{i-1,j-1})\} \\
\delta_x^2 \delta_y^2 \Phi_{ij} &= \frac{1}{hk}\{\frac{\Phi_{i+1,j+1}}{x_f y_f} + \frac{\Phi_{i-1,j+1}}{x_b y_f} - (\frac{1}{x_f y_f} + \frac{1}{x_b y_f})\Phi_{i,j+1} - (\frac{1}{x_f y_f} + \frac{1}{x_f y_b})\Phi_{i+1,j} \\
&\quad + (\frac{1}{x_f y_f} + \frac{1}{x_f y_b} + \frac{1}{x_b y_f} + \frac{1}{x_b y_b})\phi_{ij} \\
&\quad - (\frac{1}{x_f y_b} + \frac{1}{x_b y_b})\Phi_{i,j-1} - (\frac{1}{x_b y_f} + \frac{1}{x_b y_b})\Phi_{i-1,j} + \frac{\Phi_{i+1,j-1}}{x_f y_b} + \frac{\Phi_{i-1,j-1}}{x_b y_b}\}
\end{aligned}
$$