Journal of Computational Physics **148,** 663–674 (1999)
Article ID jcph.1998.6141, available online at http://www.idealibrary.com on IDEAL®

NOTE

A Three-Point Sixth-Order Nonuniform Combined Compact Difference Scheme

A three-point nonuniform combined compact difference (NCCD) scheme with sixth-order accuracy is proposed for numerical models. The NCCD scheme is a generalization of the previously proposed combined compact difference (CCD) scheme with a global Hermitan polynomial spline and has major improved features such as error and computational (CPU) time reduction. For nonperiodic boundaries, additional sixth- or fifth-order nonuniform boundary conditions are proposed. The NCCD scheme with either sixth- or fifth-order additional boundary conditions can increase the accuracy and decrease the CPU time about 1–2 orders of magnitude, compared to the CCD scheme. © 1999 Academic Press

1. INTRODUCTION

Following the trend toward highly accurate numerical schemes of partial differential equations (PDE) led by many authors (e.g., Adam [1]; Chu and Fan [2]; Hirsh [5]; Rubin and Khosla [8]; Navon and Riphagen [6]), Chu and Fan [2] proposed a three-point sixth-order uniform combined compact difference (CCD) scheme to increase accuracy. This scheme follows earlier work on the use of second derivatives in compact differencing (such as Rubin and Khosla [8]),

$$\sum_{k=-1}^{1} (a_k f_{i+k} + b_k f'_{i+k} + c_k f''_{i+k}) = 0, \tag{1}$$

which is referred as the Hermite formula. Here, f is a dependent variable.

The ocean variability is not uniform. For example, the western boundary currents (e.g., the Gulf Stream, Kuroshio) have much larger variability than the ocean interior. An ideal treatment is to use a nonuniform scheme: high resolution grids for high variability areas and low resolution grids for low variability areas. Goedheer and Potters [4] proposed a nonuniform grid for a fourth-order compact difference scheme. Following their path, we propose in this paper a three-point sixth-order nonuniform combined compact (NCCD) scheme with sixth-order or fifth-order accuracy at both interior and exterior boundaries.

0021-9991/99 $30.00

2. NCCD SCHEME

Let $f(x)$ be defined on the interval, $0 \le x \le L$. Use a nonuniform grid, $0 = x_0 < x_1 < x_2 < \cdots < x_N = L$ with a nonuniform space $h_i \equiv \Delta x_i = x_i - x_{i-1}$ $(i = 1, 2, \ldots, N)$. 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}, and f'_{i+1}, f''_{i+1}. Let $H_i(x)$ be a local Hermitian polynomial defined on a closed interval $[x_{i-1,}x_{i+1}]$ with a spacing of $(h_i + h_{i+1})$, and determined by f evaluated at x_{i-1}, x_i, x_{i+1}, and its derivatives f', and f'' evaluated at x_{i-1} and x_{i+1} (Fig. 1a),

$$\begin{aligned} H_i(x) = {} & f_i + (f_{i-1} - f_i)\Phi_1(\xi) + (f_{i+1} - f_i)\Phi_2(\xi) + f'_{i-1}h_i\Phi_3(\xi) \\ & + f'_{i+1}h_i\Phi_4(\xi) + f''_{i-1}h_i^2\Phi_5(\xi) + f''_{i+1}h_i^2\Phi_6(\xi), \end{aligned} \tag{2}$$

where $\xi = (x - x_i)/h_i$ and

$$\Phi_j(\xi) = a_j\xi + b_j\xi^2 + c_j\xi^3 + d_j\xi^4 + e_j\xi^5 + g_j\xi^6, \quad j = 1, 2, \ldots, 6, \tag{3}$$

are six elements of the local Hermitian polynomial $H_i(x)$ satisfying the following homogeneous conditions

$$\begin{gathered} \Phi_j(-1) = 0, \quad \Phi_j(k_i) = 0, \quad \Phi'_j(-1) = 0, \quad \Phi'_j(k_i) = 0, \\ \Phi''_j(-1) = 0, \quad \Phi''_j(k_i) = 0, \quad j = 1, 2, \ldots, 6, \end{gathered}$$

except for

$$\Phi_1(-1) = 1, \quad \Phi_2(k_i) = 1, \quad \Phi'_3(-1) = 1, \quad \Phi'_4(k_i) = 1, \quad \Phi''_5(-1) = 1, \quad \Phi''_6(k_i) = 1,$$

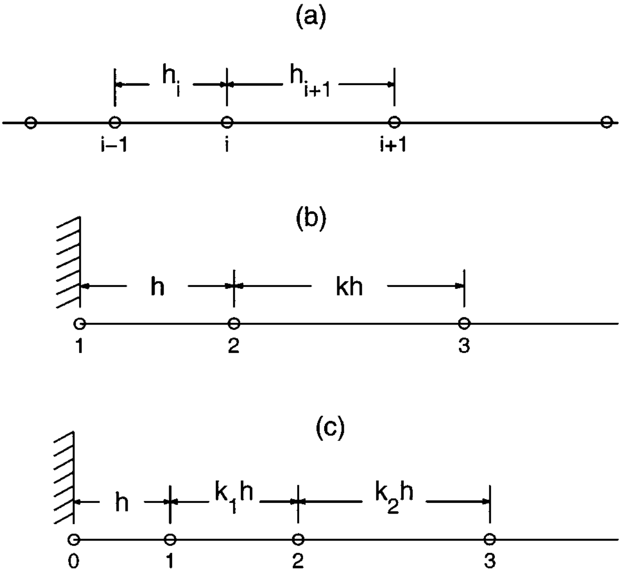


FIG. 1. Nonuniform grid systems: (a) three-point grid in the interior domain; (b) three-point grid at the left boundary; and (c) four-point grid at the left boundary.

where $k_i = h_{i+1}/h_i$.

If we define

$$\left(\frac{\delta f}{\delta x}\right)_i = H_i'(x_i), \quad \left(\frac{\delta^2 f}{\delta x^2}\right)_i = H_i''(x_i),$$

$$\left(\frac{\delta f}{\delta x}\right)_{i-1} = f_{i-1}', \quad \left(\frac{\delta f}{\delta x}\right)_{i+1} = f_{i+1}', \quad \left(\frac{\delta^2 f}{\delta x^2}\right)_{i-1} = f_{i-1}'', \quad \left(\frac{\delta^2 f}{\delta x^2}\right)_{i+1} = f_{i-1}'',$$

the NCCD scheme is given by

$$\begin{aligned}
&\frac{k_i^3(2k_i+5)}{(k_i+1)^4}\left(\frac{\delta f}{\delta x}\right)_{i-1} + \left(\frac{\delta f}{\delta x}\right)_i + \frac{(5k_i+2)}{(k_i+1)^4}\left(\frac{\delta f}{\delta x}\right)_{i+1} \\
&\quad + \frac{k_i^3}{2(k_i+1)^3}h_i\left(\frac{\delta^2 f}{\delta x^2}\right)_{i-1} - \frac{k_i}{2(k_i+1)^3}h_i\left(\frac{\delta^2 f}{\delta x^2}\right)_{i+1} \\
&= \frac{3\left(5k_i^2+4k_i+1\right)}{(k_i+1)^5}\frac{f_{i+1}-f_i}{k_i h_i} + \frac{3k_i^3\left(1k_i^2+4k_i+5\right)}{(k_i+1)^5}\frac{f_i-f_{i-1}}{h_i}
\end{aligned} \tag{4}$$

and

$$\begin{aligned}
&-\frac{6k_i^2\left(5-k_i-k_i^2\right)}{(k_i+1)^4}\left(\frac{\delta f}{\delta x}\right)_{i-1} + \frac{6\left(5k_i^2-k_i-1\right)}{k_i(k_i+1)^4}\left(\frac{\delta f}{\delta x}\right)_{i+1} \\
&\quad - \frac{k_i^2(3-2k_i)}{(k_i+1)^3}h_i\left(\frac{\delta^2 f}{\delta x^2}\right)_{i-1} + h_i\left(\frac{\delta^2 f}{\delta x^2}\right)_i - \frac{(3k_i-2)}{(k_i+1)^3}h_i\left(\frac{\delta^2 f}{\delta x^2}\right)_{i+1} \\
&= \frac{6}{k_i(k_i+1)^5}\Bigg(\left(15k_i^3+4k_i^2-2k_i-1\right)\frac{f_{i+1}-f_i}{k_i h_i} \\
&\quad - k_i^3\left(15+4k_i-2k_i^2-k_i^3\right)\frac{f_i-f_{i-1}}{h_i}\Bigg).
\end{aligned} \tag{5}$$

Expanding the variable f, f', and f'' into the Taylor series at points $i-1$ and $i+1$, the truncation errors are estimated as $-k_i^3 f_i^{(7)} h_i^6/7!$ for (4) and $-2k_i^2(9k_i^2-17k_i+9) f_i^{(8)} h_i^7/8!$ for (5).

3. A GLOBAL HERMITAN POLYNOMIAL SPLINE ON A NONUNIFORM GRID

Let $H_i(x)$ be a local Hermitian polynomial defined on a closed interval $[x_{i-1,}x_{i+1}]$ with a spacing of $(h_i + h_{i+1})$. We may define a global Hermitan polynomial spine by

$$\begin{aligned}
H_g(x) &= H_2(x), \quad 0 = x_1 \le x < x_2, \\
H_g(x) &= \frac{x_{i+1}-x}{h_i}H_i(x) + \frac{x-x_i}{h_i}H_{i+1}(x), \quad x_i \le x < x_{i+1}\ (i = 2, 3, \ldots, N-2) \\
H_g(x) &= H_{N-1}(x), \quad x_{N-1} \le x \le x_N,
\end{aligned} \tag{6}$$

which has up to third-order continuous derivatives at $[x_1, x_N]$.

4. BOUNDARY ALGORITHMS

For periodic boundaries, the NCCD scheme automatically provides the sixth-order accuracy at the boundaries. But for nonperiodic boundaries, we propose a sixth-order and a fifth-order boundary algorithm for solving finite difference equations (FDE) and a fifth-order boundary algorithm for difference calculation. For simplicity, we discuss left boundary ($i = 0$) as an example. The treatment of the right boundary ($i = N$) is the as same as for the left boundary.

4.1. Finite Difference Equation (FDE)

We have a choice of using sixth-order or fifth-order additional boundary conditions. The grid structure is illustrated in Fig. 1b.

4.1.1. *Sixth-Order Accuracy Formulation*

The sixth-order accuracy at the boundary is based on using a global Hermitan polynomial spline and an integration equation for the boundary cell. For a second-order partial differential equation with constant coefficients,

$$af''(x) + bf'(x) + cf(x) = s(x), \tag{7}$$

we propose an extra boundary condition with sixth-order accuracy as

$$a_1\left(\frac{\delta f}{\delta x}\right)_0 + a_2\left(\frac{\delta f}{\delta x}\right)_1 + b_1\left(\frac{\delta^2 f}{\delta x^2}\right)_0 + b_2\left(\frac{\delta^2 f}{\delta x^2}\right)_1 + c_1 f_0 + c_2 f_1 + c_3 f_2 = d. \tag{8}$$

4.1.2. *Fifth-Order Accuracy Formulation*

If we can bear a little less accuracy (fifth-order) at the boundary, the formulation will be much simpler,

$$\begin{aligned}&\left(3k^2 - 2k + 1 - \frac{1}{(1+k)^2}\right)\left(\frac{\delta f}{\delta x}\right)_0 + k(3k-1)\left(\frac{\delta f}{\delta x}\right)_1 + \frac{k^3}{2(1+k)}h\left(\frac{\delta^2 f}{\delta x^2}\right)_0\\ &\quad - \frac{k^2}{2}h\left(\frac{\delta^2 f}{\delta x^2}\right)_1 + \left(6k^2 - 3k + 1 - \frac{1}{(1+k)^3}\right)\frac{f_0}{h}\\ &\quad - (6k^2 - 3k + 1)\frac{f_1}{h} + \frac{1}{(1+k)^3}\frac{f_2}{h} = 0.\end{aligned} \tag{9}$$

4.2. Finite Difference Calculation

The NCCD can also be used to calculate the high-order finite difference. There are $2 * (N+1)$ unknown variables $[(\delta f/\delta x)_i, (\delta^2 f/\delta x^2)_i, i = 0, 1, 2, \ldots, N]$, but there only are $2 * (N-1)$ equations (4) and (5) at internal nodes. Therefore, at each boundary point two additional conditions are needed to close the system. Here, we only show the left boundary. The first additional boundary condition is (9). Expanding the variable f into the Taylor series in another form (Fig. 1c), we obtain the second additional boundary condition,

$$\left(\frac{\delta f}{\delta x}\right)_0 + a_1\left(\frac{\delta f}{\delta x}\right)_1 + a_2 h\left(\frac{\delta^2 f}{\delta x^2}\right)_1 = a_3\frac{f_1 - f_0}{h} + a_4\frac{f_2 - f_0}{h} + a_5\frac{f_3 - f_0}{h}. \quad (10)$$

5. STOMMEL OCEAN MODEL

We use the Stommel ocean model [9] to compare the accuracy and CPU between NCCD and CCD schemes.

5.1. Model Description

Stommel [9] 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. The x and y axes point eastward and northward, respectively. The boundaries of the ocean are at $x = 0$, L_x and $y = 0$, L_y. The ocean is considered as a homogeneous and incompressible layer of constant depth H when at rest. Stommel derived an equation for the streamfunction ψ,

$$\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right)\Psi + \hat{\beta}\frac{\partial \Psi}{\partial x} = -\gamma \sin\left(\frac{\pi}{L_y}y\right) \quad (11)$$

with the boundary conditions:

$$\Psi(0, y) = \Psi(L_x, y) = \Psi(x, 0) = \Psi(x, L_y) = 0. \quad (12)$$

Here, the right-hand side of (11) indicates the wind forcing, and $\hat{\beta}$, γ represent the latitudinal change of the Coriolis parameter and the strength of the surface wind stress, respectively. The analytical solution of (11) with the boundary conditions (12) is given by (Fig. 2)

$$\Psi^{(a)}(x, y) = -\gamma\left(\frac{L_y}{\pi}\right)^2 \sin\left(\frac{\pi}{L_y}y\right)(pe^{Ax} + qe^{Bx} - 1). \quad (13)$$

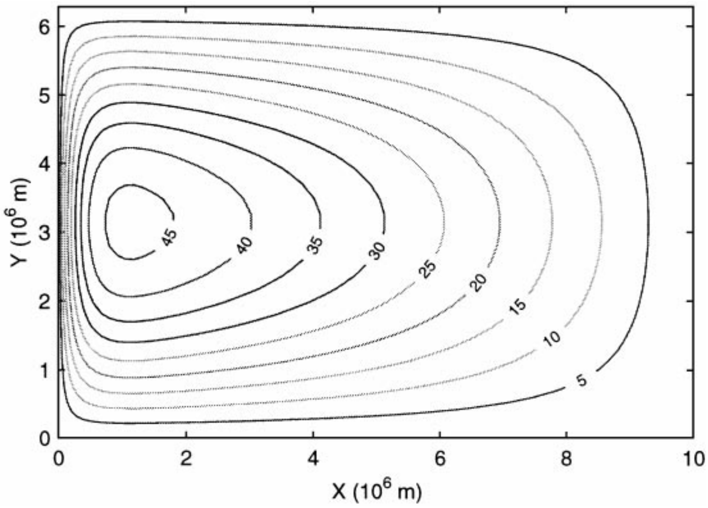


FIG. 2. Streamfunction (m^2/s) obtained from Stommel ocean model.

The physical parameters selected by Stommel [9] are

$$L_x = 10^7\,\mathrm{m}, \quad L_y = 2\pi \times 10^6\,\mathrm{m}, \quad \hat{\beta} = \frac{1}{3} \times 10^{-5}\,\mathrm{m}^{-1}, \quad \gamma = \frac{1}{4} \times 10^{-10}\,\mathrm{m}^{-2}. \qquad (14)$$

5.2. Two-Dimensional Nonuniform Grid

We discretize the domain into $0 = x_0 < x_1 < x_2 < \cdots < x_M = L_x$; $0 = y_0 < y_1 < y_2 < \cdots < y_N = L_y$ with nonuniform spacing $(\Delta x_i,\ \Delta y_j)$, that is,

$$x_i = x_{i-1} + \Delta x_i, \quad y_j = y_{j-1} + \Delta y_j, \quad i = 1, 2, \ldots, M;\ j = 1, 2, \ldots, N, \qquad (15)$$

where M and N are numbers of grid cells in the x and y directions, respectively. A two-dimensional nonuniform grid (Fig. 3) was used with left-to-right decreasing resolution in the x-direction,

$$\Delta x_i = \Delta x_\infty \left[1 - \exp\left(-\alpha_x \frac{i}{M}\right)\right], \quad i = 1, 2, \ldots, M, \qquad (16)$$

and with boundary-to-interior decreasing resolution and symmetric at the middle in the y-direction,

$$\begin{aligned} \Delta y_j &= \Delta y_\infty \left[1 - \exp\left(-\alpha_y \frac{2j}{N}\right)\right], &&\text{if } 1 \le j \le \left[\frac{N}{2}\right], \\ \Delta y_j &= \Delta y_{N-j+1}, &&\text{if } \left[\frac{N}{2}\right] \le j \le N, \end{aligned} \qquad (17)$$

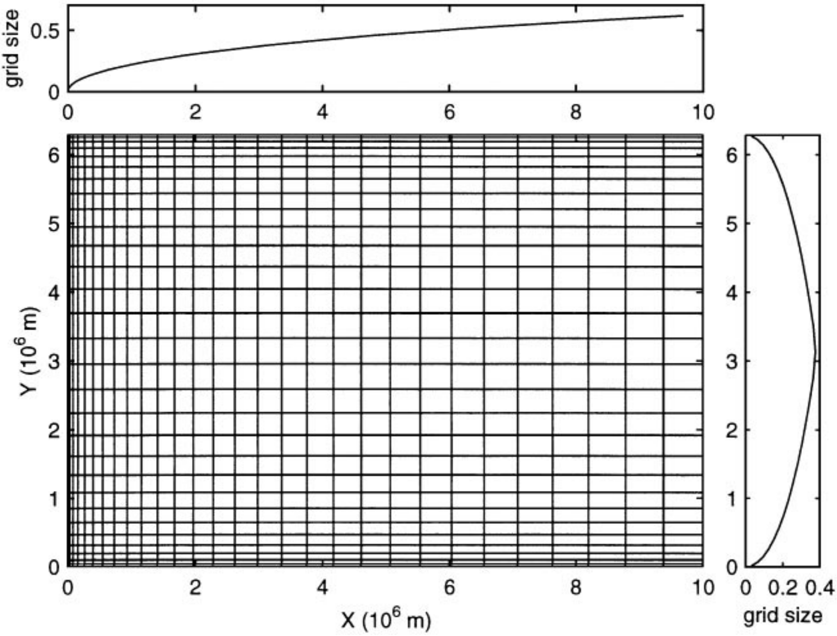


FIG. 3. Two-dimensional nonuniform grid.

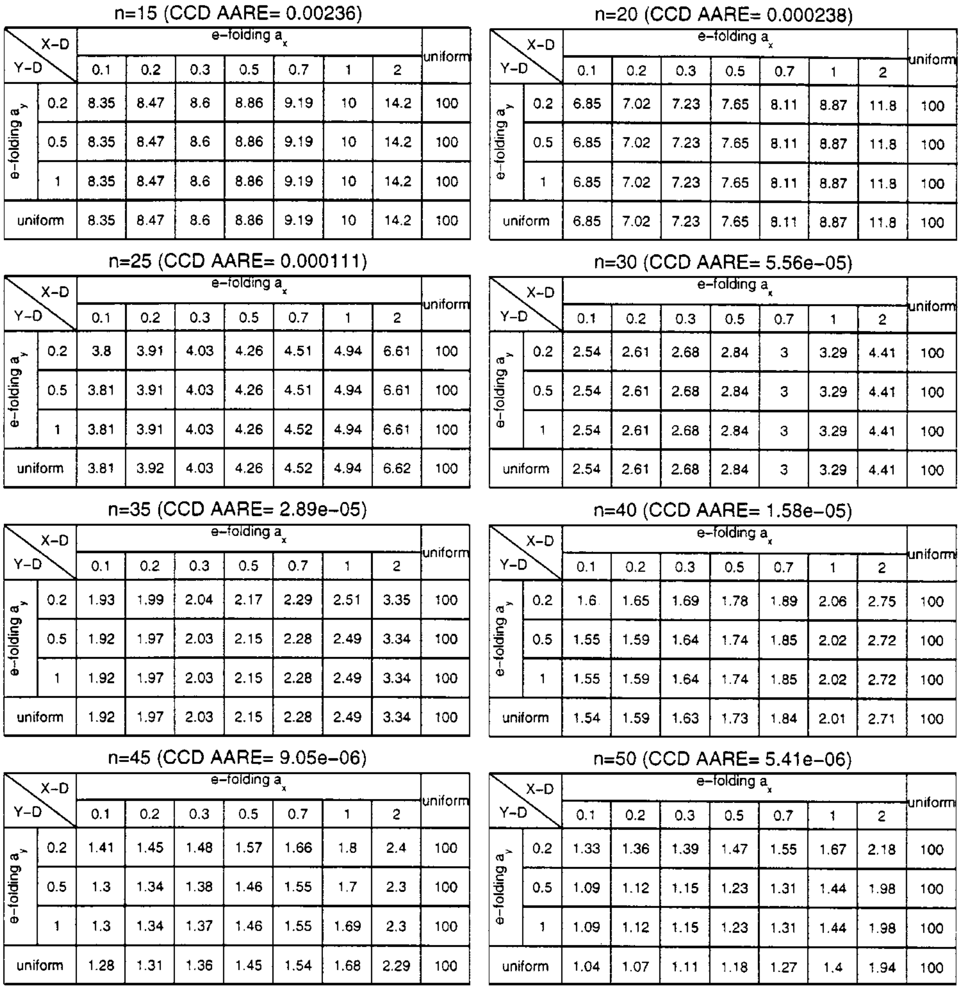
where α_x and α_y are the e-folding homogeneity parameters and Δx_∞ and Δy_∞ are the limiting grid size in the x and y directions, respectively. Increase of α_x (or α_y) means the increase of the homogeneity of the x-grid (or y-grid). The values of Δx_∞ and Δy_∞ are determined by satisfying the conditions $x_M = L_x$ and $y_N = L_y$. In this study, we used the same number of grid cells for both x and y directions $(M = N)$.

We use the same alternating direction implicit (ADI) method as described in Chu and Fan [3] to solve the two-dimensional equation (11) numerically. Such an iterative process stops when the correction at the iteration $k+1$,

$$\mathrm{corr}^{(k+1)} = \frac{\sum_{i=1}^{N-1}\sum_{j=1}^{M-1}\left|\Psi_{i,j}^{k+1} - \Psi_{i,j}^{k}\right|(\Delta x_i + \Delta x_{i+1})(\Delta y_j + \Delta y_{j+1})}{\sum_{i=1}^{N-1}\sum_{j=1}^{M-1}\left|\Psi_{i,j}^{k}\right|(\Delta x_i + \Delta x_{i+1})(\Delta y_j + \Delta y_{j+1})}, \tag{18}$$

is smaller than 10^{-8}. The numerical solution for the grid point (x_i, y_j) is $\Psi_{i,j}$. The

TABLE I
Comparison of Relative Average Error (%)



n=15 (CCD AARE= 0.00236)

X-D		e-folding a_x							uniform
Y-D		0.1	0.2	0.3	0.5	0.7	1	2	
e-folding a_y	0.2	8.35	8.47	8.6	8.86	9.19	10	14.2	100
	0.5	8.35	8.47	8.6	8.86	9.19	10	14.2	100
	1	8.35	8.47	8.6	8.86	9.19	10	14.2	100
uniform		8.35	8.47	8.6	8.86	9.19	10	14.2	100

n=20 (CCD AARE= 0.000238)

X-D		e-folding a_x							uniform
Y-D		0.1	0.2	0.3	0.5	0.7	1	2	
e-folding a_y	0.2	6.85	7.02	7.23	7.65	8.11	8.87	11.8	100
	0.5	6.85	7.02	7.23	7.65	8.11	8.87	11.8	100
	1	6.85	7.02	7.23	7.65	8.11	8.87	11.8	100
uniform		6.85	7.02	7.23	7.65	8.11	8.87	11.8	100

n=25 (CCD AARE= 0.000111)

X-D		e-folding a_x							uniform
Y-D		0.1	0.2	0.3	0.5	0.7	1	2	
e-folding a_y	0.2	3.8	3.91	4.03	4.26	4.51	4.94	6.61	100
	0.5	3.81	3.91	4.03	4.26	4.51	4.94	6.61	100
	1	3.81	3.91	4.03	4.26	4.52	4.94	6.61	100
uniform		3.81	3.92	4.03	4.26	4.52	4.94	6.62	100

n=30 (CCD AARE= 5.56e−05)

X-D		e-folding a_x							uniform
Y-D		0.1	0.2	0.3	0.5	0.7	1	2	
e-folding a_y	0.2	2.54	2.61	2.68	2.84	3	3.29	4.41	100
	0.5	2.54	2.61	2.68	2.84	3	3.29	4.41	100
	1	2.54	2.61	2.68	2.84	3	3.29	4.41	100
uniform		2.54	2.61	2.68	2.84	3	3.29	4.41	100

n=35 (CCD AARE= 2.89e−05)

X-D		e-folding a_x							uniform
Y-D		0.1	0.2	0.3	0.5	0.7	1	2	
e-folding a_y	0.2	1.93	1.99	2.04	2.17	2.29	2.51	3.35	100
	0.5	1.92	1.97	2.03	2.15	2.28	2.49	3.34	100
	1	1.92	1.97	2.03	2.15	2.28	2.49	3.34	100
uniform		1.92	1.97	2.03	2.15	2.28	2.49	3.34	100

n=40 (CCD AARE= 1.58e−05)

X-D		e-folding a_x							uniform
Y-D		0.1	0.2	0.3	0.5	0.7	1	2	
e-folding a_y	0.2	1.6	1.65	1.69	1.78	1.89	2.06	2.75	100
	0.5	1.55	1.59	1.64	1.74	1.85	2.02	2.72	100
	1	1.55	1.59	1.64	1.74	1.85	2.02	2.72	100
uniform		1.54	1.59	1.63	1.73	1.84	2.01	2.71	100

n=45 (CCD AARE= 9.05e−06)

X-D		e-folding a_x							uniform
Y-D		0.1	0.2	0.3	0.5	0.7	1	2	
e-folding a_y	0.2	1.41	1.45	1.48	1.57	1.66	1.8	2.4	100
	0.5	1.3	1.34	1.38	1.46	1.55	1.7	2.3	100
	1	1.3	1.34	1.37	1.46	1.55	1.69	2.3	100
uniform		1.28	1.31	1.36	1.45	1.54	1.68	2.29	100

n=50 (CCD AARE= 5.41e−06)

X-D		e-folding a_x							uniform
Y-D		0.1	0.2	0.3	0.5	0.7	1	2	
e-folding a_y	0.2	1.33	1.36	1.39	1.47	1.55	1.67	2.18	100
	0.5	1.09	1.12	1.15	1.23	1.31	1.44	1.98	100
	1	1.09	1.12	1.15	1.23	1.31	1.44	1.98	100
uniform		1.04	1.07	1.11	1.18	1.27	1.4	1.94	100

discretization error was represented by an area-averaged relative error (AARE),

$$\text{AARE} = \frac{\sum_{i=1}^{N-1}\sum_{j=1}^{M-1}\left|\Psi_{i,j}-\Psi_{i,j}^{(a)}\right|(\Delta x_i+\Delta x_{i+1})(\Delta y_j+\Delta y_{j+1})}{\sum_{i=1}^{N-1}\sum_{j=1}^{M-1}\left|\Psi_{i,j}^{(a)}\right|(\Delta x_i+\Delta x_{i+1})(\Delta y_j+\Delta y_{j+1})}. \tag{19}$$

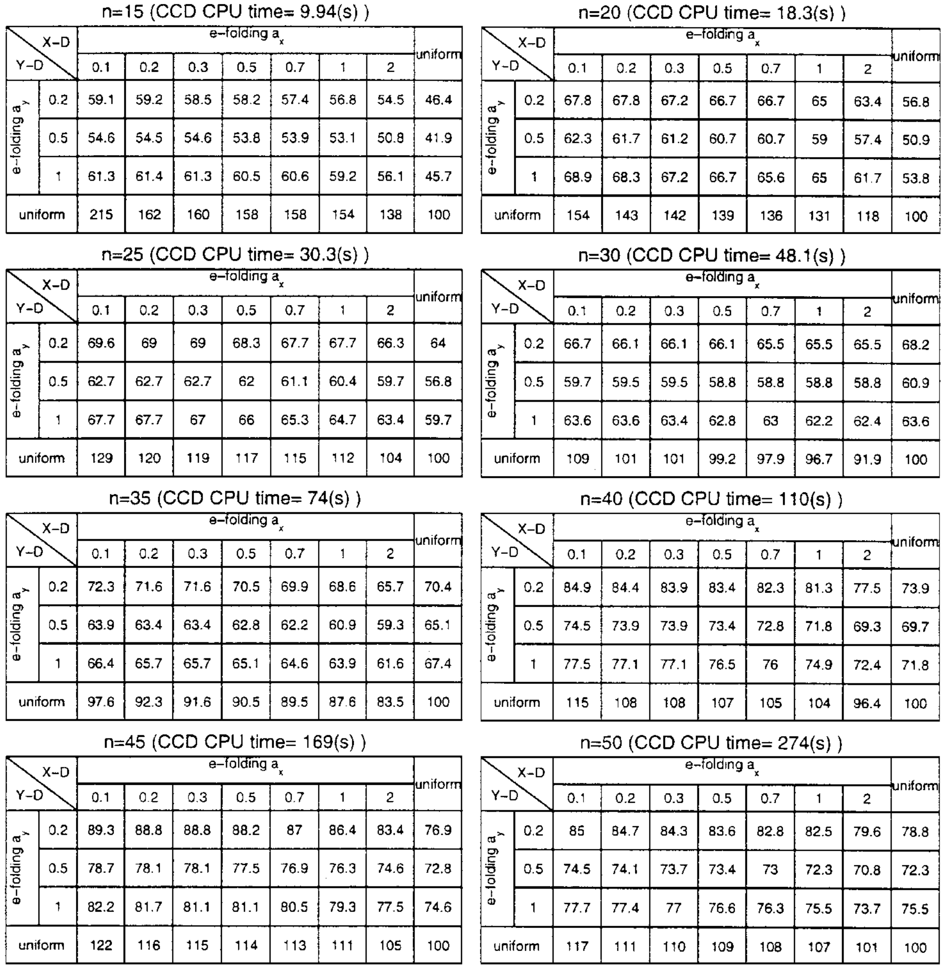
We solved (11) numerically with both NCCD and CCD schemes under various horizontal resolutions and recorded the CPU time (a SUN Sparc-20 was used) for each run. To test the sensitivity of NCCD scheme on the grid size and e-folding scale, we computed the discretization error with different cell numbers $M = N = 15, 20, 25, 30, 35, 40, 45, 50$, different x-directional e-folding scales $\alpha_x = 0.1$, 0.2, 0.3, 0.5, 0.7, 1, 2 and different y-directional e-folding scales $\alpha_y = 0.2$, 0.5, 1. Using the uniform CCD scheme as the reference, we define the error and CPU ratios (%) by

$$\varepsilon(N,\alpha_x,\alpha_y) = \frac{\text{AARE}^{(\text{NCCD})}(N,\alpha_x,\alpha_y)}{\text{AARE}^{(\text{CCD})}(N)}, \quad \lambda(\alpha_x,\alpha_y) = \frac{\text{CPU}^{(\text{NCCD})}(N,\alpha_x,\alpha_y)}{\text{CPU}^{(\text{CCD})}(N)}. \tag{20}$$

Thus, the uniform CCD scheme has a value of 100% for both ϵ and λ.

TABLE II

Comparison of Relative CPU Time (%)



n=15 (CCD CPU time= 9.94(s))

X-D Y-D		e-folding a_x							uniform
		0.1	0.2	0.3	0.5	0.7	1	2	
e-folding a_y	0.2	59.1	59.2	58.5	58.2	57.4	56.8	54.5	46.4
	0.5	54.6	54.5	54.6	53.8	53.9	53.1	50.8	41.9
	1	61.3	61.4	61.3	60.5	60.6	59.2	56.1	45.7
uniform		215	162	160	158	158	154	138	100

n=20 (CCD CPU time= 18.3(s))

X-D Y-D		e-folding a_x							uniform
		0.1	0.2	0.3	0.5	0.7	1	2	
e-folding a_y	0.2	67.8	67.8	67.2	66.7	66.7	65	63.4	56.8
	0.5	62.3	61.7	61.2	60.7	60.7	59	57.4	50.9
	1	68.9	68.3	67.2	66.7	65.6	65	61.7	53.8
uniform		154	143	142	139	136	131	118	100

n=25 (CCD CPU time= 30.3(s))

X-D Y-D		e-folding a_x							uniform
		0.1	0.2	0.3	0.5	0.7	1	2	
e-folding a_y	0.2	69.6	69	69	68.3	67.7	67.7	66.3	64
	0.5	62.7	62.7	62.7	62	61.1	60.4	59.7	56.8
	1	67.7	67.7	67	66	65.3	64.7	63.4	59.7
uniform		129	120	119	117	115	112	104	100

n=30 (CCD CPU time= 48.1(s))

X-D Y-D		e-folding a_x							uniform
		0.1	0.2	0.3	0.5	0.7	1	2	
e-folding a_y	0.2	66.7	66.1	66.1	66.1	65.5	65.5	65.5	68.2
	0.5	59.7	59.5	59.5	58.8	58.8	58.8	58.8	60.9
	1	63.6	63.6	63.4	62.8	63	62.2	62.4	63.6
uniform		109	101	101	99.2	97.9	96.7	91.9	100

n=35 (CCD CPU time= 74(s))

X-D Y-D		e-folding a_x							uniform
		0.1	0.2	0.3	0.5	0.7	1	2	
e-folding a_y	0.2	72.3	71.6	71.6	70.5	69.9	68.6	65.7	70.4
	0.5	63.9	63.4	63.4	62.8	62.2	60.9	59.3	65.1
	1	66.4	65.7	65.7	65.1	64.6	63.9	61.6	67.4
uniform		97.6	92.3	91.6	90.5	89.5	87.6	83.5	100

n=40 (CCD CPU time= 110(s))

X-D Y-D		e-folding a_x							uniform
		0.1	0.2	0.3	0.5	0.7	1	2	
e-folding a_y	0.2	84.9	84.4	83.9	83.4	82.3	81.3	77.5	73.9
	0.5	74.5	73.9	73.9	73.4	72.8	71.8	69.3	69.7
	1	77.5	77.1	77.1	76.5	76	74.9	72.4	71.8
uniform		115	108	108	107	105	104	96.4	100

n=45 (CCD CPU time= 169(s))

X-D Y-D		e-folding a_x							uniform
		0.1	0.2	0.3	0.5	0.7	1	2	
e-folding a_y	0.2	89.3	88.8	88.8	88.2	87	86.4	83.4	76.9
	0.5	78.7	78.1	78.1	77.5	76.9	76.3	74.6	72.8
	1	82.2	81.7	81.1	81.1	80.5	79.3	77.5	74.6
uniform		122	116	115	114	113	111	105	100

n=50 (CCD CPU time= 274(s))

X-D Y-D		e-folding a_x							uniform
		0.1	0.2	0.3	0.5	0.7	1	2	
e-folding a_y	0.2	85	84.7	84.3	83.6	82.8	82.5	79.6	78.8
	0.5	74.5	74.1	73.7	73.4	73	72.3	70.8	72.3
	1	77.7	77.4	77	76.6	76.3	75.5	73.7	75.5
uniform		117	111	110	109	108	107	101	100

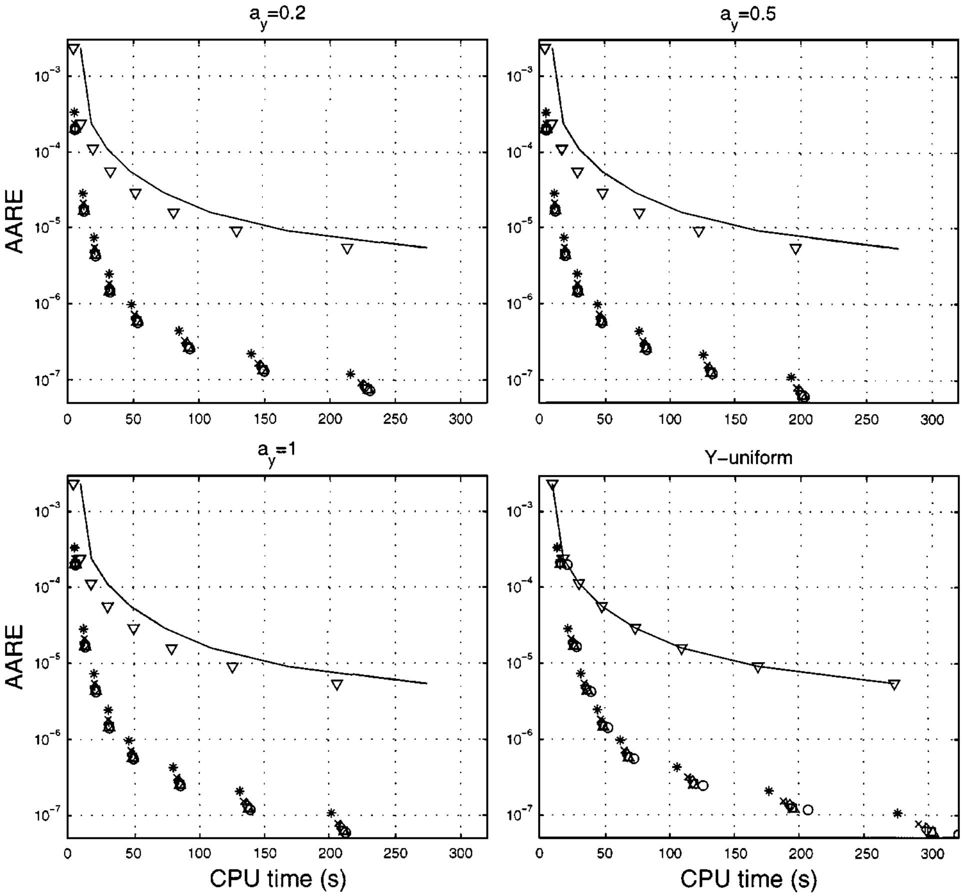


FIG. 4. The AARE-CPU diagrams for NCCD and CCD scheme comparison. Here, the solid curve indicates the CCD scheme. Eight symbols represent different α_x-values: 0.1 (○); 0.2 (△); 0.3 (□); 0.5 (◇); 0.7 (+); 1.0 (×); 2.0, (∗); and ∞ (▽).

Table I shows the distribution of $\varepsilon(N, \alpha_x, \alpha_y)$. The symbols "X-D" and "Y-D" mean the x-dependency and y-dependency, respectively. The last column (row) of the table indicates the use of the uniform grid in x-direction (y-direction). The AARE of the NCCD scheme is 1 to 14% of the AARE of the uniform CCD scheme. This error reduction enhances as N increases and as α_x or α_y decreases. We further notice that the AARE are very sensitive to the e-folding parameter α_x and very insensitive to α_y. Taking $N = 15$ as an example, ε decreases from 14.2% to 8.35% as α_x decreases from 2 to 0.1. However, ε keeps the same value as α_y varies. Such an x–y asymmetry is called by the inhomogeneous ocean variability. The analytical solution of the Stommel ocean model shows a strong variability of Ψ in the western boundary only (Fig. 2). The nonuniform scheme reduces the truncation error drastically when the fine resolution grid is applied there.

Table II shows the distribution of $\lambda(N, \alpha_x, \alpha_y)$. The CPU of the NCCD scheme is 50 to 89% of the CPU of the uniform CCD scheme. This CPU saving enhances as N decreases and as α_x increases. Taking $N = 15$, $\alpha_y = 0.5$ as an example, λ decreases from 54.6% to 50.8% as α_x increases from 0.1 to 2.

We use the AARE-CPU diagram to verify the performance of the NCCD scheme. When α_x and α_y are given, both AARE and CPU depend on N. Thus, variation of N creates a curve in

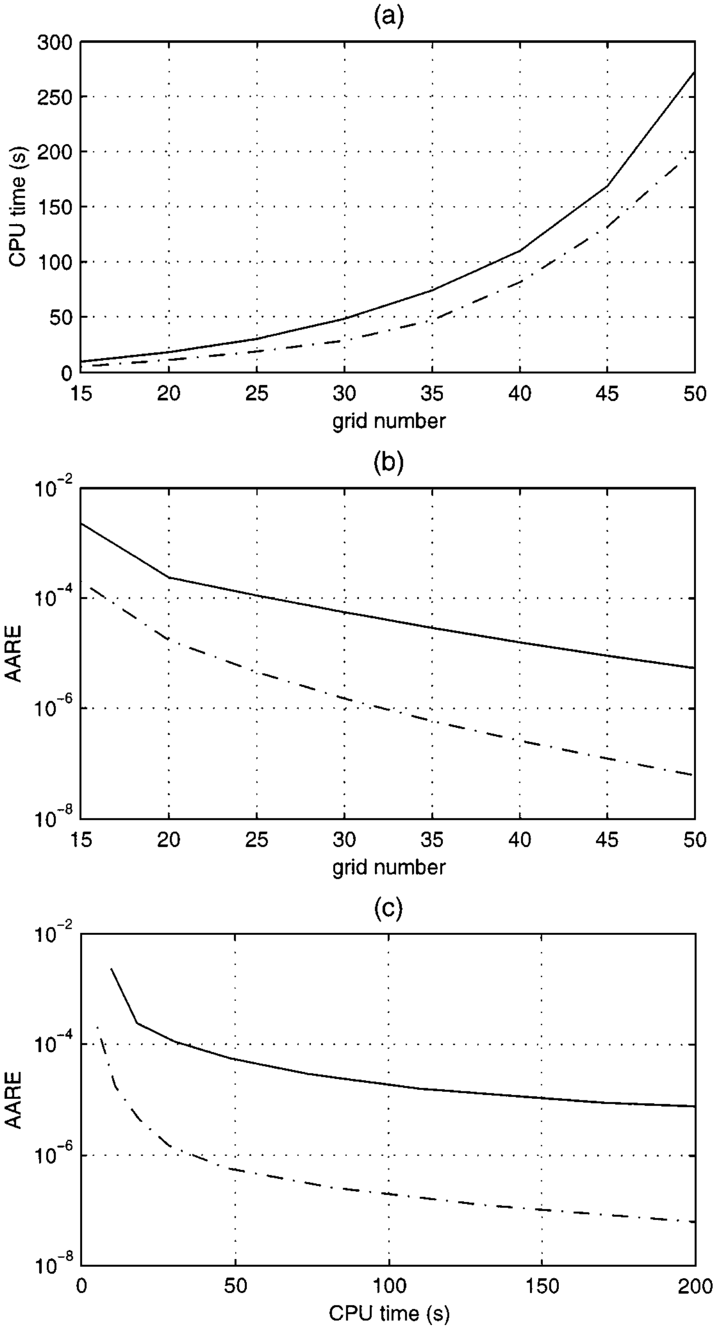


FIG. 5. Error comparison between specific NCCD scheme (with $\alpha_x = 0.3$ and $\alpha_y = 0.5$) and CCD scheme: (a) CPU time; (b) AARE; and (c) AARE-CPU diagram. Here, the solid curves denote the CCD scheme, and the dashed curves denote the NCCD scheme.

the AARE-CPU diagram. We only use eight different values for N (15, 20, 25, 30, 35, 40, 45, 50) in this study. Therefore, we have eight points given for α_x and α_y on the diagram. Figure 4 shows the AARE-CPU diagram for four different values of α_y (0.2, 0.5, 1, and ∞ (uniform)), respectively. The solid curve indicates the CCD scheme. Eight different symbols represent the eight different values used for α_x. The symbol "∇" indicates $\alpha_x = \infty$ (uniform). The other seven symbols represent the seven values we used for α_x (0.1, 0.2, 0.3, 0.5, 0.7, 1, 2). For the CCD scheme, $\alpha_x = \infty$ and $\alpha_y = \infty$. Again, we see the insensitivity of the NCCD scheme to α_y. For a given CPU time, the NCCD scheme increases the accuracy by 1–2 orders of magnitudes.

Figure 5 shows the comparison between a specific NCCD scheme (with $\alpha_x = 0.3$ and $\alpha_y = 0.5$) and the CCD scheme. The solid curves denote the CCD scheme, and the dashed curves denote the NCCD scheme. This specific NCCD scheme reduces both CPU time by 20–30% (Figure 5a) and AARE around 100 times (Fig. 5b) versus the CCD scheme. Such a reduction enhances as N increases. Such an error reduction is clearly seen in the AARE-CPU diagram for this specific NCCD scheme and the CCD scheme (Fig. 5c).

6. CONCLUSIONS

(1) This study shows that the NCCD scheme is a promising highly accurate method for both derivative computation and FDE solutions. The NCCD scheme has all the good features as the CCD scheme, such as three-point and sixth-order accuracy. Besides, the NCCD scheme represents the high variability area more accurate than the CCD scheme by using fine grid there. Our analysis shows that for the same CPU time, the NCCD scheme has an error reduction by 1–2 orders of magnitude, compared to the CCD scheme.

(2) To keep a three-point scheme, an additional boundary algorithm is needed. We propose both fifth- and sixth-order boundary algorithms. Since the fifth-order formulation is more simple than the six-order formulation and easy to use, especially in two-dimensional problems.

(3) Use of the NCCD scheme makes the convergence fast. In other words, it needs less iterations and saves CPU time.

(4) We found a global Hermitan polynomial spine in this study. Thus, the NCCD scheme is easily used in integral methods as well as in stagged grids.

ACKNOWLEDGMENTS

Many thanks to three anonymous reviewers whose comments improved the manuscript a great deal. This work was funded by the Office of Naval Research NOMP Program, the Naval Oceanographic Office, and the Naval Postgraduate School.

REFERENCES

1. Y. Adam, Highly accurate compact implicit methods and boundary conditions, *J. Comput. Phys.* **24**, 10 (1977).
2. P. C. Chu and C. Fan, Sixth-order difference scheme for sigma coordinate ocean models, *J. Phys. Oceanogr.* **27**, 2064 (1997).
3. P. C. Chu and C. Fan, A three-point combined compact difference scheme, *J. Comput. Phys.* **555**, 370 (1998).
4. W. J. Goedheer and J. H. M. Potters, A compact finite difference scheme on a nonequidistance mesh, *J. Comput. Phys.* **61**, 269 (1985).
5. R. S. Hirsh, Higher order accurate difference solutions of fluid mechanics problems by a compact differencing technique, *J. Comput. Phys.* **19**, 90 (1975).
6. I. M. Navon and H. A. Riphagen, An implicit compact fourth-order algorithm for solving the shallow-water equations in conservation-law form, *Mon. Weather Rev*. **107**, 1107 (1979).
7. P. M. Prenter, Splines and variational method (Wiley-Interscience, New York, 1974), p. 100.

8. S. G. Rubin and P. K. Khosla, Polynomial Interpolation methods for viscous flow calculations, *J. Comput. Phys.* **24**, 217 (1977).
9. H. M. Stommel, The western intensification of wind-driven ocean currents. *Trans. Amer. Geophys. Union* **29**, 202 (1948).

Received April 17, 1998; revised October 29, 1998

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