JOURNAL OF COMPUTATIONAL PHYSICS **140,** 370–399 (1998)
ARTICLE NO. CP985899

A Three-Point Combined Compact Difference Scheme

Peter C. Chu and Chenwu Fan

Department of Oceanography, Naval Postgraduate School, Monterey, California 93943
E-mail: chu@nps.navy.mil

Received February 12, 1997; revised December 3, 1997

A new three-point combined compact difference (CCD) scheme is developed for numerical models. The major features of the CCD scheme are: three point, implicit, sixth-order accuracy, and inclusion of boundary values. Due to its combination of the first and second derivatives, the CCD scheme becomes more compact and more accurate than normal compact difference schemes. The efficient twin-tridiagonal (for calculating derivatives) and triple-tridiagonal (for solving partial difference equation with the CCD scheme) methods are also presented. Besides, the CCD scheme has sixth-order accuracy at periodic boundaries and fifth-order accuracy at nonperiodic boundaries. The possibility of extending to a three-point eighth-order scheme is also included.

1. INTRODUCTION

The grid spacings $(\Delta x, \Delta y)$ in most ocean numerical models are not small. For example, a global ocean model is considered having high resolution when a horizontal grid is $(1/8)^\circ$, approximately 14.5 km. For such large grid spacing, use of highly accurate difference scheme becomes urgent. For example, McCalpin [1] used fourth-order differencing to reduce pressure gradient error in σ-coordinate ocean models.

The trend toward highly accurate numerical schemes of partial differential equations (PDE) has recently led to a renewed interest in compact difference schemes. Concurrently, Adam [2], Hirsh [3], and Kreiss [4] have proposed Hermitian compact techniques using less nodes (three instead of five) at each grid point to solve PDE. Later on, as pointed out by Adam [5], the truncation errors are usually four to six times smaller than the same order noncompact schemes. Since then, much work has been done in developing compact schemes for various applications, such as: an implicit compact fourth-order algorithm [6]; a fourth-order compact difference scheme for nonuniform grids [7]; fourth-order and sixth-order compact difference schemes for the staggered grid [8]; an early form of the sixth-order

0021-9991/98 $25.00

combined compact difference scheme [9]; compact finite difference schemes with a range of spatial scales [10]; and an upwind fifth-order compact scheme [11]. These schemes are characterized by (a) 5-point sixth-order, (b) much lower accuracy at nodes adjacent to boundaries, and (c) no requirement on PDE to be satisfied at boundaries.

Several recent work emphasizes on the improvement of boundary accuracy. For hyperbolic system, Carpenter *et al*. [12, 13] introduced a simultaneous approximation term (SAT) method that solves a linear combination of the boundary conditions and the hyperbolic equations near the boundary. This method provides fourth-order accuracy at both interior and boundary. Under the assumption that the derivative operator admits a summation-by-parts formula then the SAT method is stable in the classical sense and is also time-stable. For 2D vorticity-stream function formulation, E and Liu [14, 15] proposed a finite difference scheme with fourth-order accuracy at both interior and boundary. Question arises: can we construct a scheme (1) working for any differential equation and (2) with high-order accuracy at both interior and boundary?

A new three-point sixth-order combined compact (CCD) scheme is such a scheme with the following features: (a) 3-point sixth-order, (b) comparable accuracy at nodes adjacent to boundaries, and (c) requirement on PDE to be satisfied at boundaries. Fourier analysis of errors is used to prove the CCD scheme as having better resolution characteristics than any current (uncompact and compact) scheme. Two implicit solvers for the CCD scheme are also proposed for calculating various differences (twin-tridiagonal solver) and for solving PDEs (triple-tridiagonal solver). Furthermore, we use the one-dimensional convection-diffusion equation and two-dimensional Stommel ocean model to illustrate the application of the CCD solvers and to demonstrate the benefit of using CCD scheme.

2. CCD SCHEME

2.1. General CCD Algorithm

Let the dependent variable $f(x)$ be defined on the interval, $0 \le x \le L$. Use a uniform grid, $0 = x_1 < x_2 < x_3 < \cdots < x_N < x_{N+1} = L$ with a spacing $h = x_{i+1} - x_i = L/N$. Let the dependent variable $f(x)$ at any grid point x_i and two neighboring points x_{i-1} and x_{i+1} be given by f_i, f_{i-1}, and f_{i+1} and let its derivatives at the two neighboring points x_{i-1} and x_{i+1} be given by $f'_{i-1}, f''_{i-1}, \ldots, f^{(k)}_{i-1}$ and $f'_{i+1}, f''_{i+1}, \ldots, f^{(k)}_{i+1}$. The essence of the CCD scheme is to relate $f_i, f'_i, f''_i, \ldots, f^{(k)}_i$ to the two neighboring points: $f_{i-1}, f'_{i-1}, f''_{i-1}, \ldots, f^{(k)}_{i-1}$ and $f_{i+1}, f'_{i+1}, f''_{i+1}, \ldots, f^{(k)}_{i+1}$,

$$
\begin{aligned}
&\left(\frac{\delta f}{\delta x}\right)_i + \alpha_1\left(\left(\frac{\delta f}{\delta x}\right)_{i+1} + \left(\frac{\delta f}{\delta x}\right)_{i-1}\right) + \beta_1 h\left(\left(\frac{\delta^2 f}{\delta x^2}\right)_{i+1} - \left(\frac{\delta^2 f}{\delta x^2}\right)_{i-1}\right) + \cdots \\
&\quad = \frac{a_1}{2h}(f_{i+1} - f_{i-1}) \\
&\left(\frac{\delta^2 f}{\delta x^2}\right)_i + \alpha_2\left(\left(\frac{\delta^2 f}{\delta x^2}\right)_{i+1} + \left(\frac{\delta^2 f}{\delta x^2}\right)_{i-1}\right) + \beta_2\frac{1}{2h}\left(\left(\frac{\delta f}{\delta x}\right)_{i+1} - \left(\frac{\delta f}{\delta x}\right)_{i-1}\right) + \cdots \\
&\quad = \frac{a_2}{h^2}(f_{i+1} - 2f_i + f_{i-1}) \\
&\quad \cdots
\end{aligned}
\tag{2.1}
$$

and to compute $f_i', f_i'', \ldots, f_i^{(k)}$ by means of the values and derivatives at the two neighboring points. Moving from the one boundary to the other, CCD forms a global algorithm to compute various derivatives at all grid points. In this paper we only discuss the sixth-order CCD scheme.

2.2. Local Hermitian Polynomial

Let $H_i(x)$ be a local Hermitian polynomial defined on the closed interval $[x_{i-1}, x_{i+1}]$, representing the variable f at x_i and f and its derivatives f', f'' at the two neighboring points x_{i-1}, and x_{i+1},

$$\begin{aligned} H_i(x_{i-1}) = f_{i-1}, \quad H_i(x_i) = f_i, \quad H_i(x_{i+1}) = f_{i+1}, \\ H_i'(x_{i-1}) = f_{i-1}', \quad H_i'(x_{i+1}) = f_{i+1}', \quad H_i''(x_{i-1}) = f_{i-1}'', \quad H_i''(x_{i+1}) = f_{i+1}''. \end{aligned} \tag{2.2}$$

Expand $H_i(x)$ into Taylor series in the neighborhood of x_i with sixth-order accuracy

$$\begin{aligned} H_i(x) = H_i(x_i) + H_i'(x_i)x + \frac{H_i''(x_i)}{2!}x^2 + \frac{H_i^{(3)}(x_i)}{3!}x^3 + \frac{H_i^{(4)}(x_i)}{4!}x^4 \\ + \frac{H_i^{(5)}(x_i)}{5!}x^5 + \frac{H_i^{(6)}(x_i)}{6!}x^6. \end{aligned} \tag{2.3}$$

The seven coefficients in (2.3) are determined by the seven equations in (2.2),

$$\begin{aligned} H_i'(x_i) &= \frac{15}{16h}(f_{i+1} - f_{i-1}) - \frac{7}{16}(f_{i+1}' + f_{i-1}') + \frac{h}{16}(f_{i+1}'' - f_{i-1}'') \\ H_i''(x_i) &= \frac{3}{h^2}(f_{i+1} - 2f_i + f_{i-1}) - \frac{9}{8h}(f_{i+1}' - f_{i-1}') + \frac{1}{8}(f_{i+1}'' + f_{i-1}'') \\ H_i^{(3)}(x_i) &= -\frac{15}{4h^3}(f_{i+1} - f_{i-1}) + \frac{15}{4h^2}(f_{i+1}' + f_{i-1}') - \frac{3}{4h}(f_{i+1}'' - f_{i-1}'') \\ H_i^{(4)}(x_i) &= -\frac{36}{h^4}(f_{i+1} - 2f_i + f_{i-1}) + \frac{21}{h^3}(f_{i+1}' - f_{i-1}') - \frac{3}{h^2}(f_{i+1}'' + f_{i-1}'') \\ H_i^{(5)}(x_i) &= \frac{45}{2h^5}(f_{i+1} - f_{i-1}) - \frac{45}{2h^4}(f_{i+1}' + f_{i-1}') + \frac{15}{2h^3}(f_{i+1}'' - f_{i-1}'') \\ H_i^{(6)}(x_i) &= \frac{360}{h^6}(f_{i+1} - 2f_i + f_{i-1}) - \frac{225}{h^5}(f_{i+1}' - f_{i-1}') + \frac{45}{h^4}(f_{i+1}'' + f_{i-1}''). \end{aligned} \tag{2.4}$$

The kth derivative at the grid point x_i is approximately given by

$$f^{(k)}(x_i) \simeq H_i^{(k)}(x_i). \tag{2.5}$$

Substitution of (2.5) into (2.4) leads to

$$\begin{aligned} \frac{7}{16}(f_{i+1}' + f_{i-1}') + f_i' - \frac{h}{16}(f_{i+1}'' - f_{i-1}'') &= \frac{15}{8}\frac{1}{2h}(f_{i+1} - f_{i-1}) - \frac{1349}{7781760}f_i^{(7)}h^6 \\ \frac{9}{8h}(f_{i+1}' - f_{i-1}') - \frac{1}{8}(f_{i+1}'' + f_{i-1}'') + f_i'' &= 3\frac{1}{h^2}(f_{i+1} - 2f_i + f_{i-1}) - \frac{1}{20160}f_i^{(8)}h^6 \end{aligned} \tag{2.6}$$

which are the schemes for computing the first-order and second-order derivatives at the grid point x_i, respectively. Thus, the CCD scheme with sixth-order accuracy can be written by

$$\frac{7}{16}\left(\left(\frac{\delta f}{\delta x}\right)_{i+1}+\left(\frac{\delta f}{\delta x}\right)_{i-1}\right)+\left(\frac{\delta f}{\delta x}\right)_i-\frac{h}{16}\left(\left(\frac{\delta^2 f}{\delta x^2}\right)_{i+1}-\left(\frac{\delta^2 f}{\delta x^2}\right)_{i-1}\right)$$
$$=\frac{15}{16h}(f_{i+1}-f_{i-1}) \tag{2.7}$$

which is for the first derivative calculation, and

$$\frac{9}{8h}\left(\left(\frac{\delta f}{\delta x}\right)_{i+1}-\left(\frac{\delta f}{\delta x}\right)_{i-1}\right)-\frac{1}{8}\left(\left(\frac{\delta^2 f}{\delta x^2}\right)_{i+1}+\left(\frac{\delta^2 f}{\delta x^2}\right)_{i-1}\right)+\left(\frac{\delta^2 f}{\delta x^2}\right)_i$$
$$=\frac{3}{h^2}(f_{i+1}-2f_i+f_{i-1}) \tag{2.8}$$

which is for the second derivative calculation. Comparing (2.7) with (2.1), we find that the parameters in (2.1) for the sixth-order scheme should be

$$\alpha_1=\frac{7}{16},\quad \beta_1=-\frac{1}{16},\quad a_1=\frac{15}{8},\quad \alpha_2=-\frac{1}{8},\quad \beta_2=\frac{9}{4},\quad a_2=3.$$

For the sixth-order CCD scheme, the truncation errors in (2.6)

$$\frac{1349}{7781760}f_i^{(7)}h^6\approx 1.73*10^{-4}f_i^{(7)}h^6,\quad \frac{1}{20160}f_i^{(8)}h^6\approx 4.9*10^{-5}f_i^{(8)}h^6$$

are quite small.

Another benefit of using CCD scheme is the existence of a global Hermitian polynomial with continuous first- and second-order derivatives at each grid point. We will describe it in Appendix 1.

2.3. Error Estimation

We compare the truncation errors between the CCD scheme with current generalized schemes [10] for first-order derivatives,

$$f_i'+\alpha(f_{i+1}'+f_{i-1}')+\beta(f_{i+2}'+f_{i-2}')=a\frac{f_{i+1}-f_{i-1}}{2h}+b\frac{f_{i+2}-f_{i-2}}{4h}+c\frac{f_{i+3}-f_{i-3}}{6h} \tag{2.9}$$

and the second-order derivatives,

$$f_i''+\alpha(f_{i+1}''+f_{i-1}'')+\beta(f_{i+2}''+f_{i-2}'')$$
$$=a\frac{f_{i+1}-2f_i+f_{i-1}}{h^2}+b\frac{f_{i+2}-2f_i+f_{i-2}}{4h^2}+\frac{f_{i+3}-2f_i+f_{i-3}}{9h^2}, \tag{2.10}$$

where the parameters α, β, a, b, c take different values for various schemes (Table 1). The comparison of truncation errors is listed in the last column in Table 1. We find that the CCD scheme has the smallest truncation error among various sixth-order schemes. For

TABLE 1
Truncation Errors in Various Difference Schemes for the First and Second Derivative Calculations

Derivative approximation	Eq.	Scheme	Parameter					Truncation error
					a	b	c	
First	(2.12)	2nd-order central	0	0	1	0	0	$\frac{1}{3!} f^{(3)} h^2$
	(2.12)	Standard Padé scheme	$\frac{1}{4}$	0	$\frac{3}{2}$	0	0	$\frac{-1}{5!} f^{(5)} h^4$
	(2.12)	6th-order central	0	0	$\frac{3}{2}$	$\frac{-3}{5}$	$\frac{1}{10}$	$36 \times \frac{1}{7!} f^{(7)} h^6$
	(2.12)	6th-order tridiagonal	$\frac{1}{3}$	0	$\frac{14}{9}$	$\frac{1}{9}$	0	$4 \times \frac{1}{7!} f^{(7)} h^6$
	(2.12)	6th-order pentadiagonal	$\frac{17}{57}$	$\frac{-1}{144}$	$\frac{90}{57}$	0	0	$\frac{-100}{19} \times \frac{1}{7!} f^{(7)} h^6$
	(2.7)	6th-order CCD	/	/	/	/	/	$\frac{-1349}{1544} \times \frac{1}{7!} f^{(7)} h^6$
Second	(2.13)	2nd-order central	0	0	1	0	0	$2 \times \frac{1}{4!} f^{(4)} h^2$
	(2.13)	Standard Padé scheme	$\frac{1}{10}$	0	$\frac{6}{5}$	0	0	$\frac{18}{5} \times \frac{1}{6!} f^{(6)} h^4$
	(2.13)	6th-order central	0	0	$\frac{3}{2}$	$\frac{-3}{5}$	$\frac{1}{10}$	$72 \times \frac{1}{8!} f^{(8)} h^6$
	(2.13)	6th-order tridiagonal	$\frac{2}{11}$	0	$\frac{12}{11}$	$\frac{3}{11}$	0	$\frac{-184}{11} \times \frac{1}{8!} f^{(8)} h^6$
	(2.13)	6th-order pentadiagonal	$\frac{12}{97}$	$\frac{-1}{194}$	$\frac{120}{97}$	0	0	$\frac{-2672}{97} \times \frac{1}{8!} f^{(8)} h^6$
	(2.8)	6th-order CCD	/	/	/	/	/	$-2 \times \frac{1}{8!} f^{(8)} h^6$

example, the truncation error of the first derivative using the CCD scheme is about 41.2 times smaller than using the sixth-order central scheme, 4.6 times smaller than using the sixth-order tridiagonal (compact) schemes, and 6.0 times smaller than using the sixth-order pentadiagonal (compact) scheme. The truncation error of the second derivative using the CCD scheme is about 36 times smaller than using the sixth-order central scheme, 8.4 times smaller than using the sixth-order tridiagonal scheme (compact), and 13.8 times smaller than using the sixth-order pentadiagonal scheme (compact). Comparing the CCD scheme with the second-order central difference (SCD) scheme (most commonly used in ocean models), truncation errors for both first and second derivatives are more than four orders of magnitude smaller.

Another good feature of the CCD scheme is that the CCD scheme uses the same formulation at all grid points except at the boundaries, where some additional boundary treatment is formulated. These additional schemes at the boundaries are fifth-order accurate for the PDE with the CCD scheme (see Section 5). A CCD scheme with eighth-order accuracy will be presented in Appendix 2.

3. FOURIER ANALYSIS OF ERRORS

Fourier analysis of errors is commonly used to evaluate various difference schemes, described extensively in Swartz and Wendroff [16], Oliger and Kreiss [17], Vichnevetsky

and Bowles [18], Roberts and Weiss [19], Fromm [20], Orszag [21, 22], and Lele [10]. As pointed out by Lele [10], Fourier analysis provides an effective way to quantify the resolution characteristics of differencing approximations.

For the purpose of Fourier analysis the dependent variable $f(x)$ is assumed to be periodic over the domain $[0, L]$ of the independent variable, i.e., $f_1 = f_{N+1}$ and $h = L/N$. The dependent variable may decomposed into Fourier series,

$$f(x) = \sum_{k=-N/2}^{k=N/2} \hat{f}_k e^{(2\pi i k x/L)}, \tag{3.1}$$

where $i = \sqrt{-1}$. It is convenient to introduce a scaled wavenumber $w = 2\pi kh/L = 2\pi k/N$, and a scaled coordinate $s = x/h$. The Fourier modes in terms of these are simply $\exp(iws)$. The exact first-order and second-order derivatives of (3.1) generate a function with exact Fourier coefficients

$$\hat{f}'_k = \frac{iw}{h}\hat{f}_k, \qquad \hat{f}''_k = -\left(\frac{w}{h}\right)^2 \hat{f}_k.$$

However, the Fourier coefficients of the derivatives obtained from the differencing scheme might not be the same as the exact Fourier coefficients, i.e.,

$$(\hat{f}'_k)_{fd} = \frac{iw'}{h}\hat{f}_k, \qquad (\hat{f}''_k)_{fd} = -\left(\frac{w''}{h}\right)^2 \hat{f}_k,$$

where $w' = w'(w)$ and $w'' = w''(w)$ are the modified wavenumber (both real numbers) for the first-order and second-order differencing. The smaller the difference between the exact and modified wavenumbers, the better the difference scheme.

According to Lele [10], the modified wavenumbers of the current generalized difference schemes (2.9) and (2.10) are

$$w'(w) = \frac{a \sin w + \frac{b}{2}\sin 2w + \frac{c}{3}\sin 3w}{1 + 2\alpha \cos w + 2\beta \cos 2w} \tag{3.2}$$

and

$$w''(w) = \sqrt{\frac{2a(1 - \cos w) + \frac{b}{2}(1 - \cos 2w) + \frac{2c}{9}(1 - \cos 3w)}{1 + 2\alpha \cos w + 2\beta \cos 2w}}, \tag{3.3}$$

respectively.

For the CCD schemes (2.7) and (2.8), the modified wavenumbers w' and w'' can be calculated jointly as follows:

$$f(x) = \sum_k \hat{f}_k e^{(iw(x/h))} \tag{3.4}$$

$$f'(x) = \sum_k \hat{f}'_k e^{(iw(x/h))} \tag{3.5}$$

$$f''(x) = \sum_k \hat{f}''_k e^{(iw(x/h))} \tag{3.6}$$

and

$$[f'(x)]_{fd} = \sum_k (\hat{f}'_k)_{fd} e^{(iw(x/h))} \tag{3.7}$$

$$[f''(x)]_{fd} = \sum_k (\hat{f}''_k)_{fd} e^{(iw(x/h))} \tag{3.8}$$

$$f(x+h) = \sum_k \hat{f}_k e^{(iw(x/h))} e^{iw} \tag{3.9}$$

$$f(x-h) = \sum_k \hat{f}_k e^{(iw(x/h))} e^{-iw} \tag{3.10}$$

$$[f'(x+h)]_{fd} = \sum_k (\hat{f}'_k)_{fd} e^{(iw(x/h))} e^{iw} \tag{3.11}$$

$$[f'(x-h)]_{fd} = \sum_k (\hat{f}'_k)_{fd} e^{(iw(x/h))} e^{-iw} \tag{3.12}$$

$$[f''(x+h)]_{fd} = \sum_k (\hat{f}'_k)_{fd} e^{(iw(x/h))} e^{iw} \tag{3.13}$$

$$[f''(x-h)]_{fd} = \sum_k (\hat{f}'_k)_{fd} e^{(iw(x/h))} e^{-iw}. \tag{3.14}$$

Substitution of (3.4)–(3.14) into (2.7)–(2.8), we have

$$\frac{7}{8}[\cos w + 1]w' + \frac{1}{8}\sin w (w'')^2 = \frac{15}{8}\sin w \tag{3.15}$$

$$-\frac{9}{4}(\sin w)w' - \left[1 - \frac{1}{4}\cos w\right](w'')^2 = 6[\cos w - 1]. \tag{3.16}$$

Solving (3.15)–(3.16), we have

$$w'(w) = \frac{9\sin w[4+\cos w]}{24 + 20\cos w + \cos 2w} \tag{3.17}$$

$$w''(w) = \sqrt{\frac{81 - 48\cos w - 33\cos 2w}{48 + 40\cos w + 2\cos 2w}}. \tag{3.18}$$

Among various difference schemes, the modified wavenumbers of the first-order differencing w' (Fig. 1a) and of the second-order differencing w'' (Fig. 1b) of the CCD scheme are closest to the exact wavenumber w.

In multidimensional problems the phase error of first-order differencing scheme appear in the form of anisotropy [10, 18],

$$(C'_p)_{fd}(w,\theta) \equiv w'(w,\theta)/w = \frac{(\cos\theta)w'(w\cos\theta) + (\sin\theta)w'(w\sin\theta)}{w}. \tag{3.19}$$

Figure 1c shows polar plots of phase speed anisotropy of various schemes for first derivative approximations. The phase speed for wavenumber (magnitude) $w/\pi = \frac{1}{50}, \frac{5}{50}, \ldots, \frac{45}{50}, \frac{50}{50}$ are plotted. Here, we also see that the CCD scheme shows improvement.

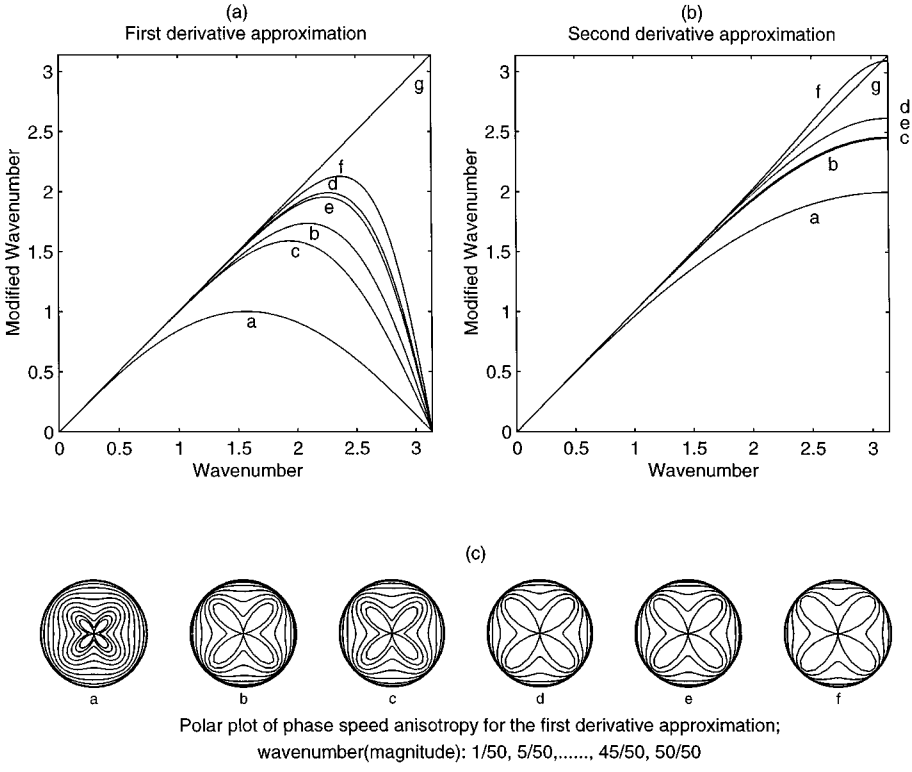

FIG. 1. Fourier analysis of error for derivative approximation: (a) second-order central scheme; (b) standard Padé scheme; (c) sixth-order central scheme; (d) sixth-order tridiagonal scheme; (e) sixth-order pentadiagonal scheme; (f) combined compact scheme; (g) exact differentiation.

4. CCD FOR DERIVATIVE CALCULATIONS

The previous section shows that the sixth-order 3-point CCD scheme is more accurate than any other sixth-order scheme including ordinary compact schemes. Nevertheless, since the CCD scheme is implicit and combines computation between the first-order and second-order differences, we should compute f' and f'' jointly and globally.

An efficient and implicit CCD solver is designed to calculate the first-order and second-order differences. Since CCD is a 3-point scheme, the difference calculation at x_i needs to use f, f', and f'' at the two neighboring points x_{i-1} and x_{i+1}. At the two boundaries x_1 and x_{N+1}, some specific treatment should be included in the CCD scheme.

4.1. Non-Periodic Boundaries

At both boundaries, $x = x_1$ and $x = x_{N+1}$, we propose a fourth-order one-sided CCD scheme instead of the two-sided scheme to keep 3-point structure,

$$\left(\frac{\delta f}{\delta x}\right)_1 + \alpha_1 \left(\frac{\delta f}{\delta x}\right)_2 + \beta_1 h \left(\frac{\delta^2 f}{\delta x^2}\right)_2 = \frac{1}{h}(a_1 f_1 + b_1 f_2 + c_1 f_3) \tag{4.1}$$

$$h\left(\frac{\delta^2 f}{\delta x^2}\right)_1 + \alpha_2 h\left(\frac{\delta^2 f}{\delta x^2}\right)_2 + \beta_2\left(\frac{\delta f}{\delta x}\right)_2 = \frac{1}{h}(a_2 f_1 + b_2 f_2 + c_2 f_3) \tag{4.2}$$

$$\left(\frac{\delta f}{\delta x}\right)_{N+1} + \alpha_1\left(\frac{\delta f}{\delta x}\right)_N - \beta_1 h\left(\frac{\delta^2 f}{\delta x^2}\right)_N = -\frac{1}{h}(a_1 f_{N+1} + b_1 f_N + c_1 f_{N-1}) \tag{4.3}$$

$$h\left(\frac{\delta^2 f}{\delta x^2}\right)_{N+1} + \alpha_2 h\left(\frac{\delta^2 f}{\delta x^2}\right)_N - \beta_2\left(\frac{\delta f}{\delta x}\right)_N = \frac{1}{h}(a_2 f_{N+1} + b_2 f_N + c_2 f_{N-1}), \tag{4.4}$$

where

$$\alpha_1 = 2, \quad \beta_1 = -1, \quad a_1 = -7/2, \quad b_1 = 4, \quad c_1 = -1/2,$$
$$\alpha_2 = 5, \quad \beta_2 = -6, \quad a_2 = 9, \quad b_2 = -12, \quad c_2 = 3.$$

At the boundaries, the first-order difference, represented by (4.1) and (4.3), has a truncation error of $-\frac{22}{5!} f^{(5)} h^4$. The second-order difference, represented by (4.2) and (4.4), has a truncation error of $-\frac{14}{5!} f^{(5)} h^4$. The accuracy at both boundaries can be further improved to fifth or sixth order.

The global CCD system, consisting of (4.1) and (4.2) for $i = 1$, (2.7) and (2.8) for $i = 2, 3, 4, \ldots, N$, and (4.3) and (4.4) for $i = N + 1$, is a well-posed system since it has $2(N + 1)$ equations with $2(N + 1)$ unknowns: $(\delta f/\delta x)_i$, $(\delta^2 f/\delta x^2)_i$, $i = 1, 2, 3, \ldots, N$, $N + 1$. We may write the $2(N + 1)$ equations (4.1)–(4.4), (2.7), and (2.8) into a more general form (global CCD system),

$$a_i^j(1)\left(\frac{\delta f}{\delta x}\right)_{i-1} + a_i^j(2)\left(\frac{\delta f}{\delta x}\right)_i + a_i^j(3)\left(\frac{\delta f}{\delta x}\right)_{i+1} + b_i^j(1)\left(\frac{\delta^2 f}{\delta x^2}\right)_{i-1}$$
$$+ b_i^j(2)\left(\frac{\delta^2 f}{\delta x^2}\right)_i + b_i^j(3)\left(\frac{\delta^2 f}{\delta x^2}\right)_{i+1} = s_i^j, \quad j = 1, 2, \tag{4.5}$$

with

$$a_1^j(1) = b_1^j(1) = a_{N+1}^j(3) = b_{N+1}^j(3) = 0, \quad j = 1, 2, \tag{4.6}$$

representing the four boundary equations (4.1)–(4.4). Here, $j = 1$ corresponds to the first-order derivative computation (2.7), and $j = 2$ corresponds to the second-order derivative computation (2.8). The two variables s_i^1 and s_i^2 are source terms.

The $2(N + 1) \times 2(N + 1)$ coefficient matrix of (4.5) has a twin-tridiagonal structure and can be directly solved by two steps: twin-forward elimination and twin-backward substitution (see Appendix 3).

4.2. Periodic Boundaries

For periodic boundaries, we have

$$f_0 = f_N, \quad f_1 = f_{N+1}, \quad f_0' = f_N', \quad f_1' = f_{N+1}', \quad f_0'' = f_N'', \quad f_1'' = f_{N+1}''. \tag{4.7}$$

Thus, the global CCD system, consisting of (2.7) and (2.8) for $i = 1, 2, 3, \ldots, N$, is well-posed since it has $2N$ equations with $2N$ unknowns: $(\delta f/\delta x)_i$, $(\delta^2 f/\delta x^2)_i$, $i = 1, 2, 3, \ldots, N$. The coefficient matrix and related algorithm are listed in Appendix 4.

5. CCD FOR SOLVING FINITE DIFFERENCE EQUATIONS (FDE)

Any PDE discretized by the CCD scheme (called here the CCD FDE) can only be solved globally since the CCD scheme is implicit. Unlike any other schemes, the CCD FDE solver requires the satisfaction of the FDE not only on the interior points, but also on the boundary nodes. Benefits of such a treatment are to decrease the truncation errors near the boundaries as well as to increase the global accuracy. Here, we propose a triple-tridiagonal solver for solving CCD FDE.

5.1. Nonperiodic Boundaries

Consider a one-dimensional differential equation,

$$a_1(x)\frac{df}{dx} + a_2(x)\frac{d^2f}{dx^2} + a_0(x)\,f(i) = s(x), \quad 0 \le x \le L, \tag{5.1}$$

with general boundary conditions

$$d_1(x)\,f'(x) + d_0(x)\,f(x) = c(x) \quad \text{at } x = 0;\ x = L, \tag{5.2}$$

which is the Dirichlet boundary condition when $d_0 = 1$, $d_1 = 0$ and the Neumann boundary condition when $d_0 = 0$, $d_1 = 1$.

The corresponding FDE can be written as

$$a_1(i)\left(\frac{\delta f}{\delta x}\right)_i + a_2(i)\left(\frac{\delta^2 f}{\delta x^2}\right)_i + a_0(i)\,f_i = s_i, \quad i = 1, 2, \ldots, N+1, \tag{5.3}$$

and the boundary conditions become

$$d_1^l\left(\frac{\delta f}{\delta x}\right)_1 + d_0^l f_1 = c^l, \quad d_1^r\left(\frac{\delta f}{\delta x}\right)_{N+1} + d_0^r f_{N+1} = c^r. \tag{5.4}$$

Notice that we applied the FDE (5.3) not only to the interior points but also to the two boundary points (x_1 and x_{N+1}). At each interior grid node i ($2 \le i \le N$) we have three equations [(5.3), (2.7), and (2.8)] with three unknown variables f_i, $(\delta f/\delta x)_i$, $(\delta^2 f/\delta x^2)_i$. However, we have only two equations [(5.3) and (5.4)] at both boundaries but three unknowns: f_1, $(\delta f/\delta x)_1$, $(\delta^2 f/\delta x^2)_1$ for the left boundary, and f_{N+1}, $(\delta f/\delta x)_{N+1}$, $(\delta^2 f/\delta x^2)_{N+1}$ for the right boundary. To close the system we need an extra condition for both the left and right boundaries.

The additional boundary conditions are obtained by constructing a new fifth-order polynomial,

$$P(x) = P_0 + P_1 x + P_2 x^2 + P_3 x^3 + P_4 x^4 + P_5 x^5. \tag{5.5}$$

For the left boundary, the six coefficients of $P(x)$ can be obtained by

$$P(x_1) = f_1, \quad P(x_2) = f_2, \quad P(x_3) = f_3, \quad P'(x_1) = f_1', \quad P'(x_2) = f_2', \quad P''(x_2) = f_2''. \tag{5.6}$$

The additional left boundary condition with fifth-order accuracy is then (Appendix 5)

$$14\left(\frac{\delta f}{\delta x}\right)_1 + 16\left(\frac{\delta f}{\delta x}\right)_2 + 2h\left(\frac{\delta^2 f}{\delta x^2}\right)_1 - 4h\left(\frac{\delta^2 f}{\delta x^2}\right)_2 + \frac{1}{h}(31 f_1 - 32 f_2 + f_3) = 0 \quad (5.7)$$

and the additional right boundary condition with fifth-order accuracy is written as

$$14\left(\frac{\delta f}{\delta x}\right)_{N+1} + 16\left(\frac{\delta f}{\delta x}\right)_N - 2h\left(\frac{\delta^2 f}{\delta x^2}\right)_{N+1} + 4h\left(\frac{\delta^2 f}{\delta x^2}\right)_N - \frac{1}{h}(31 f_{N+1} - 32 f_N + f_{N-1}) = 0. \quad (5.8)$$

Thus, we establish three equations for all grid points (interior and boundary) with three unknowns f_i, $(\delta f/\delta x)_i$, $(\delta^2 f/\delta x^2)_i$, $i = 1, 2, \ldots, N+1$. We may write the $3(N+1)$ equations (2.7), (2.8), (5.3), (5.4), (5.7), (5.8) into a more general form (global CCD FDE system),

$$a_i^j(1)\left(\frac{\delta f}{\delta x}\right)_{i-1} + a_i^j(2)\left(\frac{\delta f}{\delta x}\right)_i + a_i^j(3)\left(\frac{\delta f}{\delta x}\right)_{i+1} + b_i^j(1)\left(\frac{\delta^2 f}{\delta x^2}\right)_{i-1} + b_i^j(2)\left(\frac{\delta^2 f}{\delta x^2}\right)_i + b_i^j(3)\left(\frac{\delta^2 f}{\delta x^2}\right)_{i+1} + c_i^j(1) f_{i-1} + c_i^j(2) f_i + c_i^j(3) f_{i+1} = s_i^j, \quad (5.9)$$

where $i = 1, 2, 3, \ldots, N+1$ and $j = 1, 2, 3$. The superscript j indicates different equations used at each grid point: $j = 1$ corresponds to FDE (5.3), $j = 2$ corresponds to the first-order derivative calculation (2.7), and $j = 3$ corresponds to the second-order derivative calculation (2.8). For all the interior and boundary points, the coefficients of (5.9) satisfy

$$a_i^1(1) = a_i^1(3) = b_i^1(1) = b_i^1(3) = c_i^1(1) = c_i^1(3) = 0. \quad (5.10)$$

For the two boundaries, the coefficients of (5.9) satisfy

$$a_1^j(1) = b_1^j(1) = c_1^j(1) = 0,$$
$$a_{N+1}^j(3) = b_{N+1}^j(3) = c_{N+1}^j(3) = 0, \quad j = 1, 2, 3. \quad (5.11)$$

Thus, the coefficient matrix of (5.9) indicates a triple-tridiagonal structure and can be solved in two steps: triple-forward elimination and triple-backward substitution (Appendix 6).

5.2. Periodic Boundaries

For periodic boundaries (4.9), the global CCD system (5.9) is well-posed since it has $3N$ equations with $3N$ unknowns: f_i, $(\delta f/\delta x)_i$, $(\delta^2 f/\delta x^2)_i$, $i = 1, 2, 3, \ldots, N$. The coefficient matrix and the related algorithm are listed in Appendix 7.

6. EXAMPLES

The CCD scheme proposed here is a three-point scheme with sixth-order accuracy. Usually a three-point scheme (e.g., central difference scheme) has only second-order accuracy.

Two examples are used in this section to show the advantage of using this new three-point scheme. Comparison is made between the CCD scheme and the second-order central difference (SCD) scheme on: (a) truncation error, (b) horizontal resolution, and (c) CPU time.

6.1. One-Dimensional Convection–Diffusion Equation

Consider a one-dimensional convection–diffusion equation,

$$a(x)\psi + b(x)\frac{d\psi}{dx} - c(x)\frac{d^2\psi}{dx^2} = d(x), \quad 0 \le x \le \pi, \tag{6.1}$$

with the boundary conditions

$$\psi(0) = 0, \quad \psi(\pi) = 0. \tag{6.2}$$

If the coefficient functions in (6.1) are taken as

$$a(x) = 1, \quad b(x) = 1, \quad c(x) = 1, \quad d(x) = \cos x + 2\sin x, \quad 0 \le x \le \pi, \tag{6.3}$$

Eq. (6.1) has an analytical solution,

$$\psi^{(\mathrm{an})}(x) = \sin(x). \tag{6.4}$$

We solved (6.1) numerically with both CCD and SCD schemes under various horizontal resolutions, and we recorded the CPU time (a SUN Sparc-20 was used) for each run. Comparing the numerical results with the analytic solution (6.4), we obtain the truncation errors of the two schemes for the given resolution (represented by number of cells). We define an averaged relative error ($\mathrm{err}_{\mathrm{av}}$) by

$$\mathrm{err}_{\mathrm{av}} = \frac{\sum_{i,j} \left|\Psi_{i,j} - \Psi_{i\cdot j}^{(\mathrm{an})}\right| \Delta x \Delta y}{\sum_{i,j} |\Psi_{i,j}| \Delta x \Delta y}. \tag{6.5}$$

Thus, we have a data set consisting of truncation error, CPU time, and cell number for the two schemes.

The relationship between the cell number (N) and $\mathrm{err}_{\mathrm{av}}$ (Fig. 2a) for the CCD scheme (solid curve) and the SCD scheme (dashed curve) shows that for the same $\mathrm{err}_{\mathrm{av}}$ the cell number would be much smaller in the CCD scheme than in the SCD scheme. In other words, we may use a much coarser resolution for the CCD scheme than for the SCD scheme if the same accuracy is required. For example, the CCD scheme needs only 18 cells when $\mathrm{err}_{\mathrm{av}}$ is around 0.38×10^{-7}. However, for the same accuracy, the SCD scheme requires 9400 cells (see Table 2).

The relationship between the CPU time and the averaged relative error (Fig. 2b) for the CCD scheme (solid curve) and the SCD scheme (dashed curve) shows that for the same $\mathrm{err}_{\mathrm{av}}$ the CPU time is much shorter in the CCD scheme than in the SCD scheme.

Such striking features can also be observed in Table 2. When the relative truncation errors are on the order of 0.2×10^{-6}, the SCD scheme needs 3600 grid cells; however, the CCD

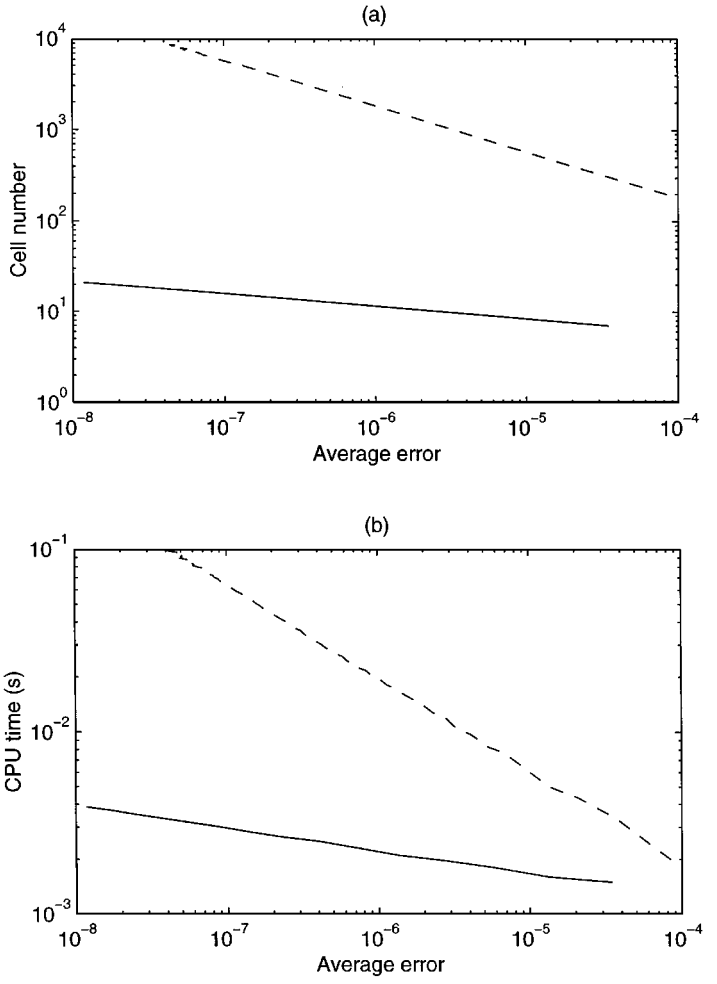

FIG. 2. Comparison between the CCD and SCD schemes in one-dimensional convection–diffusion equation: (a) cell number versus average error; (b) CPU time versus average error. Here solid curves denote the CCD scheme and the dashed curves represent the SCD scheme.

scheme requires only 14 grid cells. The CPU time is also more than an order of magnitude smaller using the CCD scheme (0.28×10^{-2} s) than using the SCD scheme (0.32×10^{-1} s). The ratio of CPU between using SCD and CCD schemes (Ra), called the CPU ratio here, is around 24.2 when the truncation errors are on the order of 4.37×10^{-7}.

6.2. Stommel Ocean Model

Stommel [23] designed an ocean model to explain the westward intensification of wind-driven ocean currents. Consider a rectangular ocean with the origin of a Cartesian coordinate system at the southwest corner (Fig. 3). The x and y axes point eastward and northward, respectively. The boundaries of the ocean are at $x = 0, \lambda$ and $y = 0, b$. The ocean is considered as a homogeneous and incompressible layer of constant depth D when at rest. When currents occur as in the real ocean, the depth differs from D everywhere by a small

TABLE 2
Comparison between the CCD and SCD Schemes in One-Dimensional Convection–Diffusion Equation

Error range	Features	CCD	SCD	Ra
$0.36\sim0.83\times10^{-4}$	Cell number	7	200	
	Average error	0.3649×10^{-4}	0.8292×10^{-4}	1.22
	CPU time (s)	0.0015	0.001833	
$0.27\sim0.35\times10^{-5}$	Cell number	10	1000	
	Average error	0.2734×10^{-5}	0.343×10^{-5}	4.42
	CPU time (s)	0.002	0.008833	
$0.23\sim0.26\times10^{-6}$	Cell number	14	3600	
	Average error	0.2395×10^{-6}	0.2577×10^{-6}	11.3
	CPU time (s)	0.002833	0.032	
$0.37\sim0.38\times10^{-7}$	Cell number	18	9400	
	Average error	0.3747×10^{-7}	0.3779×10^{-7}	24.2
	CPU time (s)	0.0035	0.08483	

perturbation. Due to the incompressibility, a streamfunction ψ is defined by

$$u=-\frac{\partial\psi}{\partial y},\quad v=\frac{\partial\psi}{\partial x},$$

where u and v are the x and y components of the velocity vector.

The surface wind stress is taken as $-F\cos(\pi y/b)$. The component frictional forces are taken as $-Ru$ and $-Rv$, where R is the frictional coefficient. The Coriolis parameter f is also introduced. In general it is a function of y. The latitudinal variation of f, $\beta=df/dy$, is called the β-effect in the ocean dynamics. Under these conditions Stommel derived an equation for the streamfunction ψ,

$$\left(\frac{\partial^2}{\partial x^2}+\frac{\partial^2}{\partial y^2}\right)\Psi+\alpha\frac{\partial\Psi}{\partial x}=-\gamma\sin\left(\frac{\pi}{b}y\right),\tag{6.6}$$

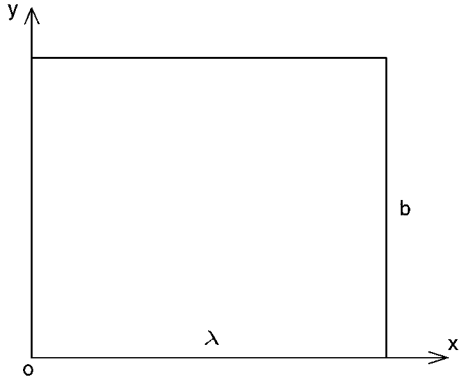

FIG. 3. Ocean basin dimensions and the coordinate system.

with the boundary conditions

$$\Psi(0, y) = \Psi(\lambda, y) = \Psi(x, 0) = \Psi(x, b) = 0. \tag{6.7}$$

Here, the two parameters α and γ are defined by

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

The analytical solution of (6.6) with the boundary conditions (6.7) is given by

$$\Psi = -\gamma \left(\frac{b}{\pi}\right)^2 \sin\left(\frac{\pi}{b} y\right)\left(pe^{Ax} + qe^{Bx} - 1\right), \tag{6.8}$$

where

$$\begin{gathered} A = -\frac{\alpha}{2} + \sqrt{\frac{\alpha^2}{4} + \left(\frac{\pi}{b}\right)^2}, \quad B = -\frac{\alpha}{2} - \sqrt{\frac{\alpha^2}{4} + \left(\frac{\pi}{b}\right)^2} \\ p = \left(1 - e^{B\lambda}\right)/\left(e^{A\lambda} - e^{B\lambda}\right), \quad q = 1 - p. \end{gathered} \tag{6.9}$$

The physical parameters are selected as [23]

$$\begin{gathered} \lambda = 10^7 \text{ m}, \quad b = 2\pi \times 10^6 \text{ m}, \quad D = 200 \text{ m}, \\ F = 0.3 \times 10^{-7} \text{ m}^2 \text{ s}^{-2}, \quad R = 0.6 \times 10^{-3} \text{ m s}^{-1}. \end{gathered}$$

The parameter β is taken as 0 for the case without the β-effect case, and it is taken as 10^{-11} m^{-1} s^{-1} for the case with the β-effect case.

6.2.1. *Computational Algorithm*

Use a uniform grid, $0 = x_1 < x_2 < \cdots < x_{N_x} < x_{N_x+1} = \lambda$, and $0 = y_1 < y_2 < \cdots < y_{N_y} < y_{N_y+1} = b$ with grid spacing $\Delta x = x_{i+1} - x_i = \lambda/N_x$ and $\Delta y = y_{j+1} - y_j = b/N_y$. For simplicity and no loss of generality, we assume that the cell number in both the x and y directions are the same, $N_x = N_y = N$. The alternating direction implicit (ADI) method is used for solving FDE. The iteration k to $k+1$ can be separated into two parts: (a) iteration along the x-axis to obtain "intermediate variables" $\Psi^*_{i,j}$, $(\delta\Psi/\delta x)^*_{i,j}$, and $(\delta^2\Psi/\delta x^2)^*_{i,j}$,

$$\begin{aligned} &\left(\frac{\delta^2\Psi}{\delta x^2}\right)^*_{i,j} + \alpha\left(\frac{\delta\Psi}{\delta x}\right)^*_{i,j} - \frac{6}{\Delta y^2}\Psi^*_{i,j} = s_{i,j} - \frac{3}{\Delta y^2}\left(\Psi^k_{i,j+1} + \Psi^k_{i,j-1}\right) + \frac{1}{8}\left(\frac{\delta^2\Psi}{\delta y^2}\right)^k_{i,j+1} \\ &\quad + \frac{1}{8}\left(\frac{\delta^2\Psi}{\delta y^2}\right)^k_{i,j-1} + \frac{9}{8\Delta y}\left(\left(\frac{\delta\Psi}{\delta y}\right)^k_{i,j-1} - \left(\frac{\delta\Psi}{\delta y}\right)^k_{i,j+1}\right) \end{aligned} \tag{6.10}$$

$$\begin{aligned} &\frac{7}{16}\left(\left(\frac{\delta\Psi}{\delta x}\right)^*_{i+1,j} + \left(\frac{\delta\Psi}{\delta x}\right)^*_{i-1,j}\right) + \left(\frac{\delta\Psi}{\delta x}\right)^*_{i,j} - \frac{\Delta x}{16}\left(\left(\frac{\delta^2\Psi}{\delta x^2}\right)^*_{i+1,j} - \left(\frac{\delta^2\Psi}{\delta x^2}\right)^*_{i-1,j}\right) \\ &\quad - \frac{15}{8}\frac{1}{2\Delta x}(\Psi^*_{i+1,j} - \Psi^*_{i-1,j}) = 0 \end{aligned} \tag{6.11}$$

$$\begin{aligned} &\frac{9}{8\Delta x}\left(\left(\frac{\delta\Psi}{\delta x}\right)^*_{i+1,j} - \left(\frac{\delta\Psi}{\delta x}\right)^*_{i-1,j}\right) - \frac{1}{8}\left(\left(\frac{\delta^2\Psi}{\delta x^2}\right)^*_{i+1,j} + \left(\frac{\delta^2\Psi}{\delta x^2}\right)^*_{i-1,j}\right) \\ &\quad + \left(\frac{\delta^2\Psi}{\delta x^2}\right)^*_{i,j} - 3\frac{1}{\Delta x^2}(\Psi^*_{i+1,j} - 2\Psi^*_{i,j} + \Psi^*_{i-1,j}) = 0 \end{aligned} \tag{6.12}$$

and (b) iteration along the y-axis to obtain variables at the next iteration $k+1$, $\Psi_{i,j}^{k+1}$, $(\delta\Psi/\delta x)_{i,j}^{k+1}$, and $(\delta^2\Psi/\delta x^2)_{i,j}^{k+1}$,

$$\left(\frac{\delta^2\Psi}{\delta y^2}\right)_{i,j}^{k+1} - \frac{6}{\Delta x^2}\Psi_{i,j}^{k+1} = s_{i,j} - \alpha\left(\frac{\delta\Psi}{\delta x}\right)_{i,j}^{*} - \frac{3}{\Delta x^2}(\Psi_{i+1,j}^{*} + \Psi_{i-1,j}^{*}) + \frac{1}{8}\left(\frac{\delta^2\Psi}{\delta x^2}\right)_{i+1,j}^{*}$$
$$+ \frac{1}{8}\left(\frac{\delta^2\Psi}{\delta x^2}\right)_{i-1,j}^{*} + \frac{9}{8\Delta x}\left(\left(\frac{\delta\Psi}{\delta x}\right)_{i-1,j}^{*} - \left(\frac{\delta\Psi}{\delta x}\right)_{i+1,j}^{*}\right) \tag{6.13}$$

$$\frac{7}{16}\left(\left(\frac{\delta\Psi}{\delta y}\right)_{i,j+1}^{k+1} + \left(\frac{\delta\Psi}{\delta y}\right)_{i,j-1}^{k+1}\right) + \left(\frac{\delta\Psi}{\delta y}\right)_{i,j}^{k+1} - \frac{\Delta y}{16}\left(\left(\frac{\delta^2\Psi}{\delta y^2}\right)_{i,j+1}^{k+1} - \left(\frac{\delta^2\Psi}{\delta y^2}\right)_{i,j-1}^{k+1}\right)$$
$$- \frac{15}{8}\frac{1}{2\Delta y}\left(\Psi_{i,j+1}^{k+1} - \Psi_{i,j-1}^{k+1}\right) = 0 \tag{6.14}$$

$$\frac{9}{8\Delta y}\left(\left(\frac{\delta\Psi}{\delta y}\right)_{i,j+1}^{k+1} - \left(\frac{\delta\Psi}{\delta y}\right)_{i,j-1}^{k+1}\right) - \frac{1}{8}\left(\left(\frac{\delta^2\Psi}{\delta y^2}\right)_{i,j+1}^{k+1} + \left(\frac{\delta^2\Psi}{\delta y^2}\right)_{i,j-1}^{k+1}\right)$$
$$+ \left(\frac{\delta^2\Psi}{\delta y^2}\right)_{i,j}^{k+1} - 3\frac{1}{\Delta y^2}\left(\Psi_{i,j+1}^{k+1} - 2\Psi_{i,j}^{k+1} + \Psi_{i,j-1}^{k+1}\right) = 0. \tag{6.15}$$

Such an iterative process stops when the correction at the iteration $k+1$,

$$\mathrm{corr}^{(k+1)} = \frac{\sum_{i,j}\left|\Psi_{i,j}^{k+1} - \Psi_{i,j}^{k}\right|\Delta x\Delta y}{\sum_{i,j}\left|\Psi_{i,j}^{k}\right|\Delta x\Delta y}, \tag{6.16}$$

is smaller than 10^{-6}.

6.2.2. *Case* 1: *Without the β-Effect*

The condition $\beta = 0$ leads to $\alpha = 0$ in (6.6). The analytical solution of (6.6) becomes

$$\Psi = \gamma\left(\frac{b}{\pi}\right)^2 \sin\left(\frac{\pi}{b}y\right)\left(1 - \frac{1 - e^{-\frac{\pi}{b}\lambda}}{e^{\frac{\pi}{b}\lambda} - e^{-\frac{\pi}{b}\lambda}}e^{\frac{\pi}{b}x} - \frac{e^{\frac{\pi}{b}\lambda} - 1}{e^{\frac{\pi}{b}\lambda} - e^{-\frac{\pi}{b}\lambda}}e^{-\frac{\pi}{b}x}\right) \tag{6.17}$$

which is depicted in Fig. 4.

We solved (6.6) numerically with both CCD and SCD schemes under various horizontal resolutions, and we recorded the CPU time (a SUN Sparc-20 was used) for each run. Comparing the numerical results with the analytic solution (6.17), we obtain the truncation errors of the two schemes for various resolutions (represented by the number of cells).

The relationship between N and $\mathrm{err}_{\mathrm{av}}$ (Fig. 5a) for the CCD scheme (solid curve) and the SCD scheme (dashed curve) shows that for the same $\mathrm{err}_{\mathrm{av}}$ the cell number (N) would be much smaller for the CCD scheme than for the SCD scheme. This is to say that we may use a much coarser resolution for the CCD scheme than for the SCD scheme for the same accuracy. The relationship between the CPU time and the averaged relative error (Fig. 5b) for the CCD scheme (solid curve) and the SCD scheme (dashed curve) shows that for the same $\mathrm{err}_{\mathrm{av}}$ the CPU time is much shorter in the CCD scheme than in the SCD scheme.

Table 3 lists $\mathrm{err}_{\mathrm{av}}$, cell number, CPU time for the two schemes, and CPU ratio (Ra). When the relative truncation errors are on the order of 0.68×10^{-4}, the SCD scheme needs

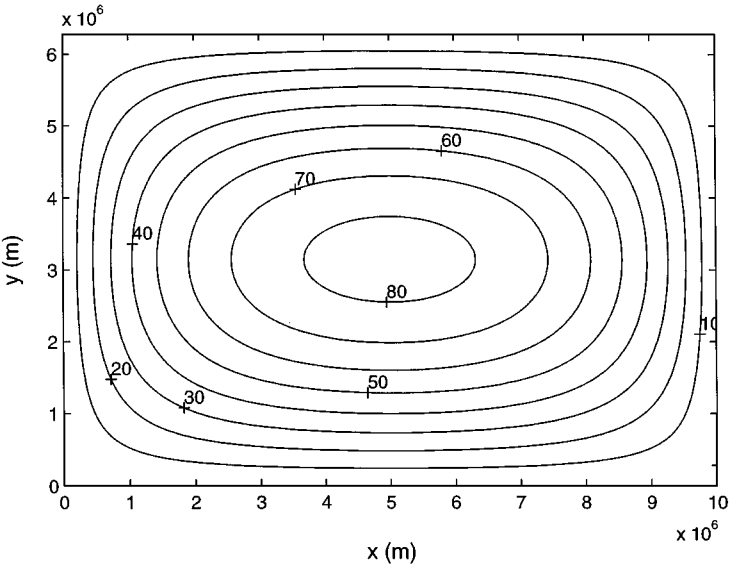

FIG. 4. Streamfunction (m^2/s) obtained from Stommel ocean model with beta = 0.

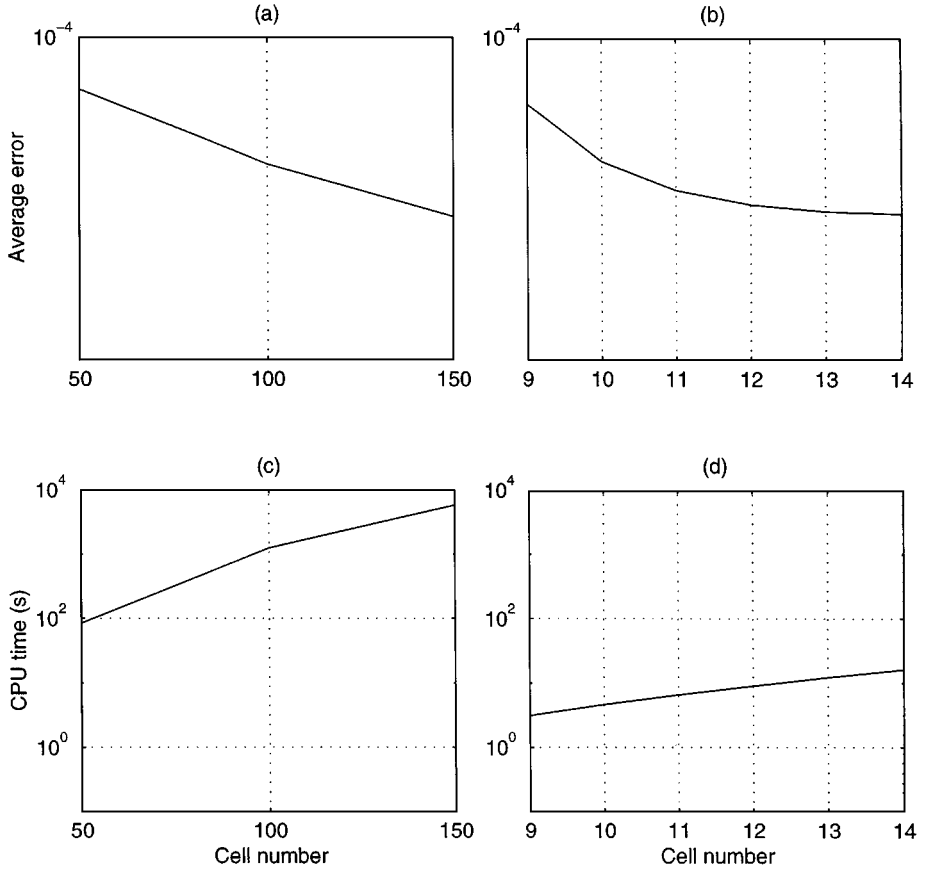

FIG. 5. Performance of the CCD and SCD schemes in Stommel ocean model (beta = 0): (a) average error versus cell number in the SCD scheme; (b) average error versus cell number in the CCD scheme; (c) CPU time versus cell number in the SCD scheme; (d) CPU time versus cell number in the CCD scheme.

TABLE 3
Comparison between the CCD and SCD Schemes in Stommel Ocean Model (beta = 0)

Error range	Features	CCD	SCD	Ra
$0.86\sim0.9 \times 10^{-4}$	Cell number	9×9	50×50	
	Average error	0.866×10^{-4}	0.894×10^{-4}	27.0
	CPU time (s)	3.10	83.8	
$0.76\sim0.77 \times 10^{-4}$	Cell number	10×10	100×100	
	Average error	0.766×10^{-4}	0.761×10^{-4}	271.7
	CPU time (s)	4.6	1250	
$0.68\sim0.69 \times 10^{-4}$	Cell number	14×14	150×150	
	Average error	0.685×10^{-4}	0.68×10^{-4}	356.8
	CPU time (s)	16.2	5780	

22,500 grid cells; however, the CCD scheme requires only 196 grid cells. The CPU ratio between using SCD and CCD schemes (Ra) is 356.8.

6.2.3. *Case* 2: *With the β-Effect*

For this case, $\beta = 10^{-11}$ m^{-1}s^{-1} is used. The analytical streamfunction, ψ^{an}, is plotted in Fig. 6. We solved (6.6) numerically with both CCD and SCD schemes under various horizontal resolutions, and we recorded the CPU time (a SUN Sparc-20 was used) for each run. Comparing the numerical results with the analytic solution (6.8), we obtain the truncation errors of the two schemes for various given resolutions (represented by the number of cells).

The relationship between N and err_{av} (Fig. 7a) for the CCD scheme (solid curve) and the SCD scheme (dashed curve) shows that for the same err_{av} the cell number (N) would be

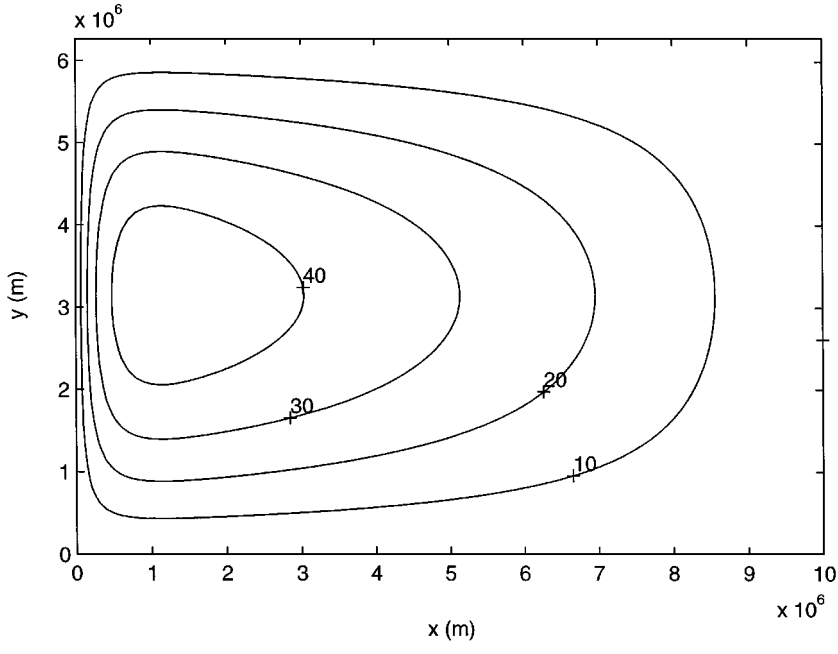

FIG. 6. Streamfunction (m^3/s) obtained from Stommel ocean model with beta $= 10^{-11}$ m^{-1} s^{-1}.

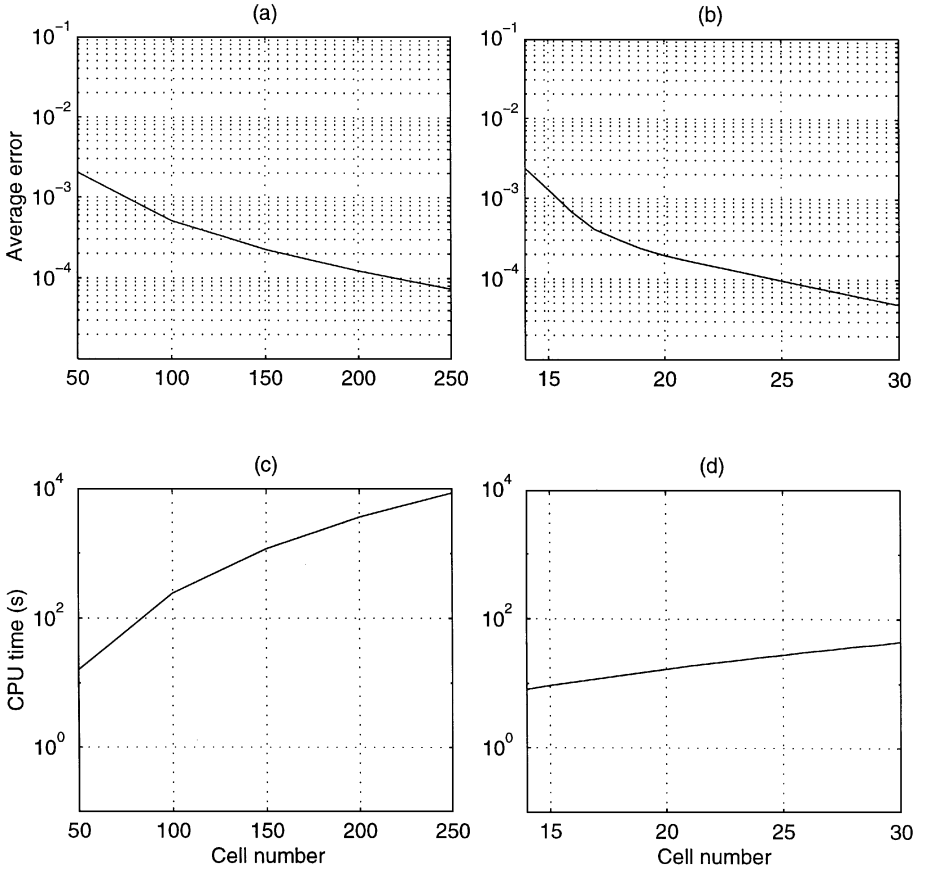

FIG. 7. Performance of the CCD and SCD schemes in Stommel ocean model (beta $= 10^{-11}$ m^{-1} s^{-1}): (a) average error versus cell number in the SCD scheme; (b) average error versus cell number in the CCD scheme; (c) CPU time versus cell number in the SCD scheme; (d) CPU time versus cell number in the CCD scheme.

much smaller in the CCD scheme than in the SCD scheme. The relationship between the CPU time and the averaged relative error (Fig. 7b) for the CCD scheme (solid curve) and the SCD scheme (dashed curve) shows that for the same err_{av} the CPU time is much shorter in the CCD scheme than in the SCD scheme.

Table 4 lists err_{av}, cell number, CPU time, and Ra for the two schemes. When the relative truncation errors are on the order of 0.73×10^{-4}, the SCD scheme needs 22,500 grid cells; however, the CCD scheme requires only 729 grid cells. The CPU ratio between using SCD and CCD schemes (Ra) is 254.87.

7. CONCLUSIONS

(1) From this study, it can be stated that the three-point sixth-order CCD scheme is a promising highly accurate method for both derivative computation and FDE solutions. The advantage of this scheme is the existence of a global sixth-order polynomial which not only satisfies the FDE at all the grid nodes including boundary points but also the boundary conditions.

TABLE 4
Comparison between the CCD and SCD Schemes in Stommel Ocean Model (beta = 10^{-11} m^{-1} s^{-1})

Error range	Features	CCD	SCD	Ra
0.20–0.24 $\times 10^{-2}$	Cell number	14×14	50×50	
	Average error	0.236×10^{-2}	0.204×10^{-2}	1.98
	CPU time (s)	8.12	16.1	
0.22–0.24 $\times 10^{-3}$	Cell number	19×19	150×150	
	Average error	0.238×10^{-3}	0.225×10^{-3}	78.79
	CPU time (s)	14.9	1174	
0.73–0.74 $\times 10^{-4}$	Cell number	27×27	250×250	
	Average error	0.73×10^{-4}	0.735×10^{-4}	254.87
	CPU time (s)	33.9	8640	

(2) Fourier analysis shows that the CCD scheme has the least error among other same order schemes, including the normal compact scheme. Also, the CCD scheme has the smallest truncation error among various sixth-order schemes. The truncation error of the first derivative using the CCD scheme is about 41.2 times smaller than using the sixth-order central scheme, 4.6 times smaller than using the sixth-order tridiagonal (compact) scheme, and 6.0 times smaller than using the sixth-order pentadiagonal (compact) scheme. The truncation error of the second derivative using the CCD scheme is about 36 times smaller than using the sixth-order central scheme, 8.4 times smaller than using the sixth-order tridiagonal scheme (compact), and 13.8 times smaller than using the sixth-order pentadiagonal scheme (compact). Comparing the CCD scheme with the second-order central difference (SCD) scheme (most commonly used in ocean models), the truncation errors for both first and second derivatives are more than four orders of magnitude smaller.

(3) For periodic boundaries, the CCD scheme has sixth-order accuracy at all grid points including boundary nodes. For nonperiodic boundaries, the CCD scheme has sixth-order accuracy at all interior grid points, fourth-order accuracy in the derivative computation, and fifth-order accuracy in the FDE solutions at the boundary nodes.

(4) Both twin-tridiagonal and triple-tridiagonal techniques are proposed for the CCD scheme for calculating derivatives and solving FDEs.

(5) Two examples (the convection–diffusion model and the Stommel ocean model) show striking results (great reduction in truncation error and CPU time), which may lead to a wide application of the CCD scheme in computational geophysics.

(6) Future studies include applying the CCD scheme to nonuniform and/or staggered grid systems, as well as designing even higher order schemes such as an eighth-order CCD scheme.

APPENDICES

Appendix 1: Global Hermitian Polynomial

The first-order and second-order CCD differences are obtained implicitly and globally by the two joint equations (2.7) and (2.8). A twin-tridiagonal technique was developed to

compute f' and f'' at all grid points. As soon as the global first and second differences are obtained, the higher order ($k = 3, 4, 5, 6$) differences can easily be calculated locally with (2.5).

Since the CCD scheme is solved globally, the neighboring local Hermitian polynomials should satisfy

$$H_i'(x_i) = H_{i-1}'(x_i) = H_{i+1}'(x_i) = \left(\frac{\delta f}{\delta x}\right)_i$$

$$H_i''(x_i) = H_{i-1}''(x_i) = H_{i+1}''(x_i) = \left(\frac{\delta^2 f}{\delta x^2}\right)_i.$$

A global polynomial $H_g(x)$ can be defined by

$$\begin{aligned}
&H_g(x) = H_2(x), && a = x_1 \le x < x_2,\\
&H_g(x) = \omega_i H_i(x) + (1-\omega_i) H_{i+1}(x), && x_i \le x < x_{i+1}\ (i = 2, 3, \ldots, n-1),\\
&H_g(x) = H_n(x), && x_n \le x < x_{n+1} = b,
\end{aligned}$$

where ω_i ($i = 2, 3, \ldots, n-1$) are the local weighting factors. Notice that no matter what value of ω_i is, the global polynomial $H_g(x)$ always has continuous first- and second-order derivatives at the point x_i,

$$H_g'(x_i) = H_g'(x_i - 0) = H_g'(x_i + 0)$$

$$H_g''(x_i) = H_g''(x_i - 0) = H_g''(x_i + 0).$$

The weighting factors are recommended to be $0 \le \omega_i \le 1$. If only the first-order and second-order derivatives are computed, we may use $\omega_i = 1/2$ for simplicity. It is also possible to optimize ω_i by minimizing the discontinuity properties of the high-order ($k \ge 3$) derivatives at the node points. As soon as the global polynomial $H_g(x)$ is established, we can calculate all the derivatives and integrate. Since the values of ω_i do not affect the first-order and second-order derivatives, we will not discuss here the effect of ω_i. This paper focuses only on the first-order and second-order differentiation of the second-order PDE.

Furthermore, a higher order (higher than sixth-order) three points CCD scheme can also be defined. See Appendix 2 for description.

Appendix 2: Eighth-Order CCD Scheme

The eighth-order CCD scheme relates f_i, f_i', f_i'', $f_i^{(3)}$ to the two neighboring points: f_{i-1}, f_{i-1}', f_{i-1}'', $f_{i-1}^{(3)}$ and f_{i+1}, f_{i+1}', f_{i+1}'', $f_{i+1}^{(3)}$ and solves for f_i', f_i'', $f_i^{(3)}$. A local Hermitian polynomial $H_i(x)$ is defined on the closed interval $[x_{i-1}, x_{i+1}]$ by

$$\begin{aligned}
H_i(x) = {} & H_i(x_i) + H_i'(x_i)x + \frac{H_i''(x_i)}{2!}x^2 + \frac{H_i^{(3)}(x_i)}{3!}x^3 + \frac{H_i^{(4)}(x_i)}{4!}x^4 + \frac{H_i^{(5)}(x_i)}{5!}x^5\\
& + \frac{H_i^{(6)}(x_i)}{6!}x^6 + \frac{H_i^{(7)}(x_i)}{7!}x^7 + \frac{H_i^{(8)}(x_i)}{8!}x^8
\end{aligned}$$

with

$$H_i(x_{i-1}) = f_{i-1}, H_i(x_i) = f_i, H_i(x_{i+1}) = f_{i+1}, H_i'(x_{i-1}) = f'_{i-1}, H_i'(x_{i+1}) = f'_{i+1},$$
$$H_i''(x_{i-1}) = f''_{i-1}, H_i''(x_{i+1}) = f''_{i+1}, H_i^{(3)}(x_{i-1}) = f_{i-1}^{(3)}, H_i^{(3)}(x_{i+1}) = f_{i+1}^{(3)}.$$

The nine parameters are determined by

$$H_i(x_i) = f_i$$

$$H_i'(x_i) = \frac{35}{32h}(f_{i+1} - f_{i-1}) - \frac{19}{32}(f'_{i+1} + f'_{i-1}) + \frac{h}{8}(f''_{i+1} - f''_{i-1}) - \frac{h^2}{96}\left(f_{i+1}^{(3)} + f_{i-1}^{(3)}\right)$$

$$H_i''(x_i) = \frac{4}{h^2}(f_{i+1} - 2f_i + f_{i-1}) - \frac{29}{16h}(f'_{i+1} - f'_{i-1}) + \frac{5}{16}(f''_{i+1} + f''_{i-1}) - \frac{h}{48}\left(f_{i+1}^{(3)} - f_{i-1}^{(3)}\right)$$

$$H_i^{(3)}(x_i) = -\frac{105}{16h^3}(f_{i+1} - f_{i-1}) + \frac{105}{16h^2}(f'_{i+1} + f'_{i-1}) - \frac{15}{8h}(f''_{i+1} - f''_{i-1}) + \frac{3}{16}\left(f_{i+1}^{(3)} + f_{i-1}^{(3)}\right)$$

$$H_i^{(4)} = -\frac{72}{h^4}(f_{i+1} - 2f_i + f_{i-1}) + \frac{183}{4h^3}(f'_{i+1} - f'_{i-1}) - \frac{39}{4h^2}(f''_{i+1} + f''_{i-1}) + \frac{3}{4h}\left(f_{i+1}^{(3)} - f_{i-1}^{(3)}\right)$$

$$H_i^{(5)}(x_i) = \frac{315}{4h^5}(f_{i+1} - f_{i-1}) - \frac{315}{4h^4}(f'_{i+1} + f'_{i-1}) + \frac{30}{h^3}(f''_{i+1} - f''_{i-1}) - \frac{15}{4h^2}\left(f_{i+1}^{(3)} + f_{i-1}^{(3)}\right)$$

$$H_i^{(6)}(x_i) = \frac{1440}{h^6}(f_{i+1} - 2f_i + f_{i-1}) - \frac{1935}{2h^5}(f'_{i+1} - f'_{i-1}) + \frac{495}{h^4}(f''_{i+1} + f''_{i-1}) - \frac{45}{2h^3}\left(f_{i+1}^{(3)} - f_{i-1}^{(3)}\right)$$

$$H_i^{(7)}(x_i) = -\frac{2575}{2h^7}(f_{i+1} - f_{i-1})z + \frac{1575}{2h^6}(f'_{i+1} + f'_{i-1}) - \frac{315}{h^5}(f''_{i+1} - f''_{i-1}) - \frac{105}{2h^4}\left(f_{i+1}^{(3)} + f_{i-1}^{(3)}\right)$$

$$H_i^{(8)}(x_i) = -\frac{21060}{h^8}(f_{i+1} - 2f_i + f_{i-1}) + \frac{13860}{2h^7}(f'_{i+1} - f'_{i-1}) - \frac{3780}{h^6}(f''_{i+1} + f''_{i-1}) + \frac{420}{h^5}\left(f_{i+1}^{(3)} - f_{i-1}^{(3)}\right).$$

The kth derivative at the grid point x_i is approximated by

$$f^{(k)}(x_i) \approx H_i^{(k)}(x_i), \quad k = 1, 2, \ldots, 8.$$

Therefore, the first-order derivative at grid point x_i is computed by

$$\frac{19}{32}(f'_{i+1} + f'_{i-1}) + f'_i - \frac{h}{8}(f''_{i+1} - f''_{i-1}) + \frac{1}{96}(f^{(3)}_{i+1} + f^{(3)}_{i-1}) = \frac{35}{16}\frac{1}{2h}(f_{i+1} - f_{i-1}) + \frac{427}{1737} f_i^{(9)} \frac{h^8}{8!};$$

the second-order derivative at grid point x_i is computed by

$$\frac{29}{16h}(f'_{i+1} - f'_{i-1}) - \frac{5}{16}(f''_{i+1} + f''_{i-1}) + f''_i + \frac{h}{48}(f^{(3)}_{i+1} - f^{(3)}_{i-1}) = 4\frac{1}{h^2}(f_{i+1} - 2f_i + f_{i-1}) + \frac{1}{45} f_i^{(10)} \frac{h^8}{8!};$$

and the third-order derivative at grid point x_i is computed by

$$-\frac{105}{16h^2}(f'_{i+1} + f'_{i-1}) + \frac{15}{8h}(f''_{i+1} - f''_{i-1}) - \frac{3}{16}(f^{(3)}_{i+1} + f^{(3)}_{i-1}) + f^{(3)}_i = -\frac{105}{8}\frac{1}{2h^3}(f_{i+1} - f_{i-1}) - \frac{1357}{16212} f_i^{(9)} \frac{h^6}{6!}.$$

Appendix 3: Nonperiodic CCD Calculation

Twin-forward elimination/backward substitution scheme is designed to solve global CCD system (4.5) with boundary conditions (4.6). The $2(N+1) \times 2(N+1)$ coefficient matrix of (4.5) has a twin-tridiagonal structure and can be directly solved by two steps: twin-forward elimination and twin-backward substitution.

A.3.1. *Twin-Forward Elimination*

The twin-forward technique is used to transform the twin-tridiagonal coefficient matrix into a twin-diagonal coefficient matrix by eliminating the four parameters, $a_i^1(1)$, $b_i^1(1)$, $a_i^2(1)$, $b_i^2(1)$ at each grid point (Fig. 8). At the left boundary ($i = 1$), these four parameters are already absent.

If the four parameters at grid node i are eliminated, it is easy to use (4.5) to eliminate $a_{i+1}^1(1)$, $a_{i+1}^2(1)$, $b_{i+1}^1(1)$, $b_{i+1}^2(1)$ at grid point $i+1$. This process continues until reaching the right boundary. The coefficient matrix of the global CCD system becomes twin-diagonal. Figure 9 shows the structure of the coefficient matrix after twin-forward elimination, where the shadowed area shows the eliminated elements.

A.3.2. *Twin-Backward Substitution*

The twin-backward substitution technique is used to obtain both $(\delta f/\delta x)_i$ and $(\delta^2 f/\delta x^2)_i$ from known $(\delta f/\delta x)_{i+1}$ and $(\delta^2 f/\delta x^2)_{i+1}$. After the twin-diagonal coefficient matrix has

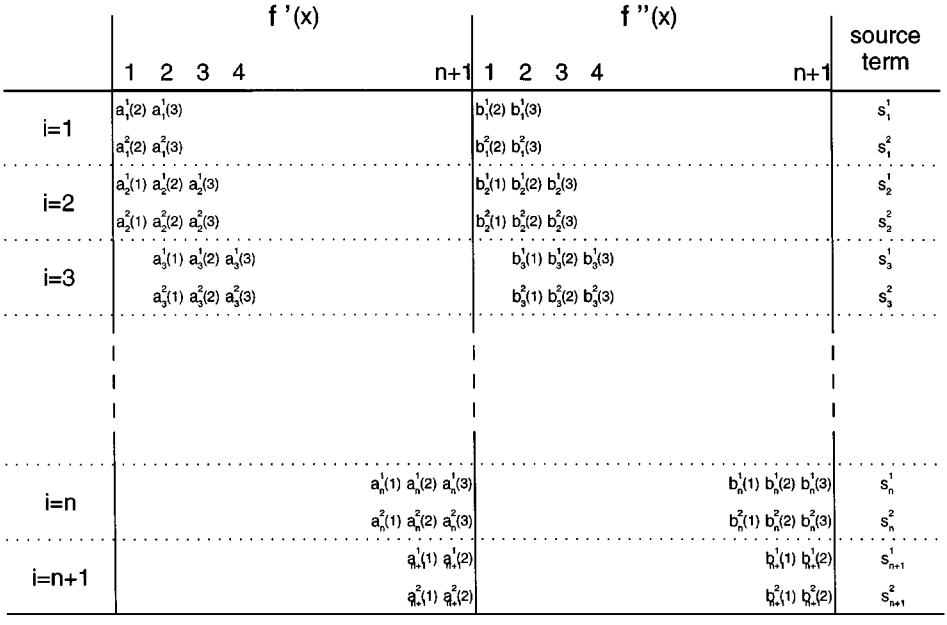

FIG. 8. Structure of the CCD coefficient matrix for nonperiodic boundaries.

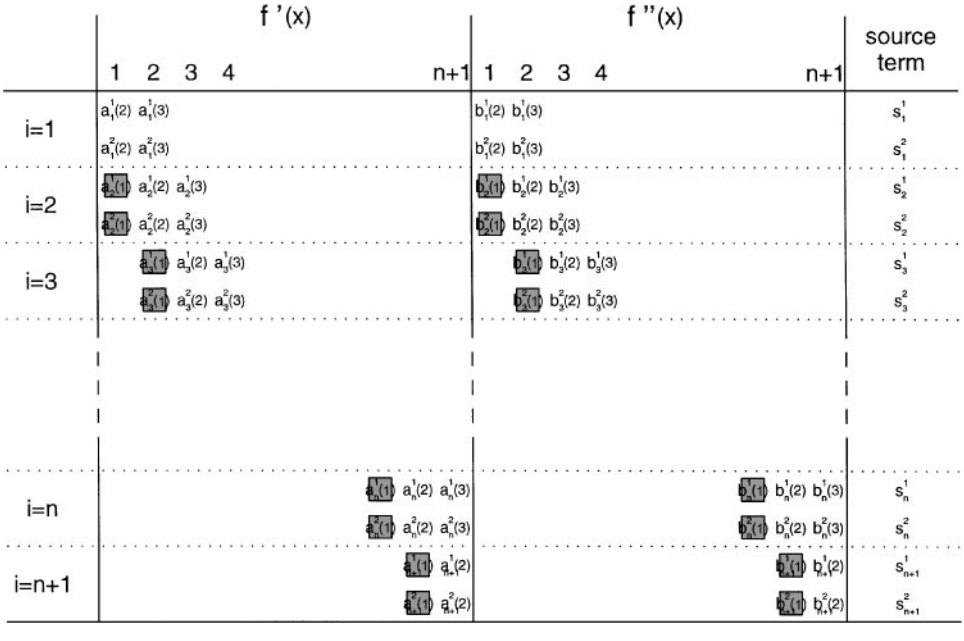

FIG. 9. The twin-forward elimination of the CCD cofficient matrix for nonperiodic boundaries. Here ■ denotes eliminated coefficients.

been established, the global CCD system (4.5) becomes two equations with two unknowns at the right boundary (x_{N+1}),

$$a^j_{N+1}(2)\left(\frac{\delta f}{\delta x}\right)_{N+1} + b^j_{N+1}(2)\left(\frac{\delta^2 f}{\delta x^2}\right)_{N+1} = s^j_{N+1}, \quad j = 1, 2.$$

Solving this set of two algebraic equations, we obtain $(\delta f/\delta x)_{N+1}$ and $(\delta^2 f/\delta x^2)_{N+1}$.

The substitution procedure starts from the second right point (x_N). The first- and second-order differences $(\delta f/\delta x)_i$ and $(\delta^2 f/\delta x^2)_i$ are computed from substitution ($i = N$, $N-1, \ldots, 1$):

$$a^j_i(2)\left(\frac{\delta f}{\delta x}\right)_i + b^j_i(2)\left(\frac{\delta^2 f}{\delta x^2}\right)_i = s^j_i - a^j_i(3)\left(\frac{\delta f}{\delta x}\right)_{i+1} - b^j_i(3)\left(\frac{\delta^2 f}{\delta x^2}\right)_{i+1}, \quad j = 1, 2.$$

Appendix 4: Periodic CCD Calculation

The structure of the periodic CCD matrix is shown in Fig. 10. Similar to nonperiodic boundaries, we construct another form of twin-forward elimination and twin-backward substitution procedures for periodic boundaries. Figure 11 shows the structure after the twin-forward elimination procedure, where the shadowed areas mean the eliminated elements.

Appendix 5: Fifth-Order Accurate Nonperiodic Boundary Conditions

Consider the left boundary with uniform grid $\Delta x = h$. Let x_1 be the left boundary node; let x_2 and x_3 be the first and second neighboring nodes. Expanding the dependent variable

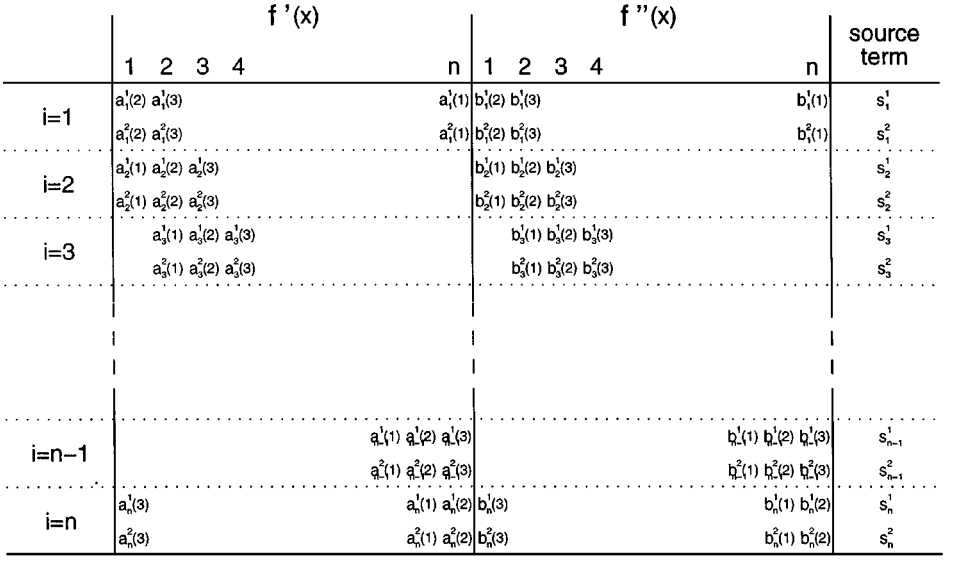

FIG. 10. Structure the CCD coefficient matrix for periodic boundaries.

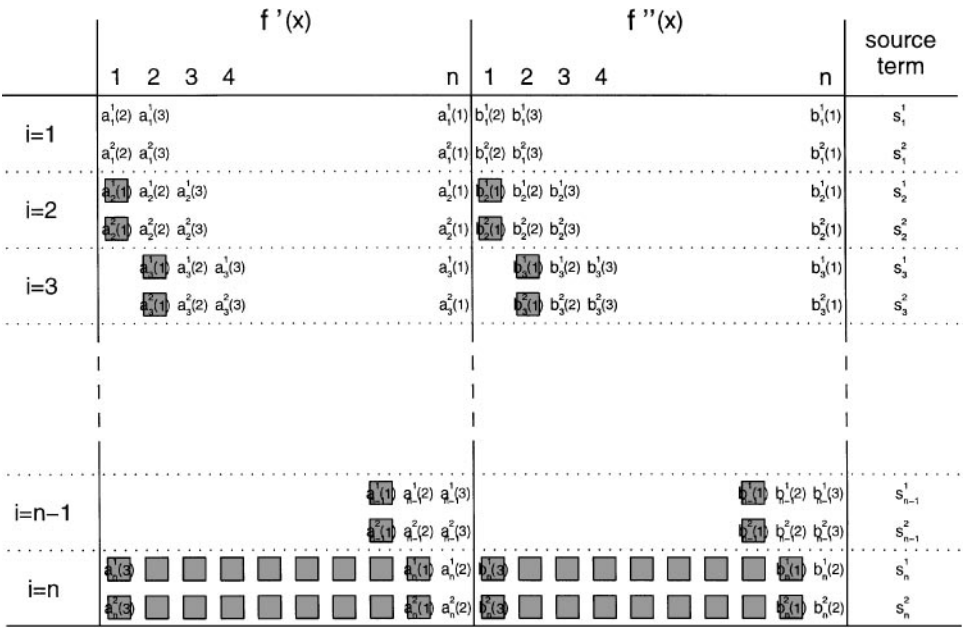

FIG. 11. The twin-forward elimination of the CCD coefficient matrix for periodic boundries. Here ■ denotes eliminated coefficients.

f and its derivatives into Taylor series at x_2, we have

$$f(x_1) = f(x_2) + \sum_{k=1}^{6} \frac{(-1)^k}{k!} f^{(k)}(x_2) h^k + O(h^7)$$

$$f(x_3) = f(x_2) + \sum_{k=1}^{6} \frac{1}{k!} f^{(k)}(x_2) h^k + O(h^7)$$

$$f'(x_1) = f'(x_2) + \sum_{k=1}^{5} \frac{(-1)^k}{k!} f^{(k+1)}(x_2) h^k + O(h^6)$$

$$f''(x_1) = f''(x_2) + \sum_{k=1}^{4} \frac{(-1)^k}{k!} f^{(k+2)}(x_2) h^k + O(h^5)$$

which lead to

$$14 f'(x_1) + 16 f'(x_2) + 2 f''(x_1) h - 4 f''(x_2) h + \frac{1}{h}(31 f(x_1) - 32 f(x_2) + f(x_3))$$

$$= \frac{h^5}{90} f^{(6)}(x_2) + O(h^6).$$

Therefore, the nonperiodic boundary condition

$$14 \left(\frac{\delta f}{\delta x}\right)_1 + 16 \left(\frac{\delta f}{\delta x}\right)_2 + 2h \left(\frac{\delta^2 f}{\delta x^2}\right)_1 - 4h \left(\frac{\delta^2 f}{\delta x^2}\right)_2 + \frac{1}{h}(31 f_1 - 32 f_2 + f_3) = 0$$

has fifth-order accuracy.

Appendix 6: Nonperiodic CCD FDE Solution

The triple-forward elimination/backward substitution scheme is designed to solve global CCD FDE system (5.9) with boundary conditions (5.10). The $3(N+1) \times 3(N+1)$ coefficient matrix of (5.9) has a triple-tridiagonal structure and can be directly solved by two steps: triple-forward elimination and triple-backward substitution.

A.6.1. *Triple-Forward Elimination*

The triple-forward technique is used to transform the triple-tridiagonal coefficient matrix into a triple-diagonal coefficient matrix by eliminating the six parameters, $a_i^1(1)$, $a_i^2(1)$, $b_i^1(1)$, $b_i^2(1)$, $c_i^1(1)$, $c_i^2(1)$ at each grid point (Fig. 12). At the left boundary ($i = 1$), these six parameters are already absent.

If the six parameters at grid node i are eliminated, it is easy to use (5.9) to eliminate $a_{i+1}^1(1)$, $a_{i+1}^2(1)$, $b_{i+1}^1(1)$, $b_{i+1}^2(1)$, $c_i^1(1)$, $c_i^2(1)$ at grid point $i+1$. This process continues until reaching the right boundary. The coefficient matrix of the global CCD FDE system becomes triple-diagonal. Figure 13 shows the structure of the coefficient matrix after triple-forward elimination, where the shadowed area shows the eliminated elements.

A.6.2. *Triple-Backward Substitution*

The triple-backward substitution technique is used to obtain f_i, $(\delta f/\delta x)_i$, and $(\delta^2 f/\delta x^2)_i$ from known f_{i+1}, $(\delta f/\delta x)_{i+1}$, and $(\delta^2 f/\delta x^2)_{i+1}$. After the triple-diagonal coefficient matrix has been established, the global CCD system (5.9) becomes three equations with three

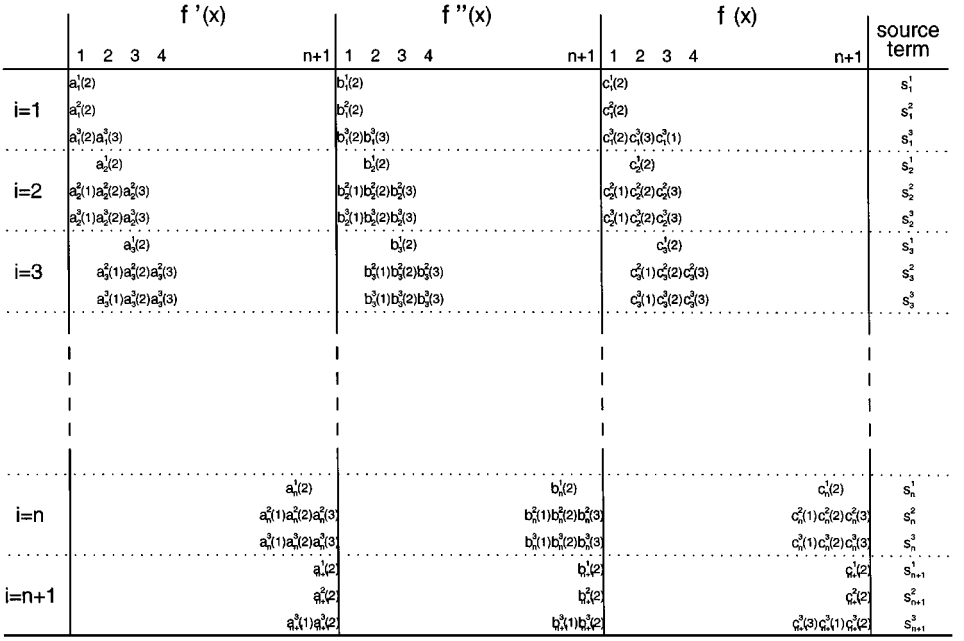

FIG. 12. Structure of the CCD coefficient matrix for FDE with nonperiodic boundaries.

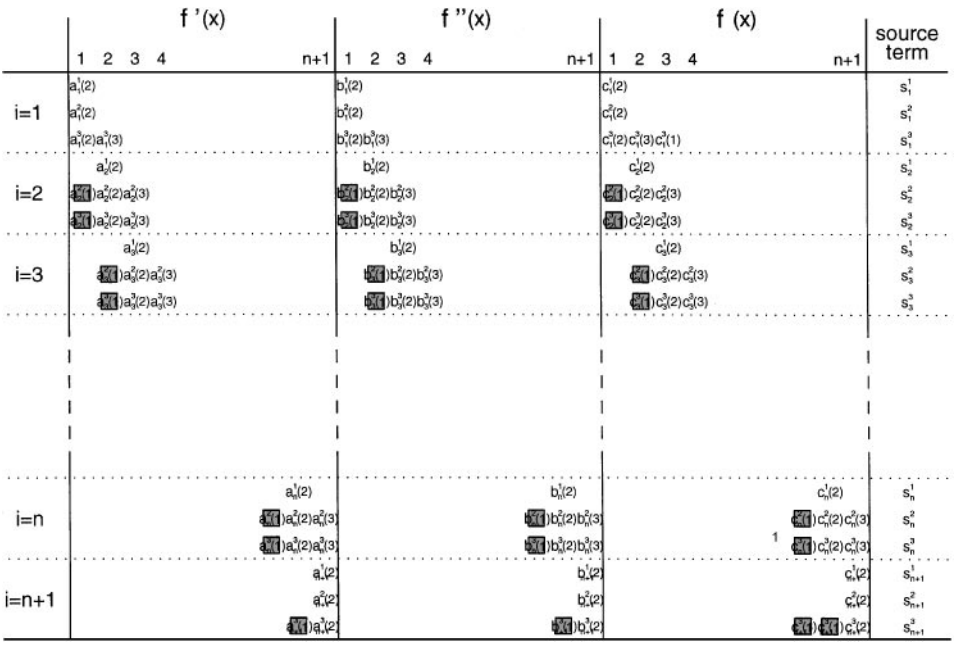

FIG. 13. The triple-forward elimination of the CCD coefficient matrix for FDE with nonperiodic boundaries. Here ■ denotes eliminated coefficients.

unknowns at the right boundary (x_{N+1}),

$$a^j_{N+1}(2)\left(\frac{\delta f}{\delta x}\right)_{N+1} + b^j_{n+1}(2)\left(\frac{\delta^2 f}{\delta x^2}\right)_{N+1} + c^j_{N+1}(2)\, f_{N+1} = s^j_{N+1}, \quad j = 1, 2, 3.$$

Solving this set of three algebraic equations, we obtain f_{N+1}, $(\delta f/\delta x)_{N+1}$ and $(\delta^2 f/\delta x^2)_{N+1}$.

The substitution procedure starts from the second right point (x_N). The dependent variable and its first- and second-order differences at any grid point (x_i) are computed from the following substitution ($i = N, N-1, \ldots, 1$):

$$a^j_i(2)\left(\frac{\delta f}{\delta x}\right)_i + b^j_i(2)\left(\frac{\delta^2 f}{\delta x^2}\right)_i + c^j_i(2)\, f_i$$

$$= s^j_i - a^j_i(3)\left(\frac{\delta f}{\delta x}\right)_{i+1} - b^j_i(3)\left(\frac{\delta^2 f}{\delta x^2}\right)_{i+1} - c^j_i(3)\, f_{i+1}, \quad j = 1, 2, 3.$$

Appendix 7: Periodic CCD FDE Solution

The structure of the periodic CCD PDE matrix is shown in Fig. 14. We can use a similar triple-forward elimination and triple-backward substitution procedures. Figure 15 shows the structure after the triple-forward elimination procedure, where the shadowed areas mean the eliminated elements.

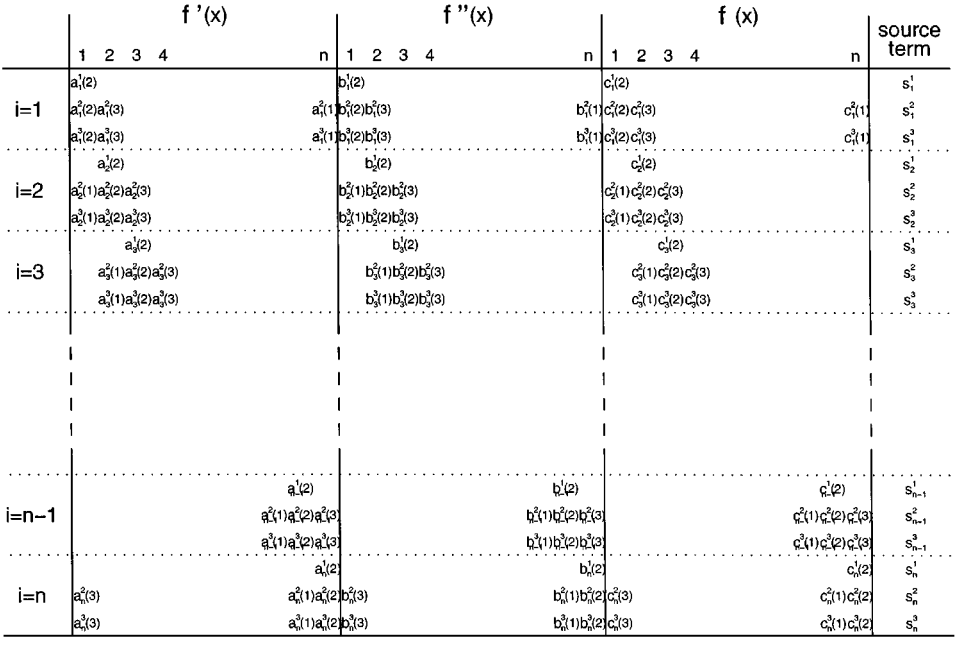

FIG. 14. Structure of the CCD coefficient matrix for FDE with periodic boundary.

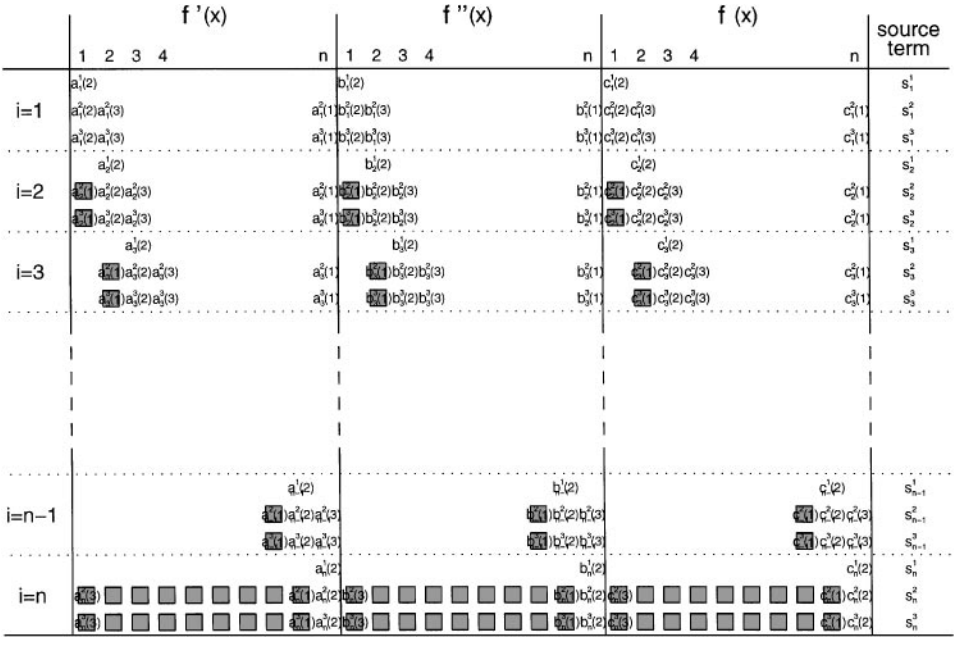

FIG. 15. The triple-forward elimination of the CCD coefficient matrix for FDE with periodic boundaries. Here ■ denotes eliminated coefficients.

ACKNOWLEDGMENTS

This work was funded by the Office of Naval Research NOMP Program, the Naval Oceanographic Office, and the Naval Postgraduate School.

REFERENCES

1. J. D. McCalpin, *Int. J. Numer. Methods Fluids* **18**, 361 (1994).
2. Y. Adam, *Math. Appl.* **1**, 393 (1975).
3. R. S. Hirsh, *J. Comput. Phys.* **19**, 90 (1975).
4. H. O. Kreiss, "Methods for the Approximate Solution of Time Dependent Problems," GARP Report No. 13, Geneva, 1975.
5. Y. Adam, *J. Comput. Phys.* **24**, 10 (1977).
6. I. M. Navon and H. A. Riphagen, *Mon. Wea. Rev.* **107**, 1107 (1979).
7. W. J. Goedheer and J. H. M. Potters, *J. Comput. Phys.* **61**, 269 (1985).
8. H.-R. Chang and H. N. Shirer, *Mon. Wea. Rev.* **113**, 409 (1985).
9. C. Fan, *Adv. in Hydrodynamics* **3**(4), 34 (1988). [Chinese]
10. S. K. Lele, *J. Comput. Phys.* **103**, 16 (1992).
11. A. I. Tolstykh, *Sov. Math. Dokl.* **44**, 69 (1992).
12. M. H. Carpenter, D. Gottlieb, and S. Abarbanel, *J. Comput. Phys.* **108**, 272 (1993).
13. M. H. Carpenter, D. Gottlieb, and S. Abarbanel, *J. Comput. Phys.* **111**, 220 (1994).
14. W. E. and J.-G. Liu, *J. Comput. Phys.* **124**, 368 (1996).
15. W. E. and J.-G. Liu, *J. Comput. Phys.* **126**, 122 (1996).
16. B. Swartz and B. Wendroff, *SIAM J. Numer. Anal.* **11**, 979 (1974).
17. H. O. Kreiss and J. Oliger, *Tellus* **3**, 99 (1972).
18. R. Vichnevetsky and J. B. Bowles, *Fourier Analysis of Numerical Approximations of Hyperbolic Equations* (SIAM, Philadelphia, 1982).
19. K. V. Roberts and N. O. Weiss, *Math. Comput.* **20**, 272 (1966).
20. J. E. Fromm, *Phys. Fluids Suppl. II* **3**, 12 (1969).
21. S. A. Orszag, *J. Fluid Mech.* **49**, 75 (1971).
22. S. A. Orszag, *Stud. Appl. Math.* **49**, 395 (1971).
23. H. M. Stommel, *Trans. Amer. Geophys. Union* **29**, 202 (1948).