An efficient multiscale-like multigrid computation for 2D convection-diffusion equations on nonuniform grids *

Ming Li†

Department of Mathematics, Honghe University, Mengzi, Yunnan 661100, P.R.China

Zhoushun Zheng

School of Mathematics and Statistics, Central South University, Changsha 410083, China

ABSTRACT: An efficient multiscale-like multigrid (MSLMG) method(is presented to solve the two-dimensional (2D) convection-diffusion equations on nonuniform grids, based on the transformation-free high order compact (HOC) difference scheme. By providing appropriate initial solutions, the discretization systems on the two finest grids are solved to obtain the MSLMG solutions with discretization-level accuracy by performing few multigrid cycles, which implemented with alternating line Gauss-Seidel smoother, interpolation and restriction on nonuniform grids. Numerical experiments of two boundary layer and local singularity problems are conducted to demonstrate the proposed algorithm is efficient and effective to decrease the computational cost.

MSC(2010): 65N55, 65N06.

Key Words: Multigrid method; convection-diffusion equation; nonuniform grid; high order compact difference scheme.

---

*This work is supported in part by the National Natural Science Foundation of China (51974377) and the National Key Research and Development Program of China (2017YFB0701700, 2017YFB0305601) and the Natural Science Foundation of Yunnan Province of China (2017FH001-012).

†Corresponding author: mathlm@csu.edu.cn

1 Introduction

High order compact (HOC) scheme is a commonly discretization method for solving the partial differential equations. This approach on uniform grids demonstrate high accuracy for smooth solutions while it fail to achieve theoretical accuracy order for computing some specialized practical cases, such as boundary layers or local singularities problems, if there are no enough mesh nodes in the local domain of steep solution gradients. Transformation-free HOC method[9, 11] is an efficient scheme for various steep gradient solution cases, by placing more grid points inside the local domain with boundary layers or local singularities, and placing comparably less grid points on the smooth regions. This approach not only keeping the good features of ordinary HOC schemes, but also has better scale resolution with smaller number of grid points, with resultant saving of memory.

Multigrid method (MG) [1] is an efficient iterative method for a wide class of partial differential equations, such as Poisson equations[2, 5, 9, 12, 14, 17, 18], convection-diffusion equations[8, 10, 15, 19] and Helmholtz equations[6, 7]. Various strategies for combining interpolator and restriction operators on uniform grids have been developed, such as multiscale multirid (MSMG)[3, 4, 13, 17] and EXCMG accelerated multiscale multigrid (EXCMG-MSMG)[5] methods.

Recently, based on the transformation-free HOC schemes, Ge and Cao discussed the V-cycle multigrid (VMG) methods on nonuniform grids, to solve the 2D convection-diffusion[8] and the 3D Poisson problems[9]. Numerical experiments shown that the VMG algorithms have high efficiency and robust for the discrete system on nonuniform grids. However, choosing an initial value to start the process of VMG algorithms[8, 9] has a certain randomness, which lead to the total computational cost maybe not optimal.

In this paper, through providing a suitable initial solution for the VMG[8], we aim to develop a multiscale-like multigrid (MSLMG) method, to solve the 2D steady-state convection-diffusion equation on nonuniform grids, based on the transformation-free HOC difference scheme [11]. In this approach, appropriate initial vectors are provided for the multigrid method on the two finest grids, respectively, which greatly reduces the number of V-cycles. Numerical experiments are conducted to verify that our MSLMG algorithm can achieve comparable accuracy and keep less cost simultaneously.

The rest of the paper is organized as follows. Section 2 introduces the model problem and a transformation-free HOC scheme. In Section 3 we introduce the restriction and interpolation operators on nonuniform grids. In Section 4 we proposed a multiscale-like multigrid (MSLMG) computation. Supporting numerical results are reported in Section 5. Concluding remarks are provided in Section 6.

2 Model problem and computational strategies

Numerical solution of convection-diffusion equations plays an important role to describe many processes in fields of fluid dynamics. In this article, we are interested in efficiently solving solution with discretization-level accuracy of 2D convection-diffusion equation of the form

$$-\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right)+c(x,y)\frac{\partial u}{\partial x}+d(x,y)\frac{\partial u}{\partial y}=f(x,y), \tag{2.1}$$

with suitable Dirichlet boundary conditions. Here the given convection-velocity $c(x,y)$, $d(x,y)$, the forcing term $f(x,y)$ and the solution $u(x,y)$ are assumed to be continuously differentiable and have the necessary continuous partial derivatives up to certain orders on the domain Ω, The $f(x,y)$ and $u(x,y)$ may have singularity in some region.

For simplicity of presentation, we assume the domain Ω is a rectangular $[D_a, D_b]\times[D_c, D_d]$, and divide the intervals $[D_a, D_b]$ and $[D_c, D_d]$ into sub-intervals by the nodes

$$D_a = x_0 < x_1 < \cdots < x_{N-1} < x_N = D_b,$$
$$D_c = y_0 < y_1 < \cdots < y_{N-1} < y_N = D_d.$$

Where N represents the number of grid intervals in each coordinate direction.

The forward and backward step lengths in the y-direction ($1 \le j \le N-1$) are defined as (see

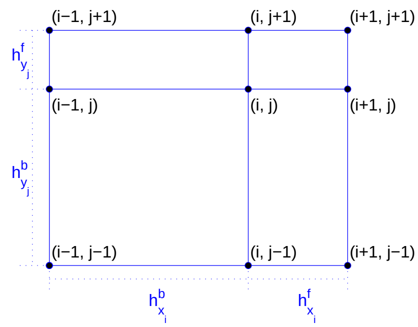


Figure 1: Nonuniform high order compact stencil

Figure 1)

$$h_{y_j}^f = y_{j+1} - y_j, \quad h_{y_j}^b = y_j - y_{j-1}.$$

Likewise for the x-direction ($1 \le i \le N-1$), we have the step lengths

$$h_{x_i}^f = x_{i+1} - x_i, \quad h_{x_i}^b = x_i - x_{i-1}.$$

Let $U_{i,j}$ denote the approximate of u at mesh point (x_i, y_j). Set $G_{i,j} = G(U_{i,j}, x_i, y_j)$ and $\gamma = h_x/h_y$. Here, we introduce the nine-point transformation-free HOC scheme [11] on nonuniform grids for the model problem (2.1) as below

$$\left[-A_{ij}\delta_x^2 - B_{ij}\delta_y^2 + C_{ij}\delta_x + D_{ij}\delta_y + C_{ij}\delta_x\delta_y - H_{ij}\delta_x\delta_y^2 - K_{ij}\delta_x^2\delta_y - L_{ij}\delta_x^2\delta_y^2\right]\Phi_{ij} = F_{ij}. \tag{2.2}$$

Here the coefficients A_{ij}, B_{ij}, C_{ij}, D_{ij}, G_{ij}, H_{ij}, K_{ij}, L_{ij}, F_{ij} and the difference operators δ_x, δ_y, δ_x^2, δ_y^2, $\delta_x\delta_y$, $\delta_x^2\delta_y$, $\delta_x\delta_y^2$, $\delta_x^2\delta_y^2$ are defined in Ref. [11].

The transformation-free HOC scheme has fourth-order accuracy on uniform grids ($h_{x_i}^f = h_{x_i}^b$ and $h_{y_j}^f = h_{y_j}^b$) and at least third-order on nonuniform grids ($h_{x_i}^f \neq h_{x_i}^b$ or $h_{y_j}^f \neq h_{y_j}^b$) (see Ref. [11] for details).

From the HOC scheme (2.2), the discretization linear system on nonuniform grids can be given

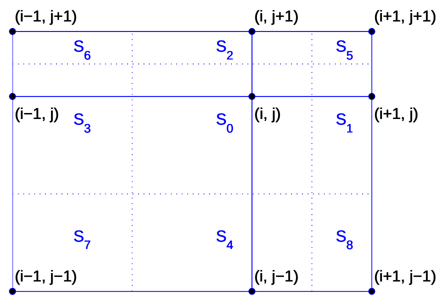


Figure 2: Areas for restriction operator [8]

as below

$$A_h u_h = F_h, \tag{2.3}$$

where A_h is a sparse matrix, F_h is a right-hand vector, and u_h is a unknown solution.

Similar to the above process, the domain Ω also can be further subdivided into a sequence of refined grids $\Omega_j (j = 1, 2, \ldots, L)$ with the number of grid intervals $N_j = 2^{j-1}N$. The corresponding discretization difference equations on grid Ω_j can be constructed as follow

$$A_j u_j = F_j, \quad j = 1, 2, \ldots, L. \tag{2.4}$$

3 Restriction and interpolation on nonuniform grids

Standard multigrid methods includes those steps: eliminating high frequency component errors using a relaxation iterative method, projecting the residuals from the fine grid to the coarse grid by a restriction operator, computing an approximate (or direct) solution of the smooth error equation on the coarse grid, prolonging a correction vector back to the fine grid by a interpolation operator, updating the previous approximation by adding the correction vector, smoothing the error again using a relaxation algorithm.

Hence the restriction and interpolation operators are two important components of multigrid ap-

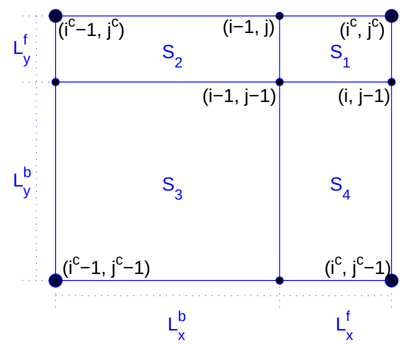


Figure 3: Areas for interpolation operator [8]

proach. In this section, we will introduce the restriction and the interpolation operators on nonuniform grids, which has been discussed by Ge and Cao in Ref. [8].

We choose a rectangle domain $[x_{i-1}, x_{i+1}] \times [y_{j-1}, y_{j+1}]$ on the fine grid Ω_h (see Figure 2) to introduce the restriction operator [8]. The symbols S_0 is the area bounded by four dashed lines around the reference mesh node (i, j) and $S_i (i = 1, \ldots, 8)$ are the area bounded by the dashed lines and the boundary lines around them. Let S is the total area of $S_i (i = 0, \ldots, 8)$. Namely,

$$
\begin{aligned}
S_0 &= \frac{(x_{i+1} - x_{i-1})(y_{j+1} - y_{j-1})}{16}, \quad & S_1 &= \frac{(x_{i+1} - x_i)(y_{j+1} - y_{j-1})}{4}, \\
S_2 &= \frac{(x_{i+1} - x_{i-1})(y_{j+1} - y_j)}{4}, \quad & S_3 &= \frac{(x_i - x_{i-1})(y_{j+1} - y_j)}{4}, \\
S_4 &= \frac{(x_{i+1} - x_{i-1})(y_j - y_{j-1})}{4}, \quad & S_5 &= \frac{(x_{i+1} - x_i)(y_{j+1} - y_j)}{4}, \\
S_6 &= \frac{(x_i - x_{i-1})(y_{j+1} - y_j)}{4}, \quad & S_7 &= \frac{(x_i - x_{i-1})(y_j - y_{j-1})}{4}, \\
S_8 &= \frac{(x_{i+1} - x_i)(y_j - y_{j-1})}{4}, \quad & S &= \sum_{i=0}^{8} S_i = (x_{i+1} - x_{i-1})(y_{j+1} - y_{j-1}).
\end{aligned}
$$

Assume $r_{i,j}$, $\bar{r}_{i^c,j^c}$ are the residuals at the fine grid nodes (i, j) and at the coarse grid nodes (i^c, j^c), respectively. The fine grid index (i, j) and the coarse grid index (i^c, j^c) satisfy that $i = 2i^c$

and $j = 2j^c$. Then, the restriction operator on nonuniform grids can be explicitly express as [8]

$$\begin{aligned}\bar{r}_{i^c,j^c} =& \frac{1}{S}(S_0 r_{i,j} + S_1 r_{i-1,j} + S_2 r_{i\,j-1} + S_3 r_{i+1\,j} + S_4 r_{i\,j+1} + \\ & S_5 r_{i-1,j-1} + S_6 r_{i+1,j-1} + S_7 r_{i+1,j+1} + S_8 r_{i-1,j+1}).\end{aligned} \tag{3.5}$$

We now turn to introduce the interpolation operator [8] on the coarse grid cell $[x_{i^c-1}, x_{i^c}] \times [y_{j^c-1}, y_{j^c}]$ with a similar strategy.

Use the big black and the small black dots to denote the grid points at coarse and fine grids (see Figure 3), respectively. It is need to notice that the coarse grid points are simultaneously some points on the fine grid.

Let S_i $(i = 1, \ldots, 4)$ are the area bounded of fine grid cell (see Figure 3), respectively. S is the area of coarse grid cell $[x_{i^c-1}, x_{i^c}] \times [y_{j^c-1}, y_{j^c}]$. The marks L_y^f and L_y^b correspond to the forward and the backward step lengths of the fine grid in y direction. L_x^f and L_x^b have a similar definitions in the x direction.

Under this condition, the explicitly express of interpolation operator [8] on nonuniform grids can be written as

$$\begin{aligned}
r_{i,j} &= \bar{r}_{i^c,j^c}, \\
r_{i-1,j} &= \frac{1}{L_x^b + L_x^f}(L_x^f \bar{r}_{i^c-1,j^c} + L_x^b \bar{r}_{i^c,j^c}), \\
r_{i,j-1} &= \frac{1}{L_y^b + L_y^f}(L_y^f \bar{r}_{i^c,j^c-1} + L_y^b \bar{r}_{i^c,j^c}), \\
r_{i-1,j-1} &= \frac{1}{S}(S_1 \bar{r}_{i^c-1,j^c-1} + S_2 \bar{r}_{i^c,j^c-1} + S_3 \bar{r}_{i^c,j^c} + S_4 \bar{r}_{i^c-1,j^c}).
\end{aligned}$$

4 Multiscale-like multigrid computation

In this section, we shall develop a multiscale-like multigrid (MSLMG) computation with suitable initial solutions for the sparse linear system (2.3) on nonuniform grids. The main idea of the

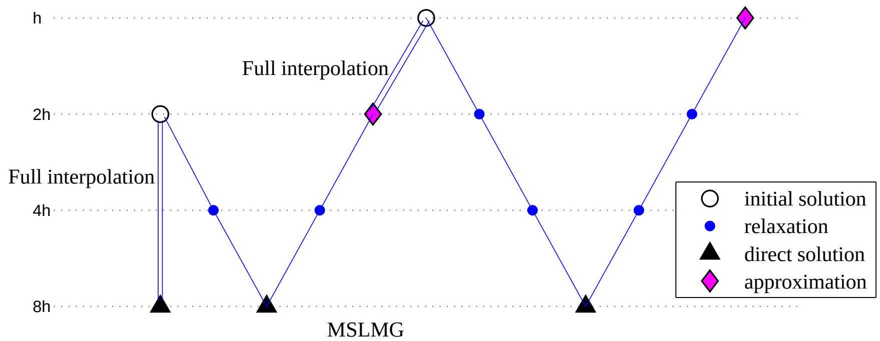


Figure 4: Illustration of the multiscale-like multigrid (MSLMG) method

algorithm is as follows, full interpolation operator is employed to construct more accurate initial guesses on two finest grids Ω_{L-1} and Ω_L. Then, with the provided initial solutions, the V-cycle multigrid solver[8] is applied to complete the computation on the grids Ω_{L-1} and Ω_L, successively. The detail of proposed algorithm is stated in Algorithm 1, and the structure of MSLMG method is shown in Figure 4.

It's worth pointing that, the proposed MSLMG computation can be regarded as the extension and application of Ref. [5, 17] on nonuniform grids.

Algorithm 1 *Multiscale-like multigrid (MSLMG) computation*

Step 1. *Compute the solution* u_1^* *on the coarsest grid* Ω_1 *by a direct solver.*

Step 2. *Construct an initial vector* u_{L-1}^0 *on the grid* Ω_{L-1} *by using a full interpolator.*

Step 3. *Compute the solution* u_{L-1}^* *by using the V-cycle multigrid with an initial guess* u_{L-1}^0.

Step 4. *Construct an initial vector* u_L^0 *on the grid* Ω_L *by using a full interpolator.*

Step 5. *Compute the solution* u_L^* *by using the V-cycle multigrid with an initial guess* u_L^0.

We now turn to estimate the computational cost of the MSLMG method in terms of *work units* (WU)[1]. Roughly speaking, here “1 WU " denotes the computational cost of performing one relaxation sweep on the finest grid Ω_L. For the sake of discussion, we neglect the amount of work required on the coarsest grid and the transfer operators (interpolation and restriction) between different nonuniform grids since it normally counts $10-20\%$ of the total cost of the entire algorithm.

Assume that the m_j is the number of V-cycles with v_1 pre-smoothing and v_2 post-smoothing, required on the grid $\Omega_j (j = L-1, L)$ when the computed approximation satisfies a given stopping criterion. Then the computational cost (WU) of MSLMG with L embedded grids can be estimated as the form of

$$WU_L^{MSLMG} \approx (v_1 + v_2)[m_L + (m_L + m_{L-1})\sum_{j=2}^{L-1} 2^{-2(L-j)}]. \tag{4.6}$$

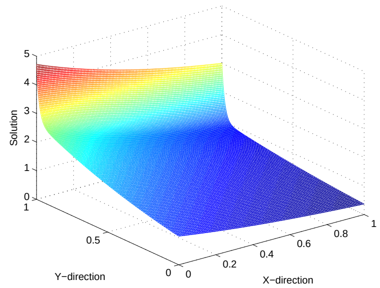
5 Numerical experiments

In our numerical experiments, we tested the proposed MSLMG algorithm for computing two 2D convection-diffusion equations with vertical boundary layer or local singularity, and compared the results with the current V-cycle multigrid (VMG) method [8]. Our codes are written in Matlab and the programs are carried out on a desktop with Inter (R) Core(TM) i5-6200U CPU (2.30GHZ,2.40GHZ) and 4GB RAM.

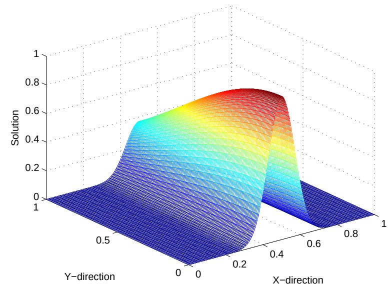
We chosen the alternating line Gauss-Seidel iterative method as smoother. The VMG method with three pre-smoothing and three post-smoothing was used in our MSLMG approach or as the compared algorithm. The iterative procedure of V-cycles was terminated when relative residuals was reduced by 10^{-10}. In our MSLMG method, the spline interpolator was used to provide initial guesses on two finest grids Ω_{L-1} and Ω_L, respectively. As for the compared VMG method, the initial guess on the finest grid was the zeros vector.

All reported errors $||E_L||_\infty$ were the L^∞-norm errors between the exact solution u and the numerical solution u_L^* obtained by VMG or MSLMG methods. The symbols $Time$ and $Iter$ denotes the computational time in seconds and the number of V-cycles (for MSLMG method on the Ω_L and the Ω_{L-1}) required, respectively. Besides, we also reported the order of solution accuracy, which is defined by

$$Rate = \frac{\log(||E_H||_\infty / ||E_h||_\infty)}{\log(H/h)},$$



(a) Example 5.1



(b) Example 5.2

Figure 5: Exact solutions

where E_H and E_h are the errors of numerical solutions with the meshsizes H and h, respectively.

Example 5.1 *[8, 16] Consider the steady-state convection-diffusion equation*

$$-u_{xx}(x,y) - u_{yy}(x,y) + \frac{100}{(1+y)}u_y(x,y) = f(x,y), \tag{5.7}$$

with suitable Dirichlet boundary conditions and the forcing term $f(x,y)$, such that the exact solution is

$$u(x,y) = \exp(y-x) + \frac{(1+y)^{101}}{2^{100}}. \tag{5.8}$$

Example 5.2 *[19] Consider the following differential equation*

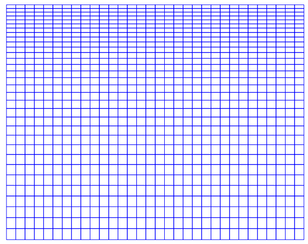
$$-u_{xx}(x,y) - u_{yy}(x,y)) + c(x,y)u_x(x,y) + d(x,y)u_y(x,y) = f(x,y), \tag{5.9}$$

with coefficients

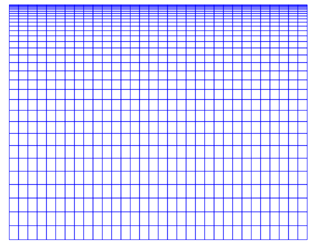
$$c(x,y) = 100x(x-1)(1-2y), \quad d(x,y) = 100y(y-1)(1-2x).$$

Where the Dirichlet boundary conditions and the forcing term $f(x,y)$ are satisfy the analytic solution

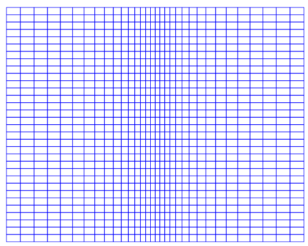
$$u(x,y) = \exp(-100(x-0.5)^2 - y^2). \tag{5.10}$$



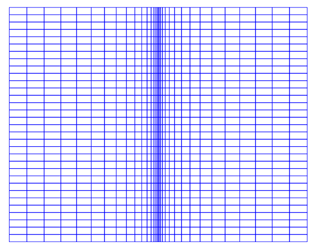
(a) Example 5.1, $\lambda_x = 0,\ \lambda_y = 0.5$



(b) Example 5.1, $\lambda_x = 0,\ \lambda_y = 0.9$



(c) Example 5.2 , $\lambda_x = 0.5,\ \lambda_y = 0$



(d) Example 5.2 , $\lambda_x = 0.9,\ \lambda_y = 0$

Figure 6: Nonuniform grid (32×32) with different stretching parameters λ_x and λ_y

The exact solution of Example 5.1 has a vertical boundary layer attached to the line $y = 1$ (see Figure 5 (a)). To obtain an appropriate discrete grid for this case, we used a nonuniform grid along the y direction with clustering near $y = 1$ and a uniform grid along the x direction, which were defined by the following grid stretching function [8]

$$x_i = \frac{i}{i_{\max}}\,(\lambda_x = 0)\,, \quad y_j = \frac{j}{j_{\max}} + \frac{\lambda_y}{\pi}\sin\left(\frac{\pi j}{j_{\max}}\right).$$

Here, stretching parameters λ_x and λ_y were used to control the density of grid nodes in the x and the y direction, respectively. For instance, the grids 32×32 with different mesh stretching parameters $(\lambda_y = 0.5, 0.9)$ were shown in Figure 6 (a) and (b).

As for Example 5.2, the exact solution has a local singularity along $x = 0.5$ (see Figure 5 (b)). Therefore, we chosen a nonuniform grid along the x direction with clustering near $x = 0.5$ and a uniform grid along the y direction, with the following stretching function [8]

$$x_i = \frac{i}{i_{\max}} + \frac{\lambda_x}{2\pi}\sin\left(\frac{2\pi i}{i_{\max}}\right), \quad y_j = \frac{j}{j_{\max}}\left(\lambda_y = 0\right).$$

Those grids 32×32 with different mesh stretching parameters ($\lambda_x = 0.5, 0.9$) were given in Figure 6 (c) and (d).

Table 1: Numerical results of Example 5.1

λ_y	N_L	VMG [8]					MSLMG				
		$\|\|E_L\|\|_\infty$	*Rate*	*Iter*	*WU*	*Time*	$\|\|E_L\|\|_\infty$	*Rate*	*Iter*	*WU*	*Time*
0.1	128	9.82(-5)	–	5	39.4	8.91	9.82(-5)	–	3, 4	31.1	7.05
	256	6.16(-6)	4.00	5	39.4	40.2	6.17(-6)	3.99	2, 4	23.3	29.7
	512	3.85(-7)	4.00	5	39.4	179	3.85(-7)	4.00	2, 3	21.4	108
0.3	128	3.46(-5)	–	5	39.4	9.02	3.46(-5)	–	3, 4	31.1	6.92
	256	2.17(-6)	4.00	5	39.4	37.8	2.17(-6)	4.00	2, 3	21.4	25.3
	512	1.35(-7)	4.00	5	39.4	168	1.35(-7)	4.00	2, 3	21.4	106
0.5	128	7.28(-6)	–	5	39.4	8.89	7.28(-6)	–	2, 3	21.4	5.84
	256	4.56(-7)	4.00	5	39.4	37.9	4.56(-7)	4.00	2, 3	21.4	25.5
	512	2.85(-8)	4.00	5	39.4	182	2.87(-8)	3.99	1, 2	11.6	88.9
0.7	128	1.43(-6)	–	5	39.4	8.91	1.43(-6)	–	2, 3	21.4	5.83
	256	8.95(-8)	4.00	5	39.4	24.6	8.92(-8)	4.01	1, 2	11.6	14.4
	512	5.59(-9)	4.00	5	39.4	177	5.59(-9)	4.00	1, 2	11.6	89.2
0.9	128	5.18(-6)	–	4	31.5	7.05	5.06(-6)	–	1, 3	13.5	5.08
	256	3.23(-7)	4.00	4	31.5	23.1	3.15(-7)	4.00	1, 2	11.6	13.5
	512	2.09(-8)	3.95	4	31.5	171	1.99(-8)	3.98	1, 1	9.75	78.9

The numerical results of the above two examples obtained by VMG [8] and MSLMG methods with different stretching parameters (λ_x, λ_y) were listed in Tables 1 and 2.

It is easy to see that MSLMG method is fourth-order accurate, as also is the VMG method.

As for efficiency, the proposed MSLMG method is much faster than the VMG method, particularly for large scale cases (i.e., $N \geq 256$). The reason is that, decreasing number of iterations on the finest grid will effectively reduce the total computational cost. Compared with the VMG method, fewer number of V-cycles are required on the finest grid for our MSLMG method (see the tenth column of Tables 1 and 2 for details). The difference in number of V-cycles became more apparent (decreases to one or two V-cycles) When $N \geq 256$. Meanwhile, the cost (WU) in Tables 1 and

Table 2: Numerical results of Example 5.2

λ_x	N_L	VMG [8]					MSLMG				
		$\|\|E_L\|\|_\infty$	*Rate*	*Iter*	*WU*	*Time*	$\|\|E_L\|\|_\infty$	*Rate*	*Iter*	*WU*	*Time*
0.1	128	2.04(-6)	–	5	39.4	9.05	2.04(-6)	–	3, 4	31.1	7.21
	256	1.27(-7)	4.00	5	39.4	38.7	1.27(-7)	4.00	2, 4	23.3	29.1
	512	7.96(-9)	4.00	5	39.4	189	8.36(-9)	3.93	1, 3	13.5	103
0.3	128	1.18(-6)	–	5	39.4	8.87	1.18(-6)	–	3, 4	31.1	7.29
	256	7.38(-8)	4.00	5	39.4	38.6	7.37(-8)	4.00	2, 3	21.4	25.5
	512	4.61(-9)	4.00	5	39.4	183	4.80(-9)	3.94	1, 3	13.5	109
0.5	128	1.97(-6)	–	5	39.4	9.20	1.97(-6)	–	3, 4	31.1	7.01
	256	1.23(-7)	4.00	5	39.4	37.8	1.23(-7)	4.00	2, 3	21.4	25.2
	512	7.71(-9)	4.00	5	39.4	194	7.93(-9)	3.96	1, 3	13.5	102
0.7	128	3.35(-6)	–	5	39.4	8.85	3.35(-6)	–	3, 4	31.1	6.95
	256	2.10(-7)	4.00	5	39.4	39.9	2.09(-7)	4.00	2, 4	23.3	27.1
	512	1.31(-8)	4.00	5	39.4	207	1.35(-8)	3.95	1, 3	13.5	104
0.9	128	5.27(-6)	–	6	47.3	10.7	5.27(-6)	–	3, 5	33.0	7.34
	256	3.31(-7)	4.00	5	39.4	40.1	3.29(-7)	4.00	2, 4	23.3	27.1
	512	2.07(-8)	4.00	6	47.3	213	2.08(-8)	3.99	1, 3	13.5	101

2 demonstrate that the proposed spent less cost than VMG method to compute a certain accurate solution. Furthermore, when a larger scale grid is required, the superiority of the proposed method on efficiency is more obvious (see the sixth and the eleventh column of Tables 1 and 2 for details).

As for numerical stability, the proposed algorithm as well as the VMG method, can effectively solve the algebraic systems (2.3) with different grid stretching (λ_x, λ_y).

Hence, MSLMG method is a cost-effective method.

6 Conclusion

We present an efficient multiscale-like multigrid (MSLMG) computation to solve a two dimensional (2D) convection-diffusion equations on nonuniform grids, based on a transformation-free HOC compact difference scheme. The main highlight of this algorithm is that the appropriate initial solutions are provided on the second finest and the finest grids, which has reduce the total computational cost for solving the discrete systems on nonuniform grids. Numerical experiments demonstrate that the superiority of the present method performs to solve the 2D convection-diffusion problem on nonuniform grids, comparison of the computational cost with current multigrid method.

It is worth pointing out that we can extend the proposed method to solving other partial differential equation with little modifications, such as the 3D case, the variable coefficient case and more general linear case. The research on these aspects will be reported in our future work.

References

[1] Briggs, W.L., Henson, V.E., McCormick, S.F.: *A multigrid tutorial (second edition)*, SIAM. Philadelphia, Pennsylvania, PA (2000)

[2] Dai, R.X., Lin, P.P., Zhang, J.: *An efficient sixth-order solution for anisotropic Poisson equation with completed Richardson extrapolation and multiscale multigrid method*, Comput. Math. Appl. **73** (8), 1865-1877 (2017)

[3] Dai, R.X., Wang, Y., Zhang J.: *Fast and high accuracy multiscale multigrid method with multiple coarse grid updating strategy for the 3D convection-diffusion equation*, Comput. Math. Appl., **66**: 542-559 (2013)

[4] Dai, R.X., Zhang J., Wang Y.: *Multiple coarse grid acceleration for multiscalezs multigrid computation*. J. Comput. Appl. Math., **269**: 75-85 (2014)

[5] Dai, R.X., Lin, P.P., Zhang, J.: *An EXCMG accelerated multiscale multigrid computaion for 3D Poisson equation*, Comput. Math. Appl., **77**, 2051-2060 (2019)

[6] Elman, H. C., Ernst, O. G., O'leary, D. P.: *A multigrid method enhanced by Krylov subspace iteration for discrete Helmholtz equations*, SIAM J. Sci. Comput. **23** (4), 1291-1315 (2001)

[7] Erlangga, Y. A., Oosterlee, C. W., Vuik, C.: *A novel multigrid based precondtioner for heterogeneous Helmholtz problems*, SIAM J. Sci. Comput. **27** (4), 1471-1492 (2006)

[8] Ge, Y.B., Cao, F.J.: *Multigrid method based on the transformation-free HOC scheme on nonuniform grids for 2D convection diffusion problems*, J. Comput. Phys. **230** (10), 4051-4070 (2011)

[9] Ge, Y.B., Cao, F.J.: *A transformation-free HOC scheme and multigrid method for solving the 3D poissson euqation on nonuniform grids*, J. Comput. Phys. **234** (18), 199-216 (2013)

[10] Gupta, M.M., Zhang, J.: *High accuracy multigrid solution of the 3D convection-diffusion euqaiton*, Appl. Math. Comput. **113** (2), 249-274 (2000)

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

[12] Li, M. Li, C.L., et al: *Cascadic multigrid methods combined with sixth-order compact scheme for Poisson equation*, Numer. Algorithms, **71**(4), 715-727 (2016)

[13] Li, M., Zheng, Z. S., Pan, K. J.: *Extrapolation multiscale multigrid method for solving 2D Poisson equation with sixth order compact scheme*, J. Appl. Math. Comput., 2019, 60: 589-604.

[14] Pan, K.J., He, D.D., Hu, H.L.: *An extrapolation cascadic multigrid method combined with a fourth-order compact scheme for 3D Poisson equation*, J. Sci. Comput. **70**, 1180-1203 (2017)

[15] Sun, H.W., Zhang, J.: *A high order finite difference discretization strategy based on extrapolation for convection-diffusion equations*, Numer. Meth. Part. D. E. **20** (1), 18-32 (2004)

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

[17] Wang, Y., Zhang, J.: *sixth-order compact scheme combined with multigrid method and extrapolation technique for 2D poisson equation*, J. Comput. Phys. **228** (1) 137-146 (2009)

[18] Zhang, J.: *Multigrid method and fourth-order compact difference scheme for 2D Poisson equation with unequal meshsize discretization*, J. Comput. Phys. **179** , 170-179 (2002)

[19] Zhang, J., Sun, H.W., Zhao, J.J.: *High order compact scheme with multigrid local mesh refinement procedure for convection diffusion problems*, Comput. Method. Appl. M. **191**, 4661-4674 (2002)