An efficient multigrid solver for 3D biharmonic equation with a discretization by 25-point difference scheme

Kejia Pan\textsuperscript{a}, Dongdong He\textsuperscript{b,∗}, Runxin Ni\textsuperscript{a}

\textsuperscript{a}School of Mathematics and Statistics, Central South University, Changsha 410083, China
\textsuperscript{b}School of Aerospace Engineering and Applied Mechanics, Tongji University, Shanghai 200092, China

Abstract

In this paper, we propose an efficient extrapolation cascadic multigrid (EXCMG) method combined with 25-point difference approximation to solve the three-dimensional biharmonic equation. First, through applying Richardson extrapolation and quadratic interpolation on numerical solutions on current and previous grids, a third-order approximation to the finite difference solution can be obtained and used as the iterative initial guess on the next finer grid. Then we adopt the bi-conjugate gradient (Bi-CG) method to solve the large linear system resulting from the 25-point difference approximation. In addition, an extrapolation method based on midpoint extrapolation formula is used to achieve higher-order accuracy on the entire finest grid. Finally, some numerical experiments are performed to show that the EXCMG method is an efficient solver for the 3D biharmonic equation.

Keywords: Richardson extrapolation, multigrid method, biharmonic equation, quadratic interpolation, high efficiency

2000 MSC: 65N06, 65N55

1. Introduction

In this paper, we consider the following three-dimensional (3D) biharmonic equation

\[ \Delta^2 u(x, y, z) = f(x, y, z), \quad (x, y, z) \in \Omega = (0, 1)^3, \]  

with Dirichlet boundary conditions of first kind

\[ u(x, y, z) = g_1(x, y, z), \quad \frac{\partial u}{\partial n} = g_2(x, y, z), \quad (x, y, z) \in \partial \Omega, \]  

or Dirichlet boundary conditions of second kind

\[ u(x, y, z) = g_1(x, y, z), \quad \frac{\partial^2 u}{\partial n^2} = g_2(x, y, z), \quad (x, y, z) \in \partial \Omega. \]  

The biharmonic operator \( \Delta^2 \) in three-dimensional (3D) Cartesian coordinates can be written as

\[ \Delta^2 u(x, y, z) = \frac{\partial^4 u}{\partial x^4} + \frac{\partial^4 u}{\partial y^4} + 2 \frac{\partial^4 u}{\partial x^2 y^2} + 2 \frac{\partial^4 u}{\partial x^2 z^2} + 2 \frac{\partial^4 u}{\partial y^2 z^2}. \]  

And the two dimensional (2D) version of Eq. (1) is

\[ \frac{\partial^4 u}{\partial x^4} + \frac{\partial^4 u}{\partial y^4} + 2 \frac{\partial^4 u}{\partial x^2 y^2} = f(x, y). \]
The biharmonic equation is a fourth-order partial differential equation which arises in areas of continuum mechanics, including linear elasticity theory, phase-field models and Stokes flows. Due to the significance of the biharmonic equation, a large number of numerical methods for solving the biharmonic equations have been proposed [1–16]. Most of these works focus on two-dimensional case. There has been very little work devoted to solving the 3D biharmonic equations. The main reason is that 3D problems require large computational power and memory storage [8, 14].

Various methods for the numerical solutions of the biharmonic equations have been considered in the literature. A popular technique is to split $\Delta^2 u = f$ into two coupled Poisson equations for $u$ and $v$: $\Delta u = v$, $\Delta v = f$, each equation can be solved by using fast Poisson solvers. The coupled method has been widely used by many authors [2, 6, 7]. As it is mentioned in [2, 6, 7], the main difficulty for the coupled (splitting or mixed) method is that the boundary conditions for the newly introduced variable $v$ are undefined and needs to be approximated accurately, and the computational results strongly depends on the choice of the approximation of missing boundary values for $v$.

Another conventional approach for solving the 3D biharmonic equations is to directly discretize Eq. (1) on a uniform grid using a 25-point computational stencil with truncation error of order $h^2$, which is derived by Ribeiro Dos Santos [5] in 1967. This conventional 25-point difference approximation connects the value of $u$ at grid point $(x, y, z)$ in terms of 24 neighboring values in a $5 \times 5 \times 5$ cube. Thus, this direct method need to be modified at grid points near the boundaries. As mentioned in [1, 13, 14], there are serious computational difficulties with solution of the linear systems obtained by the 13-point discretization of the 2D biharmonic equation and the 25-point discretization of 3D biharmonic equation. Dehghan and Mohebbi [8] also pointed that this direct method can only be used for moderate values of grid width $h$ and the well-known iterative methods such Jacobi or Gauss-Seidel either converge very slowly or diverge.

The combined compact difference method is another popular method for solving the biharmonic equation [8, 14]. For example, Altas et al. [14] proposed a fourth-order, combined compact formulation, where the unknown solution and its first derivatives are carried as unknowns at grid point and computed simultaneously, for the 3D biharmonic equation with Dirichlet boundary conditions of first kind. In 2006, Dehghan et al. [8] proposed two combined compact difference schemes for solve 3D biharmonic equation with Dirichlet boundary conditions of second kind, which use the known solution and its second derivatives as unknowns. In these combined compact difference methods, there is no need to modify the difference scheme at grid points near the boundaries, and the given Dirichlet boundary conditions are exactly satisfied and no approximations need to be carried out at the boundaries, in contrary to the coupled method. However, these combined compact difference methods introduce extra amount of computation, and the classical iterations for solving the resulting linear system suffer from slow convergence. Multigrid methods give good results in [8] and [14]. However, numerical results in [8] and [14] are reported only up to $32 \times 32 \times 32$ and $64 \times 64 \times 64$ grids, respectively. To the best of our knowledge, there is no numerical results for solving the 3D biharmonic equations with large-scale discretized meshes.

In this paper, we propose an efficient extrapolation cascadic multigrid method based on the conventional 25-point approximation to solve 3D biharmonic equations with both first and second boundary conditions. In our method, the conventional 25-point difference scheme is used to approximate the 3D biharmonic equation (1). In order to overcome the serious computational difficulties with solution of the resulting linear system, by combining Richardson extrapolation and quadratic interpolation on numerical solutions on current and previous grids, we obtain quite good initial guess of the iterative solution on the next finer grid, and then adopt the bi-conjugate gradient (Bi-CG) method to solve the large linear system efficiently. Our method has been used to solve 3D biharmonic problems with more than 135 million unknowns with only several iterations.

The rest of the paper is organized as follows: Section 2 presents the 25-point difference approximation for the 3D biharmonic equation and its modification of the difference scheme at grid points near boundaries. Section 3 reviews the classical V-cycle and W-cycle multigrid methods. In Section 4, we present a new EXCMG method to solve the linear three-dimensional biharmonic equation (1). Section 5 describes the Bi-CG solver in our new EXCMG method. Section 6 provides the numerical results to demonstrate the high efficiency and accuracy of the proposed method, and conclusions are given in the final section.

2. Second-order Finite Difference Discretization

We consider a cubic domain $\Omega = [0, 1] \times [0, 1] \times [0, 1]$. Let $N = 1/h$ be the numbers of uniform intervals along all the $x$, $y$ and $z$ directions. We discretize the domain with unequal meshesizes $h = 1/N$ in all $x$, $y$ and $z$ coordinate.
directions. The grid points are \((x_i, y_j, z_k)\), with \(x_i = ih, y_j = jh\) and \(z_k = kh, i, j, k = 0, 1, \cdots, N\). The quantity \(u_{i,j,k}\) represents the numerical solution at \((x_i, y_j, z_k)\).

Then the value on the boundary points \(u_{i,j,k}\) can be evaluated directly from the Dirichlet boundary condition. For internal grid points \((i = 2, \cdots, N - 2, j = 2, \cdots, N - 2, k = 2, \cdots, N - 2)\), the 25-point second-order difference scheme for 3D biharmonic equation was derived [5, 14]:

\[
42u_{i,j,k} - 12(u_{i-1,j,k} + u_{i+1,j,k} + u_{i,j-1,k} + u_{i,j+1,k} + u_{i,j,k-1} + u_{i,j,k+1}) \\
+ u_{i-2,j,k} + u_{i+2,j,k} + u_{i,j-2,k} + u_{i,j+2,k} + u_{i,j,k-2} + u_{i,j,k+2} \\
+ 2(u_{i-1,j-1,k} + u_{i-1,j+1,k} + u_{i+1,j-1,k} + u_{i+1,j+1,k} + u_{i,j-1,k-1} + u_{i,j-1,k+1} + u_{i,j+1,k-1} + u_{i,j+1,k+1}) \\
+ u_{i-1,j,k+1} + u_{i+1,j,k+1} = h^4 f_{i,j,k}. 
\]

(6)

Note that \(u_{i,j,k}\) is connected to grid points two grids away in each direction from the point \((x_i, y_j, z_k)\). Thus, the above difference formulation (6) for the grid points near the domain boundary \(\partial \Omega\) involves at least one value of point outside the domain, and these points outside the domain are fictitious points which need to be replaced by the internal points through the boundary condition. These could be done for both first and second kind of boundary conditions.

For the first kind of boundary condition. For example, for \(i = 1\), the point \((x_{i-2}, y_j, z_k)\) lies outside the computational domain, and the value on the fictitious point \((x_{-1}, y_j, z_k)\) can be obtained through the following central difference formula called the reflection formulas [3, 4]:

\[
\frac{u_{1,j,k} - u_{-1,j,k}}{2h} = \frac{\partial u}{\partial x}_{0,j,k},
\]

(7)

where \((\frac{\partial u}{\partial x})_{0,j,k}\) can be obtained from the boundary condition (2) and \(u_{-1,j,k}\) is given by

\[
u_{-1,j,k} = u_{1,j,k} - 2h \left(\frac{\partial u}{\partial x}\right)_{0,j,k}.
\]

(8)

For the second kind of boundary condition. For example, for \(i = 1\), the point \((x_{i-2}, y_j, z_k)\) lies outside the computational domain, and the value on the fictitious point \((x_{-1}, y_j, z_k)\) can also be obtained through the following central difference formula called the reflection formulas:

\[
\frac{u_{1,j,k} - 2u_{0,j,k} + u_{-1,j,k}}{h^2} = \frac{\partial^2 u}{\partial x^2} 0,j,k, 
\]

(9)

where \(u_{0,j,k}\) and \((\frac{\partial u}{\partial x})_{0,j,k}\) can be obtained from the boundary condition (2) and \(u_{-1,j,k}\) is given by

\[
u_{-1,j,k} = -u_{1,j,k} + 2u_{0,j,k} + h^2 \left(\frac{\partial^2 u}{\partial x^2}\right)_{0,j,k}. 
\]

(10)

We use \(u_h\) and \(u_{h/2}\) to represent the finite difference solutions of equation (1) with mesh sizes \(h\) and \(\frac{h}{2}\) respectively. Afterward, a matrix form, which express the finite difference scheme (6) and an equation set including formulas of the grid points near the boundary, can be obtained as below:

\[
A_n u_0 = f_n, 
\]

(11)

Where \(A_n\) is not a symmetry positive definite matrix, and the right hand-side vector of (6) and an equation set including the formulas of the grid points near the boundary are expressed by \(f_n\).

Note that the discretization equations for grid points that away from the boundary and that near the boundary are different, one must distinguish all possible cases. Although there are a little bit troublesome to treat all cases (there are totally 27 cases with 27 different equations), by moving the known boundary values into the right hand-side of the system, it is convenient to solve these equations which only involves unknown on the grid points.
3. Classical Multigrid Method

Since the 1970s’, many scholars have done researches on the classical multigrid method. Through deep researches on it for about fifty years, the classical multigrid method gradually forms its own comprehensive system. Including the interpolation, restriction and iteration, the classical multigrid method starts from the fine grid, goes to coarse grid and then returns to the fine grid. The classical multigrid methods contain V-cycle and W-cycle.

The classical multigrid method is introduced in detail with several steps. First, the specific smoother is used to smooth the current approximation on the fine grid. To obtain more oscillatory error components, we compute the residual and transfer it to the coarser grid with restriction. Next, we solve the residual equation on the coarser grid with the application of the number (γ) of cycles. From the fine grid to the coarsest grid and back to the fine grid is called a cycle. Then, we acquire the improved approximation on the fine grid by interpolating the correction back to the fine grid. Finally, we smooth the obtained approximation on the fine grid with the smoother again. If γ=1, call it V-cycle. And if γ=2, call it W-cycle. We take the four-level structures of V-cycle and W-cycle in Fig. 1 for instances to illustrate that.

Remark 1. When the γ-cycle is performed on the coarsest grid, direct solver is used to solve the residual equation.

4. Extrapolation Cascadic Multigrid Methods

It is an important issue to find approaches to solving the linear equation with enormous unknowns, which is obtained by FE and FD discretizations. Therefore, many authors paid great attention on it and presented multigrid methods including the MG method, the CMG method and the EXCMG method. The MG method has had a nearly integrated system through many scholars’ hard work in the past several decades. However, its algorithm is complex. Then the CMG method proposed by Deuflhard and Bornemann in [17] only use the interpolation and iteration so that its algorithm which is easy to operate is appealing. Furthermore, in 2008, the EXCMG method was proposed by Chen et al. [26] and the cores of it are Richardson extrapolation and quadratic interpolation. Compared with the CMG method, the EXCMG method provides a much better initial guess for the iteration solution on the next finer grid. In this section, we propose a new EXCMG method combined with the second-order incompact FD discretization for solving the linear three-dimensional biharmonic equation.

4.1. Description of the EXCMG Algorithm

In Algorithm 1, H is the size of the coarsest grid. L, the positive integer, denotes the total number of grids except first two embedded grids and indicates that the finest grids’ size is $\frac{H}{2^L}$. For the sizes of first two coarse grids are small, DSOLVE, a direct solver, is applied on the first two coarse grids (see line 1-2 in the Algorithm 1). In addition, procedure EXP_finite($u_{2h}, u_{4h}$) represents the third-order approximation of the FD solution $u_h$ which is obtained by Richardson extrapolation and quadratic interpolation from numerical solutions $u_{2h}$ and $u_{4h}$. Meanwhile,
a selective step is presented in the Algorithm 1 above where $\text{EXP}_\text{true}(u_h, u_{2h})$ refers to a higher-order solution which is extrapolated on the finest grid with the mesh size, $\bar{h}$, from two second-order numerical solutions $u_h$ and $u_{2h}$.

The details of the procedure of extrapolation and quadratic interpolation are introduced next subsection 4.2. The difference between our new EXCMG method and existing EXCMG method are discussed below:

(1) Instead of applying the second-order linear FE method, a second-order incompact difference scheme is used to discretize the 3D biharmonic equation in our new EXCMG method.

(2) Rather than perform the fixed number of iterations used in the existing EXCMG method, we introduce a relative residual tolerance $\epsilon_h$ into the Bi-CG solver (see line 7 in the Algorithm 1), which enables us to avoid the difficulty of determining the number of iterations at every grid level and obtain numerical solutions with desired accuracy conveniently.

(3) In our new EXCMG method, we take the Bi-CG solver as smoother instead of the CG solver (see line 8 in Algorithm 1). The Bi-CG is more suitable for positive definite matrix which is not symmetric compared with the CG solver.

(4) Through $\text{EXP}_\text{true}(u_h, u_{2h})$, a higher-order extrapolated solution $\tilde{u}_h$ is obtained easily, which improves the accuracy of the numerical solution $u_h$ (see line 11 in Algorithm 1).

### 4.2. Extrapolation and Quadratic Interpolation

The Richardson extrapolation is a well-known method for producing more accurate solutions of many problems in numerical analysis. Marchuk and Shaidurov [21] researched the application of the Richardson extrapolation on the FD method systematically in 1983. Since then, this technique has been well demonstrated in the frame of the FE and FD methods [18, 20–25, 33–36].

In next three subsections, we will give the explanation for how to obtain higher-order accuracy solution on the fine grid. Moreover, how to acquire a third-order approximation of the second-order FD method on the next finer grid is illustrated as well. Meanwhile, we can regard it as another critical application of the extrapolation method which produces good initial guesses for iterative solutions.

#### 4.2.1. Extrapolation for the True Solution

For simplicity, we first consider the three-level of embedded grids $Z_i$($i=0, 1, 2$) with mesh sizes $\bar{h}_i = h_{0}/2^i$ in one dimension. In addition, let $\varepsilon' = u' - u$ be the error of the second-order incompact FD solution $u'$ with mesh size $\bar{h}_i$. We make an assumption that the error at the node has the following form:

$$\varepsilon'(x_{i}) = A(x_{i}) h_{i}^{2} + O(h_{i}^{4})$$

where $A(x)$ is a properly smooth function independent of $h_{i}$. We will verify the error expansion (12) by numerical results in Sect. 5.

Through the equation (12), the Richardson extrapolation formula at the coarse grid point is obtained.
\[ \tilde{u}_k^j := \frac{4u_k^j - u_k^0}{3} = u(x_k) + O(h_0^4), \quad k = j, j + 1. \]  

Then, a midpoint extrapolation formula is obtained by linear interpolation

\[ \tilde{u}_{j+1/2}^j := u_{j+1/2}^j + \frac{1}{6}(u_j^1 - u_j^0 + u_{j+1}^1 - u_{j+1}^0) = u(x_{j+1/2}) + O(h_0^4), \]  

whose accuracy is fourth-order at fine grid points.

From equation (12), it is easy to obtain

\[ A(x_k) = \frac{4}{3h_0^2}(u_k^0 - u_k^1) + O(h_0^4), \quad k = j, j + 1. \]  

Through the error estimate of the linear interpolation

\[ A(x_{j+1/2}) = \frac{1}{2}(A(x_j) + A(x_{j+1})) + O(h_0^2). \]  

and substituting equation (15) to equation (16), yield

\[ A(x_{j+1/2}) = \frac{2}{3h_0^2}(u_j^0 - u_j^1) + \frac{2}{3h_0^2}(u_{j+1}^0 - u_{j+1}^1) + O(h_0^2). \]  

Since

\[ u_{j+1/2}^1 = u(x_{j+1/2}) + \frac{1}{4}A(x_{j+1/2})h_0^2 + O(h_0^4). \]  

The midpoint extrapolation formula (14) is obtained by the use of equation (17).

4.2.2. Extrapolation for the FD Solution

In this subsection, given solutions \( u^0 \) and \( u^1 \) of the second-order FD method, we will explain how to construct a third-order approximation \( w^2 \) of the FD solution \( u^2 \) by using extrapolation and interpolation methods in detail.

We divide the coarse element \((x_j, x_{j+1})\) into four uniform elements by adding one midpoint and two four equal points which are on the left side and right side of the midpoint. As a result, a set which contains five points and belongs to fine mesh \( Z_2 \) is obtained

\[ \{x_j, x_{j+1/4}, x_{j+1/2}, x_{j+3/4}, x_{j+1}\} \]

To acquire the more accurate approximation of FD solution \( u^2 \), the given solutions \( u^0 \) and \( u^1 \) are combined linearly. Therefore, here assume the existence of a constant \( c \) such that

---

Figure 2: Three embedded mesh in 1D.
\[ cu^1 + (1 - c)u^0 = u^2 + O(h^4_0), \]  

(19)

For obtaining the value of the constant \( c \), we substitute the error expansion (12) into (19) and obtain \( c = 5/4 \). Afterward, we obtain formulas of node extrapolation at points \( x_j \) and \( x_{j+1} \).

\[ w^2_k := \frac{5u^1_k - u^0_k}{4} = u^2_k + O(h^4_0), \quad k = j, j+1, \]  

(20)

Next, derive the midpoint \( x_{j+1/2} \)'s extrapolation formula. First, use the error expansion (12) again and obtain the formula below

\[ u^*_{j+1/2} = u^*_{j+1/2} - \frac{3}{16}A(x_{j+1/2})h^2_0 + O(h^4_0), \]  

(21)

Then substitute the (17) to (21) for eliminating the unknown \( A(x_{j+1/2}) \) and the following extrapolation formula of \( x_{j+1/2} \) is yielded

\[ w^2_{j+1/2} := u^1_{j+1/2} + \frac{1}{8}(u^1_j - u^1_{j+1} + u^0_j - u^0_{j+1}) = u^2_{j+1/2} + O(h^4_0). \]  

(22)

Finally, since the values of three points \( w^2_j, w^0_{j+1/2}, \) and \( u^0_{j+1/2} \) have been obtained, we can derive the extrapolation formulas at the points \( x^1_{j+1/4} \) and \( x^1_{j+3/4} \) blow through the use of the quadratic interpolation method

\[ w^2_{j+1/4} := \frac{1}{16}[9u^1_j + 12u^1_{j+1/2} - u^1_{j+1}] - (3u^0_j + u^0_{j+1})], \]  

(23)

\[ w^2_{j+3/4} := \frac{1}{16}[9u^1_{j+1} + 12u^1_{j+1/2} - u^1_{j+1}] - (3u^0_{j+1} + u^0_j)]. \]  

(24)

From the polynomial interpolation’s theory, it is easy to demonstrate that the third-order approximation of the FD solution can be presented by formulas (25) and (26). i.e.,

\[ w^2_{j+1/4} = u^2_{j+1/4} + O(h^4_0), \]  

(25)

\[ w^2_{j+3/4} = u^2_{j+3/4} + O(h^4_0), \]  

(26)

\[ \text{Figure 3: Three embedded hexahedral mesh} \]
4.2.3. Application of Extrapolation and Quartic Interpolation on Three-Dimension

In this subsection, how to acquire an accurate third-order approximation \( v^2 \) of the FD solution \( u^2 \) will be explained for embedded cubic mesh as shown in Fig.3. The specific steps of the construction of the third-order approximation \( u^2 \) are illustrated below:

**Step 1**: Corner nodes (1, 5, 21, 25, 101, 105, 121, 125): Use the extrapolation formula (20) to obtain the approximations of the values of 8 corner nodes on interpolation cells.

**Step 2**: Midpoints of edges (3, 11, 15, 23, 51, 55, 71, 75, 103, 111, 115, 123): Use the midpoint extrapolation formula (22) to obtain the approximations of the values of 19 midpoints of edges on interpolation cells.

**Step 3**: Centers of faces (13, 53, 65, 61, 77, 113): View the center of each face as the midpoint of the two face diagonals on interpolation cells. To obtain the approximations of the values of them, use the midpoint extrapolation formula (22) obtaining two approximations, calculate the arithmetic mean of the two obtained approximations and treat it as the approximate value at the center of each face.

**Step 4**: Center of the hexahedral element (63): View the center of the hexahedral element as the midpoint of the four space diagonals on interpolation cells. To obtain the approximation of the value of it, use the midpoint extrapolation formula (22) obtaining four approximations, calculate the arithmetic mean of the four obtained approximations and treat it as the approximate value at the hexahedral element.

**Step 5**: Other 98 fine grid points: The approximations of remaining 98 (5^3 - 27) grid points can be obtained by using tri-quadratic interpolation with the known 27-node (8 corner nodes, 12 midpoints of edges, 6 centers of faces and 1 center of the hexahedral element) values.

The tri-quartic interpolation formula at natural coordinates \((\xi, \eta, \zeta)\) is defined as

\[
w^2(\xi, \eta, \zeta) = \sum_i N_i(\xi, \eta, \zeta)w^2_i, \tag{27}\]

where the shape functions \(N_i\) can be written below:

\[
N_i(\xi, \eta, \zeta) = l^2_i(\xi)l^2_i(\eta)l^2_i(\zeta) \tag{28}\]

where \(l_i^2(\xi) \) (0 ≤ \(i\) ≤ 2) are the Lagrange basis polynomials of degree 2 which are defined as

\[
l^2_i(\xi) = \prod_{k=0,\xi_i\neq\xi_k}^{2} \frac{\xi - \xi_k}{\xi_i - \xi_k} \tag{29}\]

and \((\xi, \eta, \zeta)\) is the natural coordinate of node \(i \) (1 ≤ \(i\) ≤ 27).

5. Bi-Conjugate Gradient Method

The Bi-Conjugate Gradient (Bi-CG) method is an algorithm which is focus on solving linear equation systems

\[
Ax = b \tag{30}\]

Compared with the Conjugate Gradient (CG) method which needs matrix \(A\) to be self-joint, the Bi-CG method does not require matrix \(A\) self-joint but require it to multiply conjugate transpose \(A^*\). In addition, the Bi-CG method replaces the residual’s orthogonal sequence in the CG method with two sequences which are mutually orthogonal. In the Bi-CG method, the residual \(r_j\) is orthogonal with a set of vectors \(\tilde{r}_0, \tilde{r}_1, ..., \tilde{r}_{j-1}\) and \(\tilde{r}_j\) is also orthogonal with \(r_0, r_1, ..., r_{j-1}\). These relationships can be achieved by two three-term recurrence relations of vectors \([\tilde{r}_j]\) and \([r_j]\). Meanwhile, the Bi-CG method terminates within at most \(n\) steps when \(A\) is an \(n\) by \(n\) matrix. The preconditioned version and the unpreconditioned version of algorithms of the Bi-CG method are described as follows:

In the Bi-CG Method with the preconditioner algorithm below, \(x^*_k\) is adjoint, \(\bar{a}\) is the complex conjugate and the calculated \(r_k\) and \(r^*_k\) satisfy the following equations respectively

\[
r_k = b - Ax_k \tag{31}\]

\[
r^*_k = b^* - x^*_kA \tag{32}\]

In this paper, we adopt the Bi-CG method with the preconditioner as the iteration solver in our new EXCMG method.
19 numerical experiments are performed with EXCMG on the finest grid. We use the same notations in all tables. Let the amount of computational cost of the extrapolated solution $u_h$ on the finest grid, and the amount of computational cost of the EXCMG method in terms of a work unit (WU) on the finest grid, which is defined as the total computation required to perform one relaxation sweep on the finest grid that we only need to perform one iteration on the finest grid. The number of iteration is reduced significantly.

For the order of $\|v_h - u\|_2$ is one higher than the order of $\|u_h - u\|_2$, the error ratio $r_h$ is almost $\frac{1}{n}$ where $n$ denotes the level of the grid. As the grid becomes finer, $w_h$ is much closer to $u_h$ especially on the finest grid. Therefore, when the grid is fine enough, the error of $\|u_h - w_h\|_2$ is smaller so that the number of iterations is reduced.

For test problem 1, on the finest grid $512 \times 512 \times 512$, the error ratio $r_h$ is 0.028. It is obvious that the $r_h$ is so small on the finest grid that we only need to perform one iteration on the finest grid. The number of iteration is reduced significantly.

## 6. Numerical Experiments

### Test Problem 1.

The exact solution of the test problem 1 introduced in [19] can be written as

$$u(x, y, z) = (1 - \cos(2\pi x))(1 - \cos(2\pi y))(1 - \cos(2\pi z)).$$

Applying the biharmonic operator on the exact solution, we can obtain the forcing term $f(x, y, z)$ as follows:

$$f(x, y, z) = -16\pi^4(\cos(2\pi x) - 4\cos(2\pi x)\cos(2\pi z) - 4\cos(2\pi x)\cos(2\pi y) + 9\cos(2\pi x)\cos(2\pi y)\cos(2\pi z) + \cos(2\pi z)).$$

Obtain first boundary data from the exact solution while obtaining second boundary data $f_2$ by taking partial derivative for the exact solution.

Results listed on the table 1 of numerical experiments are performed with EXCMG method, using 3.6 thousand unknowns on the coarsest grid $32 \times 32 \times 32$ and more than 135 million unknowns on the finest grid $512 \times 512 \times 512$. In the table 1, “Iter” denotes the number of iterations needed for the Bi-CG solver to achieve the relative residual less than $10^{-6}$.
of numerical experiments are performed on five level grids with 3.6 thousand work units. Moreover, from Table 2, we can see that numerical solution $u_h$ reaches almost full second-order accuracy, the initial guess $w_0$ is third-order approximation to numerical solution $u_h$, while the extrapolated solution $\tilde{u}_h$ increases the numerical solution’s accuracy significantly. If we use $\tilde{u}_h$ as numerical solution on the finest grid 512 $\times$ 512, the error ratio $r_h$ is already 0.27. Thus only six iterations are needed to perform to achieve the expected accuracy.

**Test Problem 3.** The exact solution of the test problem 3 can be written as

$$u(x, y, z) = \sinh(x) \sinh(y) \sinh(z).$$

Applying the biharmonic operator on the exact solution, we can obtain the forcing term $f(x, y, z)$ as follows:

$$f(x, y, z) = e^{3z} (x^4 y^4 + 2x^2 y^2 z^2 + x^4 z^2 + 8x^3 yz + 2x^2 y^2 z^2 + 2x^2 y^2 z^4 + 4x^2 + 8xy^3 z + 8xyz^3 + y^4 z^4 + 4y^2 z^2 + 4x^2 z^2)$$

Obtain first boundary data from the exact solution while obtaining second boundary data $f_2$ by taking partial derivative for the exact solution.

1 WU (work unit) is the computational cost of performing one relaxation sweep on the finest grid. Here, the EXCMG$_{bi-co}$ computation cost = $1 + 8 \times 2^{-3} + 64 \times 2^{-6} + 512 \times 2^{-9} + 474 \times 2^{-12} = 4.12$.

2 WU (work unit) is the computational cost of performing one relaxation sweep on the finest grid. Here, the EXCMG$_{bi-co}$ computation cost = $6 + 48 \times 2^{-3} + 384 \times 2^{-6} + 470 \times 2^{-9} + 259 \times 2^{-12} = 18.98$. 

| Mesh      | Iters | $\|u_h-u\|_2$ Error | $\|u_h-u\|_\infty$ Error | $\|w_{\tilde{h}}-u_h\|_2$ Error | $\|w_{\tilde{h}}-w_{\tilde{h}}\|_\infty$ Error |
|-----------|-------|---------------------|-------------------------|-------------------------------|---------------------------------|
| 32 $\times$ 32 $\times$ 32 | 474   | 1.13(-2)            | 5.16(-2)                | 5.14(-3)                      | 1.46(-5)                        |
| 64 $\times$ 64 $\times$ 64 | 512   | 2.89(-3)            | 1.29(-2)                | 6.15(-4)                      | 3.06                            |
| 128 $\times$ 128 $\times$ 128 | 64    | 7.27(-4)            | 3.21(-3)                | 7.63(-5)                      | 3.01                            |
| 256 $\times$ 256 $\times$ 256 | 8     | 1.80(-4)            | 8.01(-4)                | 9.56(-6)                      | 3.00                            |
| 512 $\times$ 512 $\times$ 512 | 1     | 4.25(-5)            | 1.99(-4)                | 1.19(-6)                      | 3.00                            |
|           |       | 4.12 WU             |                         |                               |                                 |

Table 2: Errors and convergence rates using EXCMG$_{bi-co}$ for Problem 2.

| Mesh      | Iters | $\|u_h-u\|_2$ Error | $\|u_h-u\|_\infty$ Error | $\|w_{\tilde{h}}-u_h\|_2$ Error | $\|w_{\tilde{h}}-w_{\tilde{h}}\|_\infty$ Error |
|-----------|-------|---------------------|-------------------------|-------------------------------|---------------------------------|
| 32 $\times$ 32 $\times$ 32 | 259   | 8.96(-7)            | 8.06(-6)                | 4.59(-6)                      | 9.90(-9)                        |
| 64 $\times$ 64 $\times$ 64 | 470   | 2.30(-7)            | 2.06(-6)                | 5.48(-7)                      | 2.93(-10)                       |
| 128 $\times$ 128 $\times$ 128 | 384   | 5.80(-8)            | 5.15(-7)                | 6.66(-8)                      | 3.04                            |
| 256 $\times$ 256 $\times$ 256 | 48    | 1.46(-8)            | 1.28(-7)                | 8.19(-9)                      | 3.02                            |
| 512 $\times$ 512 $\times$ 512 | 6     | 3.67(-9)            | 3.22(-8)                | 1.00(-9)                      | 3.03                            |
|           |       | 18.98 WU            |                         |                               |                                 |

Test Problem 2. The exact solution of the test problem 2 can be written as

$$u(x, y, z) = e^{3z}. \quad (36)$$

Applying the biharmonic operator on the exact solution, we can obtain the forcing term $f(x, y, z)$ as follows:

$$f(x, y, z) = e^{3z} (x^4 y^4 + 2x^2 y^2 z^2 + x^4 z^2 + 8x^3 yz + 2x^2 y^2 z^2 + 2x^2 y^2 z^4 + 4x^2 + 8xy^3 z + 8xyz^3 + y^4 z^4 + 4y^2 z^2 + 4x^2 z^2) \quad (37)$$

Obtain first boundary data from the exact solution while obtaining second boundary data $f_2$ by taking partial derivative for the exact solution.
of numerical experiments are performed with EXCMG.

Besides, from table 3, we can see that numerical solution $u_h$ reaches almost full second-order accuracy, the initial guess $w_h$ is third-order approximation to numerical solution $u_h$, while the extrapolated solution $\tilde{u}_h$ increase the numerical solution’s accuracy greatly.

Again, five level grids are used with 3.6 thousand unknowns on the coarsest grid 32×32×32 and more than 135 million unknowns on the finest grid 512×512×512. In addition, from table 3, we can see that numerical solution $u_h$ reaches almost full second-order accuracy, the initial guess $w_h$ is third-order approximation to numerical solution $u_h$, while the extrapolated solution $\tilde{u}_h$ increase the numerical solution’s accuracy greatly.

### Test Problem 4

The exact solution of the test problem 4 can be written as

$$u(x, y, z) = xyz \log(1 + x + y + z).$$

Applying the biharmonic operator on the exact solution, we can obtain the forcing term $f(x, y, z)$ as follows:

$$f(x, y, z) = \frac{- \left(2(x^3 + 8x^2 + 15xyz + 4xy + 4xz + 4x + 4y^3 + 8y^2 + 4yz + 4y + 4z^3 + 8z^2 + 4z)\right)}{(x + y + z + 1)^4}.$$

Obtain first boundary data from the exact solution while obtaining second boundary data $f_2$ by taking partial derivative for the exact solution.

Again, results listed on the table 4 of numerical experiments are performed with EXCMG, using 3.6 thousand unknowns on the coarsest grid 32×32×32 and more than 135 million unknowns on the finest grid 512×512×512.

### Test Problem 5

The exact solution of the test problem 5 can be written as

$$u(x, y, z) = e^{10(x-0.5)^2 + 10(y-0.5)^2 + 10(z-0.2)^2}(-x^2 + x)(-y^2 + y)(-z^2 + z).$$
Obtain first boundary data from the exact solution while obtaining second boundary data \( f_2 \) by taking partial derivative for the exact solution.

Again, we use five level grids which have 3.6 thousand unknowns on the coarsest grid \( 32 \times 32 \times 32 \) and more than 135 million unknowns on the finest grid \( 512 \times 512 \times 512 \). Additionally, from table 5, we can see that numerical solution \( u_h \) reaches almost full second-order accuracy, the initial guess \( w_h \) is third-order approximation to numerical solution \( u_h \), while the extrapolated solution \( \hat{u}_h \) increase the numerical solution’s accuracy significantly.

### 7. Conclusion

In this work, we propose a new extrapolation cascadic multigrid method \( EXCMG_{bi-cg} \) to solve the linear three-dimensional biharmonic equation. By applying the Richardson extrapolation and quadratic interpolation methods on numerical solutions which are on current and previous grids, much better initial guesses of iterative solutions are obtained on the next finer grid so that the iterative time for Bi-CG solver is reduced. It is the main advantage of our work. Additionally, the introduction of the relative residual tolerance into our work enables us to obtain the desired accuracy conveniently. Furthermore, reducing computational time and the number of iteration, the numerical results of tests demonstrate that the \( EXCMG_{bi-cg} \) method is efficient and particularly suitable for solving large scale problems.

### References

**References**

[1] M.M. Gupta, R. Manohar, Direct solution of biharmonic equation using noncoupled approach, J. Comput. Phys. 33 (1979) 236-248.

[2] I. Altas, J. Dym, M.M. Gupta, R. Manohar, Multigrid solution of automatically generated high order discretisation for the biharmonic equation, SIAM J. Sci. Comput. 19 (1998) 1575-1585.

[3] L. Bauer, E. L. Reiss, Block five diagonal matrices and the fast numerical solution of the biharmonic equation, Math. Comp. 26 (1972) 311-326.

[4] B.L. Buzbee, F.W. Dorr, The direct solution of the biharmonic equation on rectangular regions and the Poisson equation on irregular regions, SIAM J. Numer. Anal. (1974) 753-763.

[5] J. Ribeiro Dos Santos, Équations aux différences finies pour l’équation biharmonique dans l’espace à trois dimensions, C.R. Acad. Sci. Paris Sér. A-B 264 A291-A293 (1967).

[6] M.M. Gupta, Discretization error estimates for certain splitting procedures for solving first biharmonic boundary value problems, SIAM J. Numer. Anal. 12 (1975) 364-377.

[7] M.M. Gupta and L.W. Ehrlich, Some difference schemes for the biharmonic equation, SIAM J. Numer. Anal. 12 (1975) 773-790.

[8] M. Dehghan, A. Mohebbi, Multigrid solution of high order discretisation for three-dimensional biharmonic equation with Dirichlet boundary conditions of second kind, Appl. Math. Comput. 180 (2006) 575-593.

[9] N.A. Gumerov, R. Duraiswami, Fast multipole method for the biharmonic equation in three dimensions, J. Comput. Phys. 215 (2006) 363-383.

[10] A. Gómez-Polanco, J.M. Guevara-Jordan, B. Molina, A mimetic iterative scheme for solving biharmonic equations, Math. Comput. Modelling 57 (2013) 2132-2139.

[11] L. Jones Tarcius Doss, N. Kousalya, Finite Pointset Method for biharmonic equations, Appl. Math. Comput. 75 (2018) 3756-3785.

[12] O. Karakashian, C. Collins, Two-Level Additive Schwarz Methods for Discontinuous Galerkin Approximations of the Biharmonic Equation, J. Sci. Comput. 74 (2018) 573-604.

[13] S.D. Conte, R.T. Dames, On an alternating direction method for solving the plate problem with mixed boundary conditions, J. Assoc. Comput. Mach. 7 (1960) 264-273.
[14] I. Altas, J. Erhel, M.M. Gupta, High accuracy solution of three-dimensional biharmonic equations, Numer. Algorithm 29(2002) 1-19.
[15] Qingqu Zhuang, Lizhen Chen, Legendre-Galerkin spectral-element method for the biharmonic equations and its applications, Appl. Math. Comput. 74 (2017) 2958-2968.
[16] Bishnu P. Lamichhane, A finite element method for a biharmonic equation based on gradient recovery operators, BIT Numer. Math. 54 (2014) 469-484.
[17] F.A. Bornemann, P. Deuflhard, The cascadic multigrid method for elliptic problems, Numer. Math. 75 (1996) 135-152.
[18] Wang, Y., Zhang, J., Fast and robust sixth-order multigrid computation for the three-dimensional convection-diffusion equation, J. Comput. Appl. Math. 234 (2010) 3496-3506.
[19] Marchuk, G.I., Shaidurov, V.V., Difference Methods and Their Extrapolations, Springer. New York (1983).
[20] Neittaanmaki, P., Lin, Q., Acceleration of the convergence in finite difference method by predictor corrector and splitting extrapolation methods, J. Comput. Math. 5 (1987) 181-190.
[21] G.I. Marchuk, V.V. Shaidurov, Difference Methods and Their Extrapolations, Springer-Verlag, New York, 1983.
[22] Fišůmeier, R., On Richardson extrapolation for finite difference methods on regular grids, Numer. Math. 55 (1989) 451-462.
[23] Han, G.Q., Spline finite difference methods and their extrapolation for singular two-point boundary value problems, J. Comput. Math. 11(1993) 289-296.
[24] Sun, H., Zhang, J., A high order finite difference discretization strategy based on extrapolation for convection diffusion equations, Numer. Methods Part. Differ. Equ. 20 (2004) 18-32.
[25] Rahul, K., Bhattacharyya, S.N., One-sided finite-difference approximations suitable for use with Richardson extrapolation, J. Comput. Phys. 219 (2006) 13-20.
[26] C.M. Chen, H.L. Hu, Z.Q. Xie, C.L. Li, Analysis of extrapolation cascadic multigrid method (EXCMG), Sci. China Ser. A-Math. 51 (2008) 1349-1360.
[27] Chen, C.M., Shi, Z.C., Hu, H.L., On extrapolation cascadic multigrid method, J. Comput. Math. 29 (2011) 684-697.
[28] H.L. Hu, C.M. Chen, K.J. Pan, Asymptotic expansions of finite element solutions to Robin problems in $H^1$ and their application in extrapolation cascadic multigrid method, Sci. China Math. 57 (2014) 687-698.
[29] K.J. Pan, D.D. He, C.M. Chen, An extrapolation cascadic multigrid method for elliptic problems on reentrant domains, Adv. Appl. Math. Mech. 9 (2017) 1347-1363.
[30] H.L. Hu, C.M. Chen, K.J. Pan, Time-extrapolation algorithm (TEA) for linear parabolic problems, J. Comput. Math. 32 (2014) 183-194.
[31] K.J. Pan, I.T. Tang, H.L. Hu, et al., Extrapolation cascadic multigrid method for 2D current resistivity modeling, Chin. J. Geophys. 55 (2012) 2769-2778 (in Chinese).
[32] K.J. Pan, I.T. Tang, 2.5-D and 3-D DC resistivity modelling using an extrapolation cascadic multigrid method, Geophys. J. Int. 197 (2014) 1459-1470.
[33] Munyakazi, J.B., Patidar, K.C., On Richardson extrapolation for fitted operator finite difference methods, Appl. Math. Comput. 201 (2008) 465-480.
[34] Tam, C.K.W., Kurbatskii, K.A., A wavenumber based extrapolation and interpolation method for use in conjunction with high-order finite difference schemes, J. Comput. Phys. 157 (2000) 588-617.
[35] Ma, Y., Ge, Y., A high order finite difference method with Richardson extrapolation for 3D convection diffusion equation, Appl. Math. Comput. 215 (2010) 3408-3417.
[36] Marchi, C.H., Novak, L.A., Santiago, C.D., et al., Highly accurate numerical solutions with repeated Richardson extrapolation for 2D Laplace equation, Appl. Math. Model. 37 (2013) 7386-7397.