A High-Order Finite Difference Discretization Strategy Based on Extrapolation for Convection Diffusion Equations

Haiwei Sun, Jun Zhang
Laboratory for High Performance Scientific Computing and Computer Simulation, Department of Computer Science, University of Kentucky, 773 Anderson Hall, Lexington, Kentucky 40506-0046

Received 18 November 2002; accepted 23 April 2003

DOI 10.1002/num.10075

We propose a new high-order finite difference discretization strategy, which is based on the Richardson extrapolation technique and an operator interpolation scheme, to solve convection diffusion equations. For a particular implementation, we solve a fine grid equation and a coarse grid equation by using a fourth-order compact difference scheme. Then we combine the two approximate solutions and use the Richardson extrapolation to compute a sixth-order accuracy coarse grid solution. A sixth-order accuracy fine grid solution is obtained by interpolating the sixth-order coarse grid solution using an operator interpolation scheme. Numerical results are presented to demonstrate the accuracy and efficacy of the proposed finite difference discretization strategy, compared to the sixth-order combined compact difference (CCD) scheme, and the standard fourth-order compact difference (FOC) scheme. © 2003 Wiley Periodicals, Inc. Numer Methods Partial Differential Eq 20: 18–32, 2004

Keywords: convection diffusion equation; compact difference scheme; CCD scheme; Richardson extrapolation

I. INTRODUCTION

In many scientific and engineering modeling applications, such as in the global ocean modeling and wide area weather forecasting, the computational domains are huge and the

Correspondence to: Jun Zhang (e-mail: jzhang@cs.uky.edu)
Contract grant sponsor: U.S. National Science Foundation; contract grant number: CCR-9988165 (to H.S.)
Contract grant sponsor: U.S. National Science Foundation; contract grant numbers: CCR-9902022, CCR-9988165, CCR-0092532 and ACR-0202934 (to J.Z.)
Contract grant sponsor: U.S. Department of Energy; contract grant number: DE-FG02-02ER45961 (to J.Z.)
Contract grant sponsor: Japanese Research Organization for Information Science & Technology (RIST) (to J.Z.)
Contract grant sponsor: University of Kentucky Research Committee (to J.Z.)
Mathematics Subject Classification: 65N06, 65N55, 65F10

© 2003 Wiley Periodicals, Inc.

grid space is not small [1]. In the context of finite difference discretization, the standard second-order discretization schemes may need fine mesh griddings to yield approximate solutions of acceptable accuracy. The resulting large size linear systems have to be solved, which may consume a lot of memory space and CPU cycles on present generation supercomputers.

One approach to reducing computational cost in very large scale modelings and simulations is to use higher order discretization methods, which use relatively coarser mesh griddings to yield approximate solutions of comparable accuracy, relative to the lower-order discretization methods using finer mesh griddings. Although most high-order discretization schemes involve more complicated derivation procedures and higher preprocessing costs to compute the matrix coefficients, the reported computational results have demonstrated that they are more efficacious than the lower-order counterparts [2–4]. One important factor affecting the computational efficacy of a discretization method is to solve the resulting linear systems. The higher-order methods usually generate linear systems of much smaller size, compared with that from the lower-order methods.

Because of this and other advantages of the high-order methods, there has been growing interest in developing and using highly accurate numerical schemes for solving partial differential equations, which has sparked renewed interest in high-order compact difference schemes [5–11]. Recently, Chu and Fan [1] propose a new three-point combined compact difference (CCD) scheme for solving one-dimensional convection diffusion equations and two-dimensional Stommel Ocean model, which is a special two-dimensional convection diffusion equation. By using Hermitian polynomial approximation, they achieve sixth-order accuracy for the inner grid points and fifth-order accuracy for the boundary grid points. The global accuracy of the CCD scheme has been shown numerically to reach the sixth order. For the two-dimensional case with zero boundary conditions, they apply Alternating Direction Implicit (ADI) [12] method and reduce and solve it as a series of one-dimensional problems. However, no sixth-order CCD scheme for solving the two-dimensional problems with nonzero boundary conditions is given in their article. The sparse linear system arising from their one-dimensional CCD discretization scheme is block tridiagonal, which is more complicated and more expensive to solve than a tridiagonal linear system.

We propose a new finite difference discretization strategy, which is based on the Richardson extrapolation technique and an operator interpolation scheme, to solve the convection diffusion equations. The resulting sparse linear systems consist of two independent tridiagonal linear systems that can be solved easily. The proposed finite difference discretization strategy is general in nature and can be used to improve the accuracy of a computed solution of a certain order to a higher order. In this article, we restrict our attention to the case in which we improve the solutions computed from the fourth-order compact difference schemes (FOC) and raise the accuracy order of the computed solution from four to six. Our computational strategy can be applied to solve general two-dimensional convection diffusion equations without the zero boundary condition assumption. The numerical results show that our new computational strategy with the FOC scheme is more efficacious than the CCD scheme or the classical FOC scheme.

This article is arranged as follows. In Section 2 we outline a new sixth-order compact difference discretization strategy for the one-dimensional case. In Section 3 we extend the methodology to the two-dimensional case, by making use of an ADI-type technique. We present some numerical results and comparisons in Section 4. Concluding remarks are included in Section 5.

II. ONE-DIMENSIONAL CASE

We first make a brief introduction to the fourth-order compact difference scheme for solving a one-dimensional convection diffusion equation of the form

$$\frac{d^2u}{dx^2} + b(x)\frac{du}{dx} + c(x)u = f(x), \qquad 0 \le x \le l, \tag{1}$$

where the known functions $b(x)$, $c(x)$, and $f(x)$ are assumed to be sufficiently smooth. For more detailed discussions on the methodologies of the fourth-order compact difference schemes, a reader is urged to consult [7, 13, 14]. We will derive a sixth-order compact computational strategy based on the fourth-order compact difference scheme.

Let $h = l/n$ be the mesh size of the uniform discretization, where n is the number of uniform intervals. For convenience, we assume that n is an even number and Ω_h is the discretized computational domain. We denote $x_j = jh$ and $u_j = u(x_j)$, $a_j = a(x_j)$, $b_j = b(x_j)$, $c_j = c(x_j)$, and $f_j = f(x_j)$, where $j = 0, 1, \ldots, n$. We write the central difference operators for the first and second derivatives as

$$\delta_x^h u_j = \frac{u_{j+1} - u_{j-1}}{2h}, \quad \delta_{xx}^h u_j = \frac{u_{j+1} + u_{j-1} - 2u_j}{h^2}, \quad j = 1, 2, \ldots, n-1,$$

with respect to a smooth function $u(x)$.

Using the Taylor series expansions, we have

$$\delta_{xx}^h u_j = u_{xx} + \frac{h^2}{12}u_{x^4} + \frac{h^4}{360}u_{x^6} + O(h^6), \tag{2}$$

and

$$\delta_x^h u_j = u_x + \frac{h^2}{6}u_{x^3} + \frac{h^4}{120}u_{x^5} + O(h^6), \tag{3}$$

in which we denoted the mth derivative of the function $u(x)$ as

$$u_{x^m} = \frac{d^m u}{dx^m}.$$

By taking derivatives with respect to x on both sides of Eq. (1), we have

$$u_{x^3} = f_x - c_x u - (b_x + c)u_x - bu_{xx}, \tag{4}$$

and

$$\begin{aligned} u_{x^4} &= f_{xx} - c_{xx}u - (b_{xx} + 2c_x)u_x - (2b_x + c)u_{xx} - bu_{x^3} \\ &= (f_{xx} - bf_x) - (c_{xx} - bc_x)u - (b_{xx} + 2c_x - bb_x - bc)u_x - (2b_x + c - b^2)u_{xx}. \end{aligned} \tag{5}$$

It follows that, at the grid point j, we have

$$(u_{x^3})_j = \delta_x^h f_j - (\delta_x^h c_j)u_j - (\delta_x^h b_j + c_j)\delta_x^h u_j - b_j \delta_{xx}^h u_j + \tau_3 h^2 + O(h^4), \tag{6}$$

and

$$(u_{x^4})_j = (\delta_{xx}^h - b_j\delta_x^h)f_j - (\delta_{xx}^h c_j - b_j\delta_x^h c_j)u_j - (\delta_{xx}^h b_j + 2\delta_x^h c_j - b_j\delta_x^h b_j - b_j c_j)\delta_x^h u_j - (2\delta_x^h b_j + c_j - b_j^2)\delta_{xx}^h u_j + \tau_4 h^2 + O(h^4). \tag{7}$$

In Eqs. (6) and (7), the notations τ_3 and τ_4 are used to denote some complicated representations, which will be truncated in obtaining the fourth-order scheme and will play no role in the final computational algorithm.

By utilizing Eqs. (2), (3), (6), and (7), the fourth-order compact difference approximation for the Eq. (1) can be written as

$$\begin{aligned}
\delta_{xx}^h u_j + b_j\delta_x^h u_j + c_j u_j &= f_j + \frac{h^2}{12}u_{x^4} + \frac{b_j h^2}{6}u_{x^3} + \left(\frac{1}{360}u_{x^6} + \frac{b_j}{120}u_{x^5}\right)h^4 + O(h^6)\\
&= f_j + \frac{h^2}{12}[(\delta_{xx}^h - b_j\delta_x^h)f_j - (\delta_{xx}^h c_j - b_j\delta_x^h c_j)u_j - (\delta_{xx}^h b_j + 2\delta_x^h c_j - b_j\delta_x^h b_j\\
&\quad - b_j c_j)\delta_x^h u_j - (2\delta_x^h b_j + c_j - b_j^2)\delta_{xx}^h u_j] + \frac{b_j h^2}{6}[\delta_x^h f_j - (\delta_x^h c_j)u_j - (\delta_x^h b_j\\
&\quad + c_j)\delta_x^h u_j - b_j\delta_{xx}^h u_j] + \left(\frac{\tau_4}{12} + \frac{b_j\tau_3}{6} + \frac{1}{360}u_{x^6} + \frac{b_j}{120}u_{x^5}\right)h^4 + O(h^6)\\
&= f_j + \frac{h^2}{12}(\delta_{xx}^h + b_j\delta_x^h)f_j - \frac{h^2}{12}(\delta_{xx}^h c_j + b_j\delta_x^h c_j)u_j - \frac{h^2}{12}(\delta_{xx}^h b_j + 2\delta_x^h c_j\\
&\quad + b_j\delta_x^h b_j + b_j c_j)\delta_x^h u_j - \frac{h^2}{12}(2\delta_x^h b_j + c_j + b_j^2)\delta_{xx}^h u_j\\
&\quad + \left(\frac{\tau_4}{12} + \frac{b_j\tau_3}{6} + \frac{1}{360}u_{x^6} + \frac{b_j}{120}u_{x^5}\right)h^4 + O(h^6).
\end{aligned} \tag{8}$$

After collecting terms, we have

$$(A_j\delta_{xx}^h + B_j\delta_x^h + C_j)u_j = F_j + \tau h^4 + O(h^6), \tag{9}$$

where

$$A_j = 1 + \frac{h^2}{12}(2\delta_x^h b_j + c_j + b_j^2),$$

$$B_j = b_j + \frac{h^2}{12}(\delta_{xx}^h b_j + 2\delta_x^h c_j + b_j\delta_x^h b_j + b_j c_j),$$

$$C_j = c_j + \frac{h^2}{12}(\delta_{xx}^h c_j + b_j \delta_x^h c_j),$$

$$F_j = f_j + \frac{h^2}{12}(\delta_{xx}^h f_j + b_j \delta_x^h f_j),$$

$$\tau = \frac{\tau_4}{12} + \frac{b_j \tau_3}{6} + \frac{1}{360} u_{x^6} + \frac{b_j}{120} u_{x^5}.$$

By solving a tridiagonal system in Eq. (9), we have

$$u_j^h = u_j = (A_j \delta_{xx}^h + B_j \delta_x^h + C_j)^{-1}(F_j + \tau h^4) + O(h^6). \tag{10}$$

This is a fourth-order approximate solution on the Ω_h grid. A corresponding fourth-order approximate solution u_j^{2h} on the Ω_{2h} grid can be computed analogously. Because the approximate solution u_j is of fourth-order accuracy, we can use the Richardson extrapolation technique [15], from both u^h and u^{2h}, to compute a sixth-order solution $\tilde{u}_j^{2h}$ on Ω_{2h} as

$$\tilde{u}_j^{2h} = \frac{(16u_{2j}^h - u_j^{2h})}{15}. \tag{11}$$

By direct interpolation, $\tilde{u}_{2j}^h = \tilde{u}_j^{2h}$ is a sixth-order approximate solution at the even indexed grid points on Ω_h.

To establish the relationship between the values of the approximate solution at the odd and even indexed grid points, we have, from (9)

$$A_j(u_{j+1} + u_{j-1} - 2u_j) + \frac{h}{2} B_j(u_{j+1} - u_{j-1}) + C_j h^2 u_j = F_j h^2 + O(h^6),$$

or

$$(C_j h^2 - 2A_j)u_j = \left(\frac{h}{2} B_j - A_j\right) u_{j-1} - \left(\frac{h}{2} B_j + A_j\right) u_{j+1} + F_j h^2 + O(h^6).$$

Thus, we can compute the approximate solution at the odd indexed grid points $(2j - 1)$ with $j = 1, 2, \ldots, n/2$, as

$$\tilde{u}_{2j-1} = \frac{1}{C_{2j-1}h^2 - 2A_{2j-1}} \left[\left(\frac{h}{2} B_{2j-1} - A_{2j-1}\right) \tilde{u}_{2j-2} - \left(\frac{h}{2} B_{2j-1} + A_{2j-1}\right) \tilde{u}_{2j} + F_{2j-1}h^2 \right] + O(h^6), \tag{12}$$

where $\tilde{u}_{2j}$ is a sixth-order solution on Ω_{2h} computed by (11). It follows that a sixth-order solution $\tilde{u}_j$ on Ω_h results.

We summarize the computational algorithm as follows.

Algorithm 2.1. *Compute a sixth-order solution using the Richardson extrapolation technique.*

- *Solve* u^h *on* Ω_h *with the fourth-order compact scheme* (9), *which needs to solve an* $(n-1)$*-by-*$(n-1)$ *tridiagonal matrix.*
- *Solve* u^{2h} *on* Ω_{2h} *with the fourth-order compact scheme* (9), *which needs to solve an* $(n/2-1)$*-by-*$(n/2-1)$ *tridiagonal matrix.*
- *Compute a sixth-order solution* $\tilde{u}_j^{2h}$ *on* Ω_{2h}, *which is also* $\tilde{u}_{2j}^h$ *on* Ω_h, *by the Richardson extrapolation technique* (11) *from the two fourth-order solutions* u_{2j}^h *and* u_j^{2h}.
- *Compute a sixth-order solution* $\tilde{u}_{2j-1}^h$ *on* Ω_h *from the sixth-order interpolation formula* (12).

III. TWO-DIMENSIONAL CASE

We study a two-dimensional convection diffusion equation in the form of

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + p(x, y)\frac{\partial u}{\partial x} + q(x, y)\frac{\partial u}{\partial y} = f(x, y), \qquad (x, y) \in [0, l_x] \times [0, l_y]. \tag{13}$$

Let Ω_{h_x,h_y} be the discretized grid space, where $h_x = l_x/N_x$ and $h_y = l_y/N_y$ are the uniform mesh sizes in the x and y coordinate directions, respectively. Here N_x and N_y are the number of uniform intervals in the x and y coordinate directions. Based on the idea used to compute a sixth-order solution in the one-dimensional case, we can use the ADI method, which is similar to a line relaxation method, to compute a sixth-order approximate solution for the two-dimensional convection diffusion equation (13). Although both use the ADI iteration method, our computational procedure is different from that used by Chu and Fan [1] for solving some two-dimensional convection diffusion equations.

The ADI method can be viewed as an iterative method to solve a higher-dimensional problem by repeatedly solving a series of lower-dimensional problems. As in [1], the iteration from k to $(k+1)$ can be separated into two parts, the x-axis sweeping and the y-axis sweeping. Similar to the one-dimensional case, our sixth-order scheme in the two-dimensional case is also based on the FOC scheme. To demonstrate the algorithm more clearly, we first introduce the fourth-order compact difference scheme for the Eq. (13). This particular fourth-order compact difference scheme with different h_x and h_y is designed in [14]. If equal mesh sizes are used, i.e., if $h_x = h_y = h$, this scheme is equivalent to the fourth-order compact difference scheme of Gupta et al. [13]. For simplicity in notations, we use the symbols h and $2h$ in most of the following coefficient and solution representations to indicate the grid spaces to which they belong. However, it should be understood that the mesh sizes in the two coordinate directions do not have to be equal.

As in [14], we assume that the fourth-order compact difference scheme has the following form:

$$\begin{aligned} A_{i,j}^h(0)u_{i,j} + A_{i,j}^h(1)u_{i+1,j} + A_{i,j}^h(2)u_{i,j+1} + A_{i,j}^h(3)u_{i-1,j} + A_{i,j}^h(4)u_{i,j-1} + A_{i,j}^h(5)u_{i+1,j+1} \\ + A_{i,j}^h(6)u_{i-1,j+1} + A_{i,j}^h(7)u_{i-1,j-1} + A_{i,j}^h(8)u_{i+1,j-1} = F_{i,j}^h, \end{aligned} \tag{14}$$

which has a nine-point computational stencil. Here the notation $A_{i,j}^h(0)$ is a simplified version of $A_{i,j}^{h_x}h_y(0)$, as we remarked in the previous paragraph. The exact representation of the coefficients can be found in [14].

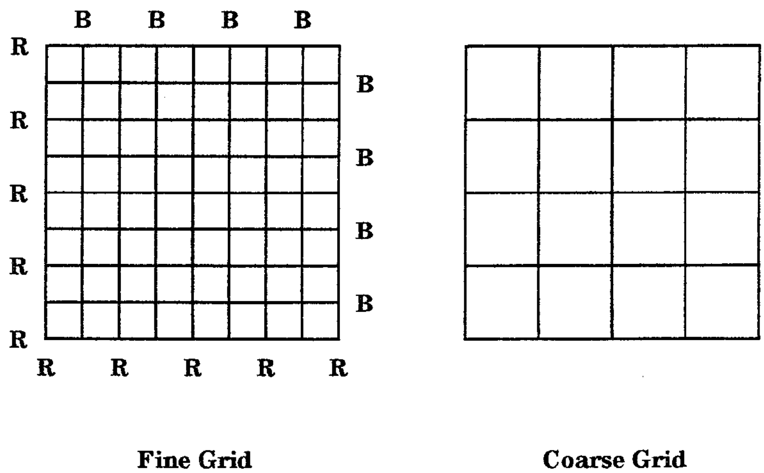

FIG. 1. An illustration of the fine and the coarse grids with the red-black line coloring.

The computational space contains three components, $\Omega^4_{h_x},h_y$, $\Omega^4_{2h_x},2h_y$, and $\Omega^6_{h_x},h_y$, where $\Omega^4_{h_x},h_y$ means the fourth-order solution space with accuracy order $O(h_x^4 + h_y^4)$, $\Omega^4_{2h_x},2h_y$ denotes the fourth-order solution space with accuracy order $O((2h_x)^4 + (2h_y)^4)$, and $\Omega^6_{h_x},h_y$ is the sixth-order solution space with accuracy order $O((2h_x)^6 + (2h_y)^6)$.

The ADI-type iterations are carried out on alternate lines of the fine grid in different coordinate directions. First, the fine grid lines are colored in red and black alternately, as in Fig. 1. The ADI iterations are carried out on the red lines only, by solving an $(N_x - 1)$-by-$(N_x - 1)$ tridiagonal linear system for each red line. The coarse grid is formed by removing all black lines of the fine grid in both coordinate directions. The corresponding equations defined on each line of the coarse grid are also solved line by line using the tridiagonal linear system solver. The approximate solution on the red lines of the fine grid and that of the corresponding coarse grid are used to obtain a sixth-order solution on the coarse grid, which is directly interpolated on to the red lines of the fine grid. The values of the approximate solution on the black lines of the fine grid are computed using an operator interpolation scheme with the fourth-order compact difference scheme (14).

Assuming that N_x and N_y are both even numbers, one ADI iteration (from the kth to the $(k + 1)$st) of the sixth-order algorithm based on the Richardson extrapolation technique is outlined as follows.

Algorithm 3.1. *One ADI iteration with the sixth order Richardson extrapolation technique.*

1. The x-axis sweeping

- Solve a tridiagonal linear system of the order $(\mathrm{N_x} - 1)$-by-$(\mathrm{N_x} - 1)$ for each x-direction red line on the fine grid $\Omega^4_{\mathrm{h_x}},\mathrm{h_y}$, i.e.,

$$\begin{aligned} A^h_{i,j}(0)u^{h,*}_{i,j} + A^h_{i,j}(1)u^{h,*}_{i+1,j} + A^h_{i,j}(3)u^{h,*}_{i-1,j} = F^h_{i,j} - (A^h_{i,j}(2)u^{h,k}_{i,j+1} + A^h_{i,j}(4)u^{h,k}_{i,j-1} \\ + A^h_{i,j}(5)u^{h,k}_{i+1,j+1} + A^h_{i,j}(6)u^{h,k}_{i-1,j+1} + A^h_{i,j}(7)u^{h,k}_{i-1,j-1} + A^h_{i,j}(8)u^{h,k}_{i+1,j-1}), \end{aligned}$$

for the fine grid x-direction red lines $j = 2, 4, \ldots, (N_y - 2)$.

- Solve a tridiagonal linear system of the order ($N_x/2 - 1$)-by-($N_x/2 - 1$) for each x-direction line on the coarse grid $\Omega^4_{2h_x,2h_y}$, i.e.,

$$A^{2h}_{i,j}(0)u^{2h,*}_{i,j} + A^{2h}_{i,j}(1)u^{2h,*}_{i+1,j} + A^{2h}_{i,j}(3)u^{2h,*}_{i-1,j} = F^{2h}_{i,j} - (A^{2h}_{i,j}(2)u^{2h,k}_{i,j+1} + A^{2h}_{i,j}(4)u^{2h,k}_{i,j-1} + A^{2h}_{i,j}(5)u^{2h,k}_{i+1,j+1} + A^{2h}_{i,j}(6)u^{2h,k}_{i-1,j+1} + A^{2h}_{i,j}(7)u^{h,k}_{i-1,j-1} + A^{2h}_{i,j}(8)u^{2h,k}_{i+1,j-1}),$$

for the coarse grid x-direction lines $j = 1, 2, \ldots, (N_y/2 - 1)$.
- From $u^{h,*}_{2i,2j} \in \Omega^4_{h_x,h_y}$ and $u^{2h,*}_{i,j} \in \Omega^4_{2h_x,2h_y}$, we compute $\tilde{u}^{h,*}_{2i,2j} \in \Omega^6_{h_x,h_y}$ by (11).
- From $\tilde{u}^{h,*}_{2i,2j}$ and $\tilde{u}^{h,k}_{i,2j-1} \in \Omega^6_{h_x,h_y}$ with (14), we obtain $\tilde{u}^{h,*}_{2i-1,2j} \in \Omega^6_{h_x,h_y}$.
- From $\tilde{u}^{h,*}_{i,2j} \in \Omega^6_{h_x,h_y}$ with (14), by solving a tridiagonal linear system of the order ($N_x - 1$)-by-($N_x - 1$) for each black line, we obtain $\tilde{u}^{h,*}_{i,2j-1} \in \Omega^6_{h_x,h_y}$. $u^{h,*}_{i,2j-1} \in \Omega^4_{h_x,h_y}$ can be obtained in the same way.

2. The y-axis sweeping

- Solve a tridiagonal linear system of the order ($N_y - 1$)-by-($N_y - 1$) for each y-direction red line on the fine grid $\Omega^4_{h_x,h_y}$, i.e.,

$$A^{h}_{i,j}(0)u^{h,k+1}_{i,j} + A^{h}_{i,j}(2)u^{h,k+1}_{i,j+1} + A^{h}_{i,j}(4)u^{h,k+1}_{i,j-1} = F^{h}_{i,j} - (A^{h}_{i,j}(1)u^{h,*}_{i+1,j} + A^{h}_{i,j}(3)u^{h,*}_{i-1,j} + A^{h}_{i,j}(5)u^{h,*}_{i+1,j+1} + A^{h}_{i,j}(6)u^{h,*}_{i-1,j+1} + A^{h}_{i,j}(7)u^{h,*}_{i-1,j-1} + A^{h}_{i,j}(8)u^{h,*}_{i+1,j-1}),$$

for the fine grid y-direction red lines $i = 2, 4, \ldots, (N_x - 2)$.
- Solve a tridiagonal linear system of the order ($N_y/2 - 1$)-by-($N_y/2 - 1$) for each y-direction line on the coarse grid $\Omega^4_{2h_x,2h_y}$, i.e.,

$$A^{2h}_{i,j}(0)u^{2h,k+1}_{i,j} + A^{2h}_{i,j}(2)u^{2h,k+1}_{i,j+1} + A^{2h}_{i,j}(4)u^{2h,k+1}_{i,j-1} = F^{2h}_{i,j} - (A^{2h}_{i,j}(1)u^{2h,*}_{i+1,j} + A^{2h}_{i,j}(3)u^{2h,*}_{i-1,j} + A^{2h}_{i,j}(5)u^{2h,*}_{i+1,j+1} + A^{2h}_{i,j}(6)u^{2h,*}_{i-1,j+1} + A^{2h}_{i,j}(7)u^{h,*}_{i-1,j-1} + A^{2h}_{i,j}(8)u^{2h,*}_{i+1,j-1}),$$

for the coarse grid y-direction lines $i = 1, 2, \ldots, (N_x/2 - 1)$.
- From $u^{h,k+1}_{2i,2j} \in \Omega^4_{h_x,h_y}$ and $u^{2h,k+1}_{i,j} \in \Omega^4_{2h_x,2h_y}$, we compute $\tilde{u}^{h,k+1}_{2i,2j} \in \Omega^6_{h_x,h_y}$ by (11).
- From $\tilde{u}^{h,k+1}_{2i,2j}$ and $\tilde{u}^{h,*}_{2i-1,j} \in \Omega^6_{h_x,h_y}$ with (14), we obtain $\tilde{u}^{h,k+1}_{2i,2j-1} \in \Omega^6_{h_x,h_y}$.
- From $\tilde{u}^{h,k+1}_{2i,j} \in \Omega^6_{h_x,h_y}$ with (14), by solving a tridiagonal linear system of the order ($N_x - 1$)-by-($N_x - 1$) for each black line, we obtain $\tilde{u}^{h,k+1}_{2i-1,j} \in \Omega^6_{h_x,h_y}$. $u^{h,k+1}_{2i-1,j} \in \Omega^4_{h_x,h_y}$ can be obtained in the same way.

The ADI iterations continue until a certain norm of the correction vector of the approximate solution is reduced to below a certain tolerance.

IV. NUMERICAL RESULTS

Four convection diffusion equations are solved using the sixth-order Richardson extrapolation discretization (computational) strategy discussed in the previous sections. Two of them are one-dimensional problems and the other two are two-dimensional problems. We compare the sixth-order Richardson extrapolation compact (REC) discretization (computational) strategy

TABLE I. Comparison of the maximum absolute errors and the CPU seconds of the solutions computed from the CCD, the REC, and the FOC schemes for solving the Problem 1.

	CCD			REC			FOC		
h	Error	CPU	Order	Error	CPU	Order	Error	CPU	Order
$\pi/4$	$4.16e-4$	0.00		$7.80e-4$	0.00		$1.03e-3$	0.00	
$\pi/8$	$4.34e-6$	0.00	6.5	$1.88e-5$	0.00	5.4	$6.01e-5$	0.00	4.1
$\pi/16$	$3.52e-8$	0.00	6.9	$3.38e-7$	0.00	5.8	$3.81e-6$	0.00	4.0
$\pi/32$	$2.76e-10$	0.01	7.0	$5.49e-9$	0.00	5.9	$2.30e-7$	0.00	4.0
$\pi/64$	$2.16e-12$	0.08	7.0	$8.68e-11$	0.01	6.0	$1.49e-8$	0.00	3.9
$\pi/128$	$1.82e-14$	0.27	6.9	$1.31e-12$	0.03	6.1	$9.29e-10$	0.01	4.0

with the CCD scheme of Chu and Fan [1], and in the two-dimensional case, also with the standard FOC scheme [16].

In the following tables with computational results, we report the maximum absolute errors of the computed solution with respect to the exact solution on a series of grids with different mesh sizes. We also report the CPU time in seconds needed to compute the approximate solution and the estimated accuracy order of a given discretization scheme. In the one-dimensional case, the CPU times are small and may not be very accurate for large mesh size h. All recorded CPU timings smaller than 0.01 seconds are reported as 0.00 second conveniently. Because the computational techniques are very accurate, we are unable to use very small mesh sizes in our double precision arithmetic. The codes are written in Fortran 77 programming language and run on a SUN Ultra 5 workstation.

In the two-dimensional case, we use the ADI iterations to solve all linear systems resulted from different discretization schemes. The iteration stopping criterion is such that the L_1 norm of the correction vector $corr^{k+1}$ of the approximate solution is less than 10^{-10}, where the correction in the $(k+1)$st iteration is defined as

$$corr^{k+1} = \sum_{i,j} |u_{i,j}^{k+1} - u_{i,j}^{k}|.$$

Problem 1. Consider the one-dimensional convection diffusion equation:

$$u_{xx} - u_x - u = -\cos x - 2\sin x, \qquad 0 \le x \le \pi,$$

with the Dirichlet boundary conditions specified as $u(0) = u(\pi) = 0$. The analytic solution of this problem is $u(x) = \sin x$. This is one of the problems studied by Chu and Fan in [1]. The computational results are listed in Table I, which show that, although the CCD scheme is more accurate for the given mesh size h, the REC scheme is cheaper to compute with. Thus the REC scheme can use a finer mesh size to compute a solution of *comparable accuracy* in *fewer* CPU seconds. For instance, the solution computed by the REC scheme using $h = 1/128$ is more accurate than that computed by the CCD scheme using $h = 1/64$. The REC scheme is faster than the CCD scheme in this comparison. The solution computed from the FOC scheme is less accurate than those from the CCD and REC schemes with the same mesh size.

Problem 2. We solve the convection diffusion equation:

TABLE II. Comparison of the maximum absolute errors and the CPU seconds of the solutions computed from the CCD, the REC, and the FOC schemes for solving the Problem 2.

	CCD			REC			FOC		
h	Error	CPU	Order	Error	CPU	Order	Error	CPU	Order
1/4	$7.37e-7$	0.00		$2.04e-7$	0.00		$2.07e-3$	0.00	
1/8	$1.47e-8$	0.00	5.6	$3.64e-9$	0.00	5.8	$1.26e-4$	0.00	4.0
1/16	$2.60e-10$	0.00	5.8	$6.10e-11$	0.00	5.9	$7.83e-6$	0.00	4.0
1/32	$4.34e-12$	0.01	5.9	$9.88e-13$	0.00	5.9	$4.90e-7$	0.00	4.0
1/64	$7.18e-14$	0.08	5.9	$3.66e-14$	0.01	4.8	$3.06e-8$	0.00	4.0
1/128	$2.79e-14$	0.27	1.4	$1.19e-14$	0.03	1.6	$1.91e-9$	0.01	4.0

$$u_{xx} - u_x = 0, \qquad 0 \le x \le 1,$$

with the Dirichlet boundary conditions specified as $u(0) = 0$ and $u(1) = 1$. The analytic solution of this problem is $u(x) = [\exp(x) - 1)/(\exp(1) - 1]$. This is a classical test problem. The test results are reported in Table II. We find that, with the same mesh size h, the REC scheme is slightly more accurate than the CCD scheme, unlike the results obtained with the Problem 1. Moreover, the REC scheme is several times faster than the CCD scheme with the *same* mesh size for solving the Problem 2. We believe that the solutions computed from the CCD and the REC schemes with $h = 1/128$ are not as accurate as they should be, because of the limited computer arithmetic precision.

Problem 3. Consider the Stommel ocean model as in [1],

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \alpha \frac{\partial u}{\partial x} = -\gamma \sin\left(\frac{\pi}{b} y\right), \qquad (x, y) \in \Omega = [0, \lambda] \times [0, b], \tag{15}$$

where the boundary conditions are

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

In this application, the two parameters α and γ are chosen as

$$\alpha = \frac{D\beta}{R}, \qquad \gamma = \frac{F\pi}{Rb}.$$

The analytic solution of Eq. (15) is given by

$$u = -\gamma\left(\frac{b}{\pi}\right)^2 \sin\left(\frac{\pi y}{b}\right)(pe^{Ax} + qe^{Bx} - 1),$$

where

$$A = -\frac{\alpha}{2} + \sqrt{\frac{\alpha^2}{4} + \left(\frac{\pi}{b}\right)^2}, \qquad B = -\frac{\alpha}{2} - \sqrt{\frac{\alpha^2}{4} + \left(\frac{\pi}{b}\right)^2},$$

TABLE III. Comparison of the maximum absolute errors and the CPU seconds of the CCD, the REC, and the FOC schemes for solving the Problem 3 with $\beta = 0$.

	CCD			REC			FOC		
n	Error	CPU	Order	Error	CPU	Order	Error	CPU	Order
4	$1.80e-3$	0.03		$2.42e-3$	0.00		$3.92e-3$	0.00	
8	$5.00e-5$	0.41	5.2	$6.69e-5$	0.03	5.2	$2.57e-4$	0.03	3.9
16	$1.07e-6$	6.44	5.5	$1.32e-6$	0.56	5.7	$1.63e-5$	0.44	4.0
32	$1.99e-8$	103.18	5.7	$2.27e-8$	10.18	5.9	$1.02e-6$	7.45	4.0
64	$3.39e-10$	1645.72	5.9	$3.68e-10$	162.87	5.9	$6.40e-8$	124.85	4.0

and

$$p = (1 - e^{B\lambda})/(e^{A\lambda} - e^{B\lambda}), \qquad q = 1 - p.$$

The physical parameters are chosen as [1]:

$$\lambda = 10^7 \text{ m}, \quad \beta = 0, \quad \text{or } 10^{-11} \text{ m}^{-1}\text{ s}^{-1}, \quad b = 2\pi \times 10^6 \text{ m}, \quad D = 200 \text{ m},$$
$$F = 0.3 \times 10^{-7} \text{ m}^2\text{s}^{-2}, \quad R = 0.6 \times 10^{-3} \text{ ms}^{-1}.$$

In the following, we define $N_x = N_y = n$. The mesh sizes h_x and h_y are equal to λ/n and b/n, respectively. In the first test with the Problem 3, we choose $\beta = 0$ as in [1], which is actually a Poisson equation. The comparison results are given in Table III.

From Table III we can see that the accuracy of the solutions computed by the CCD scheme and the REC scheme is comparable. Both solutions are about two orders of magnitude more accurate than that computed by the FOC scheme. In terms of computational cost (the CPU seconds) with the same mesh size h, the FOC scheme is the fastest. The REC scheme is several times faster than the CCD scheme.

For this test problem, if we use a coarser mesh size for the REC scheme, it is faster than the FOC scheme for computing a solution with comparable accuracy. This can be seen by comparing the solution computed by the REC scheme with $n = 32$ to that by the FOC scheme with $n = 64$.

When we choose $\beta = 10^{-11}$ (as in [1]), the accuracy of the solutions computed from the three difference schemes are comparable for small n (see Table IV). For large n, the REC scheme is seen to be the most accurate one with the same n. The CCD scheme is the most expensive one, and the FOC scheme is the least.

TABLE IV. Comparison of the maximum absolute errors and the CPU seconds of the CCD, the REC, and the FOC schemes for solving the Problem 3 with $\beta = 10^{-11}$.

	CCD			REC			FOC		
n	Error	CPU	Order	Error	CPU	Order	Error	CPU	Order
4	$7.36e-1$	0.24		$1.86e-1$	0.00		$2.00e-1$	0.00	
8	$2.53e-1$	0.64	1.5	$1.86e-2$	0.02	3.3	$2.03e-2$	0.01	3.3
16	$6.84e-4$	2.67	8.5	$1.04e-3$	0.11	4.2	$1.53e-3$	0.08	3.7
32	$3.73e-5$	12.75	4.2	$2.56e-5$	1.46	5.3	$1.01e-4$	1.14	3.9
64	$1.43e-6$	187.96	4.7	$5.19e-7$	23.53	5.6	$6.43e-6$	18.14	4.0

TABLE V. Comparison of the maximum absolute errors and the CPU seconds of the REC and the FOC schemes for solving the Problem 4.

	REC			FOC		
h	Error	CPU	Order	Error	CPU	Order
1/4	$1.35e-2$	0.00		$1.36e-2$	0.00	
1/8	$7.53e-4$	0.03	4.2	$1.00e-3$	0.02	3.7
1/16	$1.93e-5$	0.56	5.3	$6.78e-5$	0.43	3.9
1/32	$4.09e-7$	9.80	5.6	$4.30e-6$	7.39	4.0
1/64	$7.42e-9$	163.42	5.8	$2.70e-7$	122.33	4.0

Problem 4. We consider a general two dimensional convection diffusion equation

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + p\frac{\partial u}{\partial x} + q\frac{\partial u}{\partial y} = f, \qquad (x, y) \in \Omega = [0, 1] \times [0, 1]. \tag{16}$$

The exact solution is given as $u(x, y) = \cos(4x + 6y)$, and the convection coefficients are

$$p = 10x(x-1)(1-2y), \qquad q = -10y(y-1)(1-2x).$$

The Dirichlet boundary condition and the right-hand side function are given accordingly. Because the boundary value is nonzero, the CCD scheme, as it is published in [1], is not applicable. We compare the performance of the REC and the FOC schemes in Table V. For solving this problem, the REC scheme is more accurate than the FOC scheme with the same mesh size h, but the FOC scheme is faster. However, to compute a solution with comparable accuracy, the REC scheme is seen to be faster by using a coarser mesh size.

In the following, we compare the number of the ADI iterations with the CCD, the REC, and the FOC schemes. The logarithm of the number of iterations corresponding to the logarithm of the number of the grid points n are plotted in Figs. 2–4. We can see that the CCD scheme takes

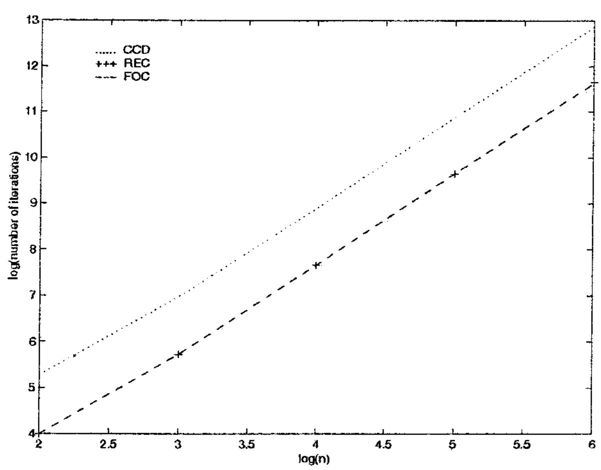

FIG. 2. Comparison of the number of the ADI iterations with the CCD, the REC, and the FOC schemes for solving the Problem 3 with $\beta = 0$.

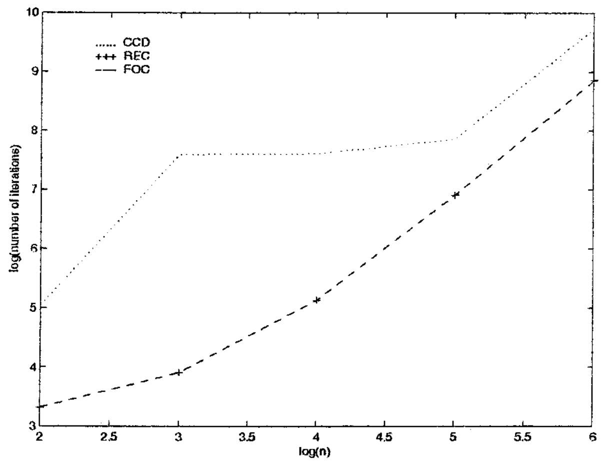

FIG. 3. Comparison of the number of the ADI iterations with the CCD, the REC, and the FOC schemes for solving the Problem 3 with $\beta = 10^{-11}$.

more iterations than the REC and the FOC schemes to converge. The convergence rates of the ADI method with the REC and FOC schemes are comparable.

V. CONCLUDING REMARKS

We presented a new sixth-order compact finite difference discretization strategy for solving the (one- and two-dimensional) convection diffusion equations. The new sixth-order compact discretization strategy results in solving tridiagonal linear systems, whereas the implicit CCD scheme needs to solve block tridiagonal linear systems of three times larger. Our numerical experiments with some one- and two-dimensional convection diffusion equations show that the

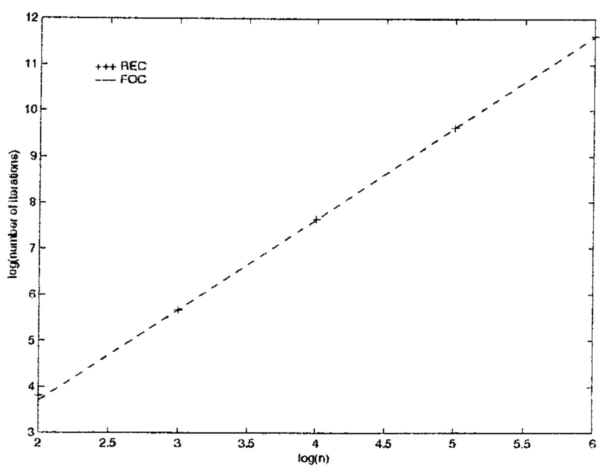

FIG. 4. Comparison of the number of the ADI iterations with the REC and the FOC schemes for solving the Problem 4.

new sixth-order discretization strategy may be advantageous, compared with the sixth-order CCD scheme and the fourth-order compact difference scheme.

We point out a major difference between the REC scheme and the CCD scheme. The REC scheme is explicit in the sense that only the approximate values of the solution function are computed. The CCD scheme is implicit in the sense that the approximate values of the first and second derivatives are computed as well as the approximate values of the solution function. For each grid point, there are three variables to be computed. This is the major reason why the CCD scheme is more expensive to compute with than the REC scheme.

We remark that the REC scheme that we implemented is based on the FOC scheme. There have been a few fast solvers (which are much faster than the ADI method) developed for solving the linear systems arising from the FOC scheme [3, 17]. These include the multigrid methods of various kinds and the preconditioned Krylov subspace methods. It is possible that some of these fast solvers may be used with the REC scheme to speedup the convergence rate of the linear system solvers in higher (two and three) dimensions. However, the computational algorithms associated with these fast solvers will be more complicated to implement than the tridiagonal linear system solver.

The two-grid implementation of the REC scheme is reminiscent of the multigrid methods used to solve discretized partial differential equations, in which the coarse grid correction is used to accelerate the convergence rate of the fine grid iterations [18, 19]. In the REC scheme, the coarse grid solution is used to improve the accuracy of the fine grid approximate solution. It may be possible to use the two-grid idea to achieve both goals, i.e., to improve the accuracy of the fine grid approximate solution and to speedup the convergence rate of some iterative methods used to solve the fine grid linear system. We are exploring this and other interesting ideas in the framework of the two-grid computations.

References

1. P. Chu and C. Fan, A three-point combined compact difference scheme, J Comput Phys 140 (1998), 370–399.
2. Y. Adam, Highly accurate compact implicit methods and boundary conditions, J Comput Phys 24 (1977), 10–22.
3. M. M. Gupta and J. Zhang, High accuracy multigrid solution of the 3D convection-diffusion equation, Appl Math Comput 113(2–3) (2000), 249–274.
4. M. M. Gupta, J. Kouatchou, and J. Zhang, Comparison of second- and fourth-order discretizations for multigrid Poisson solver, J Comput Phys 132 (1997), 123–129.
5. S. K. Lele, Compact finite difference schemes with spectral-like resolution, J Comput Phys 103 (1992), 16–42.
6. K. Sakurai, T. Aoki, W.-H. Lee, and K. Kato, Poisson equation solver with fourth-order accuracy by using interpolated differential operator scheme, Comput Math Applic 43 (2002), 621–630.
7. W. F. Spotz, High-order compact finite difference schemes for computational mechanics, Ph.D. Thesis, University of Texas at Austin, Austin, TX, 1995.
8. W. F. Spotz and G. F. Carey, High-order compact scheme for the steady stream-function vorticity equations, Int J Numer Methods Engrg 38 (1995), 3497–3512.
9. J. Zhang, An explicit fourth-order compact finite difference scheme for three dimensional convection-diffusion equation, Commun Numer Methods Engrg 14 (1998), 209–218.
10. J. Zhang, L. Ge, and J. Kouatchou, A two colorable fourth-order compact difference scheme and parallel iterative solution of the 3D convection diffusion equation, Math Comput Simulation 54(1–3) (2000), 65–80.

11. J. Zhang and J. J. Zhao, High accuracy stable numerical solution of 1D microscale heat transport equation, Commun Numer Methods Engrg 17 (2001), 821–832.

12. D. Peaceman and H. Rachford, The numerical solution of elliptic and parabolic differential equations, J SIAM 3 (1955), 28–42.

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

14. J. Zhang, H. Sun, and J. J. Zhao, High order compact scheme with multigrid local mesh refinement procedure for convection diffusion problems, Comput Methods Appl Mech Engrg 191 (2002), 4661–4674.

15. W. Cheney and D. Kincard, Numerical mathematics and computing, 4th Ed., Brooks/Cole Publishing, Pacific Grove, CA, 1999.

16. M. M. Gupta, R. P. Manohar, and J. W. Stephenson, High-order difference schemes for two-dimensional elliptic equations, Numer Methods Partial Differential Eq 1 (1985), 71–80.

17. J. Zhang, Preconditioned iterative methods and finite difference schemes for convection-diffusion, Appl Math Comput 109(1) (2000), 11–30.

18. W. L. Briggs, V. E. Henson, and S. F. McCormick, A multigrid tutorial, 2nd Ed., SIAM, Philadelphia, PA, 2000.

19. P. Wesseling, An introduction to multigrid methods, Wiley, Chichester, England, 1992.