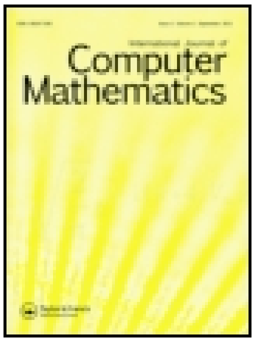


Taylor & Francis
Taylor & Francis Group

International Journal of Computer Mathematics

ISSN: 0020-7160 (Print) 1029-0265 (Online) Journal homepage: http://www.tandfonline.com/loi/gcom20

Partial Semi-Coarsening Multigrid Method Based on the HOC Scheme on Nonuniform Grids for the Convection-diffusion Problems

Fujun Cao, Yongbin Ge & Hai-Wei Sun

Publisher: Taylor & Francis & Informa UK Limited, trading as Taylor & Francis Group
Journal: *International Journal of Computer Mathematics*
DOI: 10.1080/00207160.2017.1283408

Partial Semi-Coarsening Multigrid Method Based on the HOC Scheme on Nonuniform Grids for the Convection-diffusion Problems

Fujun Cao[1,*] Yongbin Ge[2] Hai-Wei Sun[3]

1 School of Science, Inner Mongolia University of Science and Technology, Baotou 014010, China

1. *2 Institute of Applied Mathematics and Mechanics, Ningxia University, Yinchuan 750021, China*

2. *3 Department of Mathematics, University of Macau, Macao, China*

Abstract

A partial semi-coarsening multigrid method based on the high order compact (HOC) difference scheme on nonuniform grids is developed to solve the two dimensional (2D) convection-diffusion problems with boundary or internal layers. The significance of this study is that the multigrid method allows different number of grid points along different coordinate directions on nonuniform grids. Numerical experiments on some convection-diffusion problems with boundary or internal layers are conducted. They demonstrate that the partial semi-coarsening multigrid method combined with the HOC scheme on nonuniform grids, without losing the high order accuracy, is very efficient and effective to decrease the computational cost by reducing the number of grid points along the direction which does not contain boundary or internal layers.

Keywords: Convection-diffusion equation; High order compact difference scheme; Nonuniform grids; Partial semi-coarsening; Multigrid method; Boundary or internal layer

[*]Correspondence author.

Email addresses: caofujun@imust.cn (Fujun Cao), gyb@nxu.edu.cn(Yongbin Ge), haiweisun@sss.cn(Hai-wei Sun)

1. Introduction

We consider the following 2D steady convection-diffusion equation in the form of

$$-(\Phi_{xx}+\Phi_{yy})+c(x,y)\Phi_x+d(x,y)\Phi_y=f(x,y), \quad (1)$$

where $\Phi(x,y)$ is an unknown function, $c(x,y)$ and $d(x,y)$ are coefficients of convection terms in the x and the y directions, $f(x,y)$ is the right-hand side term. These functions are assumed to be sufficiently smooth and have continuous partial derivatives up to necessary orders in a rectangular computational domain Ω. $\partial\Omega$ is its boundary.

In the past three decades, many researchers have been paying attention to the high order compact (HOC) schemes for solving convection-diffusion equations and a large number of HOC difference schemes have been developed [1-16]. The schemes on uniform grid display high accuracy for smooth solutions while they fail to achieve theoretical accuracy order [13] for solving the problems with boundary or internal layers in the computational domain unless a great number of grids in the whole domain are employed. It inevitably leads to a huge waste of computational work in the physical domain in which solutions change smoothly. A reasonable strategy is to use flexible grid distribution patterns to set more grid points inside of the areas of steep solution gradients and comparably less grid points in the smooth solution regions. So developing efficient HOC difference schemes on nonuniform grids has a very important application value and actual significance.

Usually, there exist at least two methods to construct HOC schemes on nonuniform grids. One is using a grid transformation function to map nonuniform grids onto uniform grids [10, 11, 14, 16], i.e., this method transforms nonuniform grids in physical domain onto uniform grids in the computational domain. Then the equation is solved on the uniform grids. Finally, the computed results are returned to the nonuniform grids by the inverse transformation. The other method is to directly discretize the physical domain with nonuniform grids, which does not need to conduct any coordinate transformation. Kalita et al [13] used it to develop an HOC scheme to solve 2D steady convection-diffusion problems on nonuniform grids involving no coordinate transformation from the physical space to the computational space. The numerical results show that the HOC scheme has higher computed accuracy and better scale resolution for the problems with boundary layers. The majority of HOC schemes use same number of grid points in both coordinate directions on a square domain. Actually, for the problems with the boundary or internal layers just existing in one direction, using the same number of grid points in the both directions is unnecessary. We will demonstrate this viewpoint in the following part.

For solving the linear algebraic system which is arising from HOC difference schemes, the convergence speeds of the traditional iterative methods are very slow, so multigrid method is proposed to overcome this shortcoming. Since 1977, when Brandt [17] published his pioneering work on it, multigrid method has been widely used to solve various partial differential equations which are discretized by finite difference method or finite element method [18,19]. In the past two decades, multigrid method combined with fourth order compact difference schemes has been employed and developed by several authors to solve elliptic equations [9,10,11,14,15,20-28]. We notice that the most multigrid methods were performed on uniform grids because the earlier HOC schemes are constructed on the uniform grids [15,20-27]. Some computed results are obtained [10,11,14] on physically nonunifrom grids using the coordinate transformation HOC scheme. In this strategy, the multigrid method is still performed on the uniform grids. Recently, Ge and Cao [28] developed a multigrid method combined with the HOC scheme on nonuniform grids for solving the 2D convection-diffusion problems with boundary layers. The HOC scheme used in [28] is proposed by Kalita et al [13]. All numerical experiments involved in [28] were conducted using the same number of grid points in both coordinate directions in a unit square domain. Actually, if boundary or internal layers just exist in only one direction, we can use considerably fewer grid points in the other direction to correspondingly decrease the computational cost and save the CPU time and storage space. And it is important to point out that we can use the HOC scheme developed by Kalita et al [13] with the different number of grid points in the different coordinate directions if we do not use multigrid method. However, if multigrid method is considered, it is not straightforward to do so because some special multigrid strategies should be implemented to fully exploit its superiority [26,27].

In this paper, we are aiming at developing a partial semi-coarsening multigrid method combined with the HOC scheme on nonuniform grids to solve the convection-diffusion problems with boundary or internal layers. The present work allows using different number of grid points along different coordinate directions which can overcome the defect in our previous work in [28] and be more efficient and cost-effective. The reminder of this paper is organized as follows. Section 2 gives out the HOC scheme which is proposed by Kalita et al [13] for solving the 2D convection-diffusion equation on nonuniform grids. Then, a partial semi-coarsening multigrid method based on the HOC scheme is introduced and the new restriction and interpolation operators are developed in Section 3. After that, numerical experiments are conducted to show the high accuracy of the HOC scheme and the high efficiency of the partial semi-coarsening multigrid method in Section 4. Finally, concluding remarks are given in Section 5.

2. HOC difference scheme on nonuniform grids

We consider a rectangular domain $\Omega=[a_1,a_2]\times[b_1,b_2]$ and divide the intervals $[a_1,a_2]$ and $[b_1,b_2]$ into sub-intervals by the points $a_1=x_0<x_1<x_2\cdots<x_{m-1}<x_m=a_2$ and $b_1=y_0<y_1<y_2\cdots<y_{n-1}<y_n=b_2$, where m and n are numbers of sub-intervals in the x and the y directions, respectively. In the x direction, the forward and backward step lengths at point x_i are given by $x_f=x_{i+1}-x_i$ and $x_b=x_i-x_{i-1}$, respectively. Similarly, in

the y direction, $y_f = y_{j+1} - y_j$ and $y_b = y_j - y_{j-1}$, for all $1 \le j \le n-1$, as see in Fig.1. The HOC difference scheme on nonuniform grids for the 2D convection-diffusion equation (1) derived by Kalita et al. [13] is written as

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

where the coefficients A_{ij} , B_{ij} , C_{ij} , D_{ij} , G_{ij} , H_{ij} , K_{ij} , L_{ij} , F_{ij} and the finite difference operators $\delta_x, \delta_y, \delta_x^2, \delta_y^2, \delta_x\delta_y, \delta_x^2\delta_y, \delta_x\delta_y^2$ and $\delta_x^2\delta_y^2$ are defined in [13]. The order of the truncation error is four on uniform grids when $x_f = x_b$ and $y_f = y_b$ and at least three on nonuniform grids when $x_f \ne x_b$ or $y_f \ne y_b$ (More details, see Ref. [13]).

In matrix form, the system of algebraic equations given by (2) can be written as

$$\mathrm{A}\Phi=\mathrm{F} \quad (3)$$

where the coefficient matrix A is an asymmetric sparse matrix with each row containing at most nine non-zero entries. On uniform grids, Zhang [29] studied the convergence and performance of some iterative methods with HOC scheme for the 1D and 2D convection-diffusion equations. The Fourier analysis was conducted to show that for 2D problems the point line and alternating line Gauss-Seidel methods converge for small to large Reynolds numbers. For the HOC scheme of two dimensional variable coefficient convection-diffusion equations, the coefficient matrix A is showed to be an M-matrix given certain condition [30]. This result guarantees the convergence of many classical stationary iterative methods. As for the solution to the algebraic system arising from the difference equation (2), the numerical results in [28] has demonstrated that the multigrid method, the Gauss–Seidel method, the SOR method and the hybrid biconjugate gradient stabilized method (BiCGStab(2)) are convergent and the multigrid method is the most efficient and cost-effective method among these four iterative methods. So, the multigrid method is applied to solve the algebraic system in this paper.

3. Partial semi-coarsening multigrid method

Semi-coarsening multigrid method was developed by Mulder [31] to deal with convection problems and was used to simulate the whole process of flow transition in 3D flat plate boundary layers [32]. In 2002, Zhang [26] proposed a partial semi-coarsening multigrid method to solve the 2D Poisson equation. The method is constructed based on unequal-meshsize grids which means the space step of two directions are different, $h_x \ne h_y$ while in each direction the grid intervals are same (still a kind of uniform grid). For the anisotropic problems whose solution changes more rapidly in one direction (named dominant direction) than in the other direction (named non -dominant direction), more grid points are distributed in the dominant direction while less grid points are distributed in the non-dominant direction. Numerical experiments indicate that the unequal mesh size HOC difference scheme can achieve almost equivalent accuracy by decreasing the grid points in the non-dominant direction, thus the computational cost of the partial semi-coarsening multigrid method is signicantly decreased. Ge [27] generalized this method to the 3D case. In this paper, we will generalize the partial semi-coarsening multigrid method for the 2D unequal-meshsize-grid discretization developed by Zhang [26] for solving the Poisson equation to the 2D nonuniform grids to solve the convection-diffusion problems with boundary and internal layers.

According to the theory of the multigrid method, besides multigrid cycles (V type, W type, or FMV type, etc.), we must consider three components of multigrid method: relaxation operator, restriction operator and interpolation operator. Relaxation operator (the smoother) dumps the high frequency components of the errors on the current grid while leaving the low frequency components to be removed by the coarser grids. In this paper, we use the line Gauss-Seidel relaxations as smoother when the boundary or internal layers just exist in one direction. Once the boundary or internal layers exist in both x and y directions, the alternating line Gauss-Seidel relaxations (see Ref. [11]) is applied as smoother. As far as the restriction and interpolation operators are concerned, those developed by Ge and Cao [28] for nonuniform grids with the same number of grid points in both x and y directions should be reformed for new use. We will discuss them in Section 3.2.

3.1 Partial semi-coarsening nonuniform grids

For the local large gradient problems considered in this paper, we assume that the boundary or internal layers always exist in x or y direction, but not both. We also assume the physical domain is a unit square and the number of grid points in x and y directions are $N_x = 2^{n_x}, N_y = 2^{n_y}$ for some positive integers $n_x > 1$ and $n_y > 1$. If the boundary or internal layers exist in the x direction, more grid points are distributed and we get $n_x \geq n_y$. Otherwise $n_y \geq n_x$.

As described in [26], partial semi-coarsening means that grid coarsening is only performed along the dominant direction while along the non-dominant direction the grid is not coarsened. The processes of partial semi-coarsening strategy is that every other grid line along the dominant direction (boundary or internal layers' direction here) is eliminated from a fine grid to a coarse grid until there will be a coarse grid level on which $N_x = N_y$. Starting from this grid level, the standard full-coarsening strategy is implemented. The full-coarsening strategy means coarse the grids along two directions simultaneously until reach the coarsest grid level. As in standard multigrid method, the coarsest grid will have only one unknown for Dirichlet boundary problems. Suppose an internal layer exists in the x direction and at $x = 0.5$ and we use 64×16 grids in the whole computational domain, then grid coarsening process is presented in Fig.2, in which the steps ① and ② are semi-coarsening and steps ③,④ and ⑤ are full-coarsening. The total process is called the partial semi-coarsening method.

Remark 1: One of the innovation of the presented method compared to the method in [28] is that the presented method allows the number of grids at two directions to be different. This ensures that we can distribute more grids in the boundary layer or the internal layer direction, but fewer grids in the non-boundary layer or the internal direction. For the grid coarsening process of multigrid method, the partial semi-coarsening strategy is composed of the semi-coarsening process (the steps ① and ② in Fig.2) and the full-coarsening process (the steps ③,④ and ⑤ in

Fig.2). When the number of grids at x and y directions are the same on the finest grid, the partial semi-coarsening multigrid method is reduced to the full-coarsening multigrid method.

3.2 Restriction and interpolation operators

To simplify our discussion without loss of generality, we assume that the dominant direction where the boundary or internal layers exist is the x direction and the y direction is not the dominant direction. So we need to distribute more grid points in the x direction than in the y direction. Because of possible different number of grid points in the x and the y directions, we need to classify situation to construct restriction and interpolation operators. First, we define $r_{i,j}$ to be the residual at the fine grid point (i,j), $\bar{r}_{\bar{i},\bar{j}}$ is the corresponding residual at coarse grid point $(\bar{i},\bar{j})$. It is easy to see that $i=2\bar{i}$ and $j=2\bar{j}$. For convenience, we mark the lengths for restriction operators on nonuniform grids as Fig.3, where L_{bx} and L_{fx} are the backward and forward step lengths of the fine grids in the x direction. Define $L_x = L_{bx} + L_{fx}$, $L_1 = L_{fx}/2$, $L_0 = (L_{bx}+L_{fx})/2 = L_x/2$ and $L_3 = L_{bx}/2$.

(1) If $N_x > N_y$, the grid coarsening is only performed along the x direction. So, the one-direction weighting average strategy is used to construct the residual restriction operator. The residual at the coarse grid point (small black point) is computed by averaging the residual at the corresponding fine grid point (small black point) and its two neighboring fine grid points (big black points, which will not appear in the coarse grid level) in the x direction. The reference point (i,j) on the fine grid has the most contribution to the coarse grid point $(\bar{i},\bar{j})$, so the corresponding weighting coefficient is evaluated by L_0/L_x. At the same time, we notice that among two neighbors of reference point (i,j), the point $(i-1,j)$ is more near the reference point than $(i+1,j)$. It is easy to conclude that the near point has more contribution than the farther one. Thus, the inverse distance weighting coefficient is adopted to construct the residual restriction operator which can be expressed as following:

$$\bar{r}_{\bar{i},\bar{j}} = \frac{1}{L_x}(L_1 r_{i-1,j} + L_0 r_{i,j} + L_3 r_{i+1,j}).$$

(2) If $N_x = N_y$, the grid coarsening is performed along both x and y directions. A two-direction weighting average operator is designed. The residual at the coarse grid point is computed by averaging the residual at the corresponding fine grid point and its eight neighboring grid points in both x and y directions. Under this condition, the full weighting restriction operator on nonuniform grids by the area law [33] developed in [28] can be used as follows:

$$\bar{r}_{\bar{i},\bar{j}} = \frac{1}{S}(S_0 r_{i,j} + S_1 r_{i-1,j} + S_2 r_{i,j-1} + S_3 r_{i+1,j} + S_4 r_{i,j+1}$$
$$+ S_5 r_{i-1,j-1} + S_6 r_{i+1,j-1} + S_7 r_{i+1,j+1} + S_8 r_{i-1,j+1}).$$

where S_0 is the area bounded by four half meshsize lines (dashed lines) around the reference grid point (i,j) and $S_i\,(i=1,\cdots,8)$ is the area bounded by the dashed lines and the boundary lines around them, respectively, and S is the total area of S_i ($i = 0,\cdots,8$), $S = \sum_{i=0}^{8} S_i$. See Fig. 4. (More details see [28]).

Then, we use the same strategy to construct the interpolation operator.

(1) If $N_x > N_y$, corrections for the approximate solution at the fine grid points (big black points) corresponding to the coarse grid points are transferred directly. Correction for other fine grid point (small black point) takes the average of the neighboring two coarse grid points (big black points) in the x direction only. Denote L_x is the step length of the present coarse grid stencil in the x direction. L_{bx} and L_{fx} are the backward and forward step lengths of the fine grid in the x direction. See Fig. 5. The interpolation operator can be explicitly written as:

$$r_{i,j} = \overline{r}_{\overline{i},j} ,$$

$$r_{i-1,j} = \frac{1}{L_x}(L_{fx}\overline{r}_{\overline{i}-1,j} + L_{bx}\overline{r}_{\overline{i},j}) .$$

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

$$r_{i,j} = \overline{r}_{\overline{i},\overline{j}} ,$$

$$r_{i-1,j} = \frac{1}{L_x}(L_{fx}\overline{r}_{\overline{i}-1,\overline{j}} + L_{bx}\overline{r}_{\overline{i},\overline{j}}) ,$$

$$r_{i,j-1} = \frac{1}{L_y}(L_{fy}\overline{r}_{\overline{i},\overline{j}-1} + L_{by}\overline{r}_{\overline{i},\overline{j}}) ,$$

$$r_{i-1,j-1} = \frac{1}{S}(S_1\overline{r}_{\overline{i}-1,\overline{j}-1} + S_2\overline{r}_{\overline{i},\overline{j}-1} + S_3\overline{r}_{\overline{i},\overline{j}} + S_4\overline{r}_{\overline{i}-1,\overline{j}}) .$$

where L_y , L_{by} and L_{fy} have the similar definitions in the y direction. S_i ($i = 1,\cdots,4$) is the area bounded by all solid lines, respectively. S is the total area of S_i ($i = 1,\cdots,4$), $S = \sum_{i=1}^{4} S_i$. See Fig. 6. (More details, see [28]).

Remark 2: The another innovation of the presented method is that the restriction and interpolation operators in Ref. [28] are reformed for new use in this paper. Owing to the using of the partial semi-coarsening multigrid method, the situation of $N_x \neq N_y$ must emerge in the process of the multigrid V cycles . In this case the semi-coarsening strategy is necessary and the inverse distance weighting coefficient is used to construct the residual restriction and interpolation operators for one dimension. When $N_x = N_y$, the full-coarsening multigrid method is implemented and the restriction and interpolation operators proposed in [28] are still adopted.

4. Numerical experiments

In order to demonstrate the high accuracy and high efficiency of the present method, we give the following three convection-diffusion problems with boundary layers or internal layers. The right-hand function $f(x,y)$ and the Dirichlet boundary conditions on $\partial\Omega$ are prescribed to satisfy the given exact solution. All computation is started with zero initial guesses and terminated when the residuals in L_2 norm on the finest grids are reduced by 10^{10}. For each problem, we give the multigrid V cycles (Num), the CPU time in seconds and maximum absolute errors (Error) using different fine number of grid points in the tables. The source code is written in Fortran 77 programming language with double precision arithmetic and run on a Pentium IV/Dual-core/3 GHz private computer with 2GB memory.

4.1 Problem 1

We consider the following 2D convection-diffusion equation [15,28]

$$-(\Phi_{xx}+\Phi_{yy})+c(x,y)\Phi_x+d(x,y)\Phi_y=f(x,y),$$

where

$$c(x,y)=\mathrm{Re}\,x(x-1)(1-2y),\quad d(x,y)=\mathrm{Re}\,y(y-1)(1-2x).$$

The exact solution is:

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

The problem has a steep internal layer along $x=0.5$ when σ is large, so we consider the x direction is the dominant direction and a nonuniform grid is used along it. In the y direction, we just use a uniform grid. So we choose the following grid distribution function:

$$x_i=\frac{i}{m}+\frac{\lambda_x}{2\pi}\sin\left(\frac{2\pi i}{m}\right),\qquad y_j=\frac{j}{n}.$$

where λ_x $(-1<\lambda_x<1)$ is stretching parameter controlling the density of grids in the x direction. For instance, when $-1<\lambda_x<0$, the density of grids around $x=0$ and $x=1$ is more concentrate. And as the λ_x is smaller, the more grids are distributed around $x=0$ and $x=1$; when $0<\lambda_x<1$, the density of grids around $x=0.5$ is more concentrate. And as the λ_x is larger, the more grids are distributed around $x=0.5$. When $\lambda_x=0$, the grid reduce to be uniform in the physical domain. When $\lambda_x=0.85$, the grid with 64×16 is shown in Fig.7 (a).

Table 1 Maximum absolute errors, multigrid cycles and CPU time for Problem 1, $\sigma=10^3$

Grids	Uniform grids ($\lambda_x=0$)			Nonuniform grids ($\lambda_x=0.6$)		
	Num	CPU	Error	Num	CPU	Error
256×256	7	1.047	4.975(-5)	7	1.063	1.584(-6)
256×128	7	0.609	4.949(-5)	7	0.610	1.582(-6)
256×64	5	0.234	4.883(-5)	5	0.234	3.740(-6)
256×32	3	0.078	5.121(-5)	3	0.078	1.473(-5)
128×128	7	0.250	8.062(-4)	7	0.250	2.555(-5)
128×64	6	0.125	7.843(-4)	6	0.110	2.505(-5)
128×32	4	0.031	8.256(-4)	3	0.031	5.966(-5)

128×16	6	0.078	7.944(-4)	5	0.015	2.334(-4)
64×64	8	0.063	1.230(-2)	8	0.062	4.136(-4)
64×32	4	0.015	1.274(-2)	4	0.031	4.304(-4)
64×16	3	0.000	1.217(-2)	4	0.016	9.465(-4)
64×8	7	0.000	1.158(-2)	6	0.015	3.338(-3)
32×32	9	0.016	1.587(-1)	8	0.015	7.015(-3)
32×16	4	0.000	1.544(-1)	6	0.000	6.556(-3)
32×8	5	0.000	1.375(-1)	6	0.000	1.395(-3)

For this problem, we choose Re=1000 and adopt multigrid V(5,5) cycles, in which, (ν_1, ν_2) means ν_1 relaxation is performed at each grid level before projection the residual to the coarse grid space (pre-smoothing) and ν_2 relaxation after interpolating the solution back to the fine grid space (post-smoothing). Table 1 compare of the number of multigrid V cycles, the CPU time in seconds, maximum absolute errors on the uniform and on the nonuniform grids at $\sigma = 10^3$. From the table, we find that the computed solution on the nonuniform grids is more accurate than that on the uniform grids with the same number of grid points. Moreover, because the local large gradient solution only happens in the x direction, using fine grids in the y direction is not necessary; i.e., increasing the grid points in the y direction cannot increase computed accuracy any more when the grid number in this direction is over a certain threshold. For example, on uniform grid, when the grid changes from 256×32 to 256×256, the computed error does not decrease distinctly. And on nonuniform grids with $\lambda_x = 0.6$, the computed error with 256×256 grids is almost same with 256×128 grids. That means, if local gradient solution just happens in one direction, we do not need to distribute same amount of grids in the other direction; i.e., fewer grid points can be used in the non-dominant direction and the accuracy still can be guaranteed. In such a case, the computational amount would be greatly decreased to make the computation to be more cost-effective. Additionally, the multigrid method is very efficient by observing the numbers of multigrid V cycles and the CPU time.

Table 2 Maximum absolute errors, multigrid cycles and CPU time for Problem 1, $\sigma = 10^4$

Grids	Uniform grids ($\lambda_x = 0$)			Nonuniform grids ($\lambda_x = 0.8$)		
	Num	CPU	Error	Num	CPU	Error
256×256	7	1.016	1.551(-3)	8	1.187	3.433(-6)
256×128	7	0.594	1.547(-3)	7	0.610	3.412(-6)
256×64	5	0.234	1.606(-3)	5	0.219	3.544(-6)
256×32	3	0.078	1.655(-3)	3	0.078	1.459(-5)
128×128	7	0.250	2.201(-2)	7	0.250	5.540(-5)
128×64	6	0.125	2.245(-2)	6	0.125	5.734(-5)
128×32	3	0.031	2.328(-2)	3	0.046	6.037(-5)
128×16	5	0.015	2.334(-2)	3	0.016	2.314(-4)
64×64	8	0.063	5.619(+0)	8	0.078	9.628(-4)
64×32	5	0.031	5.626(+0)	4	0.031	1.008(-3)
64×16	4	0.015	5.634(+0)	4	0.015	9.983(-4)
64×8	9	0.016	5.235(+0)	5	0.016	3.334(-3)

32×32	9	0.016	6.336(+1)	10	0.016	1.992(-2)
32×16	6	0.015	6.337(+1)	6	0.015	1.760(-2)
32×8	7	0.000	5.948(+1)	6	0.000	1.833(-2)

Table 2 gives the similar computed results at $\sigma = 10^4$. We notice that when σ is bigger, the internal layer is more obvious and the superiority of using the nonuniform grids and the partial semi-coarsening multigrid method is more obvious.

Table 3 Comparison of Maximum absolute errors, multigrid cycles and CPU time at different Strategy for problem 1

	Grids	Full-Coarsening Strategy[28]			Grids	Semi-Coarsening Strategy		
		Num	CPU	Error		Num	CPU	Error
$\sigma = 10^3$	256×256	7	1.172	1.583(-6)	256×64	5	0.234	3.740(-6)
$\lambda_x = 0.6$	128×128	7	0.297	2.554(-5)	128×32	3	0.031	5.966(-5)
	64×64	8	0.094	4.135(-4)	64×16	4	0.015	9.983(-4)
$\sigma = 10^4$	256×256	8	1.187	3.433(-6)	256×64	5	0.219	3.544(-6)
$\lambda_x = 0.8$	128×128	7	0.250	5.540(-5)	128×32	3	0.046	6.037(-5)
	64×64	8	0.078	9.628(-4)	64×16	4	0.015	9.983(-4)

Table 3 compares the results from the presented method with the full-coarsening strategy in iteration number, CPU time and maximum errors. We can see that the semi-coarsening strategy obtain the numerical results in similar accuracy as the full-coarsening strategy by using one-quarter of grid number. Therefore, this strategy saves the computation cost and storage space than the method in [28].

Fig.8 presents the efficiency of the semi-coarsening and the full-coarsening strategy on the non-uniform grid. It demonstrates that the error from the semi coarsening strategy is decreasing faster than the full-coarsening strategy.

4.2 Problem 2

We consider the following 2D convection-diffusion equation [28]

$$-\varepsilon(\Phi_{xx} + \Phi_{yy}) + \frac{1}{1+y}\Phi_y = f(x,y),$$

The exact solution is:

$$\Phi(x,y) = e^{y-x} + 2^{-\frac{1}{\varepsilon}}(1+y)^{1+\frac{1}{\varepsilon}}.$$

The problem has a steep solution gradient along $y=1$ when ε is very small, so we consider the y direction is the dominant direction and a nonuniform grid is used along it. We choose the following grid distribution function to set more grid points at the boundary layer in the y direction and a uniform grid in the x direction

$$x_i = \frac{i}{m}, \qquad y_j = \frac{j}{n} + \frac{\lambda_y}{\pi}\sin(\frac{\pi j}{n}) \cdot$$

Table 4 Maximum absolute errors, multigrid cycles and CPU time for Problem 2, $\varepsilon = 0.01$

Grids	Uniform grids ($\lambda_y = 0$)			Nonuniform grids ($\lambda_y = 0.55$)		
	Num	CPU	Error	Num	CPU	Error
256×256	8	1.016	1.579(-6)	10	1.250	1.583(-8)
128×256	8	0.610	1.579(-6)	10	0.734	1.581(-8)
64×256	8	0.297	1.578(-6)	9	0.328	1.576(-8)
32×256	5	0.094	1.575(-6)	5	0.109	2.058(-8)
128×128	8	0.250	2.513(-5)	11	0.313	2.524(-7)
64×128	8	0.125	2.513(-5)	9	0.141	2.521(-7)
32×128	6	0.047	2.512(-5)	6	0.062	2.561(-7)
16×128	3	0.031	2.507(-5)	4	0.031	3.304(-7)
64×64	8	0.047	4.063(-4)	10	0.062	3.700(-6)
32×64	7	0.016	4.063(-4)	8	0.031	3.790(-6)
16×64	4	0.000	4.062(-4)	4	0.016	3.917(-6)
8×64	3	0.000	4.268(-4)	3	0.015	5.393(-6)
32×32	8	0.016	6.642(-3)	9	0.016	6.215(-5)
16×32	5	0.000	6.642(-3)	5	0.000	6.233(-5)
8×32	3	0.000	6.857(-3)	3	0.000	6.439(-5)

Table 4 presents the computed results for $\varepsilon = 0.01$. The multigrid V(2,2) cycles are used for this problem. By comparing the maximum absolute errors we find that when the grid number in the y direction is fixed, the computed accuracy does not lose with the decrease of the grid number in the x direction. That means, we can decrease the computational cost and save the CPU time and the storage space by decreasing the number of grid points in the x direction. And the computed results on the nonuniform grids are much more accurate than that on the uniform grids. The partial semi-coarsening multigrid method with fewer number of grid points in the non-dominant direction can get almost equivalent accuracy and expend less V(2,2) cycles and less CPU time than the full-coarsening multigrid method with the same number of grid points in both directions. The partial semi-coarsening multigrid method is very efficient and a great deal of computational cost and storage space is saved.

Table 5 presents the comparison of the presented method with the full-coarsening strategy on iteration number, CPU time and maximum errors. It can be seen that the semi-coarsening strategy obtains numerical results with nearly same accuracy as the full-coarsening strategy in much less grid numbers than the latter method. As a result, the presented method consumes a relatively small computational cost and CPU time.

Table 5 Comparison of Maximum absolute errors, multigrid cycles and CPU time at different Strategy for problem 2

	Grids	Full-Coarsening Strategy[28]			Grids	Semi-Coarsening Strategy		
		Num	CPU	Error		Num	CPU	Error
$\varepsilon = 10^{-2}$	256×256	10	1.625	1.582(-8)	64×256	9	0.328	1.576(-8)
	128×128	11	0.407	2.523(-7)	32×128	6	0.062	2.561(-7)

$\lambda_y = 0.55$	64×64	10	0.093	3.699(-6)	16×64	4	0.016	3.917(-6)
$\varepsilon = 10^{-3}$	256×256	14	2.688	5.583(-5)	64×256	5	0.375	5.960(-5)
$\lambda_y = 0.75$	128×128	11	0.531	5.273(-4)	32×128	7	0.234	3.788(-4)
	64×64	7	0.093	1.319(-2)	16×64	3	0.015	1.369(-2)

Fig. 9 gives the plots of the exact solution (Fig. 9(b)), the computed results on the uniform grids (Fig. 9(c)) and on the nonuniform grids (Fig. 9(d)) for $\varepsilon = 0.001$ with the grid number 8×32 (Fig. 9(a)) for Problem 2. From the figure, we observe that the computed solution can approximate the exact solution very well on the nonuniform grids while it appears very large computed errors on the uniform grids in the boundary layer. Fig.10 compared the efficiency of the semi-coarsening and the full-coarsening strategy on the non-uniform grid. From the figure, we can see that the maximum error of the semi coarsening strategy is decreasing faster than the full-coarsening strategy.

4.3 Problem 3

We now consider a convection-diffusion problem as follows [11,13]

$$-(\Phi_{xx} + \Phi_{yy}) + c(x,y)\Phi_x + d(x,y)\Phi_y = f(x,y),$$

where

$$c(x,y) = \mathrm{Re}\, x(x-1)(1-2y),\ d(x,y) = -\mathrm{Re}\, y(y-1)(1-2x).$$

The exact solution is：

$$\Phi(x,y) = [1-b(x)][1-b(y)],$$

in which

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

Unlike the previous two problems which the boundary layer just exists in one direction, this problem exists two boundary layers which are near $x = 0$ and $y = 0$, respectively. And this problem has a stagnation point at (0.5, 0.5) inside the computational domain where both convection coefficients vanish; i.e., $c(0.5,0.5) = d(0.5,0.5) = 0$, so this type of problem is very hard to solve when Re is large. We choose this problem to show that the present method can also deal with this kind of problems. Under such circumstance, the partial semi-coarsening multigrid method reduces to the full-coarsening multigrid method since we use the same number of grid points in the both x and the y directions. Here we choose the following grid distribution functions :

$$x_i = \frac{i}{m} + \frac{\lambda_x}{\pi}\sin(\frac{\pi i}{m}),\quad y_j = \frac{j}{n} + \frac{\lambda_y}{\pi}\sin(\frac{\pi j}{n}).$$

For $\lambda_x = \lambda_y = -0.95$, the grid with 64×64 is shown in Fig. 11 (a).

Table 4. Maximum absolute errors, convergence rate, multigrid cycles and CPU time for Problem 3

	Uniform grids				Nonuniform grids			
	Num	CPU	Error	Rate	Num	CPU	Error	Rate

$\mathrm{Re}=10$					$\lambda_x=\lambda_y=-0.30$			
64×64	7	0.171	8.39(-7)		7	0.188	1.08(-7)	
128×128	7	0.766	5.25(-8)	4.00	7	0.781	6.79(-9)	3.99
256×256	7	3.343	3.28(-9)	4.00	7	3.469	4.25(-10)	4.00
$\mathrm{Re}=100$					$\lambda_x=\lambda_y=-0.75$			
64×64	7	0.171	1.34(-2)		8	0.188	2.02(-5)	
128×128	7	0.766	9.36(-4)	3.83	8	0.797	1.34(-6)	3.91
256×256	7	3.360	6.03(-5)	3.96	8	3.468	8.43(-8)	3.99
$\mathrm{Re}=1000$					$\lambda_x=\lambda_y=-0.95$			
64×64	8	0.297	9.93(-1)		26	1.125	5.74(-4)	
128×128	8	1.219	8.45(-1)	0.23	27	4.641	3.44(-5)	4.06
256×256	8	4.859	2.72(-1)	1.64	28	19.53	2.12(-6)	4.02

Since for this problem $x=0$ and $y=0$ are both boundary layers, the alternating line Gauss-Seidel relaxation is used as the smoother. We use V(2,2) cycles for Re=10 and Re=100 and V(5,5) cycles for Re= 1000. Table 4 gives the multigrid V cycles, the CPU time, the maximum absolute errors and the convergence rate for this problem. The convergence rate is defined as

$$Rate=\frac{\log(Error1/Error2)}{\log(N_2/N_1)}$$

where $Error1$ and $Error2$ are the maximum absolute errors estimated for two different grids with N_1 and N_2 sub-intervals in each grid.

We can see that the computed accuracy on uniform grids is degraded dramatically with the increase of boundary layers steepness. Especially for $\mathrm{Re}=1000$, a poor solution is derived and fourth order accuracy is lost on the uniform grids while very accurate solution is obtained and fourth order convergence is kept for all cases on the nonuniform grids. Figure 11(b), (c) and (d) give the solution plots of the exact (b), computed on the uniform grids(c) and on the nonuniform girds(d) for $\mathrm{Re}=1000$ under the mesh 64×64. We note that the solution on the uniform grids is inacceptable. Whereas, a very accurate solution is obtained on the nonuniform grids. Furthermore, multigrid method on the nonuniform grids still shows very efficient for this problem although for $\mathrm{Re}=1000$, more multigrid V cycles are needed on the nonuniform grids than on the uniform grids.

5 Concluding remarks

A partial semi-coarsening multigrid method based on the HOC scheme is developed to solve the 2D convection-diffusion problems with boundary or internal layers. The highlight of this method is that it allows using different number of grid points in the two coordinate directions, so it is suitable for solving the problems with boundary or internal layers only in one direction. It overcomes the defect in our previous work [*Y. Ge and F. Cao, Multigrid method based on the* *transformation-free HOC scheme on nonuniform grids for 2D convection-diffusion problems, J. Comput. Phys. 230 (2011) 4051-4070*], in which, same number of grid points must be used along different coordinate directions. For the problems with boundary or internal layers in both directions, the partial semi-coarsening multigrid method reduces to the full-coarsening multigrid method. The numerical experiments indicate that the partial semi-coarsening multigrid algorithm is an efficient and effective tool, resulting in the saving of the computational time without losing the high order accuracy.

Recently, the full-coarsening multigrid method based on the nonuniform grid HOC difference scheme have been performed to solve the 3D Poisson equation [34] and the 3D convection diffusion equation [35]. We still use same number of grids along all directions in a cubic domain for the 3D problems. As we pointed out above, it is unnecessary to do so if the large gradient solutions exist only in one direction or in two directions (not in three directions). In other words, we can generalize the present partial semi-coarsening multigrid method for the 2D problems to the 3D cases. We will report this work in the near future.

Acknowledgments

The present work was supported by the National Science Foundation of China under Grant 11061025 and 11361045, Fok Ying-Tong Education Foundation of China under Grant 121105 and the Key Project of China Ministry of Education under grant number 210239.

Reference

[1] M. M. Gupta, R. P. Manohar, J. W. Stephenson, A single cell high order scheme for the convection diffusion equation with variable coefficients, Int. J. Numer. Meth. Fluids, 4 (1984) 641-651.

[2] Q. C. Chen, Z. Gao, Z. Yang, A perturbational h^4 exponential finite difference scheme for the convection diffusion equation, J. Comput. Phys., 104 (1993) 129-139.

[3] G. F. Carey, W. F. Spotz, Higher-order compact mixed methods, Commun. Numer. Meth. Eng., 13 (1997) 553-564.

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

[5] A. C. Radhakrishna Pillai, Fourth-order exponential finite difference methods for boundary value problems of convective diffusion type, Int. J. Numer. Meth. Fluids,37 (2001) 87 -106.

[6] S. Karaa, High-order approximation of 2D convection diffusion equation on hexagonal grids, Numer. Methods Partial Differenal. Eq., 22 (2006) 1238-1246.

[7] Z. F. Tian, S. Q. Dai, High-order compact exponential finite difference methods for convection diffusion type problems, J. Comput. Phys., 220 (2007) 952-974.

[8] Z. F. Tian, Y. B. Ge, A fourth-order compact ADI method for solving two-dimensional unsteady convection diffusion problems, J. Comput. Appl. Math., 198 (2007) 268- 286.

[9] Y. Wang, S.Yu, R. Dai and J. Zhang. A 15 point high order compact scheme with multigrid

computation for solving 3D convection diffusion equations. Int. J. Comput. Math., 92(2) (2015) 411-423.

[10] L. Ge, J. Zhang, Accuracy, robustness, and efficiency comparison in iterative computation of convection diffusion equation with boundary layers, Numer. Meth. Part. D. E., 16(2000) 379-394.

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

[12] H. F. Ding, Y. X. Zhang, A new difference scheme with high accuracy and absolute stability for solving convection diffusion equations, J. Comput. Appl. Math., 230 (2009) 600-606.

[13] 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. Meth. Fluids, 44 (2004) 33-53.

[14] R. Dai, Y. Wang. Effects of different high order compact computations for solving boundary layer problems on non-uniform grids. J. Comput. Intell. Elect. Sys., 3(3) (2014) 200-211.

[15] J. Zhang, H. W. Sun, J. J. Zhao, High order compact scheme with multigrid local mesh refinement procedure for convection diffusion problems, Comput. Meth. Appl. Mech. Eng., 191 (2002) 4661-4674.

[16] W. F. Spotz, G. F. Carey, Formulation and experiments with high-order compact schemes for nonuniform grids, Int. J. Numer. Meth. Heat & Fluid Flow, 8 (1998) 288-297.

[17] A. Brandt, Multi-level adaptive solution to boundary value problems, Math. Comput., 31(1977) 333-390.

[18] W. Hackbusch, U.Trottenberg, Multigrid Methods, Springer-verlag, Berlin, (1982).

[19] P. Wesseling, An Introduction to Multigrid Methods, Wiley, Chichester, (1992).

[20] M. M. Gupta, J. Kouatchou, J. Zhang, Comparison of second- and fourth-order discretizations for multigrid Poisson solvers, J. Comput. Phys., 132 (1997) 226-232.

[21] M. M. Gupta, J. Kouatchou, J. Zhang, A compact multigrid solver for convection diffusion equations, J. Comput. Phys., 132 (1997) 123-129.

[22] J. Zhang, Accelerated multigrid high accuracy solution of the convection diffusion equation with high Reynolds mumber, Numer. Methods Partial. Differential Eq., 13 (1997) 77-92.

[23] J. Zhang, Fast and high accuracy multigrid solution of the three dimensional Poisson equation, J. Comput. Phys., 143 (1998) 449-161.

[24] Y. Wang, J. Zhang, Fast and robust sixth-order multigrid computation for the three-dimensional convection diffusion equation, J. Comput. Appl. Math., 234(2010) 3496-3506.

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

[26] J. Zhang, Multigrid method and fourth-order compact scheme for 2D Poisson equation with unequal mesh-size discretization, J. Comput. Phys., 179 (2002) 170-179.

[27] Y. Ge, Multigrid method and fourth-order compact difference discretization scheme with

unequal meshsizes for 3D Poisson equation, J. Comput. Phys., 229 (2010) 6381-6391.

[28] Y. Ge, F. Cao, Multigrid method based on the transformation-free HOC scheme on nonuniform grids for 2D convection diffusion problems, J. Comput. Phys., 230 (2011) 4051-4070.

[29] Zhang J. On convergence and performance of iterative methods with fourth-order compact schemes. Numer. Meth. Part. D. E., 14(2)(1998) 263-280.

[30] Karaa S, Zhang J. Convergence and performance of iterative methods for solving variable coefficient convection-diffusion equation with a fourth-order compact difference scheme. Comput. & Math. Appl., 44(3)(2002)457-479.

[31] W. A. Mulder, A new multigrid approach to convection problems, J. Comput. Phys., 83 (1989) 303-317.

[32] C. Liu, Z. Liu, Multigrid mapping and box relaxation for simulation of the whole process of flow transition in 3D boundary layers, J. Comput. Phys., 119 (1995) 325-341

[33] C. Liu, Multilevel adaptive methods in computational fluid dynamics, Ph.D thesis, University of Colorado at Denver, (1989).

[34] Y. Ge, F. Cao, J. Zhang, A transformation-free HOC scheme and multigrid method for solving the 3D Poisson equation on nonuniform grids, J. Comput. Phys., 234 (2013) 199-216.

[35] Y. Ge, F. Cao. A high order compact difference scheme and multigrid method for solving the 3D convection diffusion equation on non-uniform grids. 2012 International Conference on Computational and Information Sciences, Aug 17-19, (2012)714-717, Chongqing, China

Fig.1 The stencil of HOC scheme on nonuniform mesh

Fig.2 Process of the grid coarsening

Fig. 3 Restriction only in the x direction

Fig. 4 Restriction in both the x and y directions on nonuniform grids

Fig. 5 Interpolation only in the x direction

Fig. 6 Areas for interpolation operator on nonuniform grids

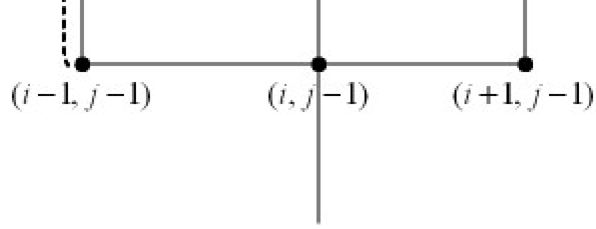
Fig.7 (a) Nonuniform grids ($\lambda_x = 0.85$,64×16); (b) Exact solution; (c) Computed solution on uniform grids; (d) Computed solution on nonuniform grids for $\sigma = 10^5$, Re=1000

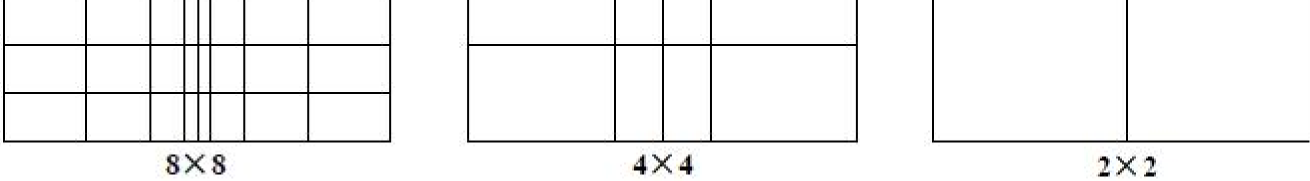
Fig.8 Comparison of the efficiency of the semi-coarsening and the full-coarsening strategy on the non uniform grid(254×64, σ=10^4, λ=0.9, Problem 1)

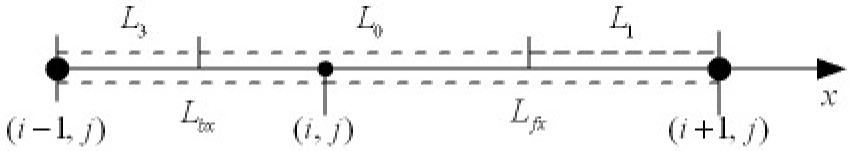
Fig.9 (a) Nonuniform grids ($\lambda_y = 0.9$, 8×32); (b) Exact solution; (c) Computed solution on the uniform grids; (d) Computed solution on the nonuniform grids for $\varepsilon = 0.001$

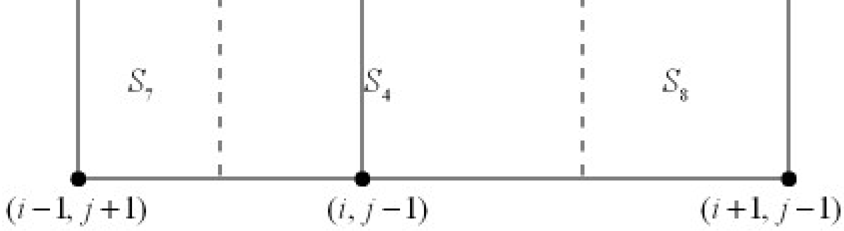
Fig.10 Comparison of the efficiency of the semi-coarsening and the full-coarsening strategy on the non uniform grid (64×254, ε=10^{-3}, λ=0.8, Problem 2)

Fig.11 (a) Nonuniform grids($\lambda_x = \lambda_y = -0.95$,64×64); (b) Exact solution; (c) Computed solution on the uniform grids; (d) Computed solution on the nonuniform grids for Re=1000

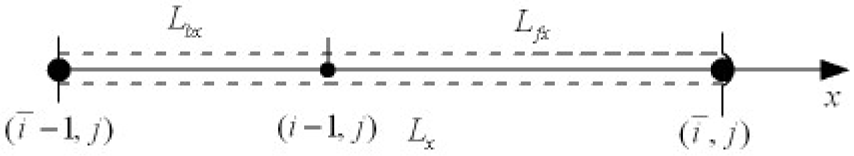

$(i-1, j-1)$
$(i, j-1)$
$(i+1, j-1)$

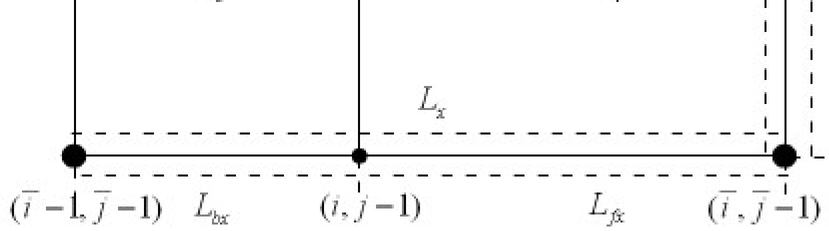

8×8
4×4
2×2


L_3
L_0
L_1
x
$(i-1, j)$
L_{bx}
(i, j)
L_{fx}
$(i+1, j)$

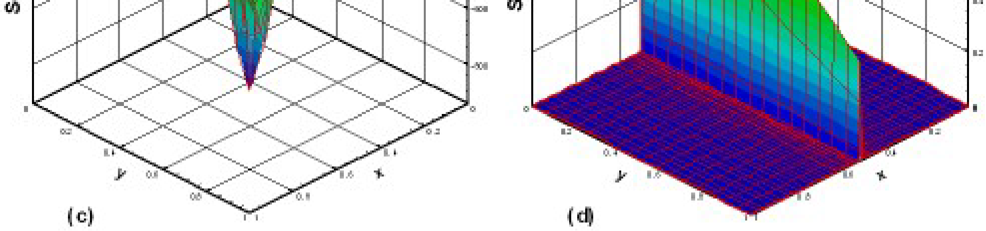


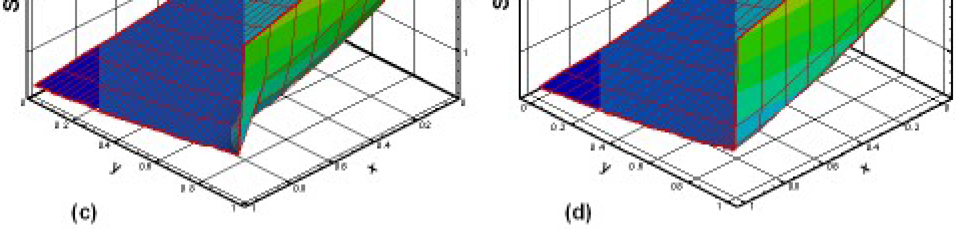
S_7
S_4
S_8
$(i-1, j+1)$
$(i, j-1)$
$(i+1, j-1)$


L_{bx}
L_{fx}
x
$(\bar{i}-1, j)$
$(i-1, j)$
L_x
$(\bar{i}, j)$

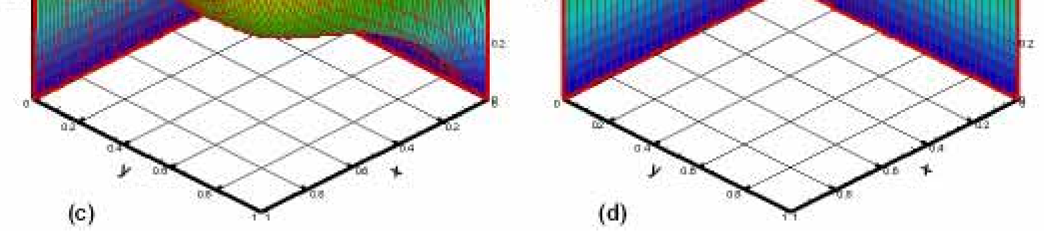


L_x
$(\bar{i}-1, \bar{j}-1)$
L_{bx}
$(i, j-1)$
L_{fx}
$(\bar{i}, \bar{j}-1)$


S
y
x
(c)
S
y
x
(d)



(c) (d)


0
0.2
0.4
y
0.6
0.8
(c)
0.2
0.4
0.6
0.8
x
(d)