Advances in Difference Equations
a SpringerOpen Journal

RESEARCH Open Access

An extrapolation full multigrid algorithm combined with fourth-order compact scheme for convection–diffusion equations

Ming Li[1,2], Zhoushun Zheng[1] and Kejia Pan[1*]

[*]Correspondence:
pankejia@hotmail.com
[1]School of Mathematics and Statistics, Central South University, Changsha, P.R. China
Full list of author information is available at the end of the article

Abstract

In this paper, we propose an extrapolation full multigrid (EXFMG) algorithm to solve the large linear system arising from a fourth-order compact difference discretization of two-dimensional (2D) convection diffusion equations. A bi-quartic Lagrange interpolation for the solution on previous coarser grid is used to construct a good initial guess on the next finer grid for V- or W-cycles multigrid solver, which greatly reduces the number of relaxation sweeps. Instead of performing a fixed number of multigrid cycles as used in classical full multigrid methods, a series of grid level dependent relative residual tolerances is introduced to control the number of the multigrid cycles. Once the fourth-order accurate numerical solutions are obtained, a simple method based on the midpoint extrapolation is employed for the fourth-order difference solutions on two-level grids to construct a sixth-order accurate solution on the entire fine grid cheaply and directly. Numerical experiments are conducted to verify that the proposed method has much better efficiency compared to classical multigrid methods. The proposed EXFMG method can also be extended to solve other kinds of partial differential equations.

MSC: 65N06; 65N55

Keywords: Extrapolation; Full multigrid method; Compact difference scheme; Convection–diffusion equation

1 Introduction

Richardson extrapolation [1, 2] is an acceleration method, used to improve the rate of convergence for a sequence. In 1983, Marchuk and Shaidurov extended Richardson extrapolation technique in the framework of the finite difference (FD) method [3]. Later on, Blum et al. discussed Richardson extrapolation strategy in conjunction with the finite element (FE) method [4–7].

Multigrid methods (MGs) [8, 9] have been shown to be one of the most efficient modern numerical strategies to solve the large linear systems arising from FE or FD discretizations of partial differential equations (PDEs), such as the Poisson equation [10–14], Helmholtz equation [15–17], convection–diffusion equation [18–20]. The main advantage of the multigrid method is that its convergence rate is independent of the number of unknowns [8, 21]. The full multigrid (FMG) method, which combines nested iteration with a V- or W-cycles multigrid method, is well known to be optimal with respect to the energy norm and the L^2-norm [8, 21–23]. It starts on the coarsest grid with a direct solver, then the solution is interpolated to form the initial guess on the next finer grid, where a few V- or W-cycles are used to produce an approximate solution whose algebraic error matches with the discretization error. The procedure goes on recursively until the finest grid.

Recently, the Richardson extrapolation technique has been applied in multigrid methods. One group of researchers used Richardson extrapolation to construct a good initial guess for the iterative solver. An extrapolation cascadic multigrid (EXCMG) method [24, 25] was proposed by Chen et al. to solve second-order elliptic problems. This method uses a new extrapolation formula to construct a quite good initial guess for the iterative solution on the next finer grid, which greatly improves the convergence rate of the original CMG algorithm (see [24–26] for details). Then the EXCMG method has been successfully applied to non-smooth elliptic problems [27, 28], parabolic problems [29], and some other related problems [30–32]. Moreover, Pan [33] and Li [34, 35] developed some EXCMG methods combined with high-order compact difference schemes to solve Poisson equations. In 2017, Hu et al. [36] obtained the super-optimality of the EXCMG method under the energy norm for $H^{2+\alpha}$-regular ($0 < \alpha \leq 1$) elliptic problems.

Another group of researchers made efforts to design an iterative procedure with Richardson extrapolation strategy to enhance the solution accuracy on the finest grid. Sun, Zhang and Dai developed a sixth-order FD scheme for solving the 2D convection–diffusion equation [37, 38]. In their approach, the ADI method is applied to compute the fourth-order accurate solution on the fine and coarse grids, respectively, then the Richardson extrapolation technique and an operator-based interpolation scheme are employed in each ADI iteration to calculate the sixth-order accurate solution on the fine grid. In 2009, Wang and Zhang designed an explicit sixth-order compact discretization strategy (MG-Six) for the 2D Poisson equation [39]. They used a V-cycle multigrid method to get the fourth-order accurate solutions on both the fine and the coarse grids first, and then chose the iterative operator with Richardson extrapolation technique to compute the sixth-order accurate solution on the fine grid.

We extend the idea described in the literature [32, 33, 39] to the original full multigrid method, and develop an extrapolation full multigrid (EXFMG) method to solve the 2D convection–diffusion equation with a fourth-order compact difference discretization. To be more precise, a bi-quartic Lagrange interpolation operator on a coarser grid is applied to get a good initial guess on the next finer grid for the multigrid V- or W-cycles solver. Besides, a stopping criterion related to relative residual is used to conveniently obtain the numerical solution with the desired accuracy. Moreover, a mid-point extrapolation strategy is used to obtain cheaply and directly a sixth-order accurate solution on the entire fine grid from two fourth-order solutions on two different grids (current fine and previous coarse grids). Finally, through two examples chosen from the literature, the computational efficiency of the EXFMG method is discussed in detail.

The rest of the paper is organized as follows. Section 2 introduces the model problem and a fourth-order compact difference scheme. Section 3 depicts the classical multigrid method. And the extrapolation full multigrid method is described in Sect. 4. Numerical results are given in Sect. 5 to show the efficiency and high-order accuracy of the new algorithm. Finally, concluding remarks are provided in Sect. 6.

2 Model problem and discretization

Consider the following two-dimensional (2D) convection–diffusion equation:

$$-u_{xx}(x,y) - u_{yy}(x,y) + p(x,y)u_x(x,y) + q(x,y)u_y(x,y) = f(x,y), \quad (x,y) \in \Omega, \tag{1}$$

with suitable Dirichlet boundary conditions. Here Ω is a rectangular domain, $p(x,y)$ and $q(x,y)$ are the given convection velocities, $f(x,y)$ is the given source term, and $u(x,y)$ is the unknown function. All these functions are assumed to be sufficiently smooth and have the necessary continuous partial derivatives up to certain orders. This equation is used to describe many processes in fields of fluid dynamics.

Let $\Omega = [0,M_x] \times [0,M_y]$ and divide Ω into $N_x \times N_y$ uniform grids with the mesh size $\Delta x = \frac{M_x}{N_x}$ and $\Delta y = \frac{M_y}{N_y}$ in the x and y coordinate directions, respectively. N_x and N_y are the numbers of uniform intervals in the x and y directions, respectively. The mesh points are (x_i,y_j) with $x_i = i\Delta x$ and $y_i = i\Delta y$, $0 \le i \le N_x$, $0 \le j \le N_y$. For simplicity, we use $U_{i,j}$ to represent the approximation of $u(x_i,y_j)$ at the mesh point (x_i,y_j), set $f_{i,j} = f(x_i,y_j)$, $p_{i,j} = p(x_i,y_j)$, and so on.

The fourth-order compact difference scheme for Eq. (1) with nine-point computational stencil is given by [40]

$$\begin{aligned}
&\left(\frac{4A_{i,j}}{\Delta x^2\Delta y^2} - \frac{2D_{i,j}}{\Delta x^2} - \frac{2E_{i,j}}{\Delta y^2}\right)U_{ij} + \left(\frac{-2A_{i,j}}{\Delta x^2\Delta y^2} - \frac{C_{i,j}}{\Delta x\Delta y^2} + \frac{D_{i,j}}{\Delta x^2} + \frac{G_{i,j}}{2\Delta x}\right)U_{i+1,j}\\
&\quad + \left(\frac{-2A_{i,j}}{\Delta x^2\Delta y^2} + \frac{C_{i,j}}{\Delta x\Delta y^2} + \frac{D_{i,j}}{\Delta x^2} - \frac{G_{i,j}}{2\Delta x}\right)U_{i-1,j}\\
&\quad + \left(\frac{-2A_{i,j}}{\Delta x^2\Delta y^2} - \frac{B_{i,j}}{2\Delta x^2\Delta y} + \frac{E_{i,j}}{\Delta y^2} + \frac{H_{i,j}}{2\Delta y}\right)U_{i,j+1}\\
&\quad + \left(\frac{-2A_{i,j}}{\Delta x^2\Delta y^2} + \frac{B_{i,j}}{\Delta x^2\Delta y} + \frac{E_{i,j}}{\Delta y^2} - \frac{H_{i,j}}{2\Delta y}\right)U_{i,j-1}\\
&\quad + \left(\frac{A_{i,j}}{\Delta x^2\Delta y^2} + \frac{B_{i,j}}{2\Delta x^2\Delta y} - \frac{C_{i,j}}{2\Delta x\Delta y^2} - \frac{K_{i,j}}{4\Delta x\Delta y}\right)U_{i-1,j+1}\\
&\quad + \left(\frac{A_{i,j}}{\Delta x^2\Delta y^2} - \frac{B_{i,j}}{2\Delta x^2\Delta y} + \frac{C_{i,j}}{2\Delta x\Delta y^2} - \frac{K_{i,j}}{4\Delta x\Delta y}\right)U_{i+1,j-1}\\
&\quad + \left(\frac{A_{i,j}}{\Delta x^2\Delta y^2} + \frac{B_{i,j}}{2\Delta x^2\Delta y} + \frac{C_{i,j}}{2\Delta x\Delta y^2} + \frac{K_{i,j}}{4\Delta x\Delta y}\right)U_{i+1,j+1}\\
&\quad + \left(\frac{A_{i,j}}{\Delta x^2\Delta y^2} - \frac{B_{i,j}}{2\Delta x^2\Delta y} - \frac{C_{i,j}}{2\Delta x\Delta y^2} + \frac{K_{i,j}}{4\Delta x\Delta y}\right)U_{i-1,j-1} = F_{i,j},
\end{aligned} \tag{2}$$

with the coefficients

$$\begin{aligned}
&A_{i,j} = -\frac{\Delta x^2 + \Delta y^2}{12}, \qquad B_{i,j} = \frac{\Delta x^2 + \Delta y^2}{12}q_{i,j}, \qquad C_{i,j} = \frac{\Delta x^2 + \Delta y^2}{12}p_{i,j},\\
&D_{i,j} = -1 + \frac{\Delta x^2}{12}\bigl[2(p_x)_{i,j}) - p_{i,j}^2\bigr], \qquad E_{i,j} = -1 + \frac{\Delta y^2}{12}\bigl(2(q_y)_{i,j} - q_{i,j}^2\bigr),\\
&K_{i,j} = \frac{\Delta x^2}{12}\bigl[2(q_x)_{i,j} - (pq)_{i,j}\bigr] + \frac{\Delta y^2}{12}\bigl[2(p_y)_{i,j} - (pq)_{i,j}\bigr],\\
&G_{i,j} = p_{i,j} + \frac{\Delta x^2}{12}\bigl[(p_{xx})_{i,j} - (pp_x)_{i,j}\bigr] + \frac{\Delta y^2}{12}\bigl[(p_{yy})_{i,j} - (p_yq)_{i,j}\bigr],
\end{aligned}$$

$$\begin{aligned} H_{i,j} &= q_{i,j} + \frac{\Delta x^2}{12}\big[(q_{xx})_{i,j} - (pq_x)_{i,j}\big] + \frac{\Delta y^2}{12}\big[(q_{yy})_{i,j} - (qq_y)_{i,j}\big], \\ F_{i,j} &= f_{i,j} + \frac{\Delta x^2}{12}\big[(f_{xx})_{i,j} - (pf_x)_{i,j}\big] + \frac{\Delta y^2}{12}\big[(f_{yy})_{i,j} - (qf_y)_{i,j}\big]. \end{aligned} \tag{3}$$

Now set $h_1 = \Delta x = \Delta y$ and assume that the domain Ω is divided into a sequence of grids $\Omega_k, k = 1, 2, \ldots, L$ with mesh size $h_k = h_1/2^{k-1}$. By applying the fourth-order compact scheme (2), we obtain the corresponding sparse linear system,

$$A_k u_k = F_k, \quad k = 1, 2, \ldots, L, \tag{4}$$

where A_k is a nine-diagonal matrix, F_k and u_k are the right-hand side vector and the solution vector, respectively.

3 Classical multigrid method

The multigrid (MG) method is a very efficient approach to solving large sparse linear systems arising from FE or FD discretizations of various PDEs [8]. The main feature behind the MG method is to divide the errors into high frequency components and low frequency components and eliminate them in different grids. To be more specific, the high frequency component errors are eliminated on the fine grid, while the remaining low frequency errors are eliminated by a recursive procedure on the coarse grid. We depict this method in Algorithm 1, in which a direct solver "DSOLVER" is used on the coarsest grid, since the size of the linear system is small. A restriction operator R_{j+1}^{j} (such as a full-weighting scheme) is employed to project residual from the fine grid to the coarse grid, and an interpolation operator I_j^{j+1} (such as bilinear interpolator) is used to transfer corrections from the coarse grid to the fine grid. The recursion parameter N_{cycles} is greater than or equal to one. If Ncycles = 1, the so-called V-cycle is generated, while Ncycles = 2 leads to the W-cycle. A cycle of Algorithm 1, means that it starts from the finest grid to the coarsest grid and back to the finest grid again. When Algorithm 1 is applied to the finest grid system (2) in the solution of which we are interested, it can be repeated until the accuracy of approximation on the finest grid is considered to be high enough.

Algorithm 1 Classical multigrid method: $u_L \Leftarrow \text{MGM}(A_L, u_L^0, F_L, L)$

```
Set $j \leftarrow L - 1$
Pre-smoothing $v_1$ times: $u_{j+1}^{v_1} \Leftarrow \text{Smooth}^{v_1}(A_{j+1}, u_{j+1}^0, F_{j+1})$
Restricting: $r_j \Leftarrow R_{j+1}^{j}(F_{j+1} - A_{j+1}u_{j+1}^{v_1})$
**if** $j == 1$ (Coarsest grid) , **then**
    $e_1 \Leftarrow \text{DSOLVER}(A_1 e_1 = r_1)$
**else**
    Set $e_j^0 \leftarrow 0$
    **for** $i = 1, N_{\text{cycles}}$ **do**
        $e_j \Leftarrow \text{MGM}(A_j, e_j^0, r_j, j)$,
Correcting: $\bar{u}_{j+1} \leftarrow u_{j+1}^{v_1} + I_j^{j+1} e_j$
Post-smoothing $v_2$ times: $u_{j+1} \Leftarrow \text{Smooth}^{v_2}(A_{j+1}, \bar{u}_{j+1}, F_{j+1})$
```

Algorithm 2 Full multigrid method (FMG)

```
$u_1^* \Leftarrow \mathrm{DSOLVER}(A_1 u_1 = F_1)$
for $j = 2, 3, \ldots, L$ do
    Compute an initial guess: $u_j^0 \Leftarrow P_{j-1}^j u_{j-1}^*$    ▷ $P_{j-1}^j$ is a FMG interpolation operator
    Solve $u_j^*$ by performing $k_j$ times $j$-grid cycles using the initial approximation $u_j^0$

        $u_j^* \Leftarrow \mathrm{MGM}^{k_j}\left(A_j, u_j^0, F_j, j\right)$
```

In fact, Algorithm 1 can be incorporated with the classical FMG method naturally. The FMG method starts with the coarsest grid, and uses an interpolation operator P_{j-1}^j to produce an initial guess for Algorithm 1 on the next finer grid; see Algorithm 2 for details. The interpolation operator P_{j-1}^j usually is of higher accuracy than the interpolation operator I_{j-1}^j used within the multigrid iteration. It is well known that FMG delivers an approximation up to discretization accuracy if the multigrid cycle converges satisfactorily and if the order of FMG interpolation is larger than the discretization order of the PDE [8].

4 Extrapolation full multigrid method

4.1 Description of the EXFMG algorithm

In this section, we shall establish an extrapolation full multigrid (EXFMG) method to solve the 2D convection–diffusion equation (1) with the fourth-order compact discretization (2). Similar to FMG method, the EXFMG method starts with the coarsest grid to compute the solution by a direct solver, and applies an interpolation operator on the current grid to provide an initial guess for Algorithm 1 on the next finer grid. Besides, EXFMG method employs a fine mesh Richardson extrapolation technique [33] to get an extrapolation solution on each grid, when the two solutions with fourth-order accuracy on current and previous grids has been obtained by the Algorithm 1. The detailed algorithm is stated in Algorithm 3, where *FMRE* denotes a fine mesh Richardson extrapolation operator which will be defined and discussed in Sect. 4.2, $\bar{u}_i^*$ and u_j^* denote the numerical solution and extrapolated solution on the grid Ω_i, respectively.

The key ingredients of this new method which are the main differences between the existing FMG methods include:

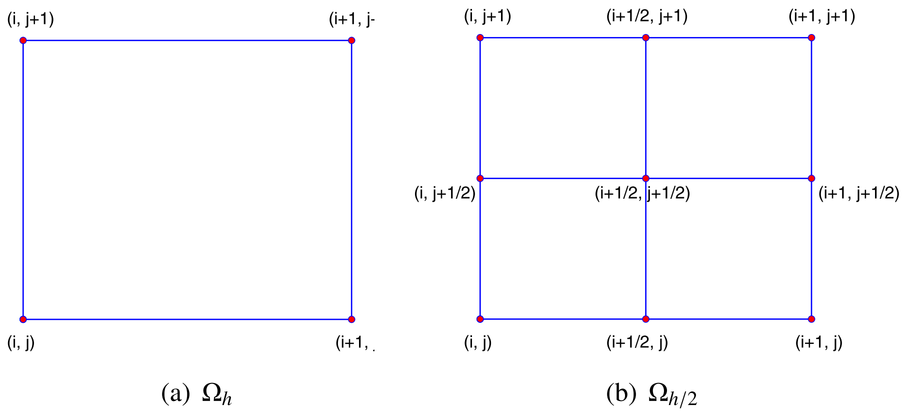
(1) A bi-quartic Lagrange interpolation operator on coarse grid is taken as a full interpolation to provide a good initial guess on the next finer grid for multigrid iteration.

(2) Instead of performing a fixed number of multigrid iterations at each level of grid as used in the usual FMG method, a tolerance ϵ_j related to the relative residual is used on jth grid level (see line 4 in Algorithm 3), which enables us to conveniently calculate the approximate solution with the desired accuracy and keep suitable computational cost.

(3) The FMG method only computes the fourth-order accuracy solution of each discretization system. But the extrapolation method based on the midpoint formula (see line 5 in Algorithm 3 and Sect. 4.2) enhances the solution accuracy on each grid (see numerical results section for details).

Algorithm 3 Extrapolation full multigrid method (EXFMG)

```
Calculate the solution $\bar{u}_1^*$ by a direct solver
for $j = 2, 3, \dots, L$ do
    Compute an initial guess: $u_j^0 \Leftarrow P_{j-1}^j \bar{u}_{j-1}^*$    ▷ $P_{j-1}^j$ is Bi-quartic Lagrange interpolation operator [33]
    Solve $\bar{u}_j^*$ by performing $j$-grid cycles until    ▷ $\epsilon_j$ denotes a given tolerance
        $\|F_j - A_j \bar{u}_j^*\| \le \epsilon_j \|F_j\|$
    Richardson Extrapolation:    ▷ $FMRE$ denotes fine mesh Richardson extrapolation operator
        $u_j^* \Leftarrow FMRE(\bar{u}_{j-1}^*, \bar{u}_j^*)$
Output all $\bar{u}_j^*$ and $u_j^*$
```



(a) Ω_h

(b) $\Omega_{h/2}$

Figure 1 Cells of coarse grid Ω_h **(a)** and refined grid$\Omega_{h/2}$ **(b)**

4.2 Fine mesh extrapolation operator

In the classical Richardson extrapolation formula, two solutions on fine and coarse grids are used to obtain an extrapolation solution on the coarse grid. Based on the Richardson extrapolation technique and interpolation theory, in 2017, Pan proposed a mid-point extrapolation technique to enhance the accuracy of approximations on fine grid directly and cheaply (see [33] for details). Now, we will introduce this strategy for 2D case in the following.

Assume that the notations u^h and $u^{h/2}$ represent the fourth-order accurate solutions on coarse grid Ω_h and fine grid $\Omega_{h/2}$ with mesh size h and $h/2$, respectively. For simplicity of description, we take a coarse cell $I_{i,j} = [x_i, x_{i+1}] \times [y_j, y_{j+1}]$ of grid Ω_h (see Fig. 1), use $u_{i,j}^h$ and $u_{i,j}^{h/2}$ to represent the fourth-order accurate solutions on the point (x_i, y_j) of the coarse grid Ω_h and the fine grid $\Omega_{h/2}$, respectively. Here $(x_i, y_{j+1/2}) \in \Omega_{h/2}$ denotes the mid-point of $(x_i, y_j) \in \Omega_h$ and $(x_i, y_{j+1}) \in \Omega_h$.

To obtain an extrapolation solution on the fine grid $\Omega_{h/2}$, we first use the classical Richardson extrapolation formula,

$$Eu_{s,t} = \frac{16}{15}u^{h/2}_{s,t} - \frac{1}{15}u^{h}_{s,t}, \quad s = i, i+1, t = j, j+1, \tag{5}$$

to get an extrapolation solution on the nodes (x_s, y_t). Then we apply the mid-point extrapolation formula [33] for those nodes $(x_s, y_{j+1/2})$, $(x_{i+1/2}, y_t)$ and $(x_{i+1/2}, y_{j+1/2})$, to obtain some formulas as follows:

$$Eu_{s,j+1/2} = u^{h/2}_{s,j+1/2} + \frac{1}{30}\bigl[\bigl(u^{h/2} - u^{h}\bigr)_{s,j} + \bigl(u^{h/2} - u^{h}\bigr)_{s,j+1}\bigr], \quad s = i, i+1, \tag{6}$$

$$Eu_{i+1/2,t} = u^{h/2}_{i+1/2,t} + \frac{1}{30}\bigl[\bigl(u^{h/2} - u^{h}\bigr)_{i,t} + \bigl(u^{h/2} - u^{h}\bigr)_{i+1,t}\bigr], \quad t = j, j+1, \tag{7}$$

$$\begin{aligned} Eu_{i+1/2,j+1/2} = u^{h/2}_{i+1/2,j+1/2} + \frac{1}{60}\bigl[&\bigl(u^{h/2} - u^{h}\bigr)_{i,j} + \bigl(u^{h/2} - u^{h}\bigr)_{i+1,j} + \bigl(u^{h/2} - u^{h}\bigr)_{i,j+1} \\ &+ \bigl(u^{h/2} - u^{h}\bigr)_{i+1,j+1}\bigr], \end{aligned} \tag{8}$$

in order to construct the corresponding extrapolation solution, respectively.

Taking the above four extrapolation formulas (5), (6), (7) and (8), we can obtain an extrapolation solution Eu, which is more accurate than the fourth-order accurate numerical solution $u^{h/2}$ on the fine grid $\Omega_{h/2}$. For simplicity, we denote this procedure from u^h, $u^{h/2}$ to Eu as an operator, we call it as fine mesh Richardson extrapolation (*FMRE*) interpolator

$$Eu \Leftarrow FMRE\bigl(u^h, u^{h/2}\bigr). \tag{9}$$

Remark 4.1 It should be noted that the extrapolation interpolator *FMRE* is a local operation, which can be done very effectively.

4.3 Computational work

Throughout this paper, when using Algorithm 1 as a solver on each grid of the EXFMG method, we always set $N_{\text{cycles}} = 1$. Now we turn to estimate the computational cost of the EXFMG method in terms of *work units* (WUs). Roughly speaking, "1 WU" is the computational cost of performing one relaxation sweep on the finest grid. For simplicity, we neglect the amount of work needed to solve the equations on the coarsest grid and the transfer operators (include interpolation and restriction) between different levels of grids since it normally counts 10–20% of the total cost of the algorithm.

Assume that the m_j is the number of the MG cycles required on the grid Ω_j when the MG solution satisfies the stopping criterion (see line 4 in Algorithm 3). Based on the above conditions, we can estimate the computational work of EXFMG with L levels of grids for d-dimensional problems as follows:

$$\begin{aligned} WU_{\text{EXFMG}} &\approx (\nu_1 + \nu_2)\bigl[m_L + (m_L + m_{L-1})2^{-d} + \cdots + (m_L + m_{L-1} + \cdots + m_2)2^{-(L-2)d}\bigr] \\ &= (\nu_1 + \nu_2)\sum_{j=2}^{L}\sum_{i=j}^{L} m_i 2^{-d(L-j)} WU. \end{aligned} \tag{10}$$

5 Numerical experiments

In this section, we conduct numerical experiments with two examples both on the unit square domain $\Omega = (0,1) \times (0,1)$. In the two test problems, the right-hand side function $f(x,y)$ and the Dirichlet boundary conditions on $\partial\Omega$ are prescribed to satisfy the given analytic solutions $u(x,y)$. Our codes are written in Matlab and the programs are carried out on a desktop with Inter (R) Core(TM) i5-6200 U CPU (2.30 GHZ, 2.40 GHZ) and 4 GB RAM.

For comparison purposes, we denote the Algorithm 1 with $N_{\text{cycles}} = 1$ as *MG*. The iterative procedure of *MG* method starts with zero initial guesses and terminates when the approximation solution u_L^* on the finest grid satisfies the same stopping criterion $\|F_L - A_L u_L^*\| \le \epsilon_L \|F_L\|$ as the EXFMG algorithm.

In MG and EXFMG methods with L embedded grids, the standard bilinear interpolation is used to transfer corrections from the coarse grid to the fine grid, the full-weighting scheme is employed to project residual from the fine grid to the coarse grid, the Conjugate Gradient (CG) method is chosen as a smoother to eliminate the high frequency error, and the number of pre-smoothing and post-smoothing steps are set to be $\nu_1 = \nu_2 = 2$. $\epsilon_L = 10^{-10}$ is taken as the stopping tolerance on the finest grid.

In all numerical experiments, we use the notations u, $\bar{u}_j^*$ and u_j^* to denote the exact solution, numerical solution and extrapolation solution on the grid Ω_j, respectively. For convenience of comparison, we choose the L^∞-norm and the L^2-norm to measure the error between u_j^* (or $\bar{u}_j^*$) and u, which are defined as

$$\left\|u - u_j^*\right\|_\infty = \|e_j\|_\infty = \max_{(x,y)\in\Omega_j} \left|e_j(x,y)\right|,$$

$$\left\|u - u_j^*\right\|_2 = \|e_j\|_2 = h_j \sqrt{\sum_{(x,y)\in\Omega_j} \left|e_j(x,y)\right|^2}.$$

We also apply the definition

$$\text{Order} = \frac{\log(\|e_{j-1}\|/\|e_j\|)}{\log(h_{j-1}/h_j)}, \quad j = 2, \cdots, L,$$

to represent the convergence order of the approximate solution. Here, $\|\cdot\|$ is a given norm (such as L^∞, L^2). Besides, we use "CPU" , "WU" and "Tflops" to record the algorithm's computational time in seconds, work units [8] and number of flops measured in *TeraFlops*, respectively. "Iter" denotes the number of multigrid cycles required.

Example 5.1 ([41, 42]) Consider the following differential equation:

$$-u_{xx}(x,y) - u_{yy}(x,y) + \frac{1}{\varepsilon(1+y)} u_y(x,y) = f(x,y), \tag{11}$$

with suitable Dirichlet boundary condition and right-hand side term $f(x,y)$, such that the exact solution is

$$u(x,y) = \exp(y-x) + 2^{-1/\varepsilon}(1+y)^{1+1/\varepsilon}. \tag{12}$$

Here, $\varepsilon = 0.01$.

Table 1 Errors and costs of algorithms with $L = 5$ for different discretization system arising from Example 5.1

Ω_L	Method	$\|u - u_L^*\|_\infty$	$\|u - u_L^*\|_2$	CPU	Iter	WU	Tflops
256 × 256	MG	6.30E–06	2.41E–06	0.92	9	47.8	1.5
	EXFMG	1.23E–07	2.17E–08	0.91	6, 8, 10, 1	45.6	1.1
512 × 512	MG	3.94E–07	1.51E–07	3.42	9	47.8	23.5
	EXFMG	2.05E–09	3.41E–10	3.05	6, 7, 9, 1	43.9	16.8
1024 × 1024	MG	2.47E–08	9.45E–09	14.3	9	47.8	377.9
	EXFMG	7.10E–11	6.33E–12	12.1	5, 7, 8, 1	38.3	229.3
2048 × 2048	MG	1.54E–09	5.91E–10	43.4	10	53.1	6736.5
	EXFMG	7.34E–12	8.68E–13	40.5	5, 7, 8, 1	38.3	3679.8

Table 2 Errors and costs of algorithms with $L = 5$ for different discretization system arising from Example 5.2

Ω_L	Method	$\|u - u_L^*\|_\infty$	$\|u - u_L^*\|_2$	CPU	Iter	WU	Tflops
256 × 256	MG	3.15E–05	3.46E–06	0.92	8	42.5	1.3
	EXFMG	3.02E–06	2.84E–07	1.21	7, 9, 11, 1	52.5	1.2
512 × 512	MG	1.94E–06	2.13E–07	4.34	9	47.8	23.5
	EXFMG	6.18E–08	4.40E–09	3.95	5, 8, 10, 1	40.3	14.4
1024 × 1024	MG	1.21E–07	1.33E–08	14.95	8	42.5	335.9
	EXFMG	1.78E–09	3.38E–10	12.47	4, 6, 9 , 1	32.0	184.8
2048 × 2048	MG	7.53E–09	8.35E–10	40.64	8	42.5	5389.2
	EXFMG	3.65E–10	8.29E–11	41.84	4, 5, 7, 1	30.1	2920.2

Table 3 Comparison of the L^∞ errors and convergence order on different grid level of MG and EXFMG methods for Example 5.1

j	Ω_j	MG		EXFMG			
		$\|u - u_j^*\|_\infty$	Order	$\|u - \bar{u}_j^*\|_\infty$	Order	$\|u - u_j^*\|_\infty$	Order
2	64 × 64	1.56E–03	–	1.56E–03	–	3.44E–04	–
3	128 × 128	1.00E–04	3.96	1.00E–04	3.96	6.96E–06	5.63
4	256 × 256	6.30E–06	3.99	6.30E–06	3.99	1.23E–07	5.83
5	512 × 512	3.94E–07	4.00	3.94E–07	4.00	2.04E–09	5.91
6	1024 × 1024	2.47E–08	4.00	2.47E–08	4.00	7.10E–11	4.84

Example 5.2 ([40]) Consider the convection–diffusion equation:

$$-u_{xx}(x,y) - u_{yy}(x,y) + x(x-1)(1-2y)u_x(x,y) + y(y-1)(1-2x)u_y(x,y) = f(x,y), \qquad (13)$$

with the analytic solution

$$u(x,y) = \exp\bigl(-2000(x-0.5)^2 - y^2\bigr). \qquad (14)$$

The numerical results for Example 5.1 and 5.2 obtained by MG and EXFMG methods with five embedded grids are listed in Tables 1 and 2. When fixing 32 × 32 as the coarsest grid Ω_1, Tables 3–8 list the numerical results of EXFMG method with the stopping criteria

$$\epsilon_j = 10^{j-L-10}, \quad j = 2, 3, \ldots, L, \qquad (15)$$

Table 4 Comparison of the L^∞ errors and convergence order on different grid level of MG and EXFMG methods for Example 5.2

j	Ω_j	MG		EXFMG			
		$\|u-u_j^*\|_\infty$	Order	$\|u-\bar{u}_j^*\|_\infty$	Order	$\|u-u_j^*\|_\infty$	Order
2	64 × 64	1.16E–02	–	1.16E–02	–	4.14E–01	–
3	128 × 128	5.36E–04	4.43	5.36E–04	4.43	2.01E–04	11.01
4	256 × 256	3.15E–05	4.09	3.15E–05	4.09	3.02E–06	6.06
5	512 × 512	1.94E–06	4.02	1.94E–06	4.02	6.18E–08	5.61
6	1024 × 1024	1.21E–07	4.01	1.21E–07	4.01	1.78E–09	5.12

Table 5 Comparison of the L^2 errors and convergence order on different grid level of MG and EXFMG methods for Example 5.1

j	Ω_j	MG		EXFMG			
		$\|u-u_j^*\|_\infty$	Order	$\|u-\bar{u}_j^*\|_\infty$	Order	$\|u-u_j^*\|_\infty$	Order
2	64 × 64	5.97E–04	–	5.97E–04	–	7.81E–05	–
3	128 × 128	3.84E–05	3.96	3.84E–05	3.96	1.35E–06	5.85
4	256 × 256	2.41E–06	3.99	2.41E–06	3.99	2.17E–08	5.96
5	512 × 512	1.51E–07	4.00	1.51E–07	4.00	3.41E–10	5.99
6	1024 × 1024	9.45E–09	4.00	9.45E–09	4.00	6.27E–12	5.77

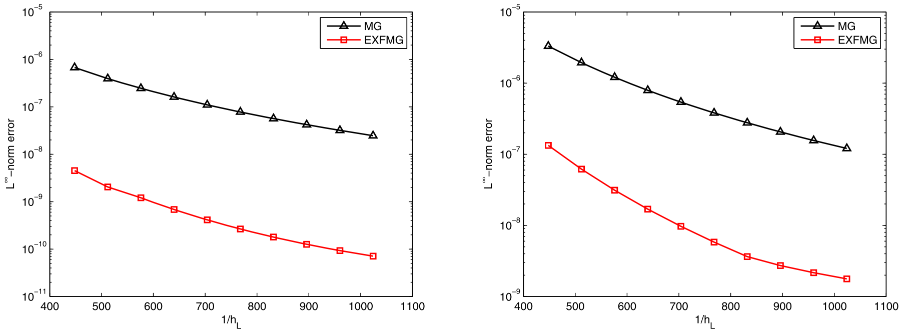
Table 6 Comparison of the L^2 errors and convergence order on different grid level of MG and EXFMG methods for Example 5.2

j	Ω_j	MG		EXFMG			
		$\|u-u_j^*\|_\infty$	Order	$\|u-\bar{u}_j^*\|_\infty$	Order	$\|u-u_j^*\|_\infty$	Order
2	64 × 64	1.37E–03	–	1.37E–03	–	1.59E–01	–
3	128 × 128	5.85E–05	4.55	5.85E–05	4.55	2.23E–05	12.80
4	256 × 256	3.46E–06	4.08	3.46E–06	4.08	2.84E–07	6.29
5	512 × 512	2.13E–07	4.02	2.13E–07	4.02	4.40E–09	6.01
6	1024 × 1024	1.33E–08	4.00	1.33E–08	4.00	3.38E–10	3.70

Table 7 Comparison of the iteration number and computational cost (WU) on different grid level of MG and EXFMG methods for Example 5.1

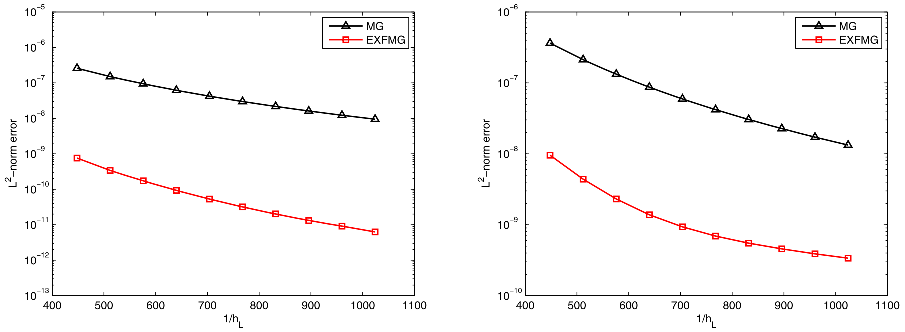
j	Ω_j	MG		EXFMG	
		Iter	WU	Iter	WU
2	64 × 64	1	0.02	1	0.02
3	128 × 128	9	0.70	10	0.78
4	256 × 256	9	2.95	8	2.63
5	512 × 512	9	11.95	7	9.30
6	1024 × 1024	9	47.95	5	26.64

on each grid Ω_j, $j = 2, 3, \dots, L$. We also present the numerical errors and computational costs (time and WU) in Figs. 2, 3, 4 and 5, respectively. Now we discuss those results in the following paragraphs.



(a) Example 5.1 (b) Example 5.2

Figure 2 Comparison of the L^∞-norm error $\|u - u_L^*\|_\infty$ with different mesh size h_L ($L = 6$)



(a) Example 5.1 (b) Example 5.2

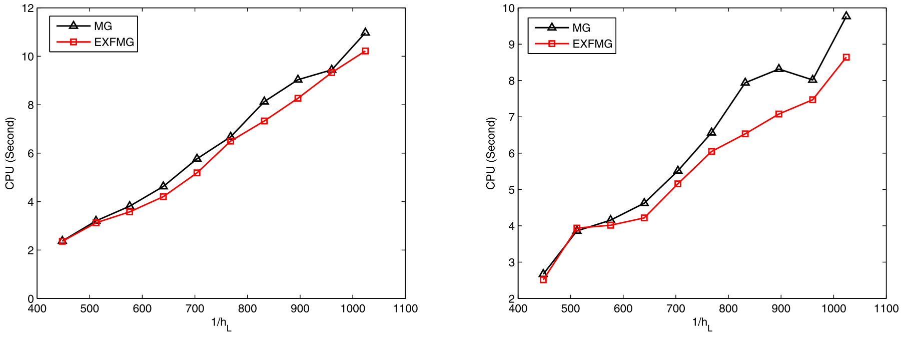
Figure 3 Comparison of the L^2-norm error $\|u - u_L^*\|_2$ with different mesh size h_L ($L = 6$)

Table 8 Comparison of the iteration number and computational cost (WU) on different grid level of MG and EXFMG methods for Example 5.2

j	Ω_j	MG		EXFMG	
		Iter	WU	Iter	WU
2	64 × 64	1	0.02	1	0.02
3	128 × 128	9	0.70	11	0.86
4	256 × 256	8	2.63	9	2.95
5	512 × 512	9	11.95	6	7.97
6	1024 × 1024	8	42.63	4	21.31

For the accuracy, we find that the numerical solutions from the EXFMG method are much more accurate than those from the MG method (see Tables 1 and 2, Figs. 2 and 3). This means that, to achieve the same accuracy, fewer grid points are needed for the EXFMG method than for the MG method. For instance, the numerical solution by the EXFMG method on the 256 × 256 grid is comparable to the numerical solution obtained by the MG method on the 512 × 512 grid in Tables 1 and 2.

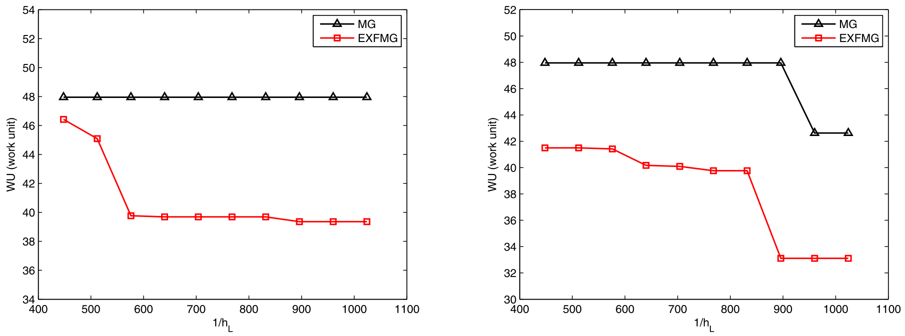
From Tables 3, 4, 5 and 6, we can see that the numerical solution $\bar{u}_j^*$ of the EXFMG method as well as MG method, converges with fourth-order in both L^∞-norm and L^2-



(a) Example 5.1

(b) Example 5.2

Figure 4 Comparison of the computational time with different mesh size h_L ($L = 6$)



(a) Example 5.1

(b) Example 5.2

Figure 5 Comparison of the WU of MG and EXFMG methods with $L = 6$

norm while the extrapolated solution u_j^* achieves sixth-order accuracy on most levels of grids and starts to lose convergent order on fine grids since the requirement to achieve sixth-order accuracy on fine grids is much higher.

Tables 7 and 8 list computational costs on each level of grids for the EXFMG method and MG methods, we find that WU as well as the total work cost of EXFMG method depends greatly on the number of multigrid iterations on the finest grid Ω_L. From Tables 1 and 2, we noticed that the EXFMG method takes fewer iterations on the finest grid, which is more efficient than the MG method. Therefore, the total computational cost (WU, Tflop and CPU time) of EXFMG method are much lower than that of the MG method, which is particularly true when mesh size is less or equal to $\frac{1}{512}$ (see Tables 1, 2 , 7, 8 and Figs. 4, 5 for details).

From the above discussion, we can see that the EXFMG method not only achieves high-order accuracy but also keeps a low computational cost. Hence, it is a cost-effective numerical solver.

6 Conclusion

We present an EXFMG method combined with a fourth-order compact difference scheme, a bi-quartic Lagrange interpolation and a simple extrapolation technique, to solve the 2D convection–diffusion equation. Numerical experiments were conducted for two test problems to demonstrate that the proposed EXFMG method improves the solution accuracy and keeps less computational costs, compared to the classical MG method.

We need point out that the fourth-order compact difference method, which is employed to discretize the differential equation, can be replaced by other numerical methods, such as the FE method and the finite volume method. Furthermore, we also plan to extend this study and apply EXFMG algorithm with sixth-order combined compact difference scheme [43–45] to solve unsteady three-dimensional convection–diffusion equations.

Acknowledgements
The research was supported by the National Natural Science Foundation of China (Nos. 41474103, 11661027, 11601012), the National Key Research and Development Program of China (Nos. 2017YFB0701700, 2017YFB0305601), the Excellent Youth Foundation of Hunan Province of China (No. 2018JJ1042), the Natural Science Foundation of Yunnan Province of China (No. 2017FH001-012), the Innovation-Driven Project of Central South University (No. 2018CX042) and the Reserve Talents Foundation of Honghe University (No. 2015HB0304). We are grateful to the two anonymous reviewers for their helpful comments.

Competing interests
The authors declare that they have no competing interests.

Authors' contributions
The authors declare that the work was realized in collaboration with the same responsibility. All authors read and approved the final manuscript.

Author details
[1]School of Mathematics and Statistics, Central South University, Changsha, P.R. China. [2]Department of Mathematics, Honghe University, Mengzi, P.R. China.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 11 January 2018 Accepted: 3 May 2018 Published online: 10 May 2018

References

1. Richardson, L.F.: The approximate solution of physical problems involving differential equations using finite differences, with an application to the stress in a masonry dam. Philos. Trans. R. Soc. Lond. Ser. A **210**, 307–357 (1910)
2. Richardson, L.F.: The deferred approach to the limit. I: the single lattice. Philos. Trans. R. Soc. Lond. Ser. A **226**, 299–349 (1927)
3. Marchuk, G.I., Shaidurov, V.V.: Difference Methods and Their Extrapolations. Springer, New York (1983)
4. Blum, H., Lin, Q., Rannacher, R.: Asymptotic error expansion and Richardson extrapolation for linear finite elements. Numer. Math. **49**(1), 11–37 (1986)
5. Chen, C.M., Lin, Q.: Extrapolation of finite element approximation in a rectangular domain. J. Comput. Math. **7**(3), 227–233 (1989)
6. Fairweather, G., Lin, Q., Lin, Y.P., et al.: Asymptotic expansions and Richardson extrapolation of approximate solutions for second order elliptic problems on rectangular domains by mixed finite element methods. SIAM J. Numer. Anal. **44**(3), 1122–1149 (2006)
7. Lin, Q., Lu, T., Shen, S.M.: Maximum norm estimate, extrapolation and optimal point of stresses for finite element methods on strogle regular triangulation. J. Comput. Math. **1**(4), 376–383 (1983)
8. Trottenberg, U., Oosterlee, C.W., Schller, A.: Multigrid. Academic Press, London (2001)
9. Hackbusch, W.: Multi-Grid Methods and Applications. Springer, Berlin (1985)
10. Othman, M., Abdullah, A.R.: An efficient multigrid Poisson solver. Int. J. Comput. Math. **71**(4), 541–553 (1999)
11. Ge, Y.B.: Multigrid method and fourth-order compact difference discretization scheme with unequal meshsizes for 3D poissson equation. J. Comput. Phys. **229**(18), 6381–6391 (2010)
12. Gupta, M.M., Kouatchou, J., Zhang, J.: Comparison of second and fourth order discretization for multigrid Poisson solver. J. Comput. Phys. **132**, 663–674 (1997)
13. Shi, Z.C., Xu, X.J., Huang, Y.Q.: Economical cascadic multigrid methods (ECMG). Sci. China Ser. A, Math. **50**, 1765–1780 (2007)
14. Li, C.L.: A new parallel cascadic multigrid method. Appl. Math. Comput. **219**, 10150–10157 (2013)
15. 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)
16. 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)
17. Stolk, C.C., Ahmed, M., Bhowmik, S.K.: A multigrid method for the Helmholtz equation with optimized coarse grid corrections. SIAM J. Sci. Comput. **36**(6), A2819–A2841 (2013)
18. Ge, Y.B.: 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)

19. Gupta, M.M., Zhang, J.: High accuracy multigrid solution of the 3D convection-diffusion euqaiton. Appl. Math. Comput. **113**(2), 249–274 (2000)
20. Bhowmik, S.K., Stolk, C.C.: Preconditioners based on windowed Fourier frames applied to elliptic partial differential equations. J. Pseudo-Differ. Oper. Appl. **2**(3), 317–342 (2011)
21. Brandt, A.: Multi-level adaptive solutions to boundary-value problems. Math. Comput. **31**(138), 333–390 (1977)
22. Mandel, J., Parter, S.V.: On the multigrid F-cycle. Appl. Math. Comput. **37**(1), 19–36 (1990)
23. Thekale, A., Gradl, T., et al.: Optimizing the number of multigrid cycle in the full multigrid algorithm. Numer. Linear Algebra Appl. **17**(2–3), 199–210 (2010)
24. Chen, C.M., Hu, H.L., et al.: Analysis of extrapolation cascadic multigrid method (EXCMG). Sci. China Ser. A **51**, 1349–1360 (2008)
25. Chen, C.M., Shi, Z.C., Hu, H.L.: On extrapolation cascadic multigrid method. J. Comput. Math. **29**(6), 684–697 (2011)
26. Chen, C.M., Hu, H.L.: Extrapolation cascadic multigrid method on piecewise uniform grid. Sci. China Math. **56**(12), 2711–2722 (2013)
27. Hu, H.L., Chen, C.M., Pan, K.J.: Asymptotic expansions of finite element solutions to Robin problems in H^3 and their application in extrapolation cascadic multigrid method. Sci. China Math. **57**, 687–698 (2014)
28. Pan, K.J., He, D.D., Chen, C.M.: An extrapolation cascadic multigrid method for elliptic problems on reentrant domains. Adv. Appl. Math. Mech. **9**(6), 1347–1363 (2017)
29. Hu, H.L., Chen, C.M., Pan, K.J.: Time-extrapolation algorithm (TEA) for linear parabolic problems. J. Comput. Math. **32**, 183–194 (2014)
30. Pan, K.J., Tang, J.T., et al.: Extrapolation cascadic multigrid method for 2.5D direct current resistivity modeling. Chin. J. Geophys. **55**, 2769–2778 (2012) (in Chinese)
31. Pan, K.J., Tang, J.T.: 2.5-D and 3-D DC resistivity modelling using an extrapolation cascadic multigrid method. Geophys. J. Int. **197**(3), 1459–1470 (2014)
32. Pan, K.J., He, D.D., et al.: A new extrapolation cascadic multigrid method for three dimensional elliptic boundary value problems. J. Comput. Phys. **344**, 499–515 (2017)
33. 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)
34. 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)
35. Li, M., Li, C.L.: New cascadic multigrid methods for two-dimensional Poisson problem based on the fourth-order compact difference scheme. Math. Methods Appl. Sci. **41**(3), 920–928 (2018)
36. Hu, H.L., Ren, Z.Y., et al.: On the convergence of an extrapolation cascadic multigrid method for elliptic problems. Comput. Math. Appl. **74**, 759–771 (2017)
37. Sun, H., Zhang, J.: A high order finite difference discretization strategy based on extrapolation for convection-diffusion equations. Numer. Methods Partial Differ. Equ. **20**(1), 18–32 (2004)
38. Dai, R.X., Zhang, J., Wang, Y.: Higher order ADI method with completed Richardson extrapolation for solving unsteady convection-diffusion equations. Comput. Math. Appl. **71**, 431–442 (2016)
39. Wang, Y., Zhang, J.: Sixth order compact scheme combined with multigrid method and extrapolation technique for 2D Poisson equation. J. Comput. Phys. **228**, 137–146 (2009)
40. Zhang, J., Sun, H.W., Zhao, J.J.: High order compact scheme with multigrid local mesh refinement procedure for convection diffusion problems. Comput. Methods Appl. Mech. Eng. **191**, 4661–4674 (2002)
41. 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**, 4051–4070 (2011)
42. 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)
43. He, D.D., Pan, K.J.: An unconditionally stable linearized CCD-ADI method for generalized nonlinear Schrödinger equations with variable coefficients in two and three dimensions. Comput. Math. Appl. **73**, 2360–2374 (2017)
44. Lee, S.T., Liu, J., Sun, H.W.: Combined compact difference scheme for linear second-order partial differential equations with mixed derivative. J. Comput. Appl. Math. **264**, 23–37 (2014)
45. Wang, Q.H., Pan, K.J., Hu, H.L.: Unique solvability of the CCD scheme for convection–diffusion equations with variable convection coefficients. Adv. Differ. Equ. **2018**, 163 (2018)