

High Order Compact Difference Scheme and Multigrid Method for 2D Elliptic Problems with Variable Coefficients and Interior/Boundary Layers on Nonuniform Grids

Bin Lan¹, Yongbin Ge^{1*}, Yan Wang¹, Yong Zhan²

¹Institute of Applied Mathematics and Mechanics, Ningxia University, Yinchuan, China ²Department of Environment and Architecture, University of Shanghai for Science and Technology, Shanghai, China

Email: ^{*}gyb@nxu.edu.cn

Received 7 April 2015; accepted 16 May 2015; published 19 May 2015

Copyright © 2015 by authors and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY). http://creativecommons.org/licenses/by/4.0/

CC O Open Access

Abstract

In this paper, a high order compact difference scheme and a multigrid method are proposed for solving two-dimensional (2D) elliptic problems with variable coefficients and interior/boundary layers on nonuniform grids. Firstly, the original equation is transformed from the physical domain (with a nonuniform mesh) to the computational domain (with a uniform mesh) by using a coordinate transformation. Then, a fourth order compact difference scheme is proposed to solve the transformed elliptic equation on uniform girds. After that, a multigrid method is employed to solve the linear algebraic system arising from the difference equation. At last, the numerical experiments on some elliptic problems with interior/boundary layers are conducted to show high accuracy and high efficiency of the present method.

Keywords

Elliptic Equation, Coordinate Transformation, High Order Compact Difference Scheme, Multigrid Method, Interior/Boundary Layer

^{*}Corresponding author.

How to cite this paper: Lan, B., Ge, Y.B., Wang, Y. and Zhan, Y. (2015) High Order Compact Difference Scheme and Multigrid Method for 2D Elliptic Problems with Variable Coefficients and Interior/Boundary Layers on Nonuniform Grids. *Journal of Applied Mathematics and Physics*, **3**, 509-523. <u>http://dx.doi.org/10.4236/jamp.2015.35063</u>

1. Introduction

Elliptic equations are widely used in the fields of solid mechanics, material science, and image processing and so on. So it is both theoretically and practically important to investigate numerical methods for such equations. Finite difference method is a general and effective method to solve elliptic equations. In the past three decades, a large number of high order compact (HOC) difference schemes [1]-[14] have been developed to overcome the deficiencies (lower accuracy or numerical oscillation, etc.) of the conventional difference schemes. These HOC schemes own high accuracy with small mesh stencil and very effective for solving the problems with smooth solutions. And they can suppress numerical oscillations that may occur if standard second order central difference scheme is used to solve the convection-dominated problem or high Reynolds number problems [12]. But we notice that many HOC schemes among them are constructed directly on the uniform grids because it is easy to be implemented in practice. However, for the computation of many problems whose physical quantities unevenly distributed in spaces or with areas of steep solution gradients, the advantages of the HOC scheme may be lost if there were no enough grid points inside the large gradient areas. Very fine discretization could be used in the whole domain to yield an approximate solution of acceptable accuracy with uniform grids. But such a fine discretization results in a very large linear system that demands a large computational cost. In other words, it would lead huge waste of computational amount if uniform grids are used in the whole physical domain. To avoid too many grid points in the computational domain and to reduce the total computational cost, we can place clustered mesh grids in the area of large gradient while relatively few grids in the smooth region. So, developing efficient difference schemes on nonuniform grids has a very important application value and actual significance.

Coordinate transformation method [11]-[15] is a commonly-used method to achieve computation on physical nonuniform grids. This method needs to transform the nonuniform grids in the physical domain to the uniform grids in the computational domain by using reversible coordinate transformation functions. After computation, it returns the computed results back to the physical domain by the inverse transformation. The method has its advantages. Firstly, it can be used to construct HOC schemes more easily on uniform grids than on nonuniform grids by discretizing the derivative terms directly; secondly, such transformations are also used to reflect many interior/boundary layer phenomena without refining the mesh near to the interior/boundary layers in the computational domain. A few researchers have used this method to deal with convection diffusion equations or Navier-Stokes equations. For instance, Choo and Schultz [11] used the transformation method and developed a fourth order compact difference scheme to solve the steady Navier-Stokes equations. The results show that the method is accurate and stable. Spotz [12] developed a class of HOC finite difference schemes for steady convection diffusion equation on uniform grids. And then he extended them to nonuniform grids by using the coordinate transformation method. Ge and Zhang [13] also solved the 2D convection diffusion equations with boundary layers using the coordinate transformation method and a fourth order scheme was applied on the uniform computational grids. The authors extended the coordinate transformation method to the three-dimensional (3D) case [14] to resolve 3D boundary layer problems. Liu C. and Liu Z. [15] employed the coordinate transformation and combined it with a fourth order finite difference scheme and multigrid method to simulate the whole process of flow transition in 3D boundary layers. If we fix our attention on elliptic equations, we notice that the coefficients of the first or second order derivatives in the original model equations considered in the literature are constant [11]-[14]. Although after coordinate transformation, the coefficients turn to be variable, the high order difference schemes, which are developed based on it, could not be used to compute the solutions of original model equations in which the coefficients of the first or second order derivatives are variable. So, the potential advantages of applying coordinate transformation method and HOC schemes to solve variable coefficients elliptic problems have not been fully investigated.

In this paper, we consider the 2D elliptic equation with variable coefficients as follows:

$$Lu = Au_{xx} + Cu_{yy} + Du_{x} + Eu_{y} + Fu = G$$
(1)

where u(x, y) is unknown function and the coefficients *A*, *C*, *D*, *E*, *F* and the right hand term *G* are the functions of *x*, *y* and are assumed to be continuously differentiable. We are aiming at developing an HOC scheme which is based on coordinate transformation and multigrid method to solve Equation (1) on nonuniform grids. The remainder of this paper is arranged as follows. Section 2 gives the method of coordinate transformation, which transforms nonuniform grids in physical domain onto uniform grids in computational domain. In Section 3, an HOC scheme on uniform grids is constructed to solve the transformed equation in the computational do-

main. After that, a multigrid method based on the HOC scheme is introduced in Section 4. In Section 5, numerical experiments are carried out to show the accuracy and efficiency of the present method. Especially, some problems with interior or boundary layers are considered in this section. Finally, some concluding remarks are given in Section 6.

2. Coordinate Transformation

We consider a rectangular physical domain Ω and $\partial \Omega$ is its boundary. We transform the independent variables x and y in the physical domain into new independent variables ξ and η in the computational domain. The transformation functions are written as:

$$x = x(\xi, \eta), \quad y = y(\xi, \eta). \tag{2}$$

Then, Equation (1) is transformed into the form as:

$$a\hat{u}_{\xi\xi} + b\hat{u}_{\xi\eta} + c\hat{u}_{\eta\eta} + d\hat{u}_{\xi} + e\hat{u}_{\eta} + f\hat{u} = g$$
(3)

where the coefficients a, b, c, d, e, f and the right hand term g are the functions of ξ, η , and

$$a(\xi,\eta) = \hat{A}\xi_x^2 + \hat{C}\xi_y^2, \tag{4}$$

$$b(\xi,\eta) = 2\hat{A}\xi_x\eta_x + 2\hat{C}\xi_y\eta_y, \tag{5}$$

$$c(\xi,\eta) = \hat{A}\eta_x^2 + \hat{C}\eta_y^2, \tag{6}$$

$$d\left(\xi,\eta\right) = \hat{A}\xi_{xx} + \hat{C}\xi_{yy} + \hat{D}\xi_{x} + \hat{E}\xi_{y},\tag{7}$$

$$e(\xi,\eta) = \hat{A}\eta_{xx} + \hat{C}\eta_{yy} + \hat{D}\eta_x + \hat{E}\eta_y, \qquad (8)$$

$$f\left(\xi,\eta\right) = \hat{F}, \ g\left(\xi,\eta\right) = \hat{G} \tag{9}$$

where \hat{A} , \hat{C} , \hat{D} , \hat{E} , \hat{F} and \hat{G} are the functions of ξ and η . By use of the coordinate transformation, Equation (1) is transformed into Equation (3) and physical domain Ω is transformed into another domain which is called computational domain and we mark it as $\hat{\Omega}$. Then, we suppose to build an HOC scheme for Equation (3) in the computational domain. In Ref. [5], the authors point out that there does not exist fourth order compact difference scheme for the general elliptic equations like Equation (3). However, if we adopt the 1D transformation grids to respectively discrete the difference equation in the two directions (x- and y-direction); *i.e.*, if we use $\xi = \xi(x)$ and $\eta = \eta(y)$, by this transformation, we get orthogonal grids in the computational domain and the coefficient $b(\xi, \eta)$ is identically zero throughout $\hat{\Omega}$. Under such circumstance, Equation (3) is simplified as

$$a\hat{u}_{\xi\xi} + c\hat{u}_{\eta\eta} + d\hat{u}_{\xi} + e\hat{u}_{\eta} + f\hat{u} = g.$$

$$\tag{10}$$

Then, for Equation (10), a fourth order compact difference scheme can be derived. We will give the derivation process of the fourth order compact scheme in the next section.

3. HOC Difference Scheme

Firstly, we divide the computational domain $\hat{\Omega}$ with uniform grid and assume h_1 and h_2 are step lengths in the ξ - and η -direction. To keep compactness of the scheme, we use reference grid point (ξ,η) and its eight neighbor grid points $(\xi \pm h_1,\eta)$, $(\xi,\eta \pm h_2)$, $(\xi \pm h_1,\eta \pm h_2)$. Correspondingly, the values of \hat{u} are denoted by $\hat{u}_{i,j}, \hat{u}_{i+1,j}, \hat{u}_{i,j+1}, \hat{u}_{i-1,j}, \hat{u}_{i+1,j+1}, \hat{u}_{i+1,j-1}, \hat{u}_{i-1,j+1}$ and $\hat{u}_{i-1,j-1}$. Then, we use Taylor's series expansions and central difference operators (the detailed expressions about these central difference operators are given in Appendix), and the following derivatives approximations can be obtained:

$$\left(\hat{u}_{\xi}\right)_{ij} = \delta_{\xi}\hat{u}_{ij} - \frac{h_{1}^{2}}{6}\left(\hat{u}_{\xi\xi\xi}\right)_{ij} + O\left(h_{1}^{4}\right),\tag{11}$$

$$\left(\hat{u}_{\eta}\right)_{ij} = \delta_{\eta}\hat{u}_{ij} - \frac{h_{2}^{2}}{6}\left(\hat{u}_{\eta\eta\eta}\right)_{ij} + O\left(h_{2}^{4}\right), \tag{12}$$

$$\left(\hat{u}_{\xi\xi}\right)_{ij} = \delta_{\xi}^{2} \hat{u}_{ij} - \frac{h_{l}^{2}}{12} \left(\hat{u}_{\xi\xi\xi\xi}\right)_{ij} + O\left(h_{l}^{4}\right), \tag{13}$$

$$\left(\hat{u}_{\eta\eta}\right)_{ij} = \delta_{\eta}^{2} \hat{u}_{ij} - \frac{h_{2}^{2}}{12} \left(\hat{u}_{\eta\eta\eta\eta}\right)_{ij} + O\left(h_{2}^{4}\right), \tag{14}$$

$$\left(\hat{u}_{\xi\eta}\right)_{ij} = \delta_{\xi}\delta_{\eta}\hat{u}_{ij} - \frac{1}{6}\left(h_{1}^{2}\hat{u}_{\xi\xi\xi\eta} + h_{2}^{2}\hat{u}_{\xi\eta\eta\eta}\right)_{ij} + O\left(h_{1}^{4} + h_{1}^{3}h_{2} + h_{1}^{2}h_{2}^{2} + h_{1}h_{2}^{3} + h_{2}^{4}\right).$$
(15)

Then the following difference equation is got if we substitute Equations (11)-(15) into Equation (10):

$$a_{ij}\delta_{\xi}^{2}\hat{u}_{ij} + c_{ij}\delta_{\eta}^{2}\hat{u}_{ij} + d_{ij}\delta_{\xi}\hat{u}_{ij} + e_{ij}\delta_{\eta}\hat{u}_{ij} + f_{ij}\hat{u}_{ij} + T_{ij} = g_{ij}.$$
 (16)

Equation (16) is actually the second order central difference scheme and T_{ij} is the truncation error, which can be written as:

$$T_{ij} = -\frac{1}{12} \left[a_{ij} h_1^2 \left(\hat{u}_{\xi\xi\xi\xi} \right)_{ij} + c_{ij} h_2^2 \left(\hat{u}_{\eta\eta\eta\eta} \right)_{ij} + 2d_{ij} h_1^2 \left(\hat{u}_{\xi\xi\xi} \right)_{ij} + 2e_{ij} h_2^2 \left(\hat{u}_{\eta\eta\eta} \right)_{ij} \right] + O\left(h_1^4 + h_1^3 h_2 + h_1^2 h_2^2 + h_1 h_2^3 + h_2^4 \right).$$
(17)

In order to get an HOC scheme, the third order and fourth order derivatives in T_{ij} need to be discretized, so we use Equation (10) to get the following equations:

$$\hat{u}_{\xi\xi\xi} = \frac{1}{a} \Big[g_{\xi} - (a_{\xi} + d) \hat{u}_{\xi\xi} - e \hat{u}_{\xi\eta} - c \hat{u}_{\xi\eta\eta} - c_{\xi} \hat{u}_{\eta\eta} - (d_{\xi} + f) \hat{u}_{\xi} - e_{\xi} \hat{u}_{\eta} - f_{\xi} \hat{u} \Big],$$
(18)

$$\hat{u}_{\eta\eta\eta} = \frac{1}{c} \Big[g_{\eta} - (c_{\eta} + e) \hat{u}_{\eta\eta} - d\hat{u}_{\xi\eta} - a\hat{u}_{\xi\xi\eta} - a_{\eta}\hat{u}_{\xi\xi} - (e_{\eta} + f) \hat{u}_{\eta} - d_{\eta}\hat{u}_{\xi} - f_{\eta}\hat{u} \Big],$$
(19)

$$\hat{u}_{\xi\xi\xi\xi} = \frac{1}{a} \Big[g_{\xi\xi} - (2a_{\xi} + d) \hat{u}_{\xi\xi\xi} - (a_{\xi\xi} + 2d_{\xi} + f) \hat{u}_{\xi\xi} - e \hat{u}_{\xi\xi\eta} - 2c_{\xi} \hat{u}_{\xi\eta\eta} - c_{\xi\xi} \hat{u}_{\eta\eta} - 2e_{\xi} \hat{u}_{\xi\eta} - c \hat{u}_{\xi\xi\eta\eta} - (d_{\xi\xi} + 2f_{\xi}) \hat{u}_{\xi} - e_{\xi\xi} \hat{u}_{\eta} - f_{\xi\xi} \hat{u} \Big],$$
(20)

$$\hat{u}_{\eta\eta\eta\eta} = \frac{1}{c} \Big[g_{\eta\eta} - (2c_{\eta} + e) \hat{u}_{\eta\eta\eta} - (c_{\eta\eta} + 2e_{\eta} + f) \hat{u}_{\eta\eta} - d\hat{u}_{\xi\eta\eta} - 2a_{\eta} \hat{u}_{\xi\xi\eta} - a_{\eta\eta} \hat{u}_{\xi\xi} - 2d_{\eta} \hat{u}_{\xi\eta} - a\hat{u}_{\xi\xi\eta\eta} - (e_{\eta\eta} + 2f_{\eta}) \hat{u}_{\eta} - d_{\eta\eta} \hat{u}_{\xi} - f_{\eta\eta} \hat{u} \Big].$$
(21)

Using the central differences, the difference approximation of $\hat{u}_{\xi\xi\xi\xi}$, $\hat{u}_{\eta\eta\eta}$, $\hat{u}_{\xi\xi\xi\xi}$ and $\hat{u}_{\eta\eta\eta\eta}$ at the grid point (ξ_i, η_j) can be obtained as follows:

$$\begin{pmatrix} \hat{u}_{\xi\xi\xi\xi} \end{pmatrix}_{ij} = \frac{1}{a_{ij}} \Big[\delta_{\xi} g_{ij} - \left(\delta_{\xi} a_{ij} + d_{ij} \right) \delta_{\xi}^{2} \hat{u}_{ij} - e_{ij} \delta_{\xi} \delta_{\eta} \hat{u}_{ij} - c_{ij} \delta_{\xi} \delta_{\eta}^{2} \hat{u}_{ij} - \delta_{\xi} c_{ij} \delta_{\eta}^{2} \hat{u}_{ij} \\ - \left(\delta_{\xi} d_{ij} + f_{ij} \right) \delta_{\xi} \hat{u}_{ij} - \delta_{\xi} e_{ij} \delta_{\eta} \hat{u}_{ij} - \delta_{\xi} f_{ij} \hat{u}_{ij} \Big] + O \Big(h_{1}^{2} + h_{1} h_{2} + h_{2}^{2} \Big),$$

$$(22)$$

$$\begin{pmatrix} \hat{u}_{\eta\eta\eta} \end{pmatrix}_{ij} = \frac{1}{c_{ij}} \Big[\delta_{\eta} g_{ij} - (\delta_{\eta} c_{ij} + e_{ij}) \delta_{\eta}^{2} \hat{u}_{ij} - d_{ij} \delta_{\xi} \delta_{\eta} \hat{u}_{ij} - a_{ij} \delta_{\xi}^{2} \delta_{\eta} \hat{u}_{ij} - \delta_{\eta} a_{ij} \delta_{\xi}^{2} \hat{u}_{ij} \\ - (\delta_{\eta} e_{ij} + f_{ij}) \delta_{\eta} \hat{u}_{ij} - \delta_{\eta} d_{ij} \delta_{\xi} \hat{u}_{ij} - \delta_{\eta} f_{ij} \hat{u}_{ij} \Big] + O(h_{1}^{2} + h_{1} h_{2} + h_{2}^{2}),$$

$$(23)$$

$$\begin{pmatrix} \hat{u}_{\xi\xi\xi\xi} \end{pmatrix}_{ij} = \frac{1}{a_{ij}} \bigg[\delta_{\xi}^{2} g_{ij} - (2\delta_{\xi}a_{ij} + d_{ij}) (\hat{u}_{\xi\xi\xi} \Big)_{ij} - (\delta_{\xi}^{2}a_{ij} + 2\delta_{\xi}d_{ij} + f_{ij}) \delta_{\xi}^{2} \hat{u}_{ij} - e_{ij} \delta_{\xi}^{2} \delta_{\eta} \hat{u}_{ij} - 2\delta_{\xi} c_{ij} \delta_{\xi} \delta_{\eta}^{2} \hat{u}_{ij} - \delta_{\xi}^{2} c_{ij} \delta_{\eta}^{2} \hat{u}_{ij} - 2\delta_{\xi} e_{ij} \delta_{\xi} \delta_{\eta} \hat{u}_{ij} - c_{ij} \delta_{\xi}^{2} \delta_{\eta}^{2} \hat{u}_{ij} - (\delta_{\xi}^{2} d_{ij} + 2\delta_{\xi} f_{ij}) \delta_{\xi} \hat{u}_{ij} - \delta_{\xi}^{2} e_{ij} \delta_{\eta} \hat{u}_{ij} - \delta_{\xi}^{2} f_{ij} \hat{u}_{ij} \bigg] + O(h_{1}^{2} + h_{1}h_{2} + h_{2}^{2}),$$

$$(24)$$

$$\begin{pmatrix} \hat{u}_{\eta\eta\eta\eta} \end{pmatrix}_{ij} = \frac{1}{c_{ij}} \begin{bmatrix} \delta_{\eta}^{2} g_{ij} - \left(2\delta_{\eta}c_{ij} + e_{ij}\right) \left(\hat{u}_{\eta\eta\eta}\right)_{ij} - \left(\delta_{\eta}^{2}c_{ij} + 2\delta_{\eta}e_{ij} + f_{ij}\right) \delta_{\eta}^{2}\hat{u}_{ij} \\ - d_{ij}\delta_{\xi}\delta_{\eta}^{2}\hat{u}_{ij} - 2\delta_{\eta}a_{ij}\delta_{\xi}^{2}\delta_{\eta}\hat{u}_{ij} - \delta_{\eta}^{2}a_{ij}\delta_{\xi}^{2}\hat{u}_{ij} - 2\delta_{\eta}d_{ij}\delta_{\xi}\delta_{\eta}\hat{u}_{ij} - a_{ij}\delta_{\xi}^{2}\delta_{\eta}^{2}\hat{u}_{ij} \\ - \left(\delta_{\eta}^{2}e_{ij} + 2\delta_{\eta}f_{ij}\right)\delta_{\eta}\hat{u}_{ij} - \delta_{\eta}^{2}d_{ij}\delta_{\xi}\hat{u}_{ij} - \delta_{\eta}^{2}f_{ij}\hat{u}_{ij} \end{bmatrix} + O\left(h_{1}^{2} + h_{1}h_{2} + h_{2}^{2}\right).$$

$$(25)$$

Finally, substituting Equations (22)-(25) into Equation (17), combining it with Equation (16) and neglecting high order truncation error term, we have:

$$A_{ij}^{*}\delta_{\xi}^{2}\hat{u}_{ij} + B_{ij}^{*}\delta_{\xi}\delta_{\eta}\hat{u}_{ij} + C_{ij}^{*}\delta_{\eta}^{2}\hat{u}_{ij} + D_{ij}^{*}\delta_{\xi}\hat{u}_{ij} + E_{ij}^{*}\delta_{\eta}\hat{u}_{ij} + F_{ij}^{*}\hat{u}_{ij} + P_{ij}^{*}\delta_{\xi}^{2}\delta_{\eta}^{2}\hat{u}_{ij} + Q_{ij}^{*}\delta_{\xi}^{2}\delta_{\eta}\hat{u}_{ij} + R_{ij}^{*}\delta_{\xi}\delta_{\eta}^{2}\hat{u}_{ij} = G_{ij}^{*},$$
(26)

where

$$A_{ij}^{*} = a_{ij} + \frac{1}{12} \Big[h_{1}^{2} \Big(\delta_{\xi}^{2} a_{ij} + 2\delta_{\xi} d_{ij} + f_{ij} + \alpha_{ij} \Big(\delta_{\xi} a_{ij} + d_{ij} \Big) \Big) + h_{2}^{2} \Big(\delta_{\eta}^{2} a_{ij} + \beta_{ij} \delta_{\eta} a_{ij} \Big) \Big],$$
(27)

$$B_{ij}^{*} = \frac{1}{12} \Big[h_{1}^{2} \Big(\alpha_{ij} e_{ij} + 2\delta_{\xi} e_{ij} \Big) + h_{2}^{2} \Big(\beta_{ij} d_{ij} + 2\delta_{\eta} d_{ij} \Big) \Big],$$
(28)

$$C_{ij}^{*} = c_{ij} + \frac{1}{12} \Big[h_{2}^{2} \Big(\delta_{\eta}^{2} c_{ij} + 2\delta_{\eta} e_{ij} + f_{ij} + \beta_{ij} \Big(\delta_{\eta} c_{ij} + e_{ij} \Big) \Big) + h_{1}^{2} \Big(\delta_{\xi}^{2} c_{ij} + \alpha_{ij} \delta_{\xi} c_{ij} \Big) \Big) \Big],$$
(29)

$$D_{ij}^{*} = d_{ij} + \frac{1}{12} \Big[h_{1}^{2} \Big(\delta_{\xi}^{2} d_{ij} + 2\delta_{\xi} f_{ij} + \alpha_{ij} \Big(\delta_{\xi} d_{ij} + f_{ij} \Big) \Big) + h_{2}^{2} \Big(\delta_{\eta}^{2} d_{ij} + \beta_{ij} \delta_{\eta} d_{ij} \Big) \Big],$$
(30)

$$E_{ij}^{*} = e_{ij} + \frac{1}{12} \Big[h_{2}^{2} \Big(\delta_{\eta}^{2} e_{ij} + 2\delta_{\eta} f_{ij} + \beta_{ij} \Big(\delta_{\eta} e_{ij} + f_{ij} \Big) \Big) + h_{1}^{2} \Big(\delta_{\xi}^{2} e_{ij} + \alpha_{ij} \delta_{\xi} e_{ij} \Big) \Big], \tag{31}$$

$$F_{ij}^{*} = f_{ij} + \frac{1}{12} \Big[h_{1}^{2} \Big(\delta_{\xi}^{2} f_{ij} + \alpha_{ij} \delta_{\xi} f_{ij} \Big) + h_{2}^{2} \Big(\delta_{\eta}^{2} f_{ij} + \beta_{ij} \delta_{\eta} f_{ij} \Big) \Big],$$
(32)

$$P_{ij}^{*} = \frac{1}{12} \left(h_{2}^{2} a_{ij} + h_{1}^{2} c_{ij} \right), \tag{33}$$

$$Q_{ij}^{*} = \frac{1}{12} \Big[h_{1}^{2} e_{ij} + h_{2}^{2} \Big(2\delta_{\eta} a_{ij} + \beta_{ij} a_{ij} \Big) \Big],$$
(34)

$$R_{ij}^{*} = \frac{1}{12} \Big[h_{2}^{2} d_{ij} + h_{1}^{2} \Big(2\delta_{\xi} c_{ij} + \alpha_{ij} c_{ij} \Big) \Big],$$
(35)

$$G_{ij}^{*} = g_{ij} + \frac{1}{12} \Big[h_{1}^{2} \Big(\delta_{\xi}^{2} g_{ij} + \alpha_{ij} \delta_{\xi} g_{ij} \Big) + h_{2}^{2} \Big(\delta_{\eta}^{2} g_{ij} + \beta_{ij} \delta_{\eta} g_{ij} \Big) \Big],$$
(36)

$$\alpha_{ij} = \frac{d_{ij} - 2\delta_{\xi}a_{ij}}{a_{ij}}, \quad \beta_{ij} = \frac{e_{ij} - 2\delta_{\eta}c_{ij}}{c_{ij}}.$$
(37)

So Equations (26) with (27)-(37) is the HOC difference scheme based on the coordinate transformation for solving Equation (1) on nonuniform grids. The present HOC scheme can be written in the form of nine-point scheme and the corresponding coefficients of them can be written as follows:

$$l_{0}\hat{u}_{ij} + l_{1}\hat{u}_{i+1,j} + l_{2}\hat{u}_{i,j+1} + l_{3}\hat{u}_{i-1,j} + l_{4}\hat{u}_{i,j-1} + l_{5}\hat{u}_{i+1,j+1} + l_{6}\hat{u}_{i-1,j+1} + l_{7}\hat{u}_{i-1,j-1} + l_{8}\hat{u}_{i+1,j-1} = G_{ij}^{*},$$
(38)

where

$$l_0 = -\frac{2}{h_1^2} A_{ij}^* - \frac{2}{h_2^2} C_{ij}^* + F_{ij}^* + \frac{4}{h_1^2 h_2^2} P_{ij}^*,$$
(39)

$$l_{1} = \frac{1}{h_{1}^{2}} A_{ij}^{*} + \frac{1}{2h_{1}} D_{ij}^{*} - \frac{2}{h_{1}^{2} h_{2}^{2}} P_{ij}^{*} - \frac{1}{h_{1} h_{2}^{2}} R_{ij}^{*}, \qquad (40)$$

$$I_{2} = \frac{1}{h_{2}^{2}}C_{ij}^{*} + \frac{1}{2h_{2}}E_{ij}^{*} - \frac{2}{h_{1}^{2}h_{2}^{2}}P_{ij}^{*} - \frac{1}{h_{1}^{2}h_{2}}Q_{ij}^{*},$$
(41)

$$l_{3} = \frac{1}{h_{1}^{2}} A_{ij}^{*} - \frac{1}{2h_{1}} D_{ij}^{*} - \frac{2}{h_{1}^{2} h_{2}^{2}} P_{ij}^{*} + \frac{1}{h_{1} h_{2}^{2}} R_{ij}^{*}, \qquad (42)$$

$$l_4 = \frac{1}{h_2^2} C_{ij}^* - \frac{1}{2h_2} E_{ij}^* - \frac{2}{h_1^2 h_2^2} P_{ij}^* + \frac{1}{h_1^2 h_2} Q_{ij}^*,$$
(43)

$$l_{5} = \frac{1}{4h_{1}h_{2}}B_{ij}^{*} + \frac{1}{h_{1}^{2}h_{2}^{2}}P_{ij}^{*} + \frac{1}{2h_{1}^{2}h_{2}}Q_{ij}^{*} + \frac{1}{2h_{1}h_{2}^{2}}R_{ij}^{*},$$
(44)

$$l_{6} = -\frac{1}{4h_{1}h_{2}}B_{ij}^{*} + \frac{1}{h_{1}^{2}h_{2}^{2}}P_{ij}^{*} + \frac{1}{2h_{1}^{2}h_{2}}Q_{ij}^{*} - \frac{1}{2h_{1}h_{2}^{2}}R_{ij}^{*},$$
(45)

$$l_7 = \frac{1}{4h_1h_2} B_{ij}^* + \frac{1}{h_1^2 h_2^2} P_{ij}^* - \frac{1}{2h_1^2 h_2} Q_{ij}^* - \frac{1}{2h_1 h_2^2} R_{ij}^*,$$
(46)

$$l_8 = -\frac{1}{4h_1h_2}B_{ij}^* + \frac{1}{h_1^2h_2^2}P_{ij}^* - \frac{1}{2h_1^2h_2}Q_{ij}^* + \frac{1}{2h_1h_2^2}R_{ij}^*.$$
(47)

From the process of derivation, it is easy to know that the truncation error of this scheme is $O(h_1^4 + h_1^3 h_2 + h_1^2 h_2^2 + h_1 h_2^3 + h_2^4)$; *i.e.*, the present HOC scheme has fourth order accuracy.

4. Multigrid Method

In order to solve the linear algebraic systems which are arising from various difference schemes, generally, some iterative methods are used. But the convergence speed of traditional iterative methods is very slow, so appears multigrid method [16]-[18]. It has been proved that multigrid method is an optimal numerical method at least for solving the linear elliptic problems. Its main characteristics is to use the traditional iterative methods to solve the residual equations on different coarse grid levels by gradually transferring the errors to coarser grids until the error is convergent, then to return the corrected results to the finer grid levels by using the interpolation.

Multigrid method is achieved by some circulation algorithms such as V cycle, W cycle or Full Multigrid V cycle (FMV) etc. The whole process has three elements: relaxation operator, projection operator and interpolation operator. The function of relaxation operator (or iteration) is to dump the high frequency components of the errors on the current grid. The function of projection and interpolation operators is to transfer error residuals from finer grids to coarser grids and to return the corrected errors from the coarser grids to the finer grids. The multigrid $V(v_1, v_2)$ cycle means v_1 relaxations are performed at each level before projecting the residual to the coarse grid space (pre-smoothing) and v_2 relaxations after interpolating the solution back to the fine grid space (post-smoothing).

Multigrid method has been used to solve various linear elliptic equations such as Poisson equation [19]-[21], convection-diffusion equations [22]-[25] and so on. Gupta *et al.* [19] [22] [24] and Zhang [20] [23] used it to solve the 2D and 3D Poisson equations and the convection diffusion equations discretized by the fourth order compact scheme on uniform grids. Ge and Cao [25] and Ge *et al.* [21] developed multigrid method on nonuniform grids to solve 2D convection diffusion equation and 3D Poisson equation with boundary layer problems based on the transformation-free HOC difference schemes. In terms of the method of coordinate transformation, Ge and Zhang [13] [14] used it to map the nonuniform grid to a uniform grid, and then employed a fourth order compact difference scheme to the transformed uniform grid and a multigrid method to solve the 2D and 3D convection diffusion equation with boundary layers.

In this paper, we adopt the multigrid V cycle method to solve the linear algebraic system arising from the difference schemes. In order to match the HOC scheme, we choose the full weighting projection operator on uniform grids [18]

$$\overline{r_{i,j}} = \frac{1}{16} \Big[4r_{i,j} + 2\Big(r_{i-1,j} + r_{i,j-1} + r_{i+1,j} + r_{i,j+1}\Big) + r_{i-1,j-1} + r_{i+1,j-1} + r_{i+1,j+1} + r_{i-1,j+1} \Big]$$

where $r_{i,j}$ is residual at the fine grid points (i, j) and $\overline{r_{\overline{i},\overline{j}}}$ is the corresponding residuals at coarse grid points $(\overline{i}, \overline{j})$, so it has $i = 2\overline{i}$ and $j = 2\overline{j}$. For the interpolation operator, we use the conventional bilinear interpolation operator on uniform grids [18]

$$\begin{split} r_{i,j} &= r_{\overline{i},\overline{j}}, \\ r_{i-1,j} &= \frac{1}{2} \Big(\overline{r_{\overline{i}}}_{-1,\overline{j}} + \overline{r_{\overline{i}}}_{,\overline{j}} \Big), \\ r_{i,j-1} &= \frac{1}{2} \Big(\overline{r_{\overline{i}}}_{,\overline{j}-1} + \overline{r_{\overline{i}}}_{,\overline{j}} \Big), \\ r_{i-1,j-1} &= \frac{1}{4} \Big(\overline{r_{\overline{i}}}_{-1,\overline{j}} + \overline{r_{\overline{i}}}_{,\overline{j}} + \overline{r_{\overline{i}}}_{-1,\overline{j}-1} + \overline{r_{\overline{i}}}_{,\overline{j}-1} \Big). \end{split}$$

Then, for relaxation operator, we use the alternating direction line Gauss-Seidel relaxations to remove the residuals on each coarse grid.

5. Numerical Experiments

In order to demonstrate the high accuracy and high efficiency of the present method, we use it to solve the following three elliptic problems with Dirichlet boundary conditions. All of the problems have the exact solutions. All computation is started with zero initial guesses and is terminated when the residuals in L_2 -norm on the finest grids are reduced by 10¹⁰. For each problem, we give the multigrid V cycles (Num), the CPU time in seconds, maximum absolute errors (Error) and convergence rates (Order) about different grid numbers in the tables. The procedure is written in Fortran 77 programming language with double precision arithmetic and run on a Pentium IV/Dual-core/3 GHz private computer with 2 GB memory. The convergence order can be got by the following formula:

$$Order = \frac{\log(Error(N_1)/Error(N_2))}{\log(N_2/N_1)}$$

where N_1 and N_2 represent for different grid numbers and $Error(N_1)$ and $Error(N_2)$ are correspondingly the maximum absolute errors.

5.1. Problem 1

We consider the following 2D convection diffusion problem [26]:

$$u_{xx} + u_{yy} + p(x, y)u_x + q(x, y)u_y = f(x, y), \ (x, y) \in [0, 1] \times [0, 1]$$

where

$$p(x, y) = \operatorname{Re} x(x-1)(1-2y), \ q(x, y) = \operatorname{Re} y(y-1)(1-2x).$$

The boundary conditions are:

$$u(x,0) = e^{-\sigma(x-0.5)^2}, u(x,1) = e^{-\sigma(x-0.5)^2-1}, u(0,y) = u(1,y) = e^{-0.25\sigma-y^2}.$$

The exact solution is:

$$u(x, y) = e^{-\sigma(x-0.5)^2 - y^2}$$
.

This problem has a steep solution gradient near x = 0.5. This kind of problems is also referred to as interior layer problems in the Ref. [26]. We choose the following transformation functions:

$$x(\xi) = \xi + \frac{\gamma_x}{2\pi} \sin(2\pi\xi), \quad y(\eta) = \eta$$

where γ_x $(-1 < \gamma_x < 1)$ is grid stretching parameter controlling the density of grid in the *x* direction. For instance, if $-1 < \gamma_x < 0$, the density of grids around x = 0 is more concentrate, and the γ_x is smaller, the more

grids are distributed around x = 0; if $0 < \gamma_x < 1$, the density of grids around x = 1 is more concentrate, and the γ_x is larger, the more grids are distributed around x = 1; if $\gamma_x = 0$, the grids turn to be uniform in the physical domain.

For this problem, we set Re = 1000 and choose $\gamma_x = 0.6$ for $\sigma = 10^3$, $\gamma_x = 0.8$ for $\sigma = 10^4$ and $\gamma_x = 0.85$ for $\sigma = 10^5$ for nonuniform grids. From the data in **Table 1**, we find that the HOC scheme has fourth order accuracy with $\sigma = 10^3$ and $\sigma = 10^4$ on the uniform grids ($\gamma_x = 0$). When $\sigma = 10^5$, the computational results are very poor on the uniform grids, so it can not well approximate the exact solutions. However, it can be observed that the computed results on the nonuniform grids can keep fourth order convergence for all σ computed and give very accurate solutions. So, it shows that the present scheme with nonuniform grids has high accuracy for the problems whose solutions change violently near the area of steep solution gradient. Meanwhile, **Table 1** also gives the numbers of multigrid V(1,1) iterations and the CPU time in seconds with different σ for Problem 1. We can see that multigrid method can rapidly converge in a short time with no more than 12 multigrid V(1,1) iterations for all cases.

In order to illustrate the computational accuracy in the whole domain, with the grid number 64×64 , we give the figures about the contours of the exact solution (**Figure 1(b**)), the computed results on uniform grids (**Figure 1(d**)) and on nonuniform grids (**Figure 1(d**)) for $\sigma = 10^5$ for Problem 1. As can be seen from the figures, the computed results can approximate the exact solutions very well on the nonuniform grids. This is because enough grid points are distributed in the area of large solution gradient on nonuniform grids. On the contrary, it appears very large computational error in the area of large solution gradient on uniform grids because it can not obtain enough grid points in the interior layer with 64×64 grids.

5.2. Problem 2

Next, we consider an elliptic problem as follows:

$$2(xu_{xx} + yu_{yy}) + D(x, y)u_{x} + E(x, y)u_{y} + F(x, y)u = G(x, y),$$

where

$$D(x, y) = \operatorname{Re} x(1-x)(1-2y), \ E(x, y) = \operatorname{Re} y(1-y)(1-2x), \ F(x, y) = -x^2 y^2.$$

The boundary conditions are:

$$u(x,0) = 0, u(x,1) = 1-b(x), u(0, y) = 0, u(1, y) = 1-b(y),$$

Table	1.1	Maximum	absolute	errors ar	nd the	convergence	rates fo	r Problem	1 Re =	1000
Lanc	1.	viannum	absolute	chions an	iu uic	convergence	Tates 10	1 I I U U U U U U	1, 10 -	1000.

C-: 1-		Unit	form grids		Nonuniform grids			
Grids	Num	CPU	Error	Order	Num	CPU	Error	Order
$\sigma = 10^3$					$\gamma_x = 0.6$			
64×64	11	0.042	1.28 (-2)		11	0.078	5.05 (-4)	
128 imes 128	11	0.266	8.44 (-4)	3.97	11	0.250	3.14 (-5)	4.05
256 imes 256	10	1.016	5.32 (-5)	4.01	11	1.110	1.98 (-6)	4.01
$\sigma = 10^4$					$\gamma_x = 0.8$			
64×64	11	0.063	2.72 (+0)		12	0.062	2.18 (-3)	
128 imes 128	10	0.250	3.00 (-2)	6.58	11	0.265	1.48 (-4)	3.92
256×256	10	1.079	2.15 (-3)	3.82	11	1.140	9.43 (-6)	3.99
$\sigma = 10^5$					$\gamma_x = 0.85$			
64×64	11	0.062	7.67 (+1)		10	0.063	2.46 (-2)	
128 imes 128	11	0.281	3.70 (+1)	1.06	10	0.266	1.50 (-3)	4.08
256 imes 256	10	1.125	1.36 (+0)	4.79	10	1.110	9.41 (-5)	4.02



Figure 1. (a) Nonuniform grids ($\gamma_x = 0.85$, 65 × 65); (b) Exact solution; (c) Computed solution on uniform grids; (d) Computed solution on nonuniform grids for $\sigma = 10^5$, Re = 10^3 .

in which

$$b(x) = \left[e^{-\operatorname{Re}(x-1)} - 1\right] / (e^{\operatorname{Re}} - 1), \quad b(y) = \left[e^{-\operatorname{Re}(y-1)} - 1\right] / (e^{\operatorname{Re}} - 1).$$

The exact solution is:

$$u(x, y) = [1-b(x)][1-b(y)].$$

For this problem, there are two boundary layers near x = 0 and y = 0. So, we can choose the following transformation functions:

$$x(\xi) = \xi + \frac{\gamma_x}{\pi} \sin(\pi\xi), \ y(\eta) = \eta + \frac{\gamma_y}{\pi} \sin(\pi\eta).$$

We choose Re = 10, 100 and 1000. From **Table 2** we find that when Re = 10, the computation can approximately achieve fourth order convergence on both uniform and nonuniform grids $(\gamma_x = \gamma_y = -0.8)$. But when Re increases to 100, the convergence rate on uniform grids decreases to the third order while still approximately the fourth order accuracy is kept on nonuniform grids $(\gamma_x = \gamma_y = -0.95)$. When Re = 1000, the convergence rate drops to under the first order on uniform grids while the second to third order is shown on nonuniform grids.

Cable 2. Maximum absolute errors and the convergence rates for Problem 2.										
a · 1		Unifo	orm grids		Nonuniform grids					
Grids	Num	CPU	Error	Order	Num	CPU	Error	Order		
Re = 10					$\gamma_x = \gamma_y = -0.8$					
64 imes 64	8	0.062	6.72 (-5)		8	0.063	9.01 (-6)			
128 imes 128	8	0.281	5.71 (-6)	3.60	8	0.296	6.28 (-7)	3.89		
256×256	8	1.172	4.56 (-7)	3.67	8	1.203	4.26 (-8)	3.90		
Re = 100					$\gamma_x = \gamma_y = -0.95$					
64 imes 64	8	0.062	1.36 (-1)		10	0.062	2.18 (-4)			
128×128	8	0.281	2.10 (-2)	2.73	9	0.313	1.71 (-5)	3.71		
256×256	9	1.182	2.38 (-3)	3.18	9	1.296	1.22 (-6)	3.83		
Re = 1000					$\gamma_x = \gamma_y = -0.999$					
64 imes 64	17	0.140	9.63 (-1)		22	0.188	9.03 (-4)			
128 imes 128	17	0.625	9.36 (-1)	0.04	21	0.672	2.72 (-4)	1.75		
256×256	17	2.656	7.24 (-1)	0.37	21	2.891	3.87 (-5)	2.83		

 $(\gamma_x = \gamma_y = -0.999)$; *i.e.*, the computed accuracy for large Re is not maintained on both uniform and nonuniform grids, this agrees with the findings of Zhang [14]. In such condition, the boundary layers are very steep, solutions in the boundary layers change very violently, which makes the computed results so poor on the uniform grids. On the other hand, although this violent change leads to slow down the convergence rate on nonuniform grids, the computed accuracy on nonuniform grids is very ideal. The computed results show that the solutions on nonuniform grids are more accurate than that on uniform grids. Besides, **Table 2** gives the numbers of multigrid V(1,1) iterations and the CPU time in seconds with different Re for Problem 2. We can see that the multigrid algorithm is very efficient and at most 22 multigrid V(1,1) cycles are needed to get convergence for all cases.

Figure 2, with grid number 64×64 , gives the contours of exact solution (Figure 2(b)), the computed results on uniform grids (Figure 2(c)) and on nonuniform grids (Figure 2(d)) for Re = 1000. We find that the present HOC scheme produces amazingly satisfying solution on nonuniform grids, while it appears a big computed error near the boundary layers on the uniform grids.

5.3. Problem 3

We consider the following elliptic equation:

$$e^{2y}u_{xx} + e^{x+1}u_{yy} + 2x^2yu_x + 2xy^2u_y + x^2y^2u = G(x, y).$$

The boundary conditions are:

$$u(x,0) = \left[2\operatorname{Re}(1-x)-1\right]/(e^{2\operatorname{Re}}-1), \quad u(x,1) = \left[2\operatorname{Re}(1-x)+e^{2\operatorname{Re}}-2\right]/(e^{2\operatorname{Re}}-1),$$

$$u(0,y) = \left(2\operatorname{Re}+e^{2\operatorname{Re}y}-2\right)/(e^{2\operatorname{Re}}-1), \quad u(1,y) = \left(e^{2\operatorname{Re}y}-2\right)/(e^{2\operatorname{Re}}-1).$$

The exact solution is:

$$u(x, y) = \left[2\operatorname{Re}(1-x) + e^{2\operatorname{Re}y} - 2\right] / (e^{2\operatorname{Re}} - 1)$$

There is a boundary layer near y = 1, so here we choose the following transformation functions:

$$x(\xi) = \xi, \ y(\eta) = \eta + \frac{\gamma_y}{\pi} \sin(\pi \eta).$$

For this problem, the multigrid V(2,2) cycles are used. **Table 3** gives the computed results, where $\gamma_y = 0.9$ for Re = 100, $\gamma_y = 0.97$ for Re = 1000 and $\gamma_y = 0.994$ for Re = 10,000 are chosen. From the data we find that for Re = 100, it gets the approximately fourth order accuracy on the uniform grids. And for Re = 1000 and



tion on uniform grids; (d) Computed solution on nonuniform grids for Re = 1000.

Re = 10,000, it just gets the second order accuracy on the uniform grids, the computed errors are dramatically distorted with the increase of Re, and this gives very poor solution. Especially for Re = 10,000, the solution is very bad and unacceptable. Compared with computed results on the uniform grids, it shows that the fourth order accuracy is achieved for all the Re numbers on the nonuniform grids and the computed results are very accurate. So, it demonstrates that the present transformed HOC scheme is effective for solving the boundary layer problems with nonuniform grids in the physical domain. Table 3 also gives the numbers of multigrid V(2,2) iterations and the CPU time in seconds for this problem. We can see that the multigrid algorithm is very effective and the numbers of multigrid V(2,2) cycles are no more than 10 times to get convergence for all cases.

Figure 3 gives the comparison of the exact and the computed solutions with the grid number 64×64 while Re = 10,000 and the transformation parameter is $\gamma_y = 0.994$. We find that the computed results are good to approximate the exact solutions on the nonuniform grids, while it appears very large computed errors near the boundary layer on the uniform grids.

6. Concluding Remarks

The aim of this paper is to build an efficient and high accuracy numerical method for solving 2D elliptic equations with variable coefficients and interior/boundary layers on nonuniform grids. Coordinate transformation

B. Lan *et al*.

Table 3. Maximum absolute errors and the convergence rates for Problem 3.									
~	Uniform grids				Nonuniform grids				
Grids	Num	CPU	Error	Order	Num	CPU	Error	Order	
Re = 100					$\gamma_y = 0.9$				
64×64	6	0.063	2.66 (-1)		8	0.078	5.67 (-5)		
128 imes 128	6	0.289	2.14 (-2)	3.68	7	0.313	3.54 (-6)	4.05	
256 imes 256	6	1.187	1.43 (-3)	3.93	7	1.359	2.20 (-7)	4.03	
Re = 1000					$\gamma_y = 0.97$				
64×64	6	0.094	7.79 (+1)		9	0.110	8.42 (-4)		
128 imes 128	6	0.391	1.91 (+1)	2.05	8	0.438	4.82 (-5)	4.17	
256 imes 256	6	1.703	4.08 (+0)	2.24	6	1.609	2.86 (-6)	4.15	
Re = 10000	e = 10000				$\gamma_y = 0.994$				
64×64	6	0.093	7.88 (+3)		8	0.094	1.17 (-2)		
128×128	6	0.406	2.00 (+3)	2.00	6	0.391	5.40 (-4)	4.49	
256×256	6	1.703	5.04 (+2)	2.00	5	1.515	2.59 (-5)	4.41	



Figure 3. (a) Nonuniform grids ($\gamma_x = 0, \gamma_y = 0.994$, 65 × 65); (b) Exact solution; (c) Computed solution on uniform grids; (d) Computed solution on nonuniform grids for Re = 10000.

method is employed to transfer nonuniform grids in the physical domain, which concentrates clustered grid points inside the interior/boundary layers, to uniform grids in the computational domain. A high order compact difference scheme is derived for the transformed equation to achieve the purpose of simplified calculation on uniform grids. It needs to be pointed out that when the transformation parameter is zero, the present HOC scheme reduces to the HOC difference scheme on uniform grids in the physical domain. So, it fits computation on both uniform and nonuniform grids. In order to accelerate the convergence of the traditional iterative methods and to reduce computational cost, a multigrid method is employed to solve the linear algebraic system which is arising from the difference scheme. Some numerical experiments with interior or boundary layer problems are conducted to demonstrate the performances of the present method. It indicates that a nonuniform grid is necessary for solving 2D elliptic problems with interior or boundary layers. By coordinate transformation, a certain number of grid points are clustered in the interior or boundary layers to guarantee that the HOC scheme for transformed equation obtains very accurate numerical solution with not so fine grids. Otherwise, the HOC scheme produces very poor approximation solution on uniform grids.

Acknowledgements

The present work was supported by the National Science Foundation of China under Grant 11361045 and 11161036, Fok Ying-Tong Education Foundation of China under Grant 121105.

References

- Lynch, R.E. and Rice, J.R. (1978) High Accuracy Finite Difference Approximation to Solutions of Elliptic Partial Differential Equations. *Proceedings of the National Academy of Sciences of the United States of America*, **75**, 2541-2544. <u>http://dx.doi.org/10.1073/pnas.75.6.2541</u>
- [2] Boisvert, R.F. (1981) Families of High Order Accurate Discretizations of Some Elliptic Problems. *SIAM Journal on Scientific and Statistical Computing*, **2**, 268-284. <u>http://dx.doi.org/10.1137/0902022</u>
- [3] Gupta, M.M., Manohar, R.P. and Stephenson, J.W. (1984) A Single Cell High Order Scheme for the Convection-Diffusion Equation with Variable Coefficients. *International Journal for Numerical Methods in Fluids*, 4, 641-651. http://dx.doi.org/10.1002/fld.1650040704
- [4] Gupta, M.M., Manohar, R.P. and Stephenson, J.W. (1985) High-Order Difference Scheme for Two Dimensional Elliptic Equations. *Numerical Methods for Partial Differential Equations*, 1, 71-80. http://dx.doi.org/10.1002/num.1690010108
- [5] Ananthakrishnaiah, U., Manohar, R. and Stephenson, J.W. (1987) High-Order Methods for Elliptic Equations with Variable Coefficients. *Numerical Methods for Partial Differential Equations*, 3, 219-227. http://dx.doi.org/10.1002/num.1690030306
- [6] Ananthakrishnaiah, U., Manohar, R. and Stephenson, J.W. (1987) Fourth-Order Finite Difference Methods for Three-Dimensional General Linear Elliptic Problems with Variable Coefficients. *Numerical Methods for Partial Differential Equations*, 3, 229-240. <u>http://dx.doi.org/10.1002/num.1690030307</u>
- [7] Ge, L.X. and Zhang, J. (2002) Symbolic Computation of High Order Compact Difference Schemes for Three Dimensional Linear Elliptic Partial Differential Equations with Variable Coefficients. *Journal of Computational and Applied Mathematics*, 143, 9-27. <u>http://dx.doi.org/10.1016/S0377-0427(01)00504-0</u>
- [8] Choo, J.Y. (1994) Stable High Order Methods for Elliptic Equations with Large First Order Terms. Computers and Mathematics with Applications, 27, 65-80. <u>http://dx.doi.org/10.1016/0898-1221(94)90006-X</u>
- Karaa, S. (2007) High-Order Difference Scheme for 2D Elliptic and Parabolic Problems with Mixed Derivatives. Numerical Methods for Partial Differential Equations, 23, 366-378. <u>http://dx.doi.org/10.1002/num.20181</u>
- [10] Briane, M. and Diaz, J.C. (2008) Uniform Convergence of Sequences of Solutions of Two-Dimensional Linear Elliptic Equations with Unbounded Coefficients. *Journal of Differential Equations*, 245, 2038-2054. http://dx.doi.org/10.1016/j.jde.2008.07.027
- [11] Choo, J.Y. and Schultz, D.H. (1994) A High Order Difference Method of Steady State Navier-Stokes Equation. Computers & Mathematics with Applications, 27, 105-119. <u>http://dx.doi.org/10.1016/0898-1221(94)90101-5</u>
- [12] Spotz, W.F. (1995) High-Order Finite Difference Schemes for Computational Mechanics. Ph.D. Thesis, University of Texas at Austin.
- [13] Ge, L.X. and Zhang, J. (2001) High Accuracy Iterative Solution of Convection Diffusion Equation with Boundary Layers on Nonuniform Grids. *Journal of Computational Physics*, **171**, 560-578. <u>http://dx.doi.org/10.1006/jcph.2001.6794</u>

- [14] Zhang, J., Ge, L.X. and Gupta, M.M. (2001) Fourth Order Compact Difference Scheme for 3D Convection Diffusion Equation with Boundary Layers on Nonuniform Grids. *Neural, Parallel and Scientific Computations*, 8, 373-392.
- [15] Liu, C. and Liu, Z. (1995) Multigrid Mapping and Box Relaxation for Simulation of the Whole Process of Flow Transition in 3D Boundary Layers. *Journal of Computational Physics*, **119**, 325-341. http://dx.doi.org/10.1006/jcph.1995.1138
- [16] Brandt, A. (1977) Multi-Level Adaptive Solution to Boundary Value Problems. *Mathematics of Computation*, **31**, 333-390. http://dx.doi.org/10.1090/S0025-5718-1977-0431719-X
- [17] Hackbusch, W. and Trottenberg, U. (1982) Multigrid Methods. Springer-Verlag, Berlin. <u>http://dx.doi.org/10.1007/BFb0069927</u>
- [18] Wesseling, P. (1992) An Introduction to Multigrid Methods. Wiley, Chichester.
- [19] Gupta, M.M., Kouatchou, J. and Zhang, J. (1997) Comparison of Second- and Fourth-Order Discretizations for Multigrid Poisson Solvers. *Journal of Computational Physics*, 132, 226-232. <u>http://dx.doi.org/10.1006/jcph.1996.5466</u>
- [20] Zhang, J. (1998) Fast and High Accuracy Multigrid Solution of the Three Dimensional Poisson Equation. Journal of Computational Physics, 143, 449-461. <u>http://dx.doi.org/10.1006/jcph.1998.5982</u>
- [21] Ge, Y., Cao, F. and Zhang, J. (2013) A Transformation-Free HOC Scheme and Multigrid Method for Solving the 3D Poisson Equation on Nonuniform Grids. *Journal of Computational Physics*, 234, 199-216. <u>http://dx.doi.org/10.1016/j.jcp.2012.09.034</u>
- [22] Gupta, M.M., Kouatchou, J. and Zhang, J. (1997) A Compact Multigrid Solver for Convection-Diffusion Equations. *Journal of Computational Physics*, **132**, 123-129. <u>http://dx.doi.org/10.1006/jcph.1996.5627</u>
- [23] Zhang, J. (1997) Accelerated Multigrid High Accuracy Solution of the Convection-Diffusion Equation with High Reynolds Number. *Numerical Methods for Partial Differential Equations*, 13, 77-92. http://dx.doi.org/10.1002/(SICI)1098-2426(199701)13:1<77::AID-NUM6>3.0.CO;2-J
- [24] Gupta, M.M. and Zhang, J. (2000) High Accuracy Multigrid Solution of the 3D Convection-Diffusion Equation. Applied Mathematics and Computation, 113, 249-274. http://dx.doi.org/10.1016/S0096-3003(99)00085-5
- [25] Ge, Y. and Cao, F. (2011) Multigrid Method Based on the Transformation-Free HOC Scheme on Nonuniform Grids for 2D Convection Diffusion Problems. *Journal of Computational Physics*, 230, 4051-4070. <u>http://dx.doi.org/10.1016/j.jcp.2011.02.027</u>
- [26] Farrell, P.A., Hegarty, A.F., Miller, J.J.H., O'Rordan, E. and Shishkin, G.I. (2000) Robust Computational Techniques for Boundary Layers. Chapman & Hall/CRC, Boca Raton.

Appendix

$$\begin{split} &\delta_{\xi} \hat{u}_{ij} = \frac{\hat{u}_{i,j+1} - \hat{u}_{i,j-1}}{2h_{1}} \\ &\delta_{\eta} \hat{u}_{ij} = \frac{\hat{u}_{i,j+1} - \hat{u}_{i,j-1}}{2h_{2}} \\ &\delta_{\xi}^{2} \hat{u}_{ij} = \frac{\hat{u}_{i+1,j} - 2\hat{u}_{i,j} + \hat{u}_{i-1,j}}{h_{1}^{2}} \\ &\delta_{\eta}^{2} \hat{u}_{ij} = \frac{\hat{u}_{i,j+1} - 2\hat{u}_{i,j} + \hat{u}_{i,j-1}}{h_{2}^{2}} \\ &\delta_{\xi} \delta_{\eta} \hat{u}_{ij} = \frac{\hat{u}_{i+1,j+1} - \hat{u}_{i-1,j+1} + \hat{u}_{i-1,j-1} - \hat{u}_{i+1,j-1}}{4h_{1}h_{2}} \\ &\delta_{\xi} \delta_{\eta} \hat{u}_{ij} = \frac{-2\hat{u}_{i,j+1} + 2\hat{u}_{i,j-1} + \hat{u}_{i+1,j+1} + \hat{u}_{i-1,j+1} - \hat{u}_{i-1,j-1} - \hat{u}_{i+1,j-1}}{2h_{1}^{2}h_{2}} \\ &\delta_{\xi} \delta_{\eta}^{2} \hat{u}_{ij} = \frac{-2\hat{u}_{i+1,j} + 2\hat{u}_{i-1,j} + \hat{u}_{i+1,j+1} - \hat{u}_{i-1,j+1} - \hat{u}_{i-1,j-1} + \hat{u}_{i+1,j-1}}{2h_{1}h_{2}^{2}} \\ &\delta_{\xi}^{2} \delta_{\eta}^{2} \hat{u}_{ij} = \frac{4\hat{u}_{i,j} - 2\left(\hat{u}_{i+1,j} + \hat{u}_{i,j+1} + \hat{u}_{i-1,j} + \hat{u}_{i,j-1}\right) + \hat{u}_{i+1,j+1} + \hat{u}_{i-1,j-1} + \hat{u}_{i+1,j-1}}{h_{1}^{2}h_{2}^{2}} \end{split}$$