

Analysis on Sixth-Order Compact Approximations with Richardson Extrapolation for 2D Poisson Equation

Ruxin Dai¹, Pengpeng Lin²

¹Department of Computer Science and Information Systems, University of Wisconsin River Falls, River Falls, WI, USA

²Department of Mathematics, Statistics and Computer Science, University of Wisconsin Stout, Menomonie, WI, USA

Email: ruxin.dai@uwrf.edu

How to cite this paper: Dai, R.X. and Lin, P.P. (2018) Analysis on Sixth-Order Compact Approximations with Richardson Extrapolation for 2D Poisson Equation. *Journal of Applied Mathematics and Physics*, 6, 1139-1159.

<https://doi.org/10.4236/jamp.2018.66097>

Received: April 5, 2018

Accepted: June 3, 2018

Published: June 6, 2018

Copyright © 2018 by authors and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

By using Richardson extrapolation and fourth-order compact finite difference scheme on different scale grids, a sixth-order solution is computed on the coarse grid. Other three techniques are applied to obtain a sixth-order solution on the fine grid, and thus give out three kinds of Richardson extrapolation-based sixth order compact computation methods. By carefully analyzing the truncation errors respectively on 2D Poisson equation, we compare the accuracy of these three sixth order methods theoretically. Numerical results for two test problems are discussed.

Keywords

Richardson Extrapolation, Sixth-Order Solutions, High Order Compact Finite Difference Scheme, Truncation Error Analysis, 2D Poisson Equation

1. Introduction

High order and high efficiency numerical computation for partial differential equations is very important in many scientific and engineering modeling problems. Compared to low order (second order) methods, high order methods can achieve satisfactory errors on much coarser grids and thus greatly curtail the computational cost [1]. Although the most extensively used methods to obtain high order accuracy are spectral methods [2] [3], these methods have some limitations. The most significant one is that the discretization of PDE by spectral methods leads to the solution of large systems of linear or non-linear equations involving full matrices [4]. In contrast, finite difference (and finite element) methods lead to systems with sparse matrices that can be handled by efficient

iterative methods to reduce the computation complexity and computation cost substantially [5]. Therefore, taking into account the computational efficiency, we focus on developing numerical algorithms based on high order finite difference methods here. Among various high order finite difference methods, high order compact (HOC) finite difference schemes are extremely noticeable. Compared to using straightforward central differences to obtain high order accuracy on a larger stencil which results the increase of bandwidth of the coefficient matrix and rises to a problem at the points close to the boundaries, the HOC schemes only use the center and adjacent points (*i.e.*, a 9-points stencil is used in HOC schemes in 2D) which avoid extra special treatments for those points close to the boundaries and further improve computational efficiency. In the past two decades, HOC schemes have been well studied. Various fourth order compact (FOC) finite difference schemes have been developed for Poisson equations, convection-diffusion equations, and Navier-Stokes equations [6] [7] [8] [9].

Recently, there has been growing interest in developing sixth order compact finite difference schemes. By using Taylor series expansion, Soptz and Carey [10] developed a compact scheme which can achieve sixth order accuracy only when the derivatives of the source term can be determined analytically. Sutmann used Padé approximation discussed by Lele [11] on the Taylor expansion for the discretized Laplace operator to develop sixth order compact schemes [12]. Although the schemes need less grid points than the straightforward expansion approach, they are not fully compact since other grid points, since other grid points (besides the center and adjacent points) are involved.. Chu and Fan [13] proposed a three-point combined compact difference scheme, which can achieve sixth order accuracy for the inner grid points and fifth order accuracy for the boundary grid points. However, their scheme asks to compute the dependent variables and their derivatives together which results in a triple-tridiagonal system with high computational cost for solution. In addition, it has a stability problem that, for certain problems with a large meshsize, the computed solution may be oscillatory [14]. Although a finer meshsize may avoid numerical oscillations, the use of fine mesh is contradictory to the motivation of using higher order compact schemes. There are other sixth order finite difference schemes generated similarly [15] [16], but all of them share common weak points such as:

- 1) derivatives of source term appeared in the right-hand side which require analytical forms or approximations for the derivatives with certain order accuracy;
- 2) not complete compact schemes which may cause problems near to the boundaries;
- 3) complicated resulting linear systems which increase the difficulty of choosing effective iterative solvers.

Another category of sixth order compact approximations is based on Richardson extrapolation which is a technique introduced by Lewis Fry Richardson in the early of 20th century [17]. In numerical analysis, Richardson

extrapolation is a sequence acceleration method used to improve the rate of convergence of a sequence and is also the basis of Romberg integration [18]. Although assumptions of smoothness and monotone truncation error convergence in the mesh spacing are involved, using Richardson extrapolation to get higher order accurate solutions is more convenient than developing direct discretizations [19]. We can avoid complicated stencils, wider bandwidth matrices, special considerations for near-boundary points, additional stability analysis, etc. In addition, highly efficient solvers for the resulting large sparse linear system can be easily applied in such sixth order methods. Further more, Richardson extrapolation does not require any knowledge of the underlying methodology except the order of accuracy, which guarantees the minimal intervention to the existing computational tools [20]. Sun *et al.* [21] first proposed to use Richardson extrapolation on two fourth order solutions of different grained grids to compute sixth order solutions on the fine grid. Then, Wang [22] developed a multiscale multigrid (MSMG) method as a very efficient solver to compute sixth order solutions for the 2D Poisson equation with Richardson extrapolation. Since the extrapolated sixth order solution is only generated on the coarse grid, the key issue of using Richardson extrapolation in the sixth order solution computation is how to obtain sixth order solutions on the fine grid. The most direct way is to inject sixth order coarse grid solution into the fine grid, which allows a subset of fine grid points to get sixth order solutions. Then, other strategies are required to compute sixth order solutions for the rest of the fine grid points and finally make the entire fine grid solution reach sixth order accuracy. At present, there are three such kind of strategies worthy of analysis and discussion. They are: 1) operator based interpolation; 2) multiple coarse grid computation; 3) complete Richardson extrapolation.

In this paper, our goal is to analyze and compare these Richardson extrapolation-based sixth order methods in computational accuracy. Therefore, we first describe these three Richardson extrapolation-based sixth order methods in Section 2. Then, we give a detailed analysis on their truncation errors in Section 3. After that, we use numerical experiments to verify our theoretical analysis in Section 4. At last, the conclusion and comments are given in Section 5.

2. Sixth Order Compact Approximations with Richardson Extrapolation

Consider a 2D Poisson equation in the form of

$$u_{xx} + u_{yy} = f(x, y), \quad (x, y) \in \Omega, \quad (1)$$

where Ω is a rectangular domain with suitable boundary conditions defined on $\partial\Omega$. The solution u and the forcing term $f(x, y)$ are assumed to be sufficiently smooth and have necessary continuous partial derivatives. Ω can be discretized with uniform meshsize h in the x and y coordinate directions. The mesh points are (x_i, y_j) with $x_i = ih$ and $y_j = jh$.

Assume we have p th order accurate approximate solutions $u_{i,j}^{2h}$ and $u_{i,j}^h$ on

the Ω_{2h} grid and Ω_h grid. A $(p + 2)$ th order accurate solution $\tilde{u}_{i,j}^{2h}$ on the Ω_{2h} grid can be obtained by the general Richardson extrapolation, which can be written as [18]

$$\tilde{u}_{i,j}^{2h} = \frac{2^p u_{2i,2j}^h - u_{i,j}^{2h}}{2^p - 1}. \quad (2)$$

To get a sixth order solution, FOC schemes are first used to compute fourth order accurate solutions $u_{i,j}^{2h}$ and $u_{i,j}^h$ on the coarse grid Ω_{2h} and fine grid Ω_h , respectively. Then the Richardson extrapolation formula as

$$\tilde{u}_{i,j}^{2h} = \frac{16u_{2i,2j}^h - u_{i,j}^{2h}}{15} \quad (3)$$

is used to compute the sixth order accurate solution $\tilde{u}_{i,j}^{2h}$ on the coarse grid Ω_{2h} [22].

2.1. Richardson Extrapolation with Operator Based Interpolation

One method using Richardson extrapolation to compute a sixth order accurate solution on the fine grid is to use an operator based interpolation scheme, proposed by Wang and Zhang [22], which iteratively updates the fine grid points in a specific sequence until some convergence condition is satisfied. This process is an iterative refinement procedure and similar to basic iterative methods [5]. The operator based interpolation for the 2D Poisson Equation (1) has the formula as [22]

$$\tilde{u}_{i,j}^h = -\frac{1}{20} \left[F_{i,j} - 4(u_{i+1,j}^h + u_{i-1,j}^h + u_{i,j+1}^h + u_{i,j-1}^h) - (u_{i+1,j+1}^h + u_{i+1,j-1}^h + u_{i-1,j+1}^h + u_{i-1,j-1}^h) \right]. \quad (4)$$

2.2. Richardson Extrapolation with Multiple Coarse Grid Computation

The second Richardson extrapolation sixth order computation is to use multiple coarse grids. For a 2D problem, the fine grid can be coarsened into four coarse grids, each of which is composed of one subset of fine grid points from: (even, even) fine grid points, (even, odd) fine grid points, (odd, even) fine grid points, and (odd, odd) fine grid points. The sixth order solution for (even, even) fine grid points comes from Richardson extrapolation. Dai *et al.* [23] proposed a direct sixth order computation for other three groups of fine grid points based on multiple coarse grids. First, tridiagonal systems are built on the X-odd grid view, which is constructed by (even, even) fine grid points (marked in red) and (odd, even) fine grid points (marked in black) as **Figure 1**, for computing the sixth order solution of (odd, even) fine grid points. Then, tridiagonal systems are built on the Y-odd grid view, which is constructed by (even, even) fine grid points (marked in red) and (even, odd) fine grid points (marked in blue) as **Figure 2**, for computing the sixth order solution of (even, odd) fine grid points. At last, for the (odd, odd) fine grid points (marked in green) surrounded by other fine grid points with sixth order solutions (marked in red) as **Figure 3**,

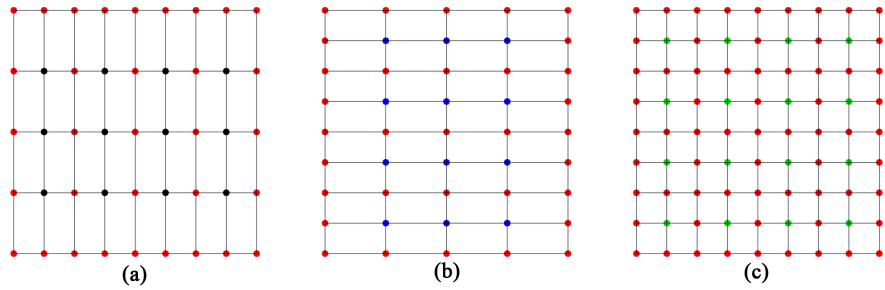


Figure 1. Illustration of the MCG updating strategy in 2D. (a) X-odd grid view: (even, even) + (odd, even) coarse grid; (b) Y-odd grid view: (even, even) + (even, odd) coarse grid; (c) The fine grid after X-odd and Y-odd grid view computation.

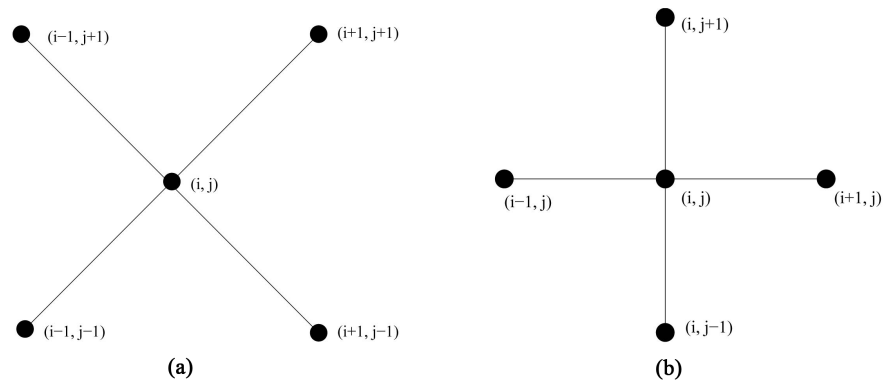


Figure 2. Illustration of the interpolation strategy in 2D. (a) Rotated grid interpolation scheme; (b) Standard grid interpolation scheme.

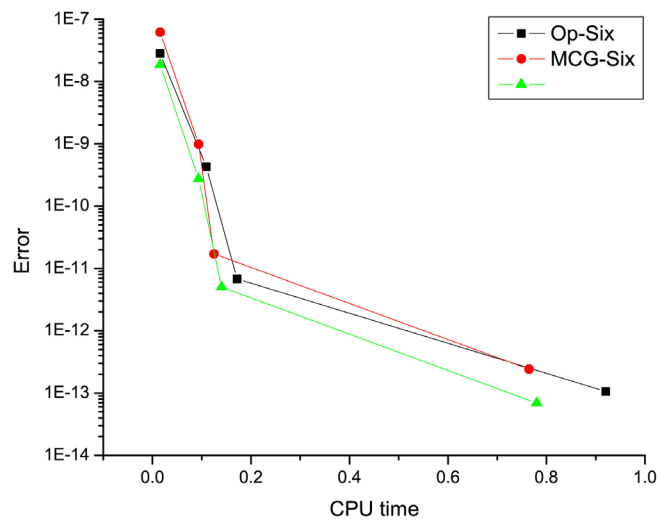


Figure 3. Comparison of the error and CPU for the Problem 1.

only one step of operator interpolation is needed to reach the sixth order solution. Computation details can be found in [23].

2.3. Completed Richardson Extrapolation

The third approach of using Richardson extrapolation for computing sixth order

solutions is Completed Richardson extrapolation. Completed Richardson extrapolation was developed by Roache and Knupp [19] and once used to produce a fourth order accurate solution on the fine grid. It did not use the extrapolated fourth order solution but rather the correction between the second order solution and the fourth order solution in the interpolation process. Similarly, by using the correction between the fourth order solution and the extrapolated sixth order solution rather than the improved solution itself, we could compute a sixth order solution on the entire fine grid.

Assume $u_{i,j}^*$ be the exact solution at node (i, j) on the fine grid. With the definition of fourth order accurate solution, we have

$$u_{i,j}^* = u_{i,j}^h + A_{i,j}h^4 + O(h^6), \tag{5}$$

where A s are the coefficients of the leading error term.

For the (even, even) fine grid points, we could directly inject extrapolated coarse grid solution to get sixth order solution. The correction between the fourth order solution and the extrapolated sixth order solution, say $c_{i,j}^h$, can be used to approximate the leading error term $A_{i,j}h^4$. By using Equation (3), we can deduce

$$A_{i,j}h^4 = c_{i,j}^h = \frac{1}{15}(u_{i,j}^h - u_{i/2,j/2}^{2h}), \quad i = \text{even}, j = \text{even} \tag{6}$$

Then, we could use the correction (6) to approximate corrections for other three subsets of fine grid points, and thus compute sixth order solutions for them.

For the (odd, odd) fine grid points, the rotated grid interpolation, as **Figure 2(a)**, is used to approximate the coefficient A of the leading error term

$$A_{i,j} = \frac{1}{4}(A_{i+1,j+1} + A_{i+1,j-1} + A_{i-1,j+1} + A_{i-1,j-1}) + O(h^2); \quad i = \text{odd}, j = \text{odd}. \tag{7}$$

is used to obtain the formula for sixth order solution computation as

$$\tilde{u}_{i,j}^h = u_{i,j}^h + c_{i,j}^h, \quad i = \text{odd}, j = \text{odd}; \tag{8}$$

where

$$c_{i,j}^h = \frac{1}{4}(c_{i+1,j+1}^h + c_{i+1,j-1}^h + c_{i-1,j+1}^h + c_{i-1,j-1}^h).$$

For the (odd, even) and (even, odd) fine grid points, the standard grid interpolation, as **Figure 2(b)**, is used to approximate the coefficient A of the leading error term

$$A_{i,j} = \frac{1}{4}(A_{i+1,j} + A_{i-1,j} + A_{i,j+1} + A_{i,j-1}) + O(h^2), \tag{9}$$

$i = \text{odd}, j = \text{even}; i = \text{even}, j = \text{odd}.$

is used to obtain the formula for sixth order solution computation as

$$\tilde{u}_{i,j}^h = u_{i,j}^h + c_{i,j}^h, \quad i = \text{odd}, j = \text{even}; i = \text{even}, j = \text{odd}; \tag{10}$$

where

$$c_{i,j}^h = \frac{1}{4}(c_{i+1,j}^h + c_{i-1,j}^h + c_{i,j+1}^h + c_{i,j-1}^h).$$

3. Truncation Error Analysis

In this section, we will give an analysis of truncation errors to compare the accuracy of three Richardson extrapolation-based sixth order methods described in Section 2. All of these methods need to use FOC schemes to get the fourth order solutions on fine and coarse grids. We first analyze the truncation error of the FOC schemes. For more general applications, we will derive the truncation error for the FOC scheme with unequal meshsizes [8].

Denote Δx and Δy to be the meshsizes in the x and y coordinate directions, respectively. The standard second order central difference operators are

$$\delta_x^2 u_{i,j} = \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{\Delta x^2}, \quad \delta_y^2 u_{i,j} = \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{\Delta y^2}$$

By using Taylor series, we have

$$\delta_x^2 u_{i,j} = u_{xx} + \frac{\Delta x^2}{12} u_{x^4} + \frac{\Delta x^4}{360} u_{x^6} + \frac{\Delta x^6}{20160} u_{x^8} + O(\Delta x^8), \quad (11)$$

and

$$\delta_y^2 u_{i,j} = u_{yy} + \frac{\Delta y^2}{12} u_{y^4} + \frac{\Delta y^4}{360} u_{y^6} + \frac{\Delta y^6}{20160} u_{y^8} + O(\Delta y^8). \quad (12)$$

From Equations (11) and (12) we can discretize Equation (1) at the grid point $x_{i,j}$ as

$$\begin{aligned} \delta_x^2 u_{i,j} + \delta_y^2 u_{i,j} = f_{i,j} + \frac{1}{12}(\Delta x^2 u_{x^4} + \Delta y^2 u_{y^4}) + \frac{1}{360}(\Delta x^4 u_{x^6} + \Delta y^4 u_{y^6}) \\ + \frac{1}{20160}(\Delta x^6 u_{x^8} + \Delta y^6 u_{y^8}) + O(\Delta^8). \end{aligned} \quad (13)$$

By taking two times partial derivatives of x and y on both sides of Equation (1), respectively, we have

$$u_{x^4} = f_{xx} - u_{yyxx}, \quad (14)$$

and

$$u_{y^4} = f_{yy} - u_{xxyy}. \quad (15)$$

Using central difference operators and Taylor series in Equations (14) and (15) gives

$$\begin{aligned} \left(u_{x^4}\right)_{i,j} = \delta_x^2 f_{i,j} - \frac{1}{\Delta y^2}(\delta_x^2 u_{i,j+1} - 2\delta_x^2 u_{i,j} + \delta_x^2 u_{i,j-1}) - \frac{\Delta x^2}{12} f_{x^4} \\ - \frac{\Delta x^4}{360} f_{x^6} - \frac{1}{\Delta y^2} \left(-\frac{\Delta x^2}{12} \left(\Delta y^2 u_{x^4 y^2} + \frac{\Delta y^4}{12} u_{x^4 y^4} \right) \right. \\ \left. - \frac{\Delta x^4}{360} \Delta y^2 u_{x^6 y^2} \right) + \frac{\Delta y^2}{12} u_{x^2 y^4} + \frac{\Delta y^4}{360} u_{x^2 y^6} + O(\Delta^6), \end{aligned} \quad (16)$$

and

$$\begin{aligned} \left(u_{y^4}\right)_{i,j} &= \delta_y^2 f_{i,j} - \frac{1}{\Delta x^2} \left(\delta_y^2 u_{i+1,j} - 2\delta_y^2 u_{i,j} + \delta_y^2 u_{i-1,j}\right) - \frac{\Delta y^2}{12} f_{y^4} \\ &\quad - \frac{\Delta y^4}{360} f_{y^6} - \frac{1}{\Delta x^2} \left(-\frac{\Delta y^2}{12} \left(\Delta x^2 u_{x^2 y^4} + \frac{\Delta x^4}{12} u_{x^4 y^4}\right) \right. \\ &\quad \left. - \frac{\Delta y^4}{360} \Delta x^2 u_{x^2 y^6}\right) + \frac{\Delta x^2}{12} u_{x^4 y^2} + \frac{\Delta x^4}{360} u_{x^6 y^2} + O(\Delta^6). \end{aligned} \tag{17}$$

By continuously taking partial derivatives of x on both sides of Equation (14), we get

$$f_{x^4} = u_{x^6} + u_{x^4 y^2} \tag{18}$$

$$f_{x^6} = u_{x^8} + u_{x^6 y^2}. \tag{19}$$

Similarly, by continuously taking partial derivatives of y on both sides of Equation (15), we get

$$f_{y^4} = u_{y^6} + u_{x^2 y^4} \tag{20}$$

$$f_{y^6} = u_{y^8} + u_{x^2 y^6}. \tag{21}$$

Substituting Equations (18) and (19) in Equation (16) gives

$$\begin{aligned} \left(u_{x^4}\right)_{i,j} &= \delta_x^2 f_{i,j} - \frac{1}{\Delta y^2} \left(\delta_x^2 u_{i,j+1} - 2\delta_x^2 u_{i,j} + \delta_x^2 u_{i,j-1}\right) - \frac{\Delta x^2}{12} u_{x^6} + \frac{\Delta y^2}{12} u_{x^2 y^4} \\ &\quad + \frac{\Delta x^2 \Delta y^2}{144} u_{x^4 y^4} - \frac{\Delta x^4}{360} u_{x^8} + \frac{\Delta y^4}{360} u_{x^2 y^6} + O(\Delta^6). \end{aligned} \tag{22}$$

And, substituting Equations (20) and (21) in Equation (17) gives

$$\begin{aligned} \left(u_{y^4}\right)_{i,j} &= \delta_y^2 f_{i,j} - \frac{1}{\Delta x^2} \left(\delta_y^2 u_{i+1,j} - 2\delta_y^2 u_{i,j} + \delta_y^2 u_{i-1,j}\right) - \frac{\Delta y^2}{12} u_{y^6} + \frac{\Delta x^2}{12} u_{x^4 y^2} \\ &\quad + \frac{\Delta x^2 \Delta y^2}{144} u_{x^4 y^4} - \frac{\Delta y^4}{360} u_{y^8} + \frac{\Delta x^4}{360} u_{x^6 y^2} + O(\Delta^6). \end{aligned} \tag{23}$$

Then, we use Equations (22) and (23) to replace the u_{x^4} and u_{y^4} terms in Equation (13) as

$$\begin{aligned} \delta_x^2 u_{i,j} + \delta_y^2 u_{i,j} &= f_{i,j} + \frac{1}{12} \left(\Delta x^2 \delta_x^2 f_{i,j} + \Delta y^2 \delta_y^2 f_{i,j}\right) \\ &\quad - \frac{1}{12} \left(\frac{\Delta x^2}{\Delta y^2} \left(\delta_x^2 u_{i,j+1} - 2\delta_x^2 u_{i,j} + \delta_x^2 u_{i,j-1}\right) \right. \\ &\quad \left. + \frac{\Delta y^2}{\Delta x^2} \left(\delta_y^2 u_{i+1,j} - 2\delta_y^2 u_{i,j} + \delta_y^2 u_{i-1,j}\right)\right) \\ &\quad + (\tau_4)_{i,j} + (\tau_6)_{i,j} + O(\Delta^8), \end{aligned} \tag{24}$$

where

$$(\tau_4)_{i,j} = \frac{1}{144} \left(u_{x^4 y^2} + u_{x^2 y^4}\right) \Delta x^2 \Delta y^2 - \frac{1}{240} \left(u_{x^6} \Delta x^4 + u_{y^6} \Delta y^4\right),$$

$$(\tau_6)_{i,j} = \frac{1}{1728}(\Delta x^4 \Delta y^2 + \Delta x^2 \Delta y^4)u_{x^4 y^4} + \frac{1}{4320}(\Delta x^2 \Delta y^4 u_{x^2 y^6} + \Delta x^4 \Delta y^2 u_{x^6 y^2}) - \frac{11}{60480}(\Delta x^6 u_{x^8} + \Delta y^6 u_{y^8}).$$

Let us use the second order central difference operators in Equation (24), multiply $6\Delta x^2$ on both sides, and denote the mesh aspect ratio $\lambda = \frac{\Delta x}{\Delta y}$, we can get a general FOC scheme like the one presented in [8] as

$$\begin{aligned} & m_1(u_{i+1,j+1} + u_{i+1,j-1} + u_{i-1,j+1} + u_{i-1,j-1}) + m_2(u_{i,j+1} + u_{i,j-1}) \\ & + m_3(u_{i+1,j} + u_{i-1,j}) - m_4 u_{i,j} \\ & = \frac{\Delta x^2}{2}(8f_{i,j} + f_{i+1,j} + f_{i-1,j} + f_{i,j+1} + f_{i,j-1}), \end{aligned} \quad (25)$$

where the coefficients are

$$m_1 = (1 + \lambda^2)/2, \quad m_2 = 5\lambda^2 - 1, \quad m_3 = 5 - \lambda^2, \quad m_4 = 10(1 + \lambda^2).$$

The fourth order truncation error of the FOC scheme (25) is

$$\tilde{\tau}_4 = \left\{ \frac{1}{24\lambda^2}(u_{x^4 y^2} + u_{x^2 y^4}) - \frac{1}{40}\left(u_{x^6} + \frac{u_{y^6}}{\lambda^4}\right) \right\} \Delta x^4. \quad (26)$$

And, the sixth order truncation error of the FOC scheme (25) is

$$\tilde{\tau}_6 = \left\{ \frac{1}{288}\left(\frac{1}{\lambda^2} + \frac{1}{\lambda^4}\right)u_{x^4 y^4} + \frac{1}{720}\left(\frac{u_{x^2 y^6}}{\lambda^4} + \frac{u_{x^6 y^2}}{\lambda^2}\right) - \frac{11}{10080}\left(u_{x^8} + \frac{u_{y^8}}{\lambda^6}\right) \right\} \Delta x^6. \quad (27)$$

In a special case with $\Delta x = \Delta y = h$, the FOC scheme has the form as

$$\begin{aligned} & u_{i+1,j+1} + u_{i+1,j-1} + u_{i-1,j+1} + u_{i-1,j-1} + 4(u_{i,j+1} + u_{i,j-1} + u_{i+1,j} + u_{i-1,j}) - 20u_{i,j} \\ & = \frac{h^2}{2}(8f_{i,j} + f_{i+1,j} + f_{i-1,j} + f_{i,j+1} + f_{i,j-1}). \end{aligned} \quad (28)$$

The fourth order and sixth order truncation errors of the FOC scheme (28) are

$$\tau_{\text{FOC4}} = \left\{ \frac{1}{24}(u_{x^4 y^2} + u_{x^2 y^4}) - \frac{1}{40}(u_{x^6} + u_{y^6}) \right\} h^4, \quad (29)$$

$$\tau_{\text{FOC6}} = \left\{ \frac{1}{144}u_{x^4 y^4} + \frac{1}{720}(u_{x^2 y^6} + u_{x^6 y^2}) - \frac{11}{10080}(u_{x^8} + u_{y^8}) \right\} h^6. \quad (30)$$

Now we can take a look at the truncation error after applying Richardson extrapolation. From the definition of the fourth order accurate solutions on the fine and coarse grids, we have

$$u_h^* = u_h^4 + \tau_{\text{FOC4}} + \tau_{\text{FOC6}}, \quad (31)$$

$$u_{2h}^* = u_{2h}^4 + 16\tau_{\text{FOC4}} + 64\tau_{\text{FOC6}}. \quad (32)$$

Using the Richardson extrapolation Formula (3) gives

$$u_{2h}^* = u_{2h}^6 - \frac{16}{5}\tau_{\text{FOC6}}. \quad (33)$$

Thus, the sixth order truncation error after applying Richardson extrapolation has the form as

$$\tau_{\text{Extrapo}} = -\frac{16}{5}\tau_{\text{FOC6}}. \tag{34}$$

For all Richardson extrapolation-based sixth order compact approximations, the truncation error of (even, even) fine grid points is τ_{Extrapo} because the corresponding sixth order solution is injected from the extrapolated solution of the standard coarse grid. For other three subsets of fine grid points, three computational strategies (operator based interpolation, multiple coarse grid computation, and completed Richardson extrapolation) could be used to obtain sixth order solutions. In the following part, the truncation error analysis for these methods are given respectively.

3.1. Truncation Error of Operator Based Interpolation

The operator based interpolation scheme is from the 9-point FOC scheme (28), which can be iteratively used to approach sixth order solutions for (odd, odd), (odd, even) and (even, odd) fine grid points. It generates a sixth order error in the form as

$$\tau_{op} = \tau_{\text{FOC4}}h^2 = \left\{ \frac{1}{24}(u_{x^4y^2} + u_{x^2y^4}) - \frac{1}{40}(u_{x^6} + u_{y^6}) \right\} h^6. \tag{35}$$

The operator based interpolation Equation (4) can be written as

$$\begin{aligned} u_{i,j}^* &= \tilde{u}_{i,j} + \tau_{op} \\ &= -\frac{1}{20} \left[F_{i,j} - 4(u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1}) \right. \\ &\quad \left. - (u_{i+1,j+1} + u_{i+1,j-1} + u_{i-1,j+1} + u_{i-1,j-1}) \right] + \tau_{op}. \end{aligned} \tag{36}$$

For any linear equation $Au = F$, there is a corresponding residual equation $Ae = r$, where e is the error and r is the residual. According to Equation (36), we could get the operator based interpolation on error as

$$\begin{aligned} 0 &= \tilde{e}_{i,j} + \tau_{op} \\ &= -\frac{1}{20} \left[r_{i,j} - 4(e_{i+1,j} + e_{i-1,j} + e_{i,j+1} + e_{i,j-1}) \right. \\ &\quad \left. - (e_{i+1,j+1} + e_{i+1,j-1} + e_{i-1,j+1} + e_{i-1,j-1}) \right] + \tau_{op}. \end{aligned} \tag{37}$$

When the iterative process of using operator based interpolation converges, the residual r tends to 0. Then Equation (37) becomes to

$$\begin{aligned} 0 &= \frac{1}{20} \left[4(e_{i+1,j} + e_{i-1,j} + e_{i,j+1} + e_{i,j-1}) \right. \\ &\quad \left. + (e_{i+1,j+1} + e_{i+1,j-1} + e_{i-1,j+1} + e_{i-1,j-1}) \right] + \tau_{op}. \end{aligned} \tag{38}$$

At this time, the main component of the error is the truncation error.

Assume the truncation errors on (odd, odd), (odd, even) and (even, odd) fine grid points as α_{op} , β_{op} and γ_{op} , respectively. A system of equations on the truncation error for three subsets of fine grid points is generated from Equation

(38) as

$$\begin{cases} 4 \times \tau_{\text{Extrapo}} + 4 \times (\beta_{op} + \beta_{op}) + 4 \times (\beta_{op} + \beta_{op}) - 20\alpha_{op} = -\tau_{op}, & i = \text{odd}, j = \text{odd}; \\ 4 \times \beta_{op} + 4 \times (\tau_{\text{Extrapo}} + \tau_{\text{Extrapo}}) + 4 \times (\alpha_{op} + \alpha_{op}) - 20\beta_{op} = -\tau_{op}, & i = \text{odd}, j = \text{even}; \\ 4 \times \gamma_{op} + 4 \times (\alpha_{op} + \alpha_{op}) + 4 \times (\tau_{\text{Extrapo}} + \tau_{\text{Extrapo}}) - 20\gamma_{op} = -\tau_{op}, & i = \text{even}, j = \text{odd}. \end{cases} \quad (39)$$

From Equation (39), we get

$$\begin{cases} \alpha_{op} = \tau_{\text{Extrapo}} + \frac{1}{6}\tau_{op}, & i = \text{odd}, j = \text{odd}; \\ \beta_{op} = \tau_{\text{Extrapo}} + \frac{7}{48}\tau_{op}, & i = \text{odd}, j = \text{even}; \\ \gamma_{op} = \tau_{\text{Extrapo}} + \frac{7}{48}\tau_{op}, & i = \text{even}, j = \text{odd}. \end{cases} \quad (40)$$

3.2. Truncation Error of Multiple Coarse Grid Computation

After applying Richardson extrapolation to get sixth order solutions for (even, even) fine grid points, in the multiple coarse grid computation, X-odd grid view and Y-odd grid view are constructed to compute sixth order solutions for (odd, even) and (even, odd) fine grid points, respectively [23]. The X-odd grid view composed by (even, even) and (odd, even) indexed fine grid points is a view of unequal meshsize grid with meshsizes h and $2h$ in the x and y coordinate directions, respectively. The Y-odd grid view composed by (even, even) and (even, odd) indexed fine grid points is a view of unequal meshsize grid with meshsizes $2h$ and h in the x and y coordinate directions, respectively. The sixth order computations on the X-odd grid view and the Y-odd grid view by using tridiagonal solvers lead to sixth order truncation errors $\tau_{x\text{-odd}}$ and $\tau_{y\text{-odd}}$, respectively. By using the general fourth order truncation error Equation (26) and setting corresponding mesh aspect ratio λ , we have explicit forms of $\tau_{x\text{-odd}}$ and $\tau_{y\text{-odd}}$ as

$$\tau_{x\text{-odd}} = \left\{ 4 \times \frac{1}{24} (u_{x^4y^2} + u_{x^2y^4}) - \frac{1}{40} (u_{x^6} + 16 \times u_{y^6}) \right\} h^6, \quad \lambda_{x\text{-odd}} = \frac{1}{2}; \quad (41)$$

$$\tau_{y\text{-odd}} = 4 \times \left\{ 4 \times \frac{1}{24} (u_{x^4y^2} + u_{x^2y^4}) - \frac{1}{40} (16 \times u_{x^6} + u_{y^6}) \right\} h^6, \quad \lambda_{y\text{-odd}} = 2. \quad (42)$$

As for the computation of (odd, even) fine grid points on the X-odd grid view with the mesh aspect ratio $\lambda_{x\text{-odd}} = \frac{1}{2}$, the coefficients in Equation (25) are set as

$$m_1 = \frac{5}{8}, \quad m_2 = \frac{1}{4}, \quad m_3 = \frac{19}{4}, \quad m_4 = \frac{50}{4}.$$

Assume the truncation error of (odd, even) fine grid points to be α_{mcg} , an equation upon the error of X-odd grid view is generated from Equation (25) with the above coefficients as

$$\frac{5}{8} \times 4\tau_{\text{Extrapo}} + \frac{1}{4} \times (\alpha_{mcg} + \alpha_{mcg}) + \frac{19}{4} \times (\tau_{\text{Extrapo}} + \tau_{\text{Extrapo}}) - \frac{50}{4} \alpha = -\tau_{x\text{-odd}}, \quad (43)$$

which gives

$$\alpha_{mcg} = \tau_{\text{Extrapo}} + \frac{\tau_{x\text{-odd}}}{12}. \tag{44}$$

As for the computation of (even, odd) fine grid points on the Y-odd grid view with the mesh aspect ratio $\lambda_{y\text{-odd}} = 2$, the coefficients in Equation (25) are set as

$$m_1 = \frac{5}{2}, \quad m_2 = 19, \quad m_3 = 1, \quad m_4 = 50.$$

Assume the truncation error of (even, odd) fine grid points to be β_{mcg} , an equation upon the error of Y-odd grid view is generated from Equation (25) with the above coefficients as

$$\frac{5}{2} \times 4\tau_{\text{Extrapo}} + 19 \times (\tau_{\text{Extrapo}} + \tau_{\text{Extrapo}}) + 1 \times (\beta_{mcg} + \beta_{mcg}) - 50\beta_{mcg} = -\tau_{y\text{-odd}}, \tag{45}$$

which gives

$$\beta_{mcg} = \tau_{\text{Extrapo}} + \frac{\tau_{y\text{-odd}}}{48}. \tag{46}$$

The update for (odd, odd) fine grid points uses the operator based interpolation Equation (4) and updated sixth order solutions of (even, even), (odd, even) and (even, odd) fine grid points. Assume the truncation error of (odd, odd) fine grid points to be γ_{mcg} , an equation upon the error of the fine grid points is generated by Equation (36) as

$$4\tau_{\text{Extrapo}} + 4 \times (\alpha_{mcg} + \alpha_{mcg}) + 4 \times (\beta_{mcg} + \beta_{mcg}) - 20\gamma_{mcg} = -\tau_{op}, \tag{47}$$

which gives

$$\gamma_{mcg} = \tau_{\text{Extrapo}} + \frac{\tau_{x\text{-odd}}}{30} + \frac{\tau_{y\text{-odd}}}{120} + \frac{\tau_{op}}{20}. \tag{48}$$

3.3. Truncation Error of Completed Richardson Extrapolation

The completed Richardson extrapolation uses two kinds of interpolation on the correction Equation (6) of (even, even) fine grid points to approximate corrections for other fine grid points. As we all know, interpolation brings error. Next, we will analyze how much of the error.

The A coefficients in Equations (7) and (9) can be viewed as a function of u which has the form of $A(u) = \tau_{\text{FOC4}}/h^4$. Based on the Taylor series expansion, the $O(h^2)$ term in Equation (7) has an explicit form as the

$$\frac{h^2}{2} \left(\frac{\partial^2 (\tau_{\text{FOC4}}/h^4)}{\partial x^2} + \frac{\partial^2 (\tau_{\text{FOC4}}/h^4)}{\partial y^2} \right);$$

while the $O(h^2)$ term in Equation (9) has

$$\text{an explicit form as } \frac{h^2}{4} \left(\frac{\partial^2 (\tau_{\text{FOC4}}/h^4)}{\partial x^2} + \frac{\partial^2 (\tau_{\text{FOC4}}/h^4)}{\partial y^2} \right).$$

The second order truncation error for Equation (7), which uses rotated grid interpolation to approximate (odd, odd) fine grid points, has the form as

$$\tau_{\text{RotateInter}} = \left\{ \frac{1}{24} u_{x^4 y^4} + \frac{1}{120} (u_{x^6 y^2} + u_{x^2 y^6}) - \frac{1}{80} (u_{x^8} + u_{y^8}) \right\} h^2; \tag{49}$$

while the second order truncation error for Equation (9), which uses standard grid interpolation to approximate (odd, even) and (even, odd) fine grid points, has the form as

$$\tau_{\text{StandInter}} = \left\{ \frac{1}{48} u_{x^4, y^4} + \frac{1}{240} (u_{x^6, y^2} + u_{x^2, y^6}) - \frac{1}{160} (u_{x^8} + u_{y^8}) \right\} h^2. \quad (50)$$

We find that $\tau_{\text{RotateInter}} = 2\tau_{\text{StandInter}}$.

Consider (odd, odd) fine grid points at first. Equation (7) can be re-written as

$$A_{i,j} = \frac{1}{4} (A_{i+1,j+1} + A_{i+1,j-1} + A_{i-1,j+1} + A_{i-1,j-1}) + \tau_{\text{RotateInter}}, \quad i = \text{odd}, j = \text{odd}. \quad (51)$$

The sixth order computation for the (odd, odd) fine grid points is only related to (even, even) fine grid points. For the (even, even) fine grid points, by using definition of fourth order solutions obtained from the FOC scheme, we have

$$A_{\text{even,even}} = \frac{1}{h^4} [u_{\text{even,even}}^* - u_{\text{even,even}}^4 - \tau_{\text{FOC6}}]. \quad (52)$$

After injecting the extrapolated coarse grid solution into the fine grid, we have

$$u_{\text{even,even}}^* = u_{\text{even,even}}^6 + \tau_{\text{Extrapo}}. \quad (53)$$

Substituting Equation (53) into Equation (52) gives

$$\begin{aligned} A_{\text{even,even}} &= \frac{1}{h^4} [u_{\text{even,even}}^6 - u_{\text{even,even}}^4 - \tau_{\text{FOC6}} + \tau_{\text{Extrapo}}] \\ &= \frac{1}{h^4} [c_{\text{even,even}} - \tau_{\text{FOC6}} + \tau_{\text{Extrapo}}]. \end{aligned} \quad (54)$$

By using Equations (8), (51) and (54), we get the truncation error of the (odd, odd) fine grid points as

$$\begin{aligned} \tau_{\text{CompEx1}} &= u_{i,j}^* - u_{i,j}^6 \\ &= (u_{i,j}^4 + A_{i,j} h^4 + \tau_{\text{FOC6}}) - \left(u_{i,j}^4 + \frac{1}{4} (c_{i+1,j+1} + c_{i+1,j-1} + c_{i-1,j+1} + c_{i-1,j-1}) \right) \\ &= (u_{i,j}^4 + A_{i,j} h^4 + \tau_{\text{FOC6}}) - \left(u_{i,j}^4 + \frac{1}{4} (A_{i+1,j+1} + A_{i+1,j-1} + A_{i-1,j+1} + A_{i-1,j-1}) h^4 \right. \\ &\quad \left. + \tau_{\text{FOC6}} - \tau_{\text{Extrapo}} \right) \\ &= \tau_{\text{RotateInter}} h^4 + \tau_{\text{Extrapo}}, \quad i = \text{odd}, j = \text{odd}. \end{aligned} \quad (55)$$

Then consider (odd, even) and (even, odd) fine grid points. Equation (9) can be re-written as

$$\begin{aligned} A_{i,j} &= \frac{1}{4} (A_{i+1,j} + A_{i-1,j} + A_{i,j+1} + A_{i,j-1}) + \tau_{\text{StandInter}}, \\ &i = \text{odd}, j = \text{even}; i = \text{even}, j = \text{odd}. \end{aligned} \quad (56)$$

The sixth order computation for the (odd, even) and (even, odd) fine grid points are related to both (even, even) and (odd, odd) fine grid points. For the updated (odd, odd) fine grid points, we have

$$u_{\text{odd,odd}}^* = u_{\text{odd,odd}}^6 + \tau_{\text{CompEx1}} = u_{\text{odd,odd}}^6 + \tau_{\text{RotateInter}} h^4 + \tau_{\text{Extrapo}}. \quad (57)$$

By using the definition of fourth order solutions obtained from the FOC scheme, we have

$$A_{\text{odd,odd}} = \frac{1}{h^4} \left[u_{\text{odd,odd}}^* - u_{\text{odd,odd}}^4 - \tau_{\text{FOC6}} \right]. \tag{58}$$

Substituting Equation (57) into Equation (58) gives

$$\begin{aligned} A_{\text{odd,odd}} &= \frac{1}{h^4} \left[u_{\text{odd,odd}}^6 - u_{\text{odd,odd}}^4 - \tau_{\text{FOC6}} + \tau_{\text{RotateInter}} h^4 + \tau_{\text{Extrapo}} \right] \\ &= \frac{1}{h^4} \left[c_{\text{odd,odd}} - \tau_{\text{FOC6}} + \tau_{\text{RotateInter}} h^4 + \tau_{\text{Extrapo}} \right]. \end{aligned} \tag{59}$$

By using Equations (10), (54), (56) and (59), we get the truncation errors of the (even, odd) and (odd, even) fine grid points as

$$\begin{aligned} \tau_{\text{CompEx2}} &= u_{i,j}^* - u_{i,j}^6 \\ &= \left(u_{i,j}^4 + A_{i,j} h^4 + \tau_{\text{FOC6}} \right) - \left(u_{i,j}^4 + \frac{1}{4} (c_{i+1,j} + c_{i-1,j} + c_{i,j+1} + c_{i,j-1}) \right) \\ &= \left(u_{i,j}^4 + A_{i,j} h^4 + \tau_{\text{FOC6}} \right) - \left(u_{i,j}^4 + \frac{1}{4} (A_{i+1,j} + A_{i-1,j} + A_{i,j+1} + A_{i,j-1}) h^4 \right. \\ &\quad \left. + \tau_{\text{FOC6}} - \frac{1}{2} \tau_{\text{RotateInter}} h^4 - \tau_{\text{Extrapo}} \right) \\ &= \tau_{\text{StandInter}} h^4 + \frac{1}{2} \tau_{\text{RotateInter}} h^4 + \tau_{\text{Extrapo}} \\ &= \tau_{\text{RotateInter}} h^4 + \tau_{\text{Extrapo}}, \quad i = \text{odd}, j = \text{even}; i = \text{even}, j = \text{odd}. \end{aligned} \tag{60}$$

We find that the truncation errors of (odd, odd), (odd, even) and (even, odd) fine grid points have the same form as $(\tau_{\text{RotateInter}} h^4 + \tau_{\text{Extrapo}})$, which is larger than the truncation error of (even, even) fine grid points (τ_{Extrapo}) generated from Richardson extrapolation as we expect. It is because another interpolation is involved, *i.e.*, Equation (55) or Equation (60).

In summary, these three Richardson extrapolation-based methods are able to compute the sixth order accurate solution on the entire fine grid. For (even, even) fine grid points, all methods use Richardson extrapolation to get the sixth order solution with truncation error τ_{Extrapo} . For other three subsets of fine grid points, different computational strategies are used to obtain sixth order solutions, which add errors of different magnitude on the truncation error τ_{Extrapo} . **Table 1** lists the truncation errors of the three different Richardson extrapolation-based sixth order compact computations, respectively. Since the error expressions involve various high-order partial derivatives of u , it is hard to conclude a quantitative relationship. By comparing the coefficients of common items, we predict that the completed Richardson extrapolation method should be more accurate than the other two methods. We think the operator based interpolation method has comparable accuracy to the multiple coarse grid computation method. We cannot estimate which one is more accurate than the other because there exists uncertainty of high order partial differential terms in the explicit truncation errors and it is hard to determine the magnitude and sign of them.

Table 1. Truncation error comparison.

Richardson extrapolation with operator based interpolation	
(even, even) points	τ_{Extrapo}
(odd, even) points	$\tau_{\text{Extrapo}} + \frac{7}{48} \left[\frac{1}{24} (u_{x^4y^2} + u_{x^2y^4}) - \frac{1}{40} (u_{x^6} + u_{y^6}) \right] h^6$
(even, odd) points	$\tau_{\text{Extrapo}} + \frac{7}{48} \left[\frac{1}{24} (u_{x^2y^4} + u_{x^4y^2}) - \frac{1}{40} (u_{x^6} + u_{y^6}) \right] h^6$
(odd, odd) points	$\tau_{\text{Extrapo}} + \frac{8}{48} \left[\frac{1}{24} (u_{x^4y^2} + u_{x^2y^4}) - \frac{1}{40} (u_{x^6} + u_{y^6}) \right] h^6$
Richardson extrapolation with multiple coarse grid computation	
(even, even) points	τ_{Extrapo}
(odd, even) points	$\tau_{\text{Extrapo}} + \frac{1}{12} \left\{ 4 \times \frac{1}{24} (u_{x^4y^2} + u_{x^2y^4}) - \frac{1}{40} (u_{x^6} + 16 \times u_{y^6}) \right\} h^6$
(even, odd) points	$\tau_{\text{Extrapo}} + \frac{1}{12} \left\{ 4 \times \frac{1}{24} (u_{x^2y^4} + u_{x^4y^2}) - \frac{1}{40} (16 \times u_{x^6} + u_{y^6}) \right\} h^6$
(odd, odd) points	$\tau_{\text{Extrapo}} + \left\{ \frac{1}{30} \left[4 \times \frac{1}{24} (u_{x^4y^2} + u_{x^2y^4}) - \frac{1}{40} (u_{x^6} + 16 \times u_{y^6}) \right] \right.$ $\left. + \frac{1}{30} \left[4 \times \frac{1}{24} (u_{x^2y^4} + u_{x^4y^2}) - \frac{1}{40} (16 \times u_{x^6} + u_{y^6}) \right] \right.$ $\left. + \frac{1}{20} \left[\frac{1}{24} (u_{x^4y^2} + u_{x^2y^4}) - \frac{1}{40} (u_{x^6} + u_{y^6}) \right] \right\} h^6$
Completed Richardson extrapolation	
(even, even) points	τ_{Extrapo}
(odd, even) points	$\tau_{\text{Extrapo}} + \left\{ \frac{1}{24} u_{x^4y^4} + \frac{1}{120} (u_{x^6y^2} + u_{x^2y^6}) - \frac{1}{120} (u_{x^8} + u_{y^8}) \right\} h^6$
(even, odd) points	$\tau_{\text{Extrapo}} + \left\{ \frac{1}{24} u_{x^4y^4} + \frac{1}{120} (u_{x^6y^2} + u_{x^2y^6}) - \frac{1}{120} (u_{x^8} + u_{y^8}) \right\} h^6$
(odd, odd) points	$\tau_{\text{Extrapo}} + \left\{ \frac{1}{24} u_{x^4y^4} + \frac{1}{120} (u_{x^6y^2} + u_{x^2y^6}) - \frac{1}{120} (u_{x^8} + u_{y^8}) \right\} h^6$

4. Numerical Results

We tested three Richardson extrapolation-based sixth order methods on two 2D Poisson equations. The 9-point FOC scheme (28) was used to get fourth order solutions on different scale grid levels.

One merit of using Richardson extrapolation for sixth order computation lies in that we can easily choose highly efficient solvers for the resulting large sparse linear systems. In our experiments, we chose a multiscale multigrid (MSMG) computational framework [22], which uses multigrid methods to speed up the linear system solution, at the same time, involves a multiscale strategy to obtain higher order accurate solution by extrapolating the computed lower order solutions. The standard V(1, 1)-cycle is selected. The initial guess for the V-cycle

on Ω_{4h} was the zero vector. The V-cycle on Ω_{2h} and Ω_h stopped when the L_2 -norm of the difference of the successive solutions was reduced by a factor of 10^{10} . The stopping criteria for the iterative operator based interpolation and Gauss-Seidel procedure was 10^{-10} . All reported errors were the maximum absolute errors over the discrete grid of the finest level.

The codes were written in Fortran 77 programming language and run on a PC, which has Intel Core i7-4770 with 3.40 GHz and 16 GB RAM.

4.1. Test Problems

Problem 1.

$$-\frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial y^2} = \alpha \sin\left(\frac{\pi}{b}y\right), \quad (x, y) \in \Omega = [0, \lambda] \times [0, b],$$

where the boundary conditions are

$$u(0, y) = u(\lambda, y) = u(x, 0) = u(x, b) = 0.$$

The parameters are chosen as

$$\alpha = \frac{F\pi}{Rb}, \quad \lambda = 10^7 \text{ m}, \quad b = 2\pi \times 10^6 \text{ m}, \quad F = 0.3 \times 10^{-7} \text{ m}^2 \cdot \text{s}^{-2}, \quad R = 0.6 \times 10^{-3} \text{ m} \cdot \text{s}^{-1}.$$

The analytical solution is

$$u = -\alpha \left(\frac{b}{\pi}\right)^2 \sin\left(\frac{\pi y}{b}\right) \left(e^{\frac{\pi x}{b}} - 1\right).$$

Problem 2.

$$-\frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial y^2} = 2\pi^2 \sin(\pi x) \cos(\pi y), \quad (x, y) \in \Omega = [0, 4] \times [0, 1],$$

which has the Dirichlet boundary condition.

The analytical solution is

$$u(x, y) = \sin(\pi x) \cos(\pi y).$$

4.2. Accuracy and Efficiency Comparison

We refined the grid from $N = 32$ to $N = 256$ for both test problems. For convenience, we used three abbreviations to represent three Richardson extrapolation-based sixth order methods to be compared. ‘‘Op-Six’’ is short for the sixth order method with Richardson extrapolation and operator based interpolation; ‘‘MCG-Six’’ means the sixth order method with Richardson extrapolation and multiple coarse grid computation; ‘‘CR-Six’’ denotes the sixth order method with completed Richardson extrapolation.

For both test problems, we compare the accuracy and efficiency of three methods by computing maximum errors and accuracy order, and recording the CPU time in seconds. The results of Problem 1 are shown in [Table 2](#) and [Table 3](#) and [Figure 3](#). The results of Problem 2 are shown in [Table 4](#) and [Table 5](#) and [Figure 4](#).

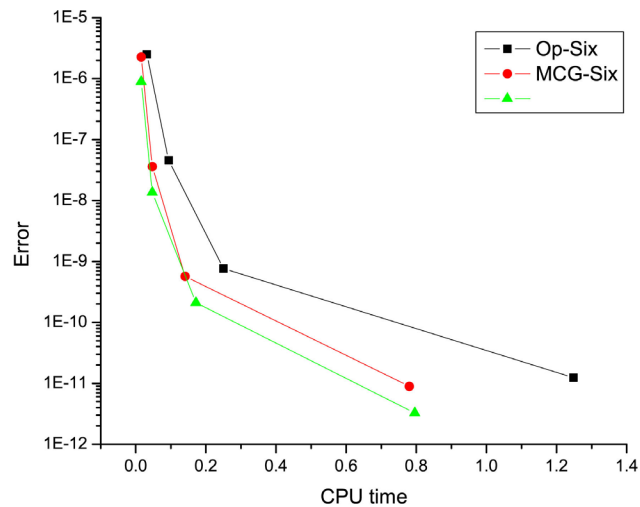


Figure 4. Comparison of the error and CPU for the Problem 2.

Table 2. Accuracy comparison results for Problem 1.

N	Op-Six		MCG-Six		CR-Six	
	Error	Order	Error	Order	Error	Order
32	2.824e-8	-	6.237e-8	-	1.861e-8	-
64	4.296e-10	6.04	9.881e-10	5.98	2.756e-10	6.08
128	6.785e-12	5.98	1.716e-11	5.85	5.016e-12	5.78
256	1.062e-13	6.00	2.429e-13	6.14	6.833e-14	6.20

Table 3. Efficiency comparison results for Problem 1.

N	Op-Six CPU	MCG-Six CPU	CR-Six CPU
32	0.0156	0.0156	0.0156
64	0.109	0.0936	0.0936
128	0.172	0.125	0.140
256	0.920	0.764	0.781

Table 4. Accuracy comparison results for Problem 2.

N	Op-Six		MCG-Six		CR-Six	
	Error	Order	Error	Order	Error	Order
32	2.498e-6	-	2.278e-6	-	8.927e-7	-
64	4.582e-8	5.77	3.624e-8	5.97	1.362e-8	6.03
128	7.663e-10	5.90	5.710e-10	5.99	2.105e-10	6.02
256	1.237e-11	5.95	8.962e-12	5.99	3.271e-12	6.01

Table 5. Efficiency comparison results for Problem 2.

N	Op-Six CPU	MCG-Six CPU	CR-Six CPU
32	0.0312	0.0156	0.0156
64	0.0936	0.0468	0.0468
128	0.250	0.140	0.172
256	1.248	0.780	0.796

As for the accuracy, it is clear that all the methods are able to obtain approximate solutions with sixth order accuracy. After comparing the errors of the three methods, we found that, for both test problems, the solutions solved by the CR-Six method were more accurate than those solved by the Op-Six method and the MCG-Six method. This observation is consistent with our analysis in Section 3. Meanwhile, we notice that the Op-Six computed a little bit more accurate solutions than the MCG-Six method for Problem 1 yet reversed for Problem 2, which shows that there is no fixed conclusion about the accuracy comparison between them.

As for the efficiency, we found that the MCG-Six method and the CR-Six method required less CPU cost than the Op-Six method. The reason is that the Op-Six method involves an iterative refinement procedure with low convergence rate. There is no evident difference between the MCG-Six method and the CR-Six method on CPU cost.

Although the CR-Six method performed better than the other two methods on accuracy and efficiency for the above two test problems, we need to point out that this advantage is for “simple” problems with “good” conditions. Here the “simple” and “good” mean those problems which are not hard to solve (e.g., small Reynolds number) and have smooth solutions, forcing functions and coefficients in the domain. Since the success of CR-Six method relies on the effectiveness of interpolation of the corrections, sufficiently smooth are necessary to guarantee an effective interpolation. We think the CR-Six method may ask for more restrictions on equations than the other two methods for the sixth order approximations on the finest grid. The robustness analysis and numerical experiments on various “difficult” problems is worth exploring in the future.

5. Concluding Remarks

Compared to the sixth order compact schemes derived by Hermitian polynomial, the Richardson extrapolation-based sixth order compact approximations have many obvious advantages, such as simple stencils, complete compact, easy implementation, suitable for high efficient linear system solvers, etc. We studied three Richardson extrapolation-based sixth order compact computation methods and analyzed the truncation errors of them respectively. All three methods were able to compute the sixth order accurate solution on the fine grid if Richardson extrapolation is applied to the fourth order solutions on fine and coarse grids successfully in the domain. From the truncation error analysis, we got a possible qualitative relationship on the accuracy among these sixth order methods, although it is not completely accurate. We also discussed the multigrid method which is suitable for the Richardson extrapolation-based sixth order methods.

Two 2D Poisson equations are tested in the numerical part. We compared the accuracy and efficiency among three sixth order methods. The test results on

accuracy are generally consistent with the observation from the theoretical analysis. The completed Richardson extrapolation method usually performs better in accuracy than the operator based interpolation with Richardson extrapolation method and the multiple coarse grid computation with Richardson extrapolation method. As expected, the efficiency of the operator based interpolation with Richardson extrapolation is lower than the other two sixth order methods.

The exploration of using Richardson extrapolation on sixth order compact computation for high dimensional problems has already begun. In fact, the operator based interpolation with Richardson extrapolation method and the multiple coarse grid computation with Richardson extrapolation method have been used to solve 3D convection-diffusion equations and show exciting performance [24]. The error analysis to 3D cases could be conducted and reported in the future.

Since Richardson extrapolation-based sixth order methods require two comparable uniform grids (*i.e.*, fine grid with meshsize h and coarse grid with meshsize $2h$), the application of this technique to more practical problems in irregular domain with various boundary conditions (*i.e.*, Dirichlet, Neumann, or mixed conditions) has limitations and is not straight forward. One undergoing study is to use the proposed methodology with finite difference ghost-cell technique to obtain high accuracy high efficiency solutions for Poisson equation with mixed boundary conditions. The finite difference ghost-cell technique [25] keeps unchanged the symmetry of the stencil and uniform meshsize through adding extra points (ghost points) outside the domain. Another thing we want to point out is Richardson extrapolation can also be used to improve the accuracy of solutions obtained from finite element methods [26], although we used finite difference methods to illustrate the idea in this paper. And, compared to high order spectral methods which ask for simple/regular geometry of the problem domain or special strategies (*i.e.*, spectral elements, domain decomposition, and etc.) for complex/irregular domains, using finite element methods to provide basic solutions and Richardson extrapolation to improve them would be a more flexible way to handle complex geometries appearing in many engineering problems.

References

- [1] Wang, Z.J., *et al.* (2013) High-Order CFD Methods: Current Status and Perspective. *International Journal for Numerical Methods in Fluids*, **72**, 811-845. <https://doi.org/10.1002/fld.3767>
- [2] Canuto, C., Hussaini, M.Y., Quarteroni, A. and Zang, T.A. (1998) *Spectral Methods in Fluid Dynamics*. Springer, New York.
- [3] Canuto, C., Hussaini, M.Y., Quarteroni, A. and Zang, T.A. (2006) *Spectral Methods: Fundamentals in Single Domains*. Springer, New York.
- [4] Bueno-Orovio, A., Pérez-García, V.M. and Fenton, F.H. (2006) *Spectral Methods for Partial Differential Equations in Irregular Domains: The Spectral Smoothed*

- Boundary Method. *SIAM Journal on Scientific Computing*, **28**, 886-900.
<https://doi.org/10.1137/040607575>
- [5] Saad, Y. (1996) Iterative Methods for Sparse Linear Systems. PWS Publishing, New York.
- [6] Kalita, J.C., Datal, D.C. and Dass, A.K. (2002) A Class of Higher Order Compact Schemes for the Unsteady Two-Dimensional Convection Diffusion Equation with Variable Convection Coefficients. *International Journal for Numerical Methods in Fluids*, **38**, 1111-1131. <https://doi.org/10.1002/flid.263>
- [7] Spitz, W.F. and Carey, G.F. (1995) High-Order Compact Scheme for the Steady Stream-Function Vorticity Equations. *International Journal for Numerical Methods in Engineering*, **38**, 3497-3512. <https://doi.org/10.1002/nme.1620382008>
- [8] Zhang, J. (2002) Multigrid Method and Fourth Order Compact Difference Scheme for 2D Poisson Equation with Unequal Mesh-Size Discretization. *Journal of Computational Physics*, **179**, 170-179. <https://doi.org/10.1006/jcph.2002.7049>
- [9] Zhang, J., Geng, Z. and Dai, R. (2012) Analysis on Two Approaches for High Order Accuracy Finite Difference Computation. *Applied Mathematics Letters*, **25**, 2081-2085. <https://doi.org/10.1016/j.aml.2012.05.003>
- [10] Spitz, W.F. and Carey, G.F. (1996) A High-Order Compact Formulation for the 3D Poisson Equation. *Numerical Methods for Partial Differential Equations*, **12**, 235-243.
[https://doi.org/10.1002/\(SICI\)1098-2426\(199603\)12:2<235::AID-NUM6>3.0.CO;2-R](https://doi.org/10.1002/(SICI)1098-2426(199603)12:2<235::AID-NUM6>3.0.CO;2-R)
- [11] Lele, S.K. (1992) Compact Finite Difference Schemes with Spectral-Like Resolution. *Journal of Computational Physics*, **103**, 16-42.
[https://doi.org/10.1016/0021-9991\(92\)90324-R](https://doi.org/10.1016/0021-9991(92)90324-R)
- [12] Sutmann, G. and Steffen, B. (2006) High-Order Compact Solvers for the Three-Dimensional Poisson Equation. *Journal of Computational and Applied Mathematics*, **187**, 142-170. <https://doi.org/10.1016/j.cam.2005.03.041>
- [13] Chu, P.C. and Fan, C. (1998) A Three-Point Combined Compact Difference Scheme. *Journal of Computational Physics*, **140**, 370-399.
<https://doi.org/10.1006/jcph.1998.5899>
- [14] Zhang, J. and Zhao, J. (2005) Truncation Error and Oscillation Property of the Combined Compact Difference Scheme. *Applied Mathematics and Computation*, **161**, 241-251. <https://doi.org/10.1016/j.amc.2003.12.023>
- [15] Nabavi, M., Kamran Siddiqui, M.H. and Dargahia, J. (2007) A New 9-Point Sixth-Order Accurate Compact Finite-Difference Method for the Helmholtz Equation. *Journal of Sound and Vibration*, **307**, 972-982.
<https://doi.org/10.1016/j.jsv.2007.06.070>
- [16] Kyei, Y., Roop, J.P. and Tang, G. (2010) A Family of Sixth-Order Compact Finite-Difference Schemes for the Three-Dimensional Poisson Equation. *Advances in Numerical Analysis*, **2010**, Article ID: 352174.
- [17] Richardson, L.F. (1910) The Approximate Arithmetical Solution by Finite Differences of Physical Problems Involving Differential Equations, with an Application to the Stresses in a Masonry Dam Trans. *Proceedings of the Royal Society of London. Series A*, **210**, 307-357.
- [18] Cheney, W. and Kincaid, E. (1999) Numerical Mathematics and Computing. 4th Edition, Cole Publishing, Pacific Grove.
- [19] Roache, P.J. and Knupp, P.M. (1993) Completed Richardson Extrapolation. *Com-*

-
- munications in Numerical Methods in Engineering*, **9**, 365-374.
<https://doi.org/10.1002/cnm.1640090502>
- [20] Burg, C. and Erwin, T. (2009) Application of Richardson Extrapolation to the Numerical Solution of Partial Differential Equations. *Numerical Methods for Partial Differential Equations*, **25**, 810-832. <https://doi.org/10.1002/num.20375>
- [21] Sun, H. and Zhang, J. (2004) A High Order Finite Difference Discretization Strategy Based on Extrapolation for Convection Diffusion Equations. *Numerical Methods for Partial Differential Equations*, **20**, 18-32. <https://doi.org/10.1002/num.10075>
- [22] Wang, Y. and Zhang, J. (2009) Sixth Order Compact Scheme Combined with Multigrid Method and Extrapolation Technique for 2D Poisson Equation. *Journal of Computational Physics*, **228**, 137-146. <https://doi.org/10.1016/j.jcp.2008.09.002>
- [23] Dai, R., Zhang, J. and Wang, Y. (2014) Multiple Coarse Grid Acceleration for Multiscale Multigrid Computation. *Journal of Computational and Applied Mathematics*, **269**, 75-85. <https://doi.org/10.1016/j.cam.2014.03.021>
- [24] Dai, R., Wang, Y. and Zhang, J. (2013) Fast and High Accuracy Multiscale Multigrid Method with Multiple Coarse Grid Updating Strategy for 3D Convection Diffusion Equation. *Computers & Mathematics with Applications*, **66**, 542-559. <https://doi.org/10.1016/j.camwa.2013.06.008>
- [25] Gibou, F. and Fedkiw, R. (2005) A Fourth Order Accurate Discretization for the Laplace and Heat Equations on Arbitrary Domains, with Applications to the Stefan Problem. *Journal of Computational Physics*, **202**, 577-601. <https://doi.org/10.1016/j.jcp.2004.07.018>
- [26] Lin, Q. and Xie, H. (2013) Extrapolation of the Finite Element Method on General Meshes. *International Journal of Numerical Analysis and Modeling*, **10**, 139-153.