

# We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index  
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?  
Contact [book.department@intechopen.com](mailto:book.department@intechopen.com)

Numbers displayed above are based on latest data collected.  
For more information visit [www.intechopen.com](http://www.intechopen.com)



# A Fourth-Order Compact Finite Difference Scheme for Solving Unsteady Convection-Diffusion Equations

Wenyuan Liao<sup>1</sup> and Jianping Zhu<sup>2</sup>

<sup>1</sup>*University of Calgary*

<sup>2</sup>*University of Texas at Arlington*

<sup>1</sup>*Canada*

<sup>2</sup>*USA*

## 1. Introduction

Convection-diffusion equations are widely used for modeling and simulations of various complex phenomena in science and engineering (Hundsdoerfer & Verwer, 2003; Morton, 1996). Since for most application problems it is impossible to solve convection-diffusion equations analytically, efficient numerical algorithms are becoming increasingly important to numerical simulations involving convection-diffusion equations.

Recently a great deal of efforts have been devoted to developing high-order compact schemes, which utilize only the grid nodes directly adjacent to the central node. In (Noye & Tan, 1989), Noye and Tan derived a class of high-order implicit schemes for solving the one-dimensional unsteady convection-diffusion equations. This method is very stable and accurate (third-order in space and second-order in time). In (Gupta et al., 1984), a fourth-order finite difference scheme for a steady convection-diffusion equation with variable coefficients was proposed. The scheme is defined on a single square cell of size  $2\Delta x$  over a nine-point stencil. In (Rigal, 1994), Rigal provided an extensive analysis of the properties of a class of two- and three-level second-order difference schemes which have been proposed in (Rigal, 1989; 1990). In (Spotz & Carey, 2001), the two-dimensional HOC (High Order Compact) scheme proposed in (Gupta et al., 1984) was extended to solve unsteady one-dimensional convection-diffusion equations with variable coefficients and two-dimensional diffusion equations. This method was further extended by Kalita et al. in (Kalita et al., 2002) to a class of HOC schemes with weighted time discretization, and successfully used to solve unsteady two-dimensional convection-diffusion equations. In (Karaa & Zhang, 2004), Karaa and Zhang proposed a novel high-order alternating direction implicit method, based on the technique in (Zhang et al., 2002), for solving unsteady two-dimensional convection-diffusion problems. This new method is second-order in time and fourth-order in space, and is computationally efficient. In (Tian & Dai, 2007), Tian and Dai proposed a class of high-order compact exponential finite difference methods for solving one- and two-dimensional steady-state convection-diffusion problems. This method is nonoscillatory, fourth-order in space, and easy to implement. Some more recent high-order ADI methods for unsteady convection-diffusion equations can be found in (Tian & Ge, 2007; You, 2006).

For simplicity, we will use the following one-dimensional equation to describe the new method developed in this chapter:

$$\begin{aligned} u_t &= au_{xx} + c(x)u_x + f(u), \\ 0 < x < 1, 0 < t &\leq T, \\ u(0, t) &= g_1(t), u(1, t) = g_2(t), \\ u(x, 0) &= h(x), \end{aligned} \quad (1)$$

where  $a$  is a constant. Extensions to more complicated equations in one-dimension (for example when  $a$  is not a constant) are straightforward; and extensions to higher-dimensional equations will be briefly discussed in Section 6.

The existence of the convection term in Eq. (1) creates several difficulties when the equation is solved numerically using the finite difference schemes. It is well-known (Hundsdorfer & Verwer, 2003; Morton, 1996) that the convection term needs to be discretized using proper upwind finite difference schemes to avoid oscillations in convection dominated problems. If the sign of  $c(x)$  changes over the solution domain, the direction of the upwind scheme must also be changed accordingly. The order of accuracy of the upwind schemes is usually lower than the central difference schemes on the same finite difference stencil.

In addition, the convection term in Eq. (1) also makes it more difficult to use the fourth-order Padé approximations. For reaction-diffusion equations ( $c(x) = 0$ ), the Padé approximation can be used to achieve fourth-order accuracy on a 3-point stencil typically used for the standard second-order algorithms (Gu et al., 2003):

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = a \frac{\delta^2 u_i^{n+1}}{\Delta x^2 (1 + \frac{\delta^2}{12})} + f(u_i^{n+1}), \quad (2)$$

where  $i = 0, 1, \dots, M$ , and  $n = 0, 1, \dots, N$  are indices for spatial and temporal grid points, respectively,  $u_i^n$  is the numerical approximation to the exact solution  $u(x_i, t^n)$ , and the central difference operator  $\delta^2$  is defined as  $\delta^2 u_i^n = u_{i-1}^n - 2u_i^n + u_{i+1}^n$ . Applying the operator  $(1 + \frac{\delta^2}{12})$  to both sides of Eq. (2), we have

$$(1 + \frac{\delta^2}{12}) \frac{u_i^{n+1} - u_i^n}{\Delta t} = a \frac{\delta^2 u_i^{n+1}}{\Delta x^2} + (1 + \frac{\delta^2}{12}) f(u_i^{n+1}). \quad (3)$$

Eq. (3) is fourth-order accurate in space but contains only the second-order central difference operator  $\delta^2$  that requires only a 3-point stencil. This approach, however, does not work when the convection term is present in the equation. This is because the fourth-order Padé approximation for Eq. (1) leads to the following equation

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = a \frac{\delta^2 u_i^{n+1}}{\Delta x^2 (1 + \frac{\delta^2}{12})} + c_i \frac{\delta u_i^{n+1}}{2\Delta x (1 + \frac{\delta^2}{6})} + f(u_i^{n+1}),$$

where the central difference operator  $\delta$  is defined as  $\delta u_i^n = u_{i+1}^n - u_{i-1}^n$ . This equation cannot be simplified to an equation that is defined on a 3-point stencil in the same way as Eq. (2) is reduced to Eq. (3).

The new method discussed in this chapter eliminates the convection term  $u_x$  from Eq. (1) and solves  $v = u_x$  directly along with  $u$ . This makes it possible to use central finite difference schemes and higher-order Padé approximations for accurate and efficient numerical solutions

of convection-diffusion equations. It is unconditionally stable, and is particularly suitable for problems that require the solution of both  $u$  and  $u_x$ . For example, when solving the Black-Scholes option pricing model (Seydel, 2002)

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS \frac{\partial V}{\partial S} - rV = 0, \quad (4)$$

both the solution  $V$  and its derivative  $V_S$  are desired. The solution  $V$  is the price of an option and its derivative  $V_S$  is called the hedge delta that represents the sensitivity of the option value to the change of the underlining stock price.

The rest of this chapter is organized as follows: The description of the new method is given in the next section. Proof of unconditional stability of the new method is given in Section 3. Computational complexity of this new method is analyzed and compared with upwind and standard central finite difference schemes in Section 4. Several numerical examples are presented in Section 5, followed by conclusions in Section 6.

## 2. The new method

In this section, we will first outline the new algorithm that eliminates the convection term in a convection-diffusion equation to facilitate the use of central finite difference schemes and then discuss how the initial and boundary conditions are handled using the new algorithm.

### 2.1 Description of the new method

Setting  $v = u_x$  in Eq. (1), we have

$$u_t = au_{xx} + c(x)v + f(u) \quad (5)$$

Differentiating both sides of Eq. (1) with respect to  $x$  leads to

$$(u_x)_t = a(u_{xx})_x + c_x(x)u_x + c(x)(u_x)_x + f_u(u)u_x \quad (6)$$

which, considering  $v = u_x$ , can be written as

$$v_t = c(x)u_{xx} + av_{xx} + c_x(x)v + f_u(u)v \quad (7)$$

Eq. (5) and (7) now form a system of equations for  $u$  and  $v$ . They only involve diffusion term  $u_{xx}$  and  $v_{xx}$ , which can be discretized by the standard central finite difference schemes. If the 3-point second-order central difference scheme is used, the discretized equations will form a block tri-diagonal algebraic equation system with  $2 \times 2$  blocks. For nonlinear  $f(u)$ , the Newton's method or its variations can be used to solve the nonlinear system of algebraic equations.

If the fourth-order Padé approximation is used for Eqs. (5) and (7), we have

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = a \frac{\delta^2 u_i^{n+1}}{\Delta x^2 (1 + \frac{\delta^2}{12})} + c_i v_i + f(u_i^{n+1}), \quad (8)$$

$$\begin{aligned} \frac{v_i^{n+1} - v_i^n}{\Delta t} = & c_i \frac{\delta^2 u_i^{n+1}}{\Delta x^2 (1 + \frac{\delta^2}{12})} + a \frac{\delta^2 v_i^{n+1}}{\Delta x^2 (1 + \frac{\delta^2}{12})} \\ & + [(c_x)_i + f_u(u_i^{n+1})]v_i^{n+1}. \end{aligned} \quad (9)$$

Applying the operator  $(1 + \frac{\delta^2}{12})$  to both sides of these two equations, we obtain

$$(1 + \frac{\delta^2}{12}) \frac{u_i^{n+1} - u_i^n}{\Delta t} = a \frac{\delta^2 u_i^{n+1}}{\Delta x^2} + (1 + \frac{\delta^2}{12}) [c_i v_i + f(u_i^{n+1})], \quad (10)$$

$$(1 + \frac{\delta^2}{12}) \frac{v_i^{n+1} - v_i^n}{\Delta t} = c_i \frac{\delta^2 u_i^{n+1}}{\Delta x^2} + a \frac{\delta^2 v_i^{n+1}}{\Delta x^2} + (1 + \frac{\delta^2}{12}) [(c_x)_i + f_u(u_i^{n+1})] v_i^{n+1}. \quad (11)$$

Eqs. (10) and (11) are fourth-order accurate in space but only contains the second-order central difference operator  $\delta^2$  that is defined on a 3-point stencil. As a result, the discretized equations form a system of block tri-diagonal algebraic equations.

## 2.2 Initial and boundary conditions

The initial condition for  $v$  can be obtained by differentiating  $h(x)$ , the initial condition for  $u$  given in Eq. (1) with respect to  $x$ , i.e.

$$v(x, 0) = \frac{dh(x)}{dx}.$$

The boundary conditions for  $v$ , or for Eq. (7), are less straight forward. In the following, we discuss three different ways to generate boundary conditions for  $v$  at the spatial grid point  $i = 0$ , assuming Dirichlet boundary conditions are given for  $u$ . The boundary conditions for  $v$  at the spatial grid point  $i = M$  and other scenarios can be dealt with in similar ways.

Standard finite difference approximation: Since  $v = u_x$ , we have

$$v_0 = \frac{u_1 - u_0}{\Delta x}, \quad (12)$$

which provides an equation to complement the equation obtained by discretizing Eq. (7) at  $i = 1$ . This approximation is first-order accurate in space. If necessary, higher-order one-sided finite difference schemes can be used to approximate  $v = u_x$ .

Padé approximation: We can use the fourth-order Padé approximation at  $i = 1$  for  $v = u_x$

$$v_1 = \frac{\delta u_1}{2\Delta x(1 + \frac{\delta^2}{6})} = \frac{u_2 - u_0}{2\Delta x(1 + \frac{\delta^2}{6})}. \quad (13)$$

Applying  $(1 + \frac{\delta^2}{6})$  to both sides of Eq. (13), we have

$$(1 + \frac{\delta^2}{6}) v_1 = \frac{u_2 - u_0}{2\Delta x}, \quad (14)$$

or

$$\frac{1}{6} v_0 + \frac{2}{3} v_1 + \frac{1}{6} v_2 = \frac{u_2 - u_0}{2\Delta x}. \quad (15)$$

Solving  $v_0$  from Eq. (15), we obtain the boundary condition

$$v_0 = \frac{3(u_2 - u_0)}{\Delta x} - 4v_1 - v_2. \quad (16)$$

Direct integration of Eq. (1): This approach is particularly convenient for special cases of Eq. (1). For example, when dealing with steady state equation with a constant convection coefficient and constant  $f(u)$ , we have

$$au_{xx} + cu_x + d = 0, \quad (17)$$

where  $a, c$  and  $d$  are constants. Integrating this equation from  $x_0$  to  $x_1$ , we have

$$au_x|_{x_0}^{x_1} + cu|_{x_0}^{x_1} + d(x_1 - x_0) = 0, \quad (18)$$

or

$$av_1 - av_0 + cu_1 - cu_0 + d\Delta x = 0. \quad (19)$$

Solving  $v_0$  from Eq. (19), we obtain the boundary condition

$$v_0 = \frac{1}{a}(av_1 + cu_1 - cu_0 + d\Delta x). \quad (20)$$

This boundary condition has no truncation error since the integrations are carried out exactly. For more general cases when some of the terms in Eq. (1) cannot be integrated exactly, various numerical integration schemes, such as the second-order Trapezoidal scheme, can be used to generate boundary condition for  $v$  in a similar way.

### 3. Stability analysis

Stability is critical to numerical methods used to solve time-dependent systems. In this section, we conduct Von Neumann stability analysis for the new method combined with the standard second-order central difference scheme. The following one-dimensional linear convection-diffusion equation is used in the analysis

$$u_t = u_{xx} + cu_x \quad (21)$$

The new method combined with the second-order central difference for solving Eq. (21) is

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \frac{1}{\Delta x^2} \delta^2 u_i^{n+1} + cv_i^{n+1}, \quad (22)$$

$$\frac{v_i^{n+1} - v_i^n}{\Delta t} = \frac{c}{\Delta x^2} \delta^2 u_i^{n+1} + \frac{\delta^2}{\Delta x^2} v_i^{n+1}. \quad (23)$$

Taking the discrete Fourier transform of the above equations, we have

$$M \begin{bmatrix} \hat{u}^{n+1} \\ \hat{v}^{n+1} \end{bmatrix} = R \begin{bmatrix} \hat{u}^n \\ \hat{v}^n \end{bmatrix}. \quad (24)$$

Thus

$$\begin{bmatrix} \hat{u}^{n+1} \\ \hat{v}^{n+1} \end{bmatrix} = M^{-1} R \begin{bmatrix} \hat{u}^n \\ \hat{v}^n \end{bmatrix}, \quad (25)$$

where

$$M = \begin{bmatrix} 1 + 2\alpha \sin^2(\frac{\theta}{2}) & -\beta \\ 2\gamma \sin^2(\frac{\theta}{2}) & 1 + 2\alpha \sin^2(\frac{\theta}{2}) \end{bmatrix},$$

and

$$R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

with  $\alpha = \frac{\Delta t}{\Delta x^2}$ ,  $\beta = c\Delta t$ , and  $\gamma = \frac{c\Delta t}{\Delta x^2}$ .

Here  $M^{-1}R$  is the amplification matrix at each time-step. In order for the numerical algorithm to be stable, the modulus of the eigenvalues of  $M^{-1}R$  must be less than or equal to unity for all possible values of  $\theta$ . For the  $2 \times 2$  matrix, it is easy to see that it has two conjugate complex eigenvalues. If the modulus of the two eigenvalues is less than or equal to 1, then the method is unconditionally stable.

The eigenvalues of  $M^{-1}R$  can be calculated as

$$\omega = \frac{1 + 2\alpha \sin^2(\frac{\theta}{2}) \pm i(c\sqrt{2\alpha\Delta t} \sin(\frac{\theta}{2}))}{1 + 4\alpha \sin^2(\frac{\theta}{2}) + 4\alpha^2 \sin^4(\frac{\theta}{2}) + 2c^2\Delta t\alpha \sin^2(\frac{\theta}{2})},$$

thus we have

$$|\omega|^2 = \frac{1}{1 + 4\alpha \sin^2(\frac{\theta}{2}) + 4\alpha^2 \sin^4(\frac{\theta}{2}) + 2c^2\Delta t\alpha \sin^2(\frac{\theta}{2})}.$$

Obviously,  $|\omega|^2 \leq 1$  since  $0 \leq \sin^2(\frac{\theta}{2}) \leq 1$  and  $\alpha > 0$ .

#### 4. Computational complexity

In this section an estimation of computational cost of the new method is obtained, and computation times for three methods (upwind, standard central finite difference or standard cfd, and fourth-order new method) are compared using the following model equation

$$u_x - u_{xx} = 0, \quad u(0) = 0, \quad u(1) = 1, \quad (26)$$

with exact solution  $u(x) = \frac{e^x - 1}{e - 1}$ .

If the above model equation is discretized over  $n$  subintervals with equal size, and the numerical solution at the  $i$ -th grid point is denoted as  $u_i$ , the solutions can be obtained by solving the linear system  $AU = b$ . For both the upwind and standard central finite difference schemes,  $A$  is an  $n \times n$  tridiagonal matrix and  $U = (u_1, u_2, \dots, u_n)$ . For the fourth-order new method the matrix  $A$  is a  $2n \times 2n$  banded matrix with a bandwidth of 7, and  $U = (u_1, v_1, u_2, v_2, \dots, u_n, v_n)$  contains the numerical solutions for both  $u$  and  $v$ , with  $v = u_x$ .

If an efficient algorithm is used to solve the tridiagonal linear system resulted from upwind and standard central finite difference schemes, a total of  $2n - 1$  divisions,  $3(n - 1)$  multiplications, and  $3(n - 1)$  subtractions are needed. The computational complexity is  $2(n - 1) + 1 + 3(n - 1) + 3(n - 1) = 8n - 7 = 8n + \mathbf{O}(1)$  flops.

If the Gaussian elimination algorithm is used to solve the linear system resulted from the fourth-order new method, a total of  $18n - 15$  divisions,  $24n - 22$  multiplications, and  $24n - 22$  subtractions are needed. The computational complexity is  $66n - 59 = 66n + \mathbf{O}(1)$  flops.

Table 1 clearly shows that to achieve the same accuracy the fourth-order new method is much faster than both upwind and the standard second-order central finite difference schemes. For instance, to obtain an error less than  $1.0E-05$ , the upwind scheme needs 68.054 seconds, and the standard second-order central finite difference scheme needs 0.0012 seconds, while the fourth-order new method needs only 0.0010 seconds to obtain an error of  $7.4E-07$ . For the



Upwind		Stand-cfd		4 <sup>th</sup> -order new	
Error	Time	Error	Time	Error	Time
9.7E-03	.0016	4.1E-04	.0011	7.4E-07	.0010
9.9E-04	.0098	9.8E-06	.0012	9.2E-09	.0012
1.0E-04	.5155	1.0E-06	.0016	1.8E-10	.0014
1.0E-05	68.05	1.0E-08	.0101	4.7E-12	.0016
1.0E-08	∞	1.0E-10	.4870	2.9E-13	.0022
1.0E-09	∞	1.5E-11	7.059	4.1E-14	.0035

Table 1. Computational cost vs accuracy for different algorithms

first-order upwind scheme, it is almost impossible to obtain an error less than  $1.0E - 08$  since a grid size of  $h = 1.0E - 08$  is needed.  
Note that the symbol  $\infty$  in Table 1 does not really represent infinity. It just represents an excessively large number. The computation time for each test case is the average of five runs. The unit for computation time is second.

5. Numerical results

Several numerical examples are presented here to compare the new method with the standard finite difference algorithms.  
**Example 1:** In this example, we compare the accuracy of four different algorithms using the same 3-point finite difference stencil. The following steady state convection-diffusion equation is used in the example:

$$- au_{xx} + cu_x = 0, \quad u(0) = 0, u(1) = 1,$$

(27)

where  $a$  and  $c$  are constants. The exact solution is given as

$$u(x) = \frac{e^{\frac{c}{a}x} - 1}{e^{\frac{c}{a}} - 1}.$$

(28)

$\Delta x$	Upwind	2 <sup>nd</sup> -stand	2 <sup>nd</sup> -new	4 <sup>th</sup> -new
1/10	1.32E-01	3.45E-02	1.41E-02	7.41E-04
1/20	7.64E-02	7.90E-03	3.70E-03	4.74E-05
1/40	4.17E-02	1.90E-03	9.52E-04	2.98E-06
1/80	2.18E-02	4.79E-04	2.39E-04	1.87E-07
1/160	1.12E-02	1.20E-04	5.98E-05	1.17E-08

Table 2. Errors between the exact solution and the numerical solution of Eq.(27) with  $a = 0.1$  and  $c = 1$ .

Table 2. shows the error  $\|u^e - u^c\|_\infty$  between the exact solution  $u^e$  and the numerical solution  $u^c$  calculated using the following four finite difference schemes with  $a = 0.1$  and  $c = 1$ :

1. Upwind: First-order upwind difference for the convection term and second-order central finite difference for the diffusion term.
2. 2<sup>nd</sup>-stand: Second-order central difference for both the diffusion and convection terms.
3. 2<sup>nd</sup>-new: New method that uses second-order central difference for Eqs. (5) and (7).



4.  $4^{th}$ -new: New method that uses fourth-order Padé approximation for Eqs. (5) and (7).

All results are obtained using a 3-point stencil. It is clear from Table 2 that the accuracies of different schemes are as expected: when the grid size  $\Delta x$  is reduced by  $\frac{1}{2}$ , the errors are reduced by approximately  $\frac{1}{2}$ ,  $(\frac{1}{2})^2$  and  $(\frac{1}{2})^4$  for the  $1^{st}$ -,  $2^{nd}$ - and  $4^{th}$ -order algorithms, respectively.

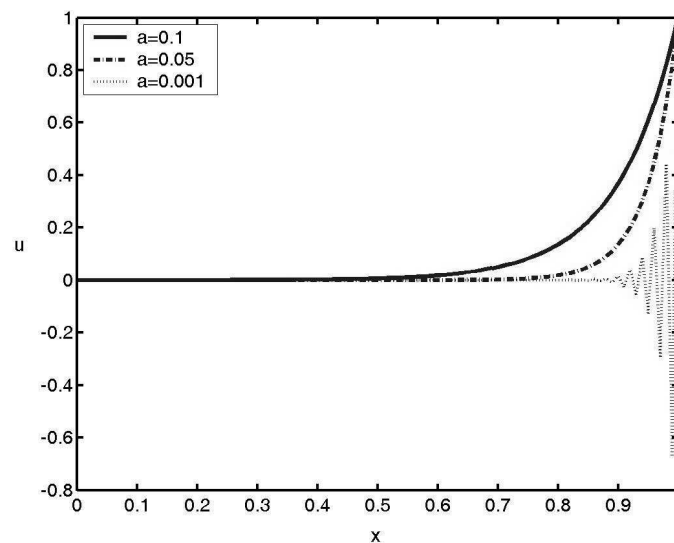


Fig. 1. Solution curves corresponding to  $a = 0.1, 0.05, 0.001$  and  $c = 1$  using the  $2^{nd}$ -stand algorithm with  $\Delta x = 0.001$ .

Although both the  $2^{nd}$ -stand and  $2^{nd}$ -new algorithms produced comparable results for the case of  $a = 0.1$  and  $c = 1$  shown in Table 2, the  $2^{nd}$ -new algorithm is much more robust when the boundary layer near  $x = 1$  becomes steeper as  $a$  decreases. Fig. 1 and Fig. 2 show the solutions calculated using the  $2^{nd}$ -stand and  $2^{nd}$ -new central difference schemes, respectively. The three curves in each figure correspond to the diffusion coefficient  $a = 0.1, 0.05$ , and  $0.001$ , respectively. It is clear from Fig. 1 that the solution calculated by the standard second-order central difference scheme becomes highly oscillatory when  $a$  reaches  $0.001$ , while the solution calculated by the new second-order central difference scheme shown in Fig. 2 describes the steep boundary layer near  $x = 1$  very well.

**Example 2:** In this example, we compare the robustness of three different algorithms. The governing equation for this example is

$$-au_{xx} + \left(x - \frac{1}{2}\right)u_x = 0, \quad u(0) = 0, \quad u(1) = 1, \quad (29)$$

where  $a$  is a constant. The convection coefficient changes sign in the middle of the domain, which makes this turning point problem difficult (Morton, 1996).

Fig. 3 shows the solution curves obtained by the new method using the second-order central difference algorithm. The four solution curves in the figure correspond to diffusion coefficient  $a = 1.0, 0.1, 0.01, 0.001$ , respectively, all calculated with  $\Delta x = 0.001$ . It is clear that as  $a$  decreases, the boundary layers at  $x = 0$  and  $x = 1$  become steeper. Also note that all four solution curves pass the same point  $x = 0.5$ , where the convection coefficient is zero.

Fig. 4 shows the solution curves obtained by the standard finite difference schemes on the same grid with  $\Delta x = 0.001$ . The diffusion term is discretized by the second-order

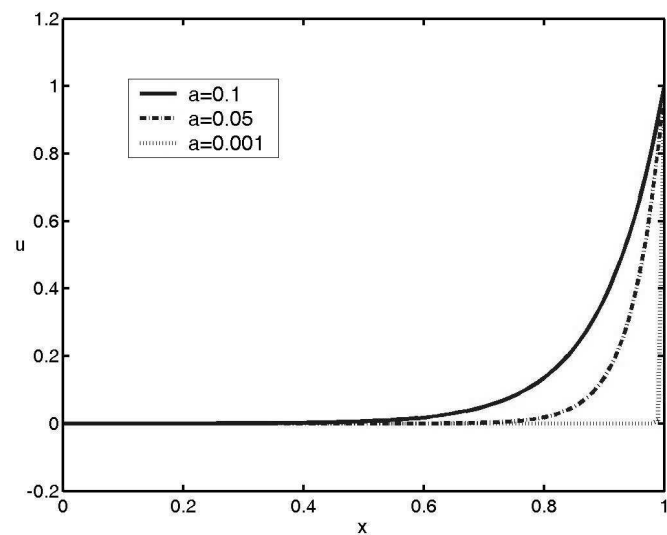


Fig. 2. Solution curves corresponding to  $a = 0.1, 0.05, 0.001$  and  $c = 1$  calculated using the 2<sup>nd</sup>–new algorithm with  $\Delta x = 0.01$ .

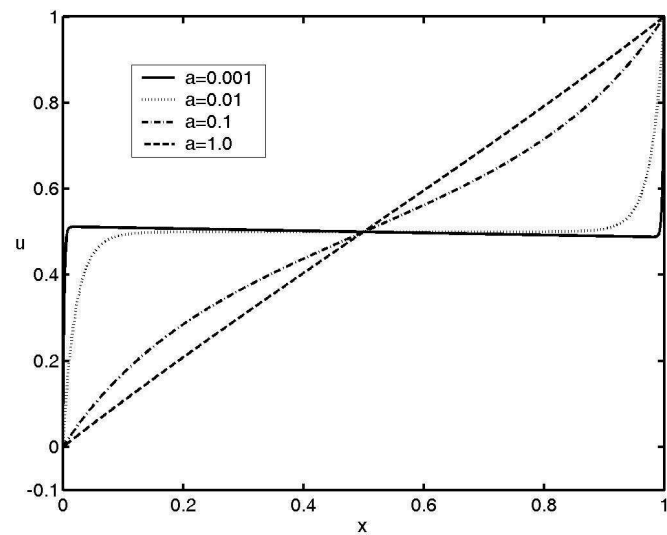


Fig. 3. Solution curves corresponding to  $a = 1.0, 0.1, 0.01$  and  $0.001$  calculated by the new method using central difference scheme

central difference scheme and the convection term is discretized by the first-order upwind scheme assuming the convection coefficient is positive. The change of sign of the convection coefficient is not taken into consideration. The solution curves correspond to  $a = 1.0$  and  $0.1$  look reasonable. But the solution curve correspond to  $a = 0.01$  is not accurate. It does not cross the solution curves corresponding to  $a = 1.0$  and  $0.1$  at  $x = 0.5$ . The solution curve corresponds to  $a = 0.001$  does not resemble the correct solution at all. This is not surprising since the convection coefficient changes sign but the upwind scheme used in the calculation does not take this into consideration.

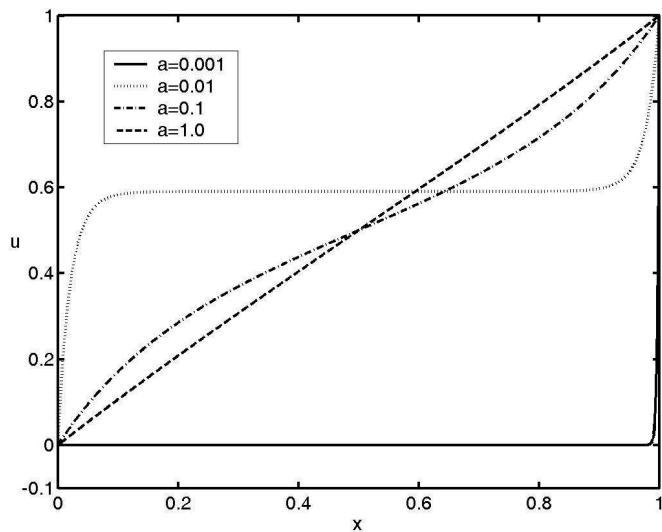


Fig. 4. Solution curves corresponding to  $a = 1.0, 0.1, 0.01$  and  $0.001$  calculated by the upwind scheme without considering the change of sign of the convection coefficient.

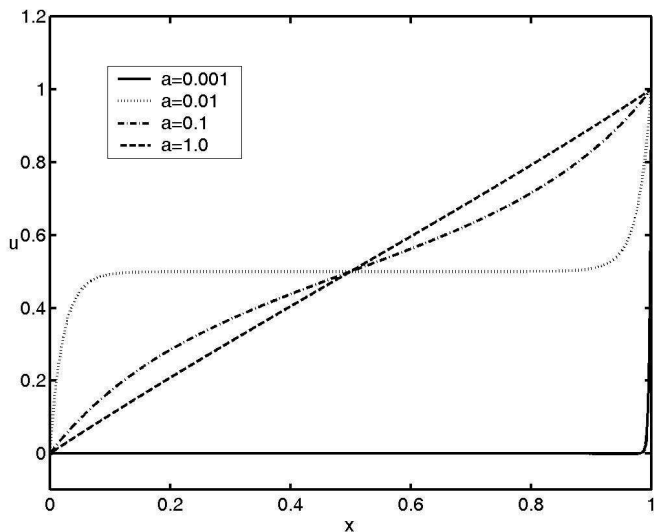


Fig. 5. Solution curves corresponding to  $a = 1.0, 0.1, 0.01$  and  $0.001$  calculated by the upwind scheme taking into consideration the sign change of the convection coefficient.

Fig. 5 shows the solution curves obtained by the standard finite difference algorithm. The diffusion term is discretized by the second-order central difference scheme and the convection term is discretized by the first-order upwind scheme taking into consideration the sign change of the convection coefficient. We can see the solution curves correspond to  $a = 1.0, 0.1$ , and  $0.01$  are very similar to those in Fig. 3. All three solution curves cross each other at the same point  $x = 0.5$ . However, the solution curve corresponding to  $a = 0.001$  does not resemble the true solution at all. This is somewhat surprising but probably can be attributed to the difficulty caused by the turning point at  $x = 0.5$  where  $c(x) = 0$ . The situation does not improve even when the grid spacing  $\Delta x$  is further reduced.

**Example 3:** In this example we solve a nonlinear Black-Scholes equation, which is widely used to model option price when transaction cost is considered. It is well known that in the area of financial mathematics, the calculations of both the solution and its first derivative are required. The derivative is the so-called hedge delta, which actually represents the number of shares (of stock) that one should hold or sale, in order to maximize the profit. The nonlinear Black-Scholes equation solved in this chapter is given below. More details about this model can be found in (Barles & Soner, 1998).

$$u_t = (1 + \Phi[e^{(Kt+x)}a^2E(u_{xx} + u_x)])(u_{xx} + u_x) - Ku_x,$$

with initial and boundary conditions

$$u(x, 0) = \max(1 - e^{-x}, 0), \quad u(-\infty, t) = 0, \quad u(\infty, t) = 1,$$

where  $a$  is transaction cost rate,  $E$  is the strike price,  $K = \frac{2\rho}{\sigma_0^2}$ ,  $\rho$  is the risk-free interest rate,  $\sigma_0$  is the volatility of the stock, and  $\Phi$  is a function defined as the solution to the following ODE:

$$\Phi'(s) = \frac{\Phi(s) + 1}{2\sqrt{s\Phi(s) - s}}, \quad \Phi(0) = 0, \quad (30)$$

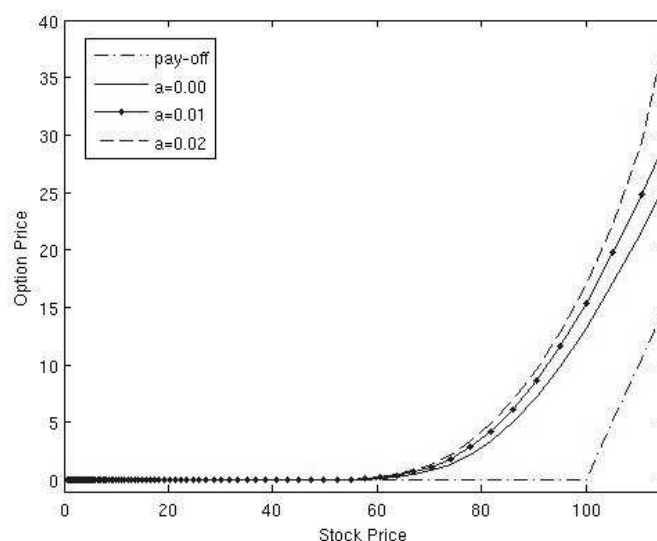


Fig. 6. Solution curves for option price corresponding to various transaction cost rates ( $a = 0.0, 0.01, 0.02$ ) and the pay-off curve.

In the numerical solution process, the infinite domain is approximated by a finite interval of sufficient length. In financial industry, one of the widely used methods for calculating hedge delta is to first solve the Black-Scholes equation to obtain numerical solution of  $u$ , and then apply finite difference schemes to the numerical solution to obtain approximations to  $u_x$ . With the new method discussed in this chapter, both the solution  $u$  and the hedge delta  $u_x$  are calculated simultaneously. Fig. 6 and Fig. 7 show the option price and the hedge delta for various transaction cost rates, respectively.

**Example 4:** In this example we solve a time dependent nonlinear equation, the Burgers' Equation:

$$u_t + uu_x = au_{xx}, \quad u(0, t) = 0, \quad u(1, t) = 1, \quad (31)$$

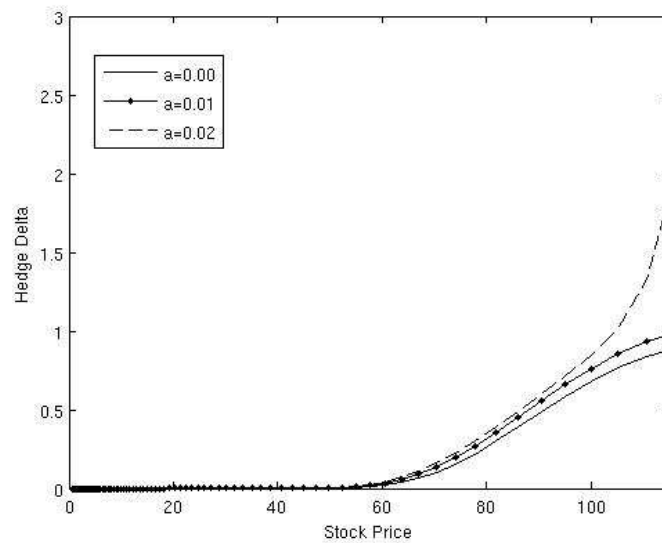


Fig. 7. Solution curves of the hedge delta  $\frac{\partial V}{\partial S}$  corresponding to various transaction cost rate ( $a = 0.0, 0.01, 0.02$ ).

where  $a$  is a constant. We first solve the equation with the initial condition  $u(x, 0) = x(1 - x)$ , for which the analytic solution is not available. Figs. 8, 9, and 10 show the solution curves for various  $a$  values obtained by the standard central difference scheme (for both the convection and diffusion terms), the upwind (for convection)-central difference (for diffusion) scheme, and the new method using second-order central difference, respectively. It is clear that as  $a$  decreases, for instance,  $a = 0.001$ , the standard central difference scheme produces oscillations in Fig. 8.

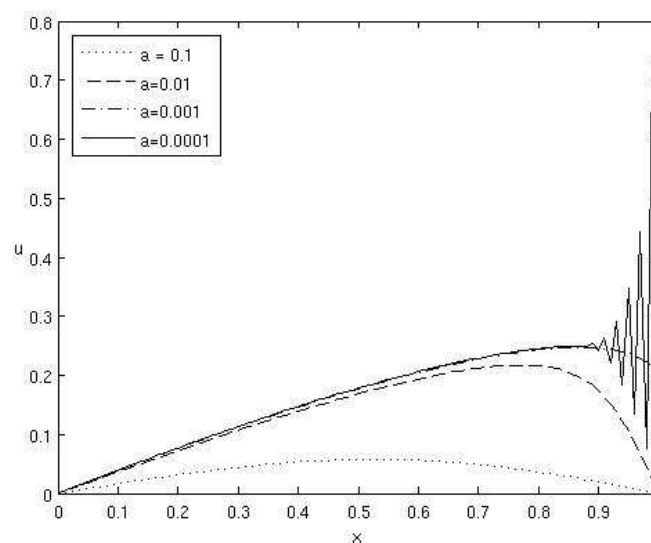


Fig. 8. Solution curves corresponding to  $a = 0.1, 0.01, 0.001, 0.0001$  using the standard central difference scheme with  $\Delta x = 0.01$ .

The new method, on the other hand, produces solutions that are oscillation free as shown in Fig. 9. These solutions are similar to those produced by the upwind-central difference scheme

shown in Fig. 10, and agree with the results reported in other works, such as (Hassanien et al., 2005).

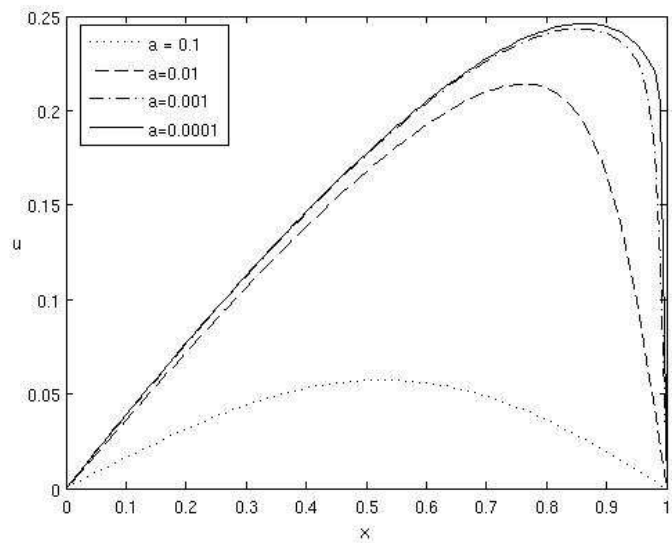


Fig. 9. Solution curves corresponding to  $a = 0.1, 0.01, 0.001, 0.0001$  using the upwind scheme with  $\Delta x = 0.01$ .

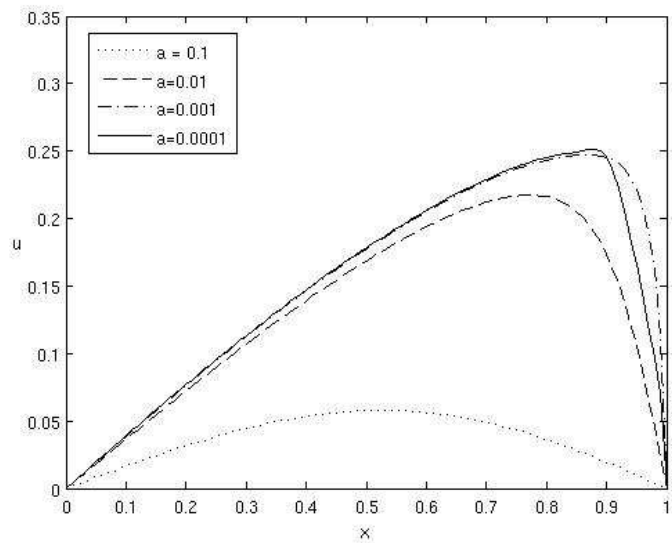


Fig. 10. Solution curves corresponding to  $a = 0.1, 0.01, 0.001, 0.0001$  using the second-order new algorithm with  $\Delta x = 0.01$ .

Next we use the initial condition

$$u(x, 0) = \frac{2a\pi \sin(\pi x)}{\kappa + \cos(\pi x)}, \tag{32}$$

with  $\kappa > 1$ , for which the exact solution to Eq. (31) is known as

$$u(x, t) = \frac{2a\pi e^{-\pi^2 at} \sin(\pi x)}{\kappa + e^{-\pi^2 at} \cos(\pi x)}. \tag{33}$$

We apply both the upwind-central scheme and the second-order new method to this example to compare the accuracy of the two algorithms. Table 3 shows the error  $\|u^e - u^c\|_\infty$  between the exact solution  $u^e$  and the numerical solution  $u^c$  calculated using the two algorithms with  $\kappa = 1.2$ , for  $a = 0.001$  and  $a = 0.0001$ . All results are obtained using a 3-point stencil. Since the main focus of the study is to compare spatial accuracy of the two algorithms, the first-order explicit time integration is used for simplicity with  $\Delta t = 0.0001$  to ensure stability. It is clear from Table 3 that the accuracies of the two schemes are as expected: When the grid size  $\Delta x$  is reduced by  $\frac{1}{2}$ , the errors are reduced by approximately  $\frac{1}{2}$  and  $(\frac{1}{2})^2$  for the upwind-central and the 2<sup>nd</sup>-new schemes, respectively.

$\Delta x$	<i>upwind</i> $a = 0.001$	2 <sup>nd</sup> -new $a = 0.001$	<i>upwind</i> $a = 0.0001$	2 <sup>nd</sup> -new $a = 0.0001$
1/20	2.69E-04	1.48E-04	3.47E-06	2.31E-06
1/40	1.67E-04	3.87E-05	2.20E-06	6.11E-07
1/80	9.40E-05	9.79E-06	1.23E-06	1.67E-07
1/160	4.98E-05	2.45E-06	6.46E-07	4.13E-08

Table 3. Errors between the exact solution and the numerical solution of Eq. (31) with  $a = 0.001$  and  $a = 0.0001$ .

The numerical examples presented in this chapter show that the standard central difference scheme is second-order accurate on a 3-point stencil but produces oscillatory solutions for convection dominated problems. The upwind scheme is more robust but is only first-order accurate on a 3-point stencil. The new method discussed in this chapter appears to combine the advantages of accuracy of the standard central difference algorithm with the robustness of the upwind scheme for convection dominated equations.

6. Conclusions

The method discussed in this chapter eliminates the convection term in Eq. (1) and makes it feasible to use central difference schemes to solve convection-diffusion equations accurately. The new method, combined with the central difference schemes, can achieve better accuracy than the upwind schemes on the same finite difference stencil, and is shown in the examples presented here to be as robust as the upwind schemes for convection dominated problems. It can also be easily combined with the Pad  approximation to achieve fourth-order accuracy in space on a 3-point finite difference stencil. The new method does incur a modest increase in computational complexity. Instead of just solving Eq. (1), the new method requires solving Eqs. (5) and (7). With a 3-point stencil, the standard upwind-central difference schemes will generate a system of tri-diagonal algebraic equations, while the new method discussed in this chapter will lead to a system of block tri-diagonal algebraic equations with  $2 \times 2$  blocks. This increased complexity, however, can be compensated by the use of fewer grid points with the increased order of accuracy of the new method. Furthermore, for problems that require calculations of both the solutions and their derivatives, the new method eliminates the need to calculate the derivatives after solving Eq. (1). The discussions of the new method in this chapter are based on one-dimensional problems. For higher-dimensional problems, a straightforward application of this method will lead to systems of four equations for two-dimensional problems and seven equations for three-dimensional problems. For better computational efficiency, operator splitting



(Gustafsson et al., 1995) should be used to first decompose the original equation into a series of one-dimensional problems. The new method discussed in this chapter can then be applied to these one-dimensional problems to calculate numerical solutions efficiently. Details will be presented in future papers.

## 7. References

- Barles, G. & Soner, H. (1998). Option pricing with transaction costs and a nonlinear Black-Scholes equation, *Finance Stochastic*, Vol. 2, 1998, pp. 369 - 397.
- Gu, Y.; Liao, W. & Zhu, J. (2003). An efficient high order algorithm for solving systems of 3-D reaction-diffusion equations, *Journal of Computational and Applied Mathematics*, Vol. 155, 2003, pp. 1 - 17.
- Gupta, M.; Manohar, R. & Stephenson, J. (1984). A single cell high-order scheme for the convection-diffusion equation with variable coefficients, *International Journal for Numerical Methods in Fluids*, Vol. 4, 1984, pp. 641 - 651.
- Gustafsson, B.; Kreiss, H. & Olinger, J. (1995). *Time-Dependent problems and difference methods*, John Wiley & Sons, New York.
- Hassanien, I; Salaman, A. & Hosham, H. (2005). Fourth-Order finite difference method for solving Burgers' equation, *Applied Mathematics and Computation*, Vol. 170, 2005, pp. 781 - 800.
- Hundsdoerfer, W. & Verwer, J. (2003). *Numerical solution of time-dependent advection-diffusion-reaction equations*, Springer, Berlin.
- Kalita, J.; Dalal, D. & Dass, A. (2002). A class of higher order compact schemes for the unsteady two-dimensional convection-diffusion equation with variable convection coefficients, *International Journal for Numerical Methods in Fluids*, Vol. 38, 2002, pp. 1111 - 1131.
- Karaa, S. & Zhang, J. (2004). High order method for solving unsteady convection-diffusion problems, *Journal of Computational Physics*, Vol. 198, no.1, 2004, pp. 1 - 9.
- Morton, K. (1996). *Numerical solution of convection-diffusion problems*, Chapman & Hall, London.
- Noye, B. & Tan, H. (1989). Finite difference methods for solving the two-dimensional advection-diffusion equation, *International Journal for Numerical Methods in Fluids*, Vol. 9, 1989, pp. 75 - 98.
- Rigal, A. (1989). Numerical analysis of two-level finite difference schemes for unsteady diffusion-convection problems, *International Journal for Numerical Methods in Engineering*, Vol. 28, 1989, pp. 1001 - 1021.
- Rigal, A. (1990). Numerical analysis of three-time-level finite difference schemes for unsteady diffusion-convection problems, *International Journal for Numerical Methods in Engineering*, Vol. 30, 1990, pp. 307 - 330.
- Rigal, A. (1994). High order difference schemes for unsteady one-dimensional diffusion-convection problems, *Journal of Computational Physics*, Vol. 114, 1994, pp. 59 - 76.
- Seydel, R. (2002). *Tools for computational finance*, Springer, Berlin.
- Spotz, W. & Carey, G. (2001). Extension of high-order compact schemes to time-dependent problems, *Numerical Methods for Partial Differential Equations*, Vol. 17, 2001, pp. 657 - 672.

- Tian, Z. & Dai, S. (2007). High-order compact exponential finite difference methods for convection-diffusion type problems, *Journal of Computational Physics*, Vol. 220, No.2, 2007, pp. 952 - 974.
- Tian, Z. & Ge, Y. (2007). A fourth-order compact ADI method for solving two-dimensional unsteady convection-diffusion problems, *Journal of Computational and Applied Mathematics*, vol. 198, 2007, pp. 268-286.
- Wood, W. (2006). An exact solution for Burgers' equation, *Communications in Numerical Methods in Engineering*, Vol. 22, 2006, pp. 797 - 798.
- You, D. (2006). A high-order Padé ADI method for unsteady convection-diffusion equations, *Journal of Computational Physics*, Vol. 214, No. 1, 2006, pp. 1-11.
- Zhang, J.; Sun, H. & Zhao, J. (2002). High order compact scheme with multigrid local mesh refinement procedure for convection diffusion problems, *Computer Methods in Applied Mechanics and Engineering*, Vol. 191, 2002, pp. 4661 - 4674.

IntechOpen



## Computational Simulations and Applications

Edited by Dr. Jianping Zhu

ISBN 978-953-307-430-6

Hard cover, 560 pages

**Publisher** InTech

**Published online** 26, October, 2011

**Published in print edition** October, 2011

The purpose of this book is to introduce researchers and graduate students to a broad range of applications of computational simulations, with a particular emphasis on those involving computational fluid dynamics (CFD) simulations. The book is divided into three parts: Part I covers some basic research topics and development in numerical algorithms for CFD simulations, including Reynolds stress transport modeling, central difference schemes for convection-diffusion equations, and flow simulations involving simple geometries such as a flat plate or a vertical channel. Part II covers a variety of important applications in which CFD simulations play a crucial role, including combustion process and automobile engine design, fluid heat exchange, airborne contaminant dispersion over buildings and atmospheric flow around a re-entry capsule, gas-solid two phase flow in long pipes, free surface flow around a ship hull, and hydrodynamic analysis of electrochemical cells. Part III covers applications of non-CFD based computational simulations, including atmospheric optical communications, climate system simulations, porous media flow, combustion, solidification, and sound field simulations for optimal acoustic effects.

### How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Wenyuan Liao and Jianping Zhu (2011). A Fourth-Order Compact Finite Difference Scheme for Solving Unsteady Convection-Diffusion Equations, Computational Simulations and Applications, Dr. Jianping Zhu (Ed.), ISBN: 978-953-307-430-6, InTech, Available from: <http://www.intechopen.com/books/computational-simulations-and-applications/a-fourth-order-compact-finite-difference-scheme-for-solving-unsteady-convection-diffusion-equations>



### InTech Europe

University Campus STeP Ri  
Slavka Krautzeka 83/A  
51000 Rijeka, Croatia  
Phone: +385 (51) 770 447  
Fax: +385 (51) 686 166  
[www.intechopen.com](http://www.intechopen.com)

### InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai  
No.65, Yan An Road (West), Shanghai, 200040, China  
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元  
Phone: +86-21-62489820  
Fax: +86-21-62489821

© 2011 The Author(s). Licensee IntechOpen. This is an open access article distributed under the terms of the [Creative Commons Attribution 3.0 License](https://creativecommons.org/licenses/by/3.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

IntechOpen

IntechOpen