Finite Difference Methods for Advection and Diffusion

Alice von Trojan, B.Sc.(Maths.Sc.)(Hons.)

Thesis submitted for the degree of
Doctor of Philosophy

Department of Applied Mathematics
University of Adelaide

April 2001

Finite Difference Methods for Advection and Diffusion

Alice von Trojan

Signed Statement

This work contains no material which has been accepted for the award of any other degree or diploma in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text. I consent to this copy of my thesis, when deposited in the University Library, being available for loan and photocopying.

SIGNED: .... DATE: .....22.10.2001.....

Acknowledgements

Special thanks go to my supervisors Associate Professor John Noye and Dr. Andrew Gill, for their interest and support. I would also like to thank the Head of the Applied Maths Department Dr. Peter Gill. Many thanks to Dr. habil. Rainer Callies, my boss at the Munich University of Technology. I would like to acknowledge the financial support I received through an Australian Postgraduate Award. Finally, I would like to thank my family.

Contents

List of Abbreviations

ADE advection-diffusion equation

BS backward-space

CPU central processing unit

CROW Crowley's advection scheme

CS centered-space

DPDE discretized partial differential equation

EPDE equivalent partial differential equation

FCT flux corrected transport

FDA finite difference approximation

FDE finite difference equation

FDM finite difference method

FEM finite element method

FS forward-space

FTCS forward-space centered-space

FVM finite volume method

IMP implicit algorithm for conservative advection

LOD locally one-dimensional

LTH Leith's base method for advection

LW2 second-order Lax-Wendroff base method

LW4 fourth-order Lax-Wendroff modified method

M4 Mitchell's fourth-order method

MDPDE modified discretized partial differential equation

MEPDE modified equivalent partial differential equation

mod_FL modification based on UW differencing where 'FL' is first letter of LTH, UW15, RUS or OPT

mod2_FL as above except modification based on CS differencing

NAT Noye and Tan method

NH2 second-order Noye-Hayman base method

NH4 fourth-order Noye-Hayman modified method

N131 three-level base method for diffusion

N151 three-level modified method for diffusion

ODE ordinary differential equation

OPT optimal base method for advection

OPT2 second-order optimal two-weight base method for transport

OPT4 fourth-order optimal two-weight modified method for transport

PDE partial differential equation

PS1 Process splitting algorithm – mod_L method for advection and FTCS method for diffusion

PS2 Process splitting algorithm – mod2_R method for advection and NH4 method for diffusion

PS3 Process splitting algorithm – mod2_O method for advection and M4 method for diffusion

RK4 fourth-order Runge-Kutta method

RMS root mean square

RUS Rusanov base method for advection

RUS2 generalized Rusanov method for conservative advection

UW upwind

UW15 upwind base method with (1,5) computational stencil for advection

Abstract

Transport phenomena are governed by the processes of advection and diffusion. This work concerns the development of high-order finite-difference methods on a uniform rectangular grid for advection and diffusion problems with smooth variable coefficients. The initial and boundary conditions are assumed to be given with sufficient smoothness to maintain the order of convergence of the scheme under consideration. High-order finite-difference methods for constant coefficients usually degenerate to first or, at best, second-order when applied to variable-coefficient problems. A technique is developed whereby the convergence rate can be increased to the constant-coefficient rate. This modification procedure is applied to finite-difference methods for both the non-conservative and conservative forms of the variable-coefficient advection equation, and to the variable-coefficient diffusion equation. Since the conservative form of the advection equation may be considered as an equation in which the two processes of non-conservative advection and decay (or growth) are taking place simultaneously, it is shown that the conservative process may be split into two separate processes, thereby simplifying the solution procedure. It is also observed that the decay may be identified as a sink term, so methods developed for the solution of the conservative equation may also be applied to the non-conservative advection equation in the presence of sinks. Likewise, the variable-coefficient diffusion equation is noted to be a special case of the variable-coefficient transport equation, so high-order methods developed to solve the former equation may also be applicable to the latter equation. Finite-difference methods can readily be extended to problems involving two or more dimensions using locally one-dimensional techniques. This is demonstrated by application to two-dimensions for the non-conservative advection equation, and to a special case of the diffusion equation. The new modified methods are particularly apt for problems involving smooth initial and boundary data, where they outperform the base methods considerably.

Chapter 1

Introduction

The advection-diffusion equation (ADE), which is commonly referred to as the transport equation, governs the way in which contaminants are transferred in a fluid due to the processes of advection and diffusion. Mass, momentum and heat transfer are all described by transport equations. Hydrologists use the ADE to model solute transport in groundwater. It has been used to describe bacterial transport in porous media (Hornberger et al. 1992) and solute transport through large uniform and layered soil columns (Porro et al. 1993). The two-dimensional ADE has been used by Portela et al. (1992) to model the fate of a pollutant released into the Tejo estuary in Portugal, and by Noye et al. (1992) to predict prawn larvae movement in coastal seas.

In many fluid flow applications, advection dominates diffusion. Meteorologists rely on accurate numerical approximations of the advection equation for weather forecasting (Staniforth and Côté 1991). In optically thin media, the time-dependent radiative transfer equation reduces to the advection equation (Stone and Mihalas 1992). The advection equation also governs acoustic or elastic wave propagation (Leveque 1997), gas discharge problems in physics (Morrow 1981, Steinle et al. 1989), and the motion of shallow free-surface flows (Hubbard and Baines 1997).

Diffusion is the governing process in problems involving flow through porous media, and conduction of heat in solids. The diffusion equation has been used to model heat flow in a thermal print head (Morris 1970), heat conduction in a thin insulated rod (Noye 1984a), and the dispersion of soluble matter in solvent flow through a tube (Taylor 1953). Equations similar to the diffusion equation have also been used to calculate heat transfer, radiation transfer and hydrostatical equilibrium in stellar evolution programs (Livne and Glasner 1986).

There are, however, few applications for which analytical solutions to the transport equation exist. Barry and Sposito (1989) discuss an analytical solution to the transport equation with time-dependent coefficients. Yates (1992) reports the development of an analytical solution to the ADE with a constant advection velocity and an exponential diffusion function. An analytical solution has been developed by Basha and El-Habel (1993) to the one-dimensional transport equation with constant advection and a

time-dependent diffusion coefficient. Zoppou and Knight (1997a, 1997b) present analytical solutions of advection and advection-diffusion equations with spatially variable coefficients. An asymptotic solution for two-dimensional flow in an estuary, where the velocity is time-varying and the diffusion coefficient varies proportionally to the flow speed, has been found by Kay (1997). Because of the limited number of analytical solutions for real-life problems, numerical techniques are usually used to approximate solutions to transport problems. The most common techniques used to solve the ADE are based on finite-difference methods (FDMs), finite-element methods (FEMs) or finite-volume methods (FVMs). Numerical approximations of the ADE generally involve the simultaneous solution of a hyperbolic operator describing the advection process and a parabolic operator describing the diffusion process (Noye 1987a).

Fluid flow can be approached from either an Eulerian or a Lagrangian viewpoint. Eulerian methods are based on the solution of the transport equation in terms of a fixed grid coordinate system (Noye 1987a). Problems associated with the Eulerian approach, which are mainly due to the hyperbolic operator, include numerical diffusion, spurious oscillation, wave-speed errors and peak-clipping (Yeh et al. 1992). Numerical diffusion, prevalent in first-order schemes (for example, the first-order upwind scheme) gives the appearance of an artificial increase in diffusion. Spurious oscillation, common in central-difference schemes, results in the development of overshoots and undershoots around the concentration front (Healy and Russell 1993). Schemes resolving the problem of numerical diffusion may thus be seen to introduce spurious oscillation, while methods which reduce oscillations may give rise to excessively damped solutions (Yeh et al. 1992). Lagrangian methods solve the transport equation by using a moving, deformable coordinate system which follows the fluid flow, thereby avoiding the explicit treatment of the advection term (Noye 1987a), leaving only the problem of solving the diffusion equation on a new variable grid each time-step. Although Lagrangian methods offer what appears to be a more natural approach to fluid flow, they have some serious drawbacks. Large grid deformations may occur in regions of rapidly varying velocity, resulting in a subsequent loss of accuracy (Vreugdenhil and Koren 1993). Additionally, where concentration fronts cross each other, such as in the presence of multiple sources, Lagrangian techniques cannot be used (Noye 1987a).

The above-mentioned problems associated with the Eulerian and Lagrangian approaches, which are mainly due to the first-order advection term, have led to the development of mixed Eulerian-Lagrangian methods for solving advection-dominated transport problems. This approach involves solving the advection and diffusion components of the transport equation separately. A fixed Eulerian grid is adopted for the solution of the diffusion process, while advection is handled by a Lagrangian formulation. Yet, these methods can also suffer from a number of problems, including an inability to treat boundary fluxes when characteristics intersect inflow or outflow boundaries, an inability to ensure conservation of mass, and the introduction of numerical diffusion due to low-order interpolation or integration (Healy and Russell 1993). If higher-order interpolation is used, numerical diffusion may be improved but at the expense of the reintroduction of spurious oscillation. Zhang et al. (1993) comment on the practical difficulties of implementing these techniques. They state that these schemes are "quite troublesome to implement and time consuming to execute because of the need to continuously track the concentration front using numerous particles at each time step." They also find criticism with Yeh's (1990) approach

in which an Eulerian-Lagrangian method with a zoomable (or adjustable) fine hidden mesh is implemented: "While this scheme reduces or virtually eliminates numerical dispersion and oscillations, the process of zooming and refining the elements at each time step is not straightforward in terms of its practical implementation. Also, the large number of elements needed for this approach requires excessive amounts of computer memory and execution time." Zhang et al. (1993) have developed what they describe as an efficient Eulerian-Lagrangian method for solving solute transport problems in steady and transient flow fields. Their work has some promising features: it eliminates spurious oscillation, reduces numerical diffusion, and reduces computational times compared to FEMs. Their method is appropriate for use with advection-dominated transport problems, but is limited to one-dimensional problems with values of the Courant number $c \leq 1$.

In this thesis, FDMs will be developed on a uniform Eulerian grid for advection and diffusion problems with smooth variable coefficients. In most applications, it is necessary to consider the fact that the fluid velocity is not constant. For example, the velocity of a river of varying cross-section is not constant, and currents due to tides fluctuate. Field and experimental studies have suggested that the diffusion coefficient in many applications is not constant but increases as a function of travel time, or equivalently, with the solute displacement distance (Basha and El-Habel 1993). However, it is not always straightforward to find appropriate representations to describe the variable velocity and diffusion fields. The choice may be restricted by the following factors. First, the lack of available experimental data for most applications due to the high cost of sampling and data analysis, as well as the practical difficulties associated with knowing the number and placement of samples to achieve a certain level of confidence in describing the physical process (Yates 1992). Second, testing the accuracy of numerical techniques generally requires an exact analytical solution. These are usually only available for very simple, highly idealised test cases.

Finite-difference methods are preferred to FEMs or FVMs in this work, because they are easier to formulate, usually require less memory capacity and computer time, and allow easier preparation of input of data (Noye 1987a). A particularly attractive feature of FDMs is that they readily lend themselves to the technique of process splitting (D'Yakonov 1963, Marchuk 1975). In this approach, the governing equation, for example, the transport equation, is divided into the separate processes of advection and diffusion. The individual processes are solved sequentially each time-step using a FDM, giving a solution to the full equation at the end of one time-step. This idea can be extended to multi-dimensional equations, using locally one-dimensional (LOD) techniques (D'Yakonov 1963, Marchuk 1975). Complicated multi-dimensional problems are split into a number of simpler one-dimensional equations, each of which can be solved separately, to give an approximation to the complete equation.

In Chapter 2, a description of the finite-difference method is given. The concepts of convergence, consistency and stability are summarized, and their connection through Lax's equivalence theorem is described. An illustration of the discretization of the one-dimensional constant-coefficient transport equation is provided, and the notation used in this work is introduced. Chapter 2 concludes with a brief discussion of some additional properties, such as conservation and non-negativity, that numerical

schemes should respect. The general one-dimensional variable-coefficient transport equation is given by

$$\frac{\partial \hat{\tau}}{\partial t} + \frac{\partial (u\hat{\tau})}{\partial x} - \frac{\partial}{\partial x}\left(\alpha \frac{\partial \hat{\tau}}{\partial x}\right) = 0, \tag{1.1}$$

in which $\hat{\tau} = \hat{\tau}(x,t)$ represents, for example, temperature, vorticity, turbulent kinetic energy or concentration of a passive solute at position x and time t in a fluid moving with variable velocity $u = u(x,t)$, subject to diffusion governed by the positive coefficient $\alpha = \alpha(x,t)$. In Chapter 3, the pure variable-coefficient advection equation written in the non-conservative form

$$\frac{\partial \hat{\tau}}{\partial t} + u(x,t)\frac{\partial \hat{\tau}}{\partial x} = 0, \tag{1.2}$$

is investigated. High-order FDMs used to approximate (1.2) for the constant velocity case generally revert to low-order for variable velocities. A technique is developed whereby the convergence rate of any FDM can be improved to its constant-coefficient order for application to variable-coefficient problems. Numerical tests in Chapter 4 illustrate the improved convergence of the new modified methods and show that they are often significantly more accurate and efficient than the original methods. In two-dimensions, the non-conservative form of the advection equation is given by

$$\frac{\partial \hat{\tau}}{\partial t} + u(x,y,t)\frac{\partial \hat{\tau}}{\partial x} + v(x,y,t)\frac{\partial \hat{\tau}}{\partial y} = 0, \tag{1.3}$$

where $\hat{\tau} = \hat{\tau}(x,y,t)$, and where u and v are the variable velocities in the x and y directions respectively. The methods developed in Chapter 3 are used in Chapter 5, in a LOD fashion, yielding highly accurate numerical solutions to (1.3). This is tested for the special case $u = u(x,t)$ and $v = v(y,t)$, for which an analytical solution is known. The most general conservative form of the one-dimensional advection equation, namely

$$\frac{\partial \hat{\tau}}{\partial t} + \frac{\partial (u\hat{\tau})}{\partial x} = 0, \tag{1.4}$$

which may also be written as

$$\frac{\partial \hat{\tau}}{\partial t} + u\frac{\partial \hat{\tau}}{\partial x} + \frac{\partial u}{\partial x}\hat{\tau} = 0, \tag{1.5}$$

is studied in Chapter 6. Various conventional techniques based on the discretization of (1.4) will be compared to several new methods based on splitting the advection and decay processes, which are seen to be taking place simultaneously in (1.5). The technique developed in Chapter 3 is then used in Chapter 7 to modify FDMs for a special case of the diffusion equation given by

$$\frac{\partial \hat{\tau}}{\partial t} - \alpha(x,t)\frac{\partial^2 \hat{\tau}}{\partial x^2} = 0. \tag{1.6}$$

Once these methods have been validated by numerical tests against an exact solution, they are used in Chapter 8 to approximate solutions to the two-dimensional version of (1.6). Furthermore, it is demonstrated in Chapter 9 that these methods can be used in conjunction with the methods developed in Chapter 3 to approximate solutions to the general variable-coefficient diffusion equation, namely

$$\frac{\partial \hat{\tau}}{\partial t} - \frac{\partial}{\partial x}\left(\alpha \frac{\partial \hat{\tau}}{\partial x}\right) = 0. \tag{1.7}$$

Expanding (1.7) gives

$$\frac{\partial \hat{\tau}}{\partial t} + u(x,t)\frac{\partial \hat{\tau}}{\partial x} - \alpha(x,t)\frac{\partial^2 \hat{\tau}}{\partial x^2} = 0, \tag{1.8}$$

in which the term $u = -\partial\alpha/\partial x$ may be regarded as a variable velocity. Thus (1.7) can be approximated by splitting (1.8) into the separate processes of advection and diffusion. Additionally, it will be shown how FDMs for the constant-coefficient transport equation can be modified to produce high-order solutions for the variable-coefficient case given by (1.8).

Analytical solutions to the variable-coefficient advection and diffusion equations are derived, and are used to assess the accuracy of the schemes. The methods are also compared in terms of their computational efficiency. For most schemes, ways to overcome difficulties near the boundaries are described, however, since a widely applicable scheme is sought, the initial and boundary conditions are usually given by the analytical solution. In this way, attention can be focussed on producing high-order methods for use away from the effects of the boundaries. Because the existence of a smooth solution is the underlying assumption of high-order methods, the test problems involve only smooth variable quantities.

Chapter 2

General Theory

The partial differential equation (PDE) governing the transport of a pollutant in a fluid, in the absence of sources and sinks, written in conservative form, is given by

$$\frac{\partial \hat{\tau}}{\partial t} + \frac{\partial}{\partial x}(u\hat{\tau}) + \frac{\partial}{\partial y}(v\hat{\tau}) + \frac{\partial}{\partial z}(w\hat{\tau}) = \frac{\partial}{\partial x}\left(\alpha_x \frac{\partial \hat{\tau}}{\partial x}\right) + \frac{\partial}{\partial y}\left(\alpha_y \frac{\partial \hat{\tau}}{\partial y}\right) + \frac{\partial}{\partial z}\left(\alpha_z \frac{\partial \hat{\tau}}{\partial z}\right), \tag{2.1}$$

in which

$\hat{\tau} = \hat{\tau}(\underline{x}, t)$ is the unknown contaminant concentration at position $\underline{x} = (x, y, z)$ and time t,
$\underline{u} = (u(\underline{x}, t), v(\underline{x}, t), w(\underline{x}, t))$ is the flow velocity vector,
$\underline{\alpha} = (\alpha_x(\underline{x}, t), \alpha_y(\underline{x}, t), \alpha_z(\underline{x}, t))$ is the diffusion coefficient vector,

and x, y and z are coordinates relative to a set of Cartesian axes.

Expanding (2.1) yields

$$\begin{aligned} &\frac{\partial \hat{\tau}}{\partial t} + \left(u - \frac{\partial \alpha_x}{\partial x}\right)\frac{\partial \hat{\tau}}{\partial x} + \left(v - \frac{\partial \alpha_y}{\partial y}\right)\frac{\partial \hat{\tau}}{\partial y} + \left(w - \frac{\partial \alpha_z}{\partial z}\right)\frac{\partial \hat{\tau}}{\partial z} + \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}\right)\hat{\tau} \\ &\qquad = \alpha_x \frac{\partial^2 \hat{\tau}}{\partial x^2} + \alpha_y \frac{\partial^2 \hat{\tau}}{\partial y^2} + \alpha_z \frac{\partial^2 \hat{\tau}}{\partial z^2}. \end{aligned} \tag{2.2}$$

In the limit as $\underline{\alpha} = 0$, (2.2) is known as the pure advection equation, and when $\underline{u} = 0$ it is known as the pure diffusion equation. If the fluid is incompressible then $\underline{\nabla}.\underline{u} = 0$, and the decay term vanishes.

The transformation of (2.2) to non-dimensional form by appropriate normalizations allows the relative importance of the various terms to be estimated. The results of numerical tests conducted on such models are then scaled to real flow conditions (Ferziger and Perić 1996).

After normalization, (2.2) appears in the same form, but it is now solved over the spatial domain $[0, 1]$ in each dimension, and the value of $\hat{\tau}$ and the coefficients of the equation are dimensionless.

Examples of the non-dimensionalization procedure for the three-dimensional transport equation are given in Noye (1984a) and Vreugdenhil (1993). All PDEs considered in this work will be assumed to have been written in non-dimensional form.

Analytical techniques such as separation of variables, integral transforms and the method of characteristics, can produce closed form expressions for the solutions of PDEs, giving the behaviour of the dependent variables continuously throughout the solution domain.

Although a large number of analytical solutions are known for general initial and boundary conditions to constant coefficient problems with simple geometries, due to the complexity of most problems, PDEs are usually solved numerically (Zoppou and Knight 1997b). Numerical techniques such as FDMs approximate the solution at specific discrete points, called grid points, in the computational domain.

Domain Discretization

Considerations will be restricted to problems involving at most two spatial dimensions. The solution domain will be discretized using a uniform rectangular grid with grid spacings of $\Delta x = 1/J$ and $\Delta y = 1/K$ in the x and y directions, which is repeated at constant time steps of $\Delta t = T/N$ in the t coordinate axis, where T is the final time of interest. Grid points occur at the intersection of grid lines, and are defined by $(x_j, y_k, t_n) = (j\Delta x, k\Delta y, n\Delta t)$ for j, k, $n = 0\,(1)\,J$, K, N, where the notation $p\,(q)\,r$ denotes the set of integers from p to r in steps of q. The exact solution $\hat{\tau}$ of the PDE is approximated at interior grid points by the numerical method. Values on the boundaries, that is, when $j = 0$ or J, $k = 0$ or K, are given by the boundary conditions, and when $n = 0$, by the initial conditions. The value of $\hat{\tau}$ at (x_j, y_k, t_n), labelled the $(j, k, n)^{th}$ grid point, will be denoted $\hat{\tau}^n_{j,k}$. A similar notation will apply to all other variable quantities.

2.1 Finite-Difference Equations

As an illustration of the discretization process, consider the following simple example, in which the one-dimensional constant-coefficient transport equation is discretized at the $(j, n)^{th}$ grid point, so that

$$\left.\frac{\partial \hat{\tau}}{\partial t}\right|^n_j + u \left.\frac{\partial \hat{\tau}}{\partial x}\right|^n_j - \alpha \left.\frac{\partial^2 \hat{\tau}}{\partial x^2}\right|^n_j = 0. \tag{2.3}$$

Each of the derivatives in (2.3) may be replaced by a discrete form. A list of all difference forms used in this work is given in Appendix A.

For example, if the time derivative is replaced by first-order forward-time differencing, namely

$$\left.\frac{\partial \hat{\tau}}{\partial t}\right|_j^n = \frac{\hat{\tau}_j^{n+1} - \hat{\tau}_j^n}{\Delta t} - \left.\frac{\Delta t}{2}\frac{\partial^2 \hat{\tau}}{\partial t^2}\right|_j^n + O\{(\Delta t)^2\}, \tag{2.4}$$

and the space derivatives are replaced by the second-order centered-space difference approximations given by (A.3) and (A.6) respectively, then the following discretized partial differential equation (DPDE) is obtained

$$\begin{aligned}\left(\frac{\hat{\tau}_j^{n+1} - \hat{\tau}_j^n}{\Delta t} - \frac{\Delta t}{2}\frac{\partial^2 \hat{\tau}}{\partial t^2} + O\{(\Delta t)^2\}\right) + u\left(\frac{\hat{\tau}_{j+1}^n - \hat{\tau}_{j-1}^n}{2\Delta x} - \frac{(\Delta x)^2}{6}\frac{\partial^3 \hat{\tau}}{\partial x^3} + O\{(\Delta x)^4\}\right) \\ -\alpha\left(\frac{\hat{\tau}_{j+1}^n - 2\hat{\tau}_j^n + \hat{\tau}_{j-1}^n}{(\Delta x)^2} - \frac{(\Delta x)^2}{12}\frac{\partial^4 \hat{\tau}}{\partial x^4} + O\{(\Delta x)^4\}\right) = 0.\end{aligned} \tag{2.5}$$

Omitting the terms of $O\{\Delta t\}$ and $O\{(\Delta x)^2\}$ in (2.5) yields the finite-difference approximation (FDA)

$$\frac{\tau_j^{n+1} - \tau_j^n}{\Delta t} + u\frac{\tau_{j+1}^n - \tau_{j-1}^n}{2\Delta x} - \alpha\frac{\tau_{j+1}^n - 2\tau_j^n + \tau_{j-1}^n}{(\Delta x)^2} = 0, \tag{2.6}$$

whose solution gives approximate values τ for the exact solution $\hat{\tau}$ of (2.3), provided (2.6) is stable (see Section 2.2). The error in using (2.6) instead of (2.5) is called the truncation error, and is given by

$$E_{\mathrm{T}} = \left[-\frac{1}{2}\Delta t\frac{\partial^2 \hat{\tau}}{\partial t^2} - \frac{1}{6}u(\Delta x)^2\frac{\partial^3 \hat{\tau}}{\partial x^3} + \frac{1}{12}\alpha(\Delta x)^2\frac{\partial^4 \hat{\tau}}{\partial x^4} + O\{(\Delta t)^2, (\Delta x)^4\}\right]_j^n, \tag{2.7}$$

which may be written

$$E_{\mathrm{T}} = O\{\Delta t, (\Delta x)^2\}. \tag{2.8}$$

Rearranging (2.6) yields the finite-difference equation (FDE)

$$\tau_j^{n+1} = \frac{1}{2}(c + 2s)\tau_{j-1}^n + (1 - 2s)\tau_j^n - \frac{1}{2}(c - 2s)\tau_{j+1}^n, \tag{2.9}$$

in which c is the Courant number and s is the diffusion number, defined by

$$c = u\Delta t/\Delta x, \quad s = \alpha\Delta t/(\Delta x)^2. \tag{2.10}$$

Equation (2.9) is known as the FTCS formula (Noye 1987b) and gives the approximate value τ_j^{n+1} at time-level $(n + 1)$ explicitly in terms of the approximate values τ_{j-1}^n, τ_j^n and τ_{j+1}^n at time-level n.

If the velocity and diffusion in (2.3) can vary in space and time, then c and s are replaced by the local Courant number and the local diffusion number, so that

$$c_j^n = u_j^n \Delta t/\Delta x, \quad s_j^n = \alpha_j^n \Delta t/(\Delta x)^2. \tag{2.11}$$

A three-level FDE may be written in the general form

$$\mathcal{F}\{\tau\} = \mathcal{L}\{\tau\} - \mathcal{R}\{\tau\} - \mathcal{P}\{\tau\} = 0, \tag{2.12}$$

where $\mathcal{L}$, $\mathcal{R}$ and $\mathcal{P}$ collect the values of τ appearing at time-level $(n+1)$, n and $(n-1)$ respectively. Methods involving more than three levels in time will not be considered here. A computational stencil (p, q, r) specifies that p grid points are involved at time-level $(n+1)$, q at level n, and r at level $(n-1)$. In two-level methods both $\mathcal{P}$ and r are omitted. If $p = 1$ the FDE is termed explicit, otherwise it is implicit.

2.1.1 Accuracy and Convergence

The accuracy of FDEs may be assessed by comparing the size of their truncation errors, which can be found by replacing the approximation τ by the exact solution $\hat{\tau}$, so that

$$\mathcal{F}\{\hat{\tau}\} = \mathcal{L}\{\hat{\tau}\} - \mathcal{R}\{\hat{\tau}\} - \mathcal{P}\{\hat{\tau}\}, \tag{2.13}$$

and then taking the Taylor series expansion of each term in (2.13) about any fixed point in the solution domain. This yields

$$\mathcal{F}\{\hat{\tau}\} = -K(c, s)\Delta t E_{\mathrm{T}}, \tag{2.14}$$

in which E_{T} is the truncation error obtained by replacing the PDE by the FDA. Note that, in the case of pure advection $K = K(c)$, and for pure diffusion $K = K(s)$. Often K is found to equal unity, in which case (2.14) is simplified to

$$\mathcal{F}\{\hat{\tau}\} = -\Delta t E_{\mathrm{T}}. \tag{2.15}$$

Since the time-dependent and spatially-dependent terms in the truncation error might cancel, the temporal derivatives will be converted to spatial derivatives using the original PDE. The order of the leading term in the truncation error then gives the order of convergence of the FDE in terms of Δx.

2.2 Properties of Finite-Difference Equations

To be useful, any finite-difference equation must be stable and consistent with the partial differential equation it is modelling, and its solution must converge to the solution of the partial differential equation. The reader is referred to Richtmyer and Morton (1967), Noye (1984a) and Hirsch (1990) for a comprehensive discussion of these concepts as only a brief summary follows.

A numerical method is said to be convergent if the discretization error e, defined as the difference between the finite-difference solution τ and the exact solution of the partial differential equation $\hat{\tau}$, approaches zero at all points in the solution domain in the limit as the grid spacings tend to zero.

A finite-difference equation is said to be consistent with a partial differential equation if, in the limit as the grid spacings tend to zero, the FDE is identical to the PDE at each point in the solution domain. In other words, the truncation error must become zero, as the grid spacings tend to zero.

A formal consistency analysis involves replacing the approximation τ by the exact solution $\hat{\tau}$ in the numerical scheme, taking the Taylor expansion of each term about some fixed point, retaining the higher order terms, and taking the limit of the resultant expression as the grid spacings approach zero. This reveals the original PDE being approximated.

The solution to a FDE found by the use of computers is not the exact solution τ of the difference equation, but an approximation $\tilde{\tau}$. A FDE is said to be stable, if the effect of the round-off errors $\xi = \tau - \tilde{\tau}$, introduced at some time level remain bounded after an infinite number of time steps.

2.2.1 Lax's Equivalence Theorem

Lax's theorem (Lax and Richtmyer 1956) states that "Given a properly posed linear initial value problem and a finite-difference approximation to it that satisfies the consistency condition, stability is the necessary and sufficient condition for convergence."

A properly posed problem is one in which the solution of the PDE depends continuously on the given initial conditions. Linearity implies that round-off errors propagate according to the homogeneous form of the given finite-difference equation.

Lax's theorem allows the complicated technique required to prove convergence of the solution of the FDE to the solution of the PDE to be replaced by the relatively easy procedures of proving stability of a FDE and its consistency with the PDE.

2.2.2 von Neumann Stability Analysis

The von Neumann stability analysis is the most commonly used method to test the stability of linear FDEs. Its main advantage is that it may be applied to any spatial grid and for any time step, which is of importance if Lax's equivalence theorem is used.

For linear initial value problems with constant coefficients, the von Neumann stability analysis provides both sufficient and necessary conditions for the stability of a FDE.

Consider a linear FDE written in the form

$$\mathcal{F}\{\tau\} = 0, \tag{2.16}$$

The error propagation equation, namely

$$\mathcal{F}\{\xi\} = 0, \tag{2.17}$$

is the same as the homogeneous part of the FDE being considered (Noye 1984a). A solution of the error propagation equation is sought in the variable separable form

$$\xi_j^n = G^n \exp\{i\beta j\}, \tag{2.18}$$

in which $i = \sqrt{-1}$ and $\beta = m\pi\Delta x$. The superscript n denotes the n^{th} power of the (possibly) complex number G. The von Neumann amplification factor $G = \xi_j^{n+1}/\xi_j^n$ must satisfy the criterion

$$|G| \leq 1 \ \ \forall\, \beta, \tag{2.19}$$

for error growth to be bounded, and hence for the FDE to be stable.

In this work, the von Neumann stability analysis will be used to determine stability conditions for FDEs developed to solve linear PDEs with non-constant coefficients. This local stability analysis, achieved by freezing the coefficients at their value at a fixed point, will provide necessary but not sufficient conditions for the stability of the FDEs considered (Hirsch 1990). A local von Neumann stability analysis is implied wherever a stability analysis is given.

2.2.3 Additional Properties

The use of implicit FDEs results in systems of linear algebraic equations which must be solved simultaneously. Direct methods (such as Gauss elimination or the Thomas algorithm for tridiagonal systems) are termed solvable if round-off errors do not magnify during the solution procedure. A necessary (but not sufficient) condition for a method to be solvable is that the coefficient matrix of the system of equations is diagonally dominant. This is so if

$$|a_{ii}| \geq \sum_{\substack{j=1 \\ j \neq i}}^{N} |a_{ij}|, \qquad i = 1(1)N, \tag{2.20}$$

where $A = [a_{ij}]$ is the $N \times N$ coefficient matrix of the system of equations to be solved.

PDEs usually model physical phenomena characterizing the conservation of a quantity such as mass or momentum. Numerical schemes should respect conservation laws, so that "at steady state and in the absence of sources, the amount of a conserved quantity leaving a closed volume is equal to the amount entering that volume" (Ferziger and Perić 1996). Although non-conservative schemes may produce artificial sources and sinks, if they are consistent and stable they can lead to accurate solutions on a very fine grid (Ferziger and Perić 1996).

If non-negative initial and boundary conditions are specified, it can be shown that the exact solution of (2.1) is also non-negative. It is important that numerical schemes preserve the non-negativity of physically non-negative quantities. Additionally, no new overshoots or undershoots should be introduced into the solution by the numerical scheme (Holmgren 1994). A linear finite-difference equation of the form

$$\tau_j^{n+1} = \sum_{m=-q}^{s} a_m \tau_{j+m}^{n}, \tag{2.21}$$

is defined as positive if all the coefficients a_m are positive (Vreugdenhil and Koren 1993). If this holds, no negative values will be introduced into the numerical solution.

Chapter 3

One-Dimensional Non-Conservative Advection

The accurate approximation of advection terms in the equations of motion governing fluid flow has been a topic of ongoing research for many years. Meteorologists, oceanographers, and scientists modelling industrial flows, have spent considerable effort developing accurate high-order numerical solutions for the advection equation with constant coefficients, with some success (see for example, Leonard 1979; Leonard and Niknafs 1990; Noye 1986, 1991; Noye and von Trojan 1996). However, little progress has been made toward developing FDMs for the case in which the advective velocity varies in time and space, as it does in most physical situations.

When a FDE developed for use in constant velocity problems is applied in a variable velocity situation, it generally degenerates at least one order of convergence (Noye and von Trojan 1998). For instance, Rusanov's explicit fourth-order formula (Rusanov 1970) reverts to first-order accuracy when the fluid velocity is variable. A technique has been developed (Noye and von Trojan 1998) which can modify any finite-difference formula which is consistent with the advection equation so that it will retain its constant velocity order in the variable velocity case.

This is illustrated by application to several well known FDEs, which will be referred to as base methods. These formulae are based on different weighted discretizations of the constant velocity advection equation. The weights are chosen to eliminate some of the dominant terms in the truncation error of the modified equivalent partial differential equation (MEPDE), described in Section 3.3. The more terms that can be removed from the truncation error, which is the difference between the MEPDE and the given PDE, the higher the accuracy of the formula. Numerical tests show the superiority of the modified method relative to the base method when the velocity is variable, and illustrate the theoretical orders of convergence.

3.1 Mathematical Formulation

The advection of a scalar quantity $\hat{\tau}(x,t)$ in a fluid moving with constant velocity u, is governed by the equation

$$\frac{\partial \hat{\tau}}{\partial t} + u\frac{\partial \hat{\tau}}{\partial x} = 0. \tag{3.1}$$

This equation is solved on the region $0 \leq x \leq 1$, $0 < t \leq T$ subject to an initial condition

$$\hat{\tau}(x,0) = f(x), \tag{3.2}$$

and a boundary condition

$$\hat{\tau}(0,t) = g_{\text{L}}(t) \text{ if } u > 0, \quad \hat{\tau}(1,t) = g_{\text{R}}(t) \text{ if } u < 0, \tag{3.3}$$

where f, g_{L} and g_{R} are known functions. Alternatively, if the problem is cyclic in space with a periodicity of one, periodic boundary conditions $\hat{\tau}(x,t) = \hat{\tau}(1+x,t)$ may be specified.

3.2 A Weighted (1,5) FDE

The advection equation can be differenced at the $(j,n)^{th}$ grid point on the (1,5) computational stencil using the weights ϕ, γ and δ in the following way (Noye 1998)

$$\begin{aligned}\frac{\tau_j^{n+1} - \tau_j^n}{\Delta t} + u\phi\left(\frac{\tau_j^n - \tau_{j-1}^n}{\Delta x}\right) + u\gamma\left(\frac{3\tau_j^n - 4\tau_{j-1}^n + \tau_{j-2}^n}{2\Delta x}\right) \\ + u\delta\left(\frac{-\tau_{j+2}^n + 8\tau_{j+1}^n - 8\tau_{j-1}^n + \tau_{j-2}^n}{12\Delta x}\right) + u(1-\phi-\gamma-\delta)\left(\frac{\tau_{j+1}^n - \tau_{j-1}^n}{2\Delta x}\right) = 0.\end{aligned} \tag{3.4}$$

Rearranging yields the explicit FDE

$$\begin{aligned}\tau_j^{n+1} = -\frac{c}{12}(\delta + 6\gamma)\tau_{j-2}^n + \frac{c}{6}(3 + \delta + 9\gamma + 3\phi)\tau_{j-1}^n + \frac{1}{2}(2 - 3\gamma c - 2\phi c)\tau_j^n \\ - \frac{c}{6}(3 + \delta - 3\gamma - 3\phi)\tau_{j+1}^n + \frac{c}{6}\delta\tau_{j+2}^n,\end{aligned} \tag{3.5}$$

in which $c = u\Delta t/\Delta x$ is the constant Courant number. Note that the second and third terms in (3.4) are backward-space difference forms, while the last two terms are centered-space difference forms. As is shown in the sections to follow, the choice of the weights in (3.4) yields the various explicit base schemes considered in this chapter, each of which has a different computational stencil.

3.3 The MEPDE

When the terms of a FDE which is consistent with the advection equation are expanded as Taylor series about a fixed point in the solution domain, an equivalent partial-differential equation (EPDE) is obtained. If this is modified by repeatedly adding a suitable multiple of the appropriate derivative of the EPDE to itself until all the temporal derivatives except $\partial\tau/\partial t$ have been eliminated, then the MEPDE (see Noye and Hayman 1986a), written as follows, is obtained

$$\frac{\partial\tau}{\partial t} + u\frac{\partial\tau}{\partial x} + \sum_{q=2}^{\infty} \frac{u(\Delta x)^{q-1}}{q!}\eta_q(c)\frac{\partial^q\tau}{\partial x^q} = 0. \tag{3.6}$$

If $\eta_q(c) = 0$ for $q = 2(1)Q$ and $\eta_{Q+1}(c) \neq 0$, the FDE is said to be order Q, since the truncation error

$$E_\tau = \sum_{q=Q+1}^{\infty} \frac{u(\Delta x)^{q-1}}{q!}\eta_q(c)\frac{\partial^q\tau}{\partial x^q}, \tag{3.7}$$

is $O\{(\Delta x)^Q\}$.

Artificial (numerical) diffusion is introduced into the solution by the use of the finite-difference equation unless $\eta_2 = 0$. It has been shown (see Noye 1984a, 1984b) that the terms involving even derivatives in the truncation error contribute to the amplitude error of the solution of the finite-difference equation, and the terms involving odd derivatives add to the wave-speed error of the numerical solution.

An analogous expression to (3.6) cannot be obtained for variable velocities because each differentiation of the EPDE would result in an increase in the number of terms that need to be considered. Instead, the exact solution $\hat{\tau}$ replaces τ in the FDE, and then each term is expanded as a Taylor series about some fixed point, yielding the discretized partial-differential equation (DPDE).

This is then modified by differentiating the original PDE successively, yielding a finite number of terms, which can be used to eliminate the temporal derivatives in the DPDE, giving the modified discretized partial-differential equation (MDPDE).

The development of (3.6) requires the function τ to be continuously differentiable on the computational domain. If there are discontinuities in the given initial or boundary conditions, the results expected from an analysis based on the MEPDE may not be found in practice. In addition to smooth initial and boundary conditions, the existence of a smooth velocity profile is assumed in the generation of the MDPDE.

The first three error terms of the MEPDE of (3.5) contain the factors

$$\begin{aligned}\eta_2 &= -\phi + c,\\ \eta_3 &= 1 + 2c^2 - \delta - 3\gamma - 3\phi c,\\ \eta_4 &= 4c + 6c^3 - 4\delta c + 6\gamma(1-2c) - \phi(1+12c^2) + 3\phi^2 c.\end{aligned} \tag{3.8}$$

If $\eta_2 = 0$, (3.5) is second-order, if $\eta_2 = \eta_3 = 0$ it is third-order, and if $\eta_2 = \eta_3 = \eta_4 = 0$ it is fourth-order. Some examples will be given in the sections to follow.

3.4 Leith's FDE

Selecting $\phi = c$, so $\eta_2 = 0$, and choosing $\gamma = \delta = 0$ in (3.5), so only a (1,3) stencil is involved, yields Leith's method for constant velocity (Leith 1964). When written in the form (2.12), Leith's method is given by

$$\begin{aligned}\mathcal{L}\{\tau\} &= \tau_j^{n+1}\\ \mathcal{R}\{\tau\} &= \frac{1}{2}c(c+1)\tau_{j-1}^n + (1-c^2)\tau_j^n + \frac{1}{2}c(c-1)\tau_{j+1}^n.\end{aligned} \tag{3.9}$$

For a variable velocity field c is replaced by the local Courant number c_j^n. For brevity, in the following, the superscript n and subscript j will be omitted from expressions involving u and c; it is understood that these are evaluated at (x_j, t_n). The order of (3.9) may be determined by conducting a formal consistency analysis. Replacing the finite-difference approximation τ by the exact solution $\hat{\tau}$, and taking the Taylor series expansion of each term in (3.9) about (x_j, t_n), gives

$$\mathcal{F}\{\hat{\tau}\} = -\Delta t E_{\mathrm{T}}, \tag{3.10}$$

where the truncation error is given by

$$E_{\mathrm{T}} = -\frac{1}{2}\Delta t\left(\frac{\partial^2\hat{\tau}}{\partial t^2} - u^2\frac{\partial^2\hat{\tau}}{\partial x^2}\right) + O\{2\}, \tag{3.11}$$

and

$$O\{p\} = O\{(\Delta x)^{p-m}(\Delta t)^m;\ m \text{ an integer}\}. \tag{3.12}$$

To determine whether the terms in (3.11) cancel, the time derivative is converted to space derivatives.

Differentiating (3.1) with respect to t gives

$$\frac{\partial^2 \hat{\tau}}{\partial t^2} = -\frac{\partial u}{\partial t}\frac{\partial \hat{\tau}}{\partial x} - u\frac{\partial^2 \hat{\tau}}{\partial t \partial x}, \tag{3.13}$$

while differentiating (3.1) with respect to x, yields

$$\frac{\partial^2 \hat{\tau}}{\partial x \partial t} = -\frac{\partial u}{\partial x}\frac{\partial \hat{\tau}}{\partial x} - u\frac{\partial^2 \hat{\tau}}{\partial x^2}. \tag{3.14}$$

Substitution of (3.14) into (3.13) to eliminate the cross derivative then produces

$$\frac{\partial^2 \hat{\tau}}{\partial t^2} = -\left(\frac{\partial u}{\partial t} - u\frac{\partial u}{\partial x}\right)\frac{\partial \hat{\tau}}{\partial x} + u^2\frac{\partial^2 \hat{\tau}}{\partial x^2}, \tag{3.15}$$

so that

$$E_{\mathrm{T}} = \frac{\Delta t}{2}\left(\frac{\partial u}{\partial t} - u\frac{\partial u}{\partial x}\right)\frac{\partial \hat{\tau}}{\partial x} + O\{2\}. \tag{3.16}$$

Leith's FDE is thus generally first-order convergent, unless u is constant, when it becomes second-order.

3.4.1 Modification

Substitution of (3.16) into (3.10) yields

$$\mathcal{F}\{\hat{\tau}\} = -d\left[\Delta x\frac{\partial \hat{\tau}}{\partial x}\right] + O\{3\}, \tag{3.17}$$

where the correction factor d is defined as

$$d = \frac{(\Delta t)^2}{2\Delta x}\left(\frac{\partial u}{\partial t} - u\,\frac{\partial u}{\partial x}\right). \tag{3.18}$$

The modification to retain the second-order convergence of the FDE is achieved by replacing the spatial derivative in (3.17) by a finite-difference approximation of appropriate order. Note that the correction factor is intrinsically first-order. To achieve second-order convergence, the approximation to the spatial derivative must therefore be at least first-order. A possible choice involves using upwind differencing, in which the major contribution to the approximation comes from one side. Although the first-order upwind scheme is inaccurate because it introduces numerical diffusion into the solution, this two-point scheme will be used here, so that Leith's (1,3) stencil is retained.

Because information travels downstream at a finite speed in advection problems, the behaviour of the quantity being approximated depends on the information it receives from a certain number of upstream points. If information is required from points more distant than the differencing allows, numerical instability results. This may be avoided by using backward-space (BS) differencing when the flow is in the positive direction, and forward-space (FS) differencing when the flow is in the negative direction. Hence, using first-order BS differencing for $d > 0$ and first-order FS differencing for $d < 0$, gives

$$\mathcal{F}\{\tau\} = \frac{1}{2}(|d| + d)\tau_{j-1}^n - |d|\tau_j^n + \frac{1}{2}(|d| - d)\tau_{j+1}^n, \tag{3.19}$$

after the error terms of order three and higher have been omitted. A second-order modified form of Leith's method is then given by expanding (3.19) using (3.9) and rearranging, yielding

$$\tau_j^{n+1} = \frac{1}{2}(c^2 + c + d + |d|)\tau_{j-1}^n + (1 - c^2 - |d|)\tau_j^n + \frac{1}{2}(c^2 - c - d + |d|)\tau_{j+1}^n, \tag{3.20}$$

which will be denoted mod_L in the following. Methods involving odd-order spatial derivatives in the leading term of their truncation error are known to be much more dispersive than those involving even-order derivatives (Leonard 1981). The presence of the first-order spatial derivative in (3.17) contributes to the wave-speed error of the numerical solution of the base method (see Table 3.1 in Section 3.12). The discretization of the spatial derivative using first-order upwind (UW) differencing introduces a second-order amplitude error into the numerical solution of the mod_L method (refer to Table 3.2). Consequently, the amplitude of the solution is better approximated by the base method, but its wave-speed is more accurately represented by the new modified method.

3.4.2 Modification

Alternatively, second-order centered-space (CS) differencing may be used to discretize $\partial\hat{\tau}/\partial x$, thereby eliminating the first-order wave-speed error present for the base method, but introducing a third-order wave-speed error into the numerical solution (see Table 3.2). An advantage of using CS differencing is that the flow direction does not need to be checked, so separate cases for $d > 0$ and $d < 0$ need not be considered. The resultant modified method will be denoted mod2_L, and is given by

$$\tau_j^{n+1} = \frac{1}{2}(c^2 + c + d)\tau_{j-1}^n + (1 - c^2)\tau_j^n + \frac{1}{2}(c^2 - c - d)\tau_{j+1}^n, \tag{3.21}$$

which is also accurate to second-order. Leonard (1981) does not advocate the use of central differencing for modelling odd-order spatial derivatives, in particular the first-order advection term, because it may introduce nonphysical oscillations and instability into the numerical solution. So, before the methods can be applied in any practical situation, stability conditions must be determined. This is discussed below.

3.4.3 Stability

The von Neumann stability analysis is the most commonly used technique to determine stability. It is based on the assumption of the existence of a Fourier decomposition of the solution over the computational domain. This implies the existence of periodic boundary conditions and constant coefficients (Hirsch 1990). The von Neumann stability analysis cannot therefore strictly be applied for variable coefficients, because single harmonics can no longer be isolated (Hirsch 1990).

The stability analysis can however be applied locally (see for example Hirsch 1990, Strikwerda 1989, Mitchell and Griffiths 1980). This involves freezing the coefficients so that the velocity and hence the Courant number are assumed to be constant within the computational stencil. If the coefficients are frozen, the corresponding constant coefficient equation is obtained (Strikwerda 1989). This is also the case for the modified methods because the correction factors, which depend on the derivatives of the variable velocity, vanish if the velocity is taken to be constant.

The stability analysis applied locally then provides a local stability condition, which is a necessary condition for the stability of the variable coefficient problem (Richtmyer and Morton 1967, Hirsch 1990). There is much numerical evidence to support the contention that if the von Neumann stability condition, derived as though the coefficients were constant, is satisfied at every point in the domain, then the variable coefficient problem is also stable (Mitchell and Griffiths 1980).

Furthermore, if the stability condition obtained from a local stability analysis is violated in a small region, the instability will not grow outside that region (Strikwerda 1989). In other words, the method can be stable even if the stability criterion $|G| \leq 1$ is violated for some values of the coefficients. This is because the condition obtained from a local stability analysis is a necessary but not sufficient condition for the stability of the variable coefficient problem. That is, the method is stable if the local stability condition is satisfied, but it is not necessarily unstable if the condition is violated.

The determination of sufficient conditions for the stability of variable coefficient problems requires additional restrictions to be placed on the von Neumann amplification factor (Richtmyer and Morton 1967, Hirsch 1990). These restrictions have only been determined for a very limited class of variable-coefficient problems (see Richtmyer and Morton 1967, Hirsch 1990); where it was found that a mild strengthening of the local stability condition is sufficient for the overall stability of the variable-coefficient problem (Richtmyer and Morton 1967, Hirsch 1990). The determination of sufficient conditions for variable-coefficient problems will not be considered in this work. Necessary conditions for the stability of the variable coefficient problem will be determined from a local stability analysis as follows:

1. Fix the coefficients, so that the velocity is constant within the computational stencil.

2. Set any correction factors, which are dependent on the derivatives of the velocity, to zero.

3. Perform the von Neumann stability analysis locally.

4. The stability of the constant coefficient equation then gives a necessary condition for the stability of the variable coefficient problem.

For the variable coefficient problem considered in this section, a necessary condition for the stability is then given by the stability of Leith's method. Leith's method is stable for $|c| \leq 1$ (Leith 1964).

3.5 The UW15 FDE

If $\delta = 0$ and $3 + \delta - 3\gamma - 3\phi = 0$ in (3.5), a formula involving no FS grid points is obtained. Setting $\phi = c$ eliminates η_2, so the scheme introduces no artificial diffusion, and the remaining weight takes the value $\gamma = 1 - c$. This formula, described by Noye (1984b), and given by

$$\tau_j^{n+1} = -\frac{1}{2}c(1-c)\tau_{j-2}^n + c(2-c)\tau_{j-1}^n + \frac{1}{2}(1-c)(2-c)\tau_j^n, \tag{3.22}$$

is unstable for $c < 0$ and so can only be used when $u > 0$. An analogous formula, which involves no BS grid points and so can only be applied when $u < 0$, may be obtained by setting $-\Delta x$ for Δx in (3.22), so that $-c$ replaces c, giving

$$\tau_j^{n+1} = \frac{1}{2}(1+c)(2+c)\tau_j^n - c(2+c)\tau_{j+1}^n + \frac{1}{2}c(1+c)\tau_{j+2}^n. \tag{3.23}$$

Summarizing (3.22) and (3.23) gives an FDE with a (1,5) stencil

$$\begin{aligned}\tau_j^{n+1} = &-\frac{1}{4}(1-c)(c+|c|)\tau_{j-2}^n + \frac{1}{2}(2-c)(c+|c|)\tau_{j-1}^n + \frac{1}{2}(2-3|c|+c^2)\tau_j^n \\ &+\frac{1}{2}(2+c)(|c|-c)\tau_{j+1}^n - \frac{1}{4}(1+c)(|c|-c)\tau_{j+2}^n,\end{aligned} \tag{3.24}$$

which may be applied for c of arbitrary sign, and will be referred to as the UW15 FDE. Its accuracy when used in a variable velocity situation is found by replacing τ by the exact solution $\hat{\tau}$, and taking

the Taylor series expansion of each term in (3.24) about (x_j, t_n), yielding

$$\mathcal{F}\{\hat{\tau}\} = -\Delta t E_{\mathrm{T}}, \tag{3.25}$$

in which

$$E_{\mathrm{T}} = \frac{\Delta t}{2}\left(\frac{\partial u}{\partial t} - u\frac{\partial u}{\partial x}\right)\frac{\partial \hat{\tau}}{\partial x} + O\{2\}. \tag{3.26}$$

The dominant term in the truncation error is identical to that in (3.16) for Leith's method, so the modification procedure will be the same. This gives the following second-order modified methods:

$$\begin{aligned}\tau_j^{n+1} = &-\frac{1}{4}(1-c)(c+|c|)\tau_{j-2}^n + \frac{1}{2}(2-c)(c+|c|)\tau_{j-1}^n + \frac{1}{2}(2-3|c|+c^2)\tau_j^n \\ &+\frac{1}{2}(2+c)(|c|-c)\tau_{j+1}^n - \frac{1}{4}(1+c)(|c|-c)\tau_{j+2}^n + \mathcal{F}\{\tau\},\end{aligned} \tag{3.27}$$

where

1. $\mathcal{F}$ = (3.19) when upwind differencing is used, giving the mod_U FDE
2. $\mathcal{F}$ = (3.19) with $|d| = 0$ when CS differencing is used, giving the mod2_U FDE

(3.28)

The stability of the base method $|c| \leq 2$ (Noye 1984b), provides a necessary condition for the stability of the variable coefficient problem (see Richtmyer and Morton 1967, Hirsch 1990).

3.6 Rusanov's FDE

Setting $\eta_2 = \eta_3 = \eta_4 = 0$ in (3.8) gives values for the weights $\phi = c$, $\gamma = c(1-c^2)/6$, $\delta = (2-c)(1-c^2)/2$. Substituting into (3.5) gives Rusanov's fourth-order FDE for constant velocity (Rusanov 1970), which is the optimal formula for the (1,5) stencil involved, in the sense that the truncation error has the highest possible order. Consider Rusanov's FDE:

$$\begin{aligned}&\mathcal{L}\{\tau\} = \tau_j^{n+1} \\ &\mathcal{R}\{\tau\} = -\frac{1}{24}c(1-c^2)(2+c)\tau_{j-2}^n + \frac{1}{6}c(1+c)(4-c^2)\tau_{j-1}^n + \frac{1}{4}(1-c^2)(4-c^2)\tau_j^n \\ &\qquad -\frac{1}{6}c(1-c)(4-c^2)\tau_{j+1}^n + \frac{1}{24}c(1-c^2)(2-c)\tau_{j+2}^n,\end{aligned} \tag{3.29}$$

applied to a variable velocity field. The DPDE obtained by conducting a consistency analysis of this

method about the point (x_j, t_n) is given by

$$\mathcal{F}\{\hat{\tau}\} = -\Delta t E_{\mathrm{T}}, \tag{3.30}$$

where the truncation error is given by

$$E_{\mathrm{T}} = -\frac{1}{2}\Delta t\left(\frac{\partial^2\hat{\tau}}{\partial t^2} - u^2\frac{\partial^2\hat{\tau}}{\partial x^2} + \frac{1}{3}\Delta t\frac{\partial^3\hat{\tau}}{\partial t^3} + \frac{1}{3}u^3\Delta t\frac{\partial^3\hat{\tau}}{\partial x^3}\right) + O\{3\}. \tag{3.31}$$

The MDPDE is obtained by converting the time derivatives to space derivatives. Differentiating (3.15) with respect to time and then eliminating the cross derivatives yields

$$\frac{\partial^3\hat{\tau}}{\partial t^3} = A\frac{\partial\hat{\tau}}{\partial x} + B\frac{\partial^2\hat{\tau}}{\partial x^2} + C\frac{\partial^3\hat{\tau}}{\partial x^3}, \tag{3.32}$$

in which

$$\begin{aligned} A &= -\frac{\partial^2 u}{\partial t^2} + 2\frac{\partial u}{\partial t}\frac{\partial u}{\partial x} + u\frac{\partial^2 u}{\partial t\partial x} - u\left(\frac{\partial u}{\partial x}\right)^2 - u^2\frac{\partial^2 u}{\partial x^2}, \\ B &= 3u\left(\frac{\partial u}{\partial t} - u\frac{\partial u}{\partial x}\right), \\ C &= -u^3. \end{aligned} \tag{3.33}$$

Substituting (3.32) and (3.15) into (3.31) gives

$$E_{\mathrm{T}} = \frac{1}{6}\Delta t\left(u^{-1}B\frac{\partial\hat{\tau}}{\partial x} - A\Delta t\frac{\partial\hat{\tau}}{\partial x} - B\Delta t\frac{\partial^2\hat{\tau}}{\partial x^2}\right) + O\{3\}. \tag{3.34}$$

The truncation error, with leading term identical to that of Leith's equation and the UW15 method, shows that Rusanov's method is also first-order convergent when u varies.

3.6.1 Modifications

Equation (3.30) can now be written as

$$\mathcal{F}\{\hat{\tau}\} = -2d\left[\Delta x\frac{\partial\hat{\tau}}{\partial x}\right] + h\left[(\Delta x)^2\frac{\partial^2\hat{\tau}}{\partial x^2}\right] + O\{4\}, \tag{3.35}$$

where the correction factors d and h are defined as

$$d = \frac{(\Delta t)^2}{12\Delta x}(u^{-1}B - A\Delta t),$$
$$h = \frac{(\Delta t)^3}{6(\Delta x)^2}B. \tag{3.36}$$

If a third-order truncation error is to be obtained, the first-order spatial derivative in (3.35) must be correct to at least second-order, and the second-order spatial derivative must be correct to at least first-order. The resultant third-order modified methods take the form

$$\begin{aligned}\tau_j^{n+1} = &-\frac{1}{24}c(1-c^2)(2+c)\tau_{j-2}^n + \frac{1}{6}c(1+c)(4-c^2)\tau_{j-1}^n + \frac{1}{4}(1-c^2)(4-c^2)\tau_j^n \\ &-\frac{1}{6}c(1-c)(4-c^2)\tau_{j+1}^n + \frac{1}{24}c(1-c^2)(2-c)\tau_{j+2}^n + \mathcal{F}\{\tau\},\end{aligned} \tag{3.37}$$

where

1. $\mathcal{F} = -\frac{1}{2}(d+|d|)\tau_{j-2}^n + \{2(d+|d|)+h\}\tau_{j-1}^n - (3|d|+2h)\tau_j^n$
$-\{2(d-|d|)-h\}\tau_{j+1}^n + \frac{1}{2}(d-|d|)\tau_{j+2}^n$, giving the mod_R FDE (3.38)

2. $\mathcal{F} = (d+h)\tau_{j-1}^n - 2h\tau_j^n - (d-h)\tau_{j+1}^n$, giving the mod2_R FDE

The mod_R method is derived by using second-order upwind differencing for the first spatial derivative and second-order centered space differencing for the second spatial derivative in (3.35). The mod2_R method is derived by using second-order centered space differencing for both spatial derivatives.

3.6.2 Remarks

The modification procedure introduces third and fourth-order dispersion into the solution of both methods, but the magnitude of these errors is halved when second-order centered space differencing instead of second-order upwind differencing is used to discretize the first-order spatial derivative. This can be seen by comparing A.3 and A.4 in Appendix A; and means that the mod2_R method should be more accurate. Both modified methods introduce a fourth-order amplitude error into the numerical solution when the second-order spatial derivative is replaced by second-order centered space differencing. The dominant error terms present for the base method have been eliminated in both modified methods.

A fourth-order method could be obtained by including the third-order error term in (3.31) into the discretization. This term is given by

$$O\{3\} = -\frac{1}{24}(\Delta t)^3 \left(\frac{\partial^4 \hat{\tau}}{\partial t^4} - u^4 \frac{\partial^4 \hat{\tau}}{\partial x^4} \right). \tag{3.39}$$

We would then have to convert the fourth-order temporal derivative to space derivatives, and introduce an additional correction factor in the modification procedure. As this adds significantly to the complexity of any modified method, this shall not be considered here.

We recall that the von Neumann stability analysis can only be used to establish necessary and sufficient conditions for initial value problems with constant coefficients. For variable coefficients, a local stability analysis leads to necessary conditions. Rusanov's method is stable for $|c| \leq 1$ (Rusanov 1970), giving a necessary condition for the stability of the variable coefficient problem.

3.7 A Weighted (3,3) FDE

Consider the advection equation evaluated at the point $(x_j, t_{n+1/2})$. Expressing the time derivative in the symmetrically weighted form:

$$\left.\frac{\partial \hat{\tau}}{\partial t}\right|_j^{n+1/2} = \phi \left.\frac{\partial \hat{\tau}}{\partial t}\right|_{j-1}^{n+1/2} + (1-\phi) \left.\frac{\partial \hat{\tau}}{\partial t}\right|_j^{n+1/2} + \phi \left.\frac{\partial \hat{\tau}}{\partial t}\right|_{j+1}^{n+1/2} + O\{(\Delta x)^2\}, \tag{3.40}$$

and the space derivative as the average of its value at the n^{th} and $(n+1)^{th}$ time levels, namely

$$\left.\frac{\partial \hat{\tau}}{\partial x}\right|_j^{n+1/2} = \frac{1}{2}\left(\left.\frac{\partial \hat{\tau}}{\partial x}\right|_j^{n} + \left.\frac{\partial \hat{\tau}}{\partial x}\right|_j^{n+1} \right) + O\{(\Delta t)^2\}, \tag{3.41}$$

and replacing all derivatives by their second-order centered-difference forms, gives the weighted method

$$\begin{aligned} \mathcal{L}\{\tau\} &= (4\phi - c)\tau_{j-1}^{n+1} + 4(1-2\phi)\tau_j^{n+1} + (4\phi + c)\tau_{j+1}^{n+1} \\ \mathcal{R}\{\tau\} &= (4\phi + c)\tau_{j-1}^{n} + 4(1-2\phi)\tau_j^{n} + (4\phi - c)\tau_{j+1}^{n}. \end{aligned} \tag{3.42}$$

If the coefficients on the left hand side of the weighted FDE are all non-zero, an implicit scheme is obtained. Implicit methods can only be used to approximate solutions of the advection equation if the boundary values are known at both ends of the computational domain, or if periodic boundary conditions are used. A more comprehensive discussion of boundary conditions is given in Section 3.10.

The resultant tridiagonal system can be solved using the Thomas algorithm provided that the system is diagonally dominant. Diagonal dominance is required so that the Thomas algorithm is stable with respect to the propagation of round-off errors (Hirsch 1990). The advantage of using the Thomas algorithm for tridiagonal systems is that it is much more efficient than using, for example, Gaussian elimination with back substitution.

From the definition given in Section 2.2.3, it can be seen that the weighted method is diagonally dominant provided $4|1-2\phi| \geq |4\phi - c| + |4\phi + c|$. Furthermore, since it is diagonally symmetrical for all c and ϕ, it is unconditionally stable. The proof of this is given in Noye (1984a).

In the constant coefficient case, the first few error terms of the MEPDE of (3.42) contain the factors

$$\begin{aligned} &\eta_2 = \eta_4 = \eta_6 = \ldots = 0, \\ &\eta_3 = (2 + c^2 - 12\phi)/2. \end{aligned} \tag{3.43}$$

Therefore, the weighted method is accurate to second-order unless $\eta_3 = 0$, when it is fourth-order convergent. Since the maximum order of convergence that can be attained for the given computational stencil is fourth-order (Noye 1986), the weighted method is optimal when $\eta_3 = 0$.

3.8 The Optimal FDE

For constant coefficients, the optimal version of the weighted method is obtained when the leading error term in the MEPDE takes the value zero. This is achieved by setting $\phi = (2 + c^2)/12$, giving the unconditionally stable fourth-order optimal method (Noye 1986), namely

$$\begin{aligned} \mathcal{L}\{\tau\} &= (2 - 3c + c^2)\tau_{j-1}^{n+1} + 2(4 - c^2)\tau_j^{n+1} + (2 + 3c + c^2)\tau_{j+1}^{n+1} \\ \mathcal{R}\{\tau\} &= (2 + 3c + c^2)\tau_{j-1}^{n} + 2(4 - c^2)\tau_j^{n} + (2 - 3c + c^2)\tau_{j+1}^{n}, \end{aligned} \tag{3.44}$$

which is diagonally dominant and hence solvable using the Thomas algorithm if $|c| \leq 1$ (Noye 1986). As the development of (3.44) involved evaluating the advection equation at $(x_j, t_{n+1/2})$, the consistency analysis to determine its convergence for variable coefficients is also taken about this point. The MDPDE obtained after converting the temporal derivatives to spatial ones in the truncation error is given by

$$\mathcal{F}\{\hat{\tau}\} = -12\Delta t E_{\mathrm{T}}, \tag{3.45}$$

where

$$E_{\mathrm{T}} = \frac{(\Delta t)^2}{24}\left(P\frac{\partial \hat{\tau}}{\partial x} - 2Q\frac{\partial^2 \hat{\tau}}{\partial x^2}\right) + O\{4\}, \tag{3.46}$$

in which

$$P = \frac{\partial^2 u}{\partial t^2} - 2\frac{\partial u}{\partial t}\frac{\partial u}{\partial x} + 2u\frac{\partial^2 u}{\partial x \partial t} - 2u\left(\frac{\partial u}{\partial x}\right)^2 + 4u^2 c^{-2}\frac{\partial^2 u}{\partial x^2},$$
$$Q = (1 - 4c^{-2})u^2\frac{\partial u}{\partial x}. \tag{3.47}$$

The truncation error indicates that the optimal method, denoted OPT, is second-order when the variable velocities are specified at each half time level. This is one order of magnitude higher than that found for the explicit base methods.

3.8.1 Modifications

Substitution of (3.46) into (3.45) yields the MDPDE

$$\mathcal{F}\{\hat{\tau}\} = -4d\left[\Delta x\frac{\partial \hat{\tau}}{\partial x}\right] + 2h\left[(\Delta x)^2\frac{\partial^2 \hat{\tau}}{\partial x^2}\right] + O\{5\}, \tag{3.48}$$

in which the correction factors are defined as

$$d = \frac{(\Delta t)^3}{8\Delta x}P,$$
$$h = \frac{(\Delta t)^3}{2(\Delta x)^2}Q. \tag{3.49}$$

To gain a third-order truncation error, both spatial derivatives must be substituted with approximations that are at least first-order. Since the derivatives are evaluated at $(x_j, t_{n+1/2})$, they are first written as the average of their values at the n^{th} and $(n+1)^{th}$ time levels, as shown in (3.41). The first-order spatial derivatives are then replaced by first-order upwind differencing, and the second-order derivatives are replaced by second-order centered-space forms. This gives

$$\begin{aligned}(2 - 3c + c^2 - d - \varphi)\tau_{j-1}^{n+1} + 2(4 - c^2 + \varphi)\tau_j^{n+1} + (2 + 3c + c^2 + d - \varphi)\tau_{j+1}^{n+1} \\ = (2 + 3c + c^2 + d + \varphi)\tau_{j-1}^{n} + 2(4 - c^2 - \varphi)\tau_j^{n} + (2 - 3c + c^2 - d + \varphi)\tau_{j+1}^{n},\end{aligned} \tag{3.50}$$

in which $\varphi = |d| + h$. This method, for which all coefficients are evaluated at the half time level, will be called the mod_O method. A fourth-order method can be obtained if all spatial derivatives are discretized using CS forms. The result, denoted mod2_O, is given by (3.50) with $\varphi = h$. The second-order wave and amplitude errors in the truncation error of the base method have been removed and replaced with third and fourth-order amplitude errors in the truncation error of the mod_O method, and by fourth-order numerical dispersion and amplitude errors in that of the mod2_O method.

3.9 Discrete Velocity Fields

In the previous sections it was assumed that the advective velocity is available as a differentiable function of x and t. However, in practice it is more likely that a discrete velocity field will be available, with values of u known only at grid points. In this case, the correction factors must be computed from the given values of u. For example, differencing the correction factor d given by (3.18) so that it is correct to first-order, does not change the order of the modified methods considered. If the derivatives in (3.18) are replaced by two-point forward-time and centered-space approximations, then d may be replaced by

$$\frac{(\Delta t)^2}{2\Delta x}\left(\frac{u_j^{n+1}-u_j^n}{\Delta t}-u_j^n\frac{u_{j+1}^n-u_{j-1}^n}{2\Delta x}\right)=\frac{1}{4}\left\{2(c_j^{n+1}-c_j^n)-c_j^n(c_{j+1}^n-c_{j-1}^n)\right\}, \tag{3.51}$$

without loss of order. If the values of u are not yet available at the time level $(n+1)$, then two-point backward-time differencing could be used instead, yielding

$$d\approx\frac{(\Delta t)^2}{2\Delta x}\left(\frac{u_j^n-u_j^{n-1}}{\Delta t}-u_j^n\frac{u_{j+1}^n-u_{j-1}^n}{2\Delta x}\right)=\frac{1}{4}\left\{2(c_j^n-c_j^{n-1})-c_j^n(c_{j+1}^n-c_{j-1}^n)\right\}. \tag{3.52}$$

The only disadvantage of this procedure is that the discrete velocity field must be known at the time level $n=-1$ if the values of d_j^0 are to be available for use at the initial time. The correction factors defined by (3.36) and (3.49) can also be discretized by replacing all derivatives of u by approximations which are accurate to at least first-order. Approximations to d must then involve three-levels in time because of the second-order temporal derivatives of u involved in each formulation. Either forward-space, backward-space or centered-space differencing could be used to approximate the second-order temporal derivative, the choice depending on which values of u are available. At most two levels in time are required to discretize h in each case.

3.10 Boundary Conditions

According to (3.3), if $u>0$ then the upstream boundary values at $x=0$ are known, but those downstream at $x=1$ will not be available. Likewise, if $u<0$ the values at $x=0$ are not given. Generally, a supplementary scheme of appropriate order must be used to provide these unknown values. For example, Leith's method could be supplemented at the left or right hand boundaries as required, using the UW15 method, which reduces to (3.23) for negative velocities or (3.22) for positive velocities. Higher-order boundary values may be obtained using the second, third or fourth-order implicit equations.

In this work, if a boundary value problem is being considered, the analytical solution will be used to provide the required initial and boundary values, as well as the downstream boundary values. The provision of an extra downstream boundary condition overspecifies the advection problem. However, this is acceptable if the advection equation is considered as the transport equation in the limit as the

diffusion tends to zero (Steinle 1994). The transport equation requires the specification of boundary conditions at both ends of the domain because of the presence of the second-order derivative.

Problems may also arise at points close to the boundaries when approximations to derivatives require data at more than three points, as in the case of the (1,5) methods. Often one-sided or implicit schemes of appropriate order are used to provide the values adjacent to the boundaries. The alternatives suggested by Steinle (1994) include interpolating the unknown values from surrounding values, or extrapolating the solution to give fictitious values outside the domain, which can be used directly in the equation.

As the performance of the schemes should be representative of a wide range of applications rather than dependent on a specific situation, periodic boundary conditions are often used in numerical tests. Although the condition $\hat{\tau}(0,t) = \hat{\tau}(1,t)$ is usually referred to as the periodic boundary condition, strictly speaking it is not a boundary condition, because for periodic problems there are no boundaries (see Strikwerda 1989).

3.11 Summary

The accuracy of four base methods (Leith's method, the UW15 method, Rusanov's method, the optimal method) was determined by conducting a formal consistency analysis, yielding the truncation error of the scheme. It was seen that the convergence rate of the base method is reduced when it is applied in a variable velocity situation. Each base method was modified in two ways in an attempt to restore the convergence rate back to (or close to) that of the base method when it is used for constant coefficients. The first modification was based on using upwind differencing for the term involving d and centered space differencing for the term involving h in the truncation error. There was a term involving h for Rusanov's method and for the optimal method. The modified methods were denoted mod_FL, where 'FL' is the first letter of the base method. The second modification involved using centered space differencing for both the term involving d and (if present) the term involving h. These modified methods were denoted mod2_FL, where again 'FL' is the first letter of the base method. A synopsis is given in Figure 3.1.

3.12 Conclusion

We have seen that FDMs may lose several orders of convergence when applied in variable velocity situations. The leading error terms, which may affect either wave speed (odd derivatives) or amplitude response (even derivatives) can be discretized and incorporated into the FDE, thereby increasing the order of convergence back to the constant coefficient rate. Although new errors are introduced by the modification procedure, they are always of a higher order than the original errors. The dominant error terms for all the base methods considered are presented in Table 3.1.

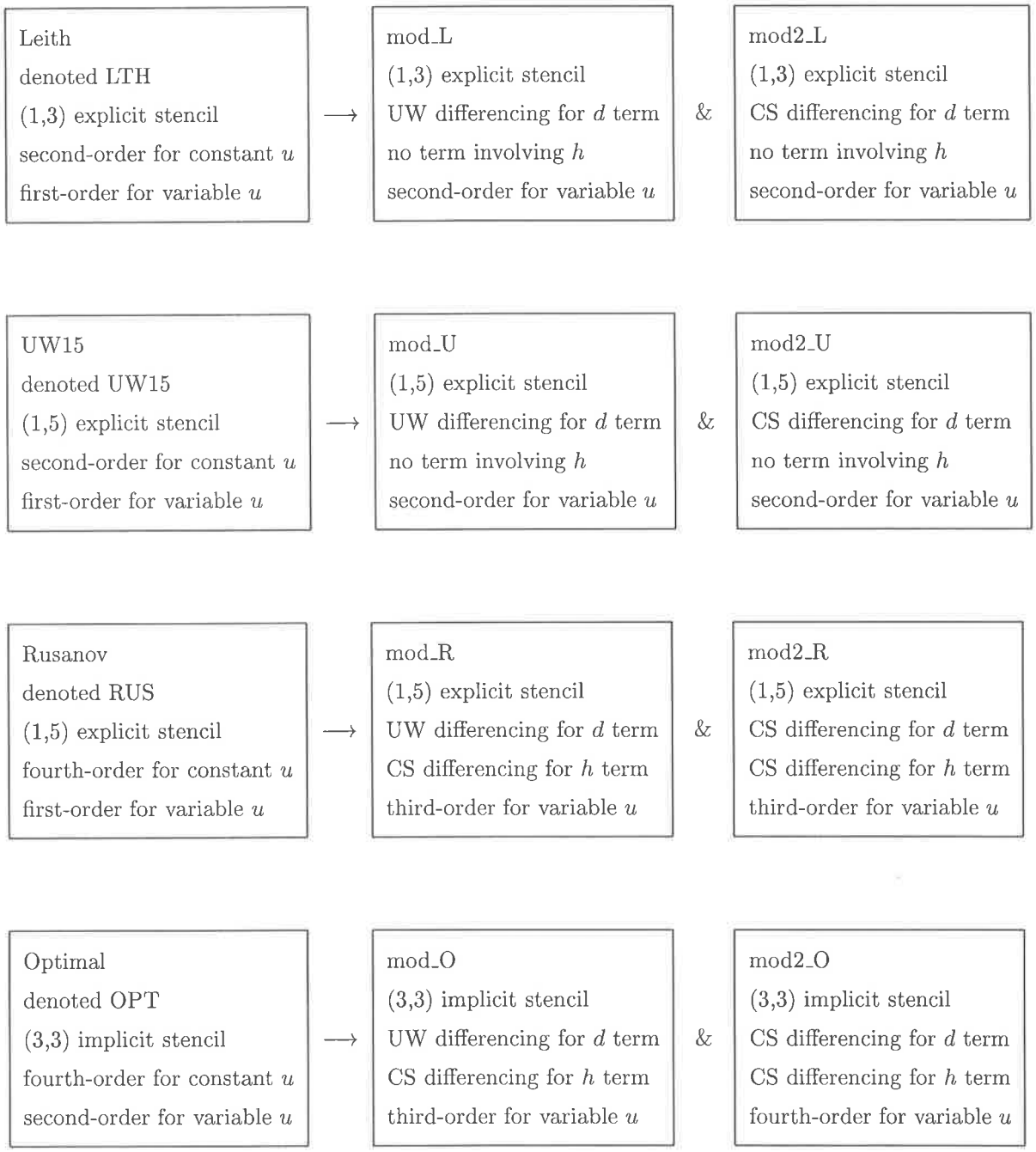

Figure 3.1: Synopsis of the properties of the methods.

Note that the variable coefficients are evaluated at (x_j, t_n) for all the explicit methods, whereas for the implicit methods they are evaluated at $(x_j, t_{n+1/2})$.

LTH	$E_{\mathrm{T}} = \Delta t \xi_1 \dfrac{\partial \hat{\tau}}{\partial x} + O\{2\}$: first-order wave-speed error
UW15	$E_{\mathrm{T}} = \Delta t \xi_1 \dfrac{\partial \hat{\tau}}{\partial x} + O\{2\}$: first-order wave-speed error
RUS	$E_{\mathrm{T}} = \Delta t \xi_1 \dfrac{\partial \hat{\tau}}{\partial x} + (\Delta t)^2 \xi_2 \dfrac{\partial \hat{\tau}}{\partial x} + (\Delta t)^2 \xi_3 \dfrac{\partial^2 \hat{\tau}}{\partial x^2} + O\{3\}$: first and second-order wave-speed errors and second-order amplitude-error (numerical diffusion)
OPT	$E_{\mathrm{T}} = (\Delta t)^2 \xi_4 \dfrac{\partial \hat{\tau}}{\partial x} + (\Delta t)^2 \xi_5 \dfrac{\partial^2 \hat{\tau}}{\partial x^2} + O\{4\}$: second-order wave-speed error and second-order amplitude error (numerical diffusion)

Table 3.1: Dominant error terms for the base methods where $\xi_1 \ldots \xi_5$ are weights.

The coefficients of the derivatives have been replaced by the weights $\xi_1 \ldots \xi_5$, to retain clarity. When the error term involving ξ_j is removed by the modification procedure, a new error term is introduced. The new terms are given in Table 3.2. Note that wherever "ξ_j" is read, the entire term involving "ξ_j" is implied.

It is not necessarily the case that any of the new errors are amongst the dominant error terms of the modified equation. For example, the mod2_L modification, in which the first-order error of the base method has been replaced by a third-order error, is still generally second-order because of the presence of the second-order term in the truncation error of the base method.

Clearly, there is the potential for the new error terms to interact with the old higher-order ones, which were not modified in any way. These interactions may act to reduce, but could possibly magnify, any existent wave or amplitude errors. It is difficult to quantify these interactions because of the variability of the velocity field, but it would be fair to say that any such effects are subtle in comparison to the magnitude of the dominant wave or amplitude errors of the base methods.

The stability of the methods was determined by conducting a local von Neumann stability analysis on the frozen coefficient problem (Strikwerda 1989, Hirsch 1990), yielding a necessary condition for the stability of the variable coefficient problem (Richtmyer and Morton 1967, Hirsch 1990).

mod_L	ξ_1 replaced by second-order amplitude error
mod2_L	ξ_1 replaced by third-order wave-speed error
mod_U	ξ_1 replaced by second-order amplitude error
mod2_U	ξ_1 replaced by third-order wave-speed error
mod_R	ξ_1 and ξ_2 replaced by third and fourth-order wave-speed errors respectively ξ_3 replaced by fourth-order amplitude error
mod2_R	ξ_1 and ξ_2 replaced by third and fourth-order wave-speed errors respectively (half magnitude of mod_R), ξ_3 replaced by fourth-order amplitude error
mod_O	ξ_4 replaced by third-order amplitude error ξ_5 replaced by fourth-order amplitude error
mod2_O	ξ_4 replaced by fourth-order wave-speed error ξ_5 replaced by fourth-order amplitude error

Table 3.2: Errors introduced by modification procedure.

By considering various base methods with different stencils, it was possible to develop several new modified methods with convergence rates ranging from two to four. It is desirable to have a broad range of methods available for several reasons. We have seen that many schemes require supplementary methods to provide values near the boundaries. By having another scheme with a different stencil, but equivalent order, this problem can be resolved without a reduction in convergence. In practice, the accuracy of the solution technique should match that of the input data. Using a less accurate scheme means that information will be lost in the solution process, while using a more accurate one may be more time consuming, but cannot add significantly to the overall accuracy of the solution.

Although we have gained some insight into the relative accuracy of the schemes by comparing their truncation errors, numerical tests are required to illustrate their performance in modelling certain aspects of advection, such as amplitude and wave speed. Numerical tests will also provide an indication of the computational time required by each scheme to produce a result. In Chapter 4, the performance of the schemes will be investigated when applied to the benchmark test problem of advecting a wavetrain of Gaussian pulses at a velocity which varies in both space and time. Because both the initial condition and velocity profile considered are smooth functions, it will be possible to verify the improved orders of convergence of the new modified methods. Additionally, the tests will show that, in general, high-order schemes are more accurate and more efficient than low-order ones.

Chapter 4

Numerical Tests for One-Dimensional Non-Conservative Advection

Numerical tests are required to validate models. A good numerical scheme for advection is one which can propagate steep fronts for a large range of Courant numbers with as little numerical diffusion as possible, with no introduction of oscillations or negative values, and requiring a minimal amount of computational effort. The accuracy of numerical schemes may be assessed by using analytical solutions, successive grid refinement or by comparing numerical schemes (Zoppou and Knight 1997b).

Few analytical solutions exist for variable coefficient problems, and those that do are often complicated, or have limited practical relevance (Zoppou and Knight 1997b). In the absence of a suitable analytical solution, successive grid refinement may be used. This technique, which will be used in Section 4.1, has the disadvantage that a very fine grid is required to obtain a solution accurate enough to be considered 'exact.' This is not only computationally expensive, but also risks the introduction of round-off errors into the numerical solution. In fact, after a certain point, further refinement of the grid will no longer reduce the error, because the round-off error has become comparable in size to the discretization error (May and Noye 1984). In Section 4.2, an analytical solution is developed, which is used in Section 4.3, to determine how the accuracy of the methods is affected by altering the size of the Courant number.

Comparing numerical schemes based on the various desirable properties mentioned above tends to be rather subjective. There is often a trade-off between factors such as overall accuracy versus simplicity and computational efficiency, amplitude errors versus wave speed errors, non-negativity versus high orders of convergence, and additionally, the imposition of stability restrictions.

4.1 Numerical Test

The aim of this section is to illustrate the improved orders of convergence of the modified methods, and to compare their performance with the application of the base methods to the solution of the non-conservative form of the advection equation. All of the methods described in Chapter 3 will be tested and examined in terms of their overall accuracy, computational efficiency, and the severity of any amplitude or wave speed errors introduced into the numerical solution.

4.1.1 Smoothness of the Data

It is known that the existence of a smooth solution is the underlying assumption of high-order methods, and that non-smoothness in the initial data (and thus the final solution) may result in reduced orders of convergence. For this reason, the test problems involve the propagation of a smooth initial profile subject to a smooth variable velocity. In other words, for a scheme of order p, at least the first $(p+1)$ derivatives of the initial condition and the velocity exist and are continuous.

4.1.2 Periodic Boundary Conditions

Periodic boundary conditions will be used for the numerical test discussed in this section. Periodic boundary conditions may be specified on a finite domain by repeating the computational domain of length L periodically so that all quantities, the solution, as well as the errors, can be expressed as a finite Fourier series over the domain 2L. Using periodic boundary conditions allows attention to be focussed on determining the order of convergence of the numerical scheme under consideration and is valid when the influence of the boundaries can be neglected or removed. It should be noted that the von Neumann stability analysis, which does not take the influence of boundary conditions into account, is based on the assumption of the existence of periodic boundary conditions (Hirsch 1990).

4.1.3 Initial Condition and Velocity Profile

The advection of a Gaussian pulse, initially given by $\hat{\tau}(x,0) = \exp\{-400(x-0.1)^2\}$, moving at a velocity of the form $u(x,t) = \kappa(0.5 + \sin^2 \pi x)\cos t$, is simulated using the numerical schemes. The coefficient κ is included to keep the maximum value of the Courant number fixed at $c\max = 1$ for both final times considered (refer to Table 4.1). The problem is solved on successively finer grids, so that the grid number ranges from $J = 50$ to 10000, and the number of time steps is set equal to the grid number, so that $N = J$. The solution is sought after a quarter cycle, that is at time $T = \pi/2$, and after one cycle.

No exact solution is known to the problem after a quarter cycle, so the exact solution is taken to be the result of the most accurate method solved on the finest grid considered. In other words, the fourth-order implicit mod2_O method solved on the grid $J = 10000$ provides the exact solution.

Since the velocity profile is temporally cyclic, the pulse is returned to its original location with its initial shape every half cycle. Hence, after one cycle, the exact solution is given by the initial condition.

4.1.4 Implementation

The numerical test described in Section 4.1 will be referred to as the Gauss test. A summary of the data required to implement the Gauss test is given in Table 4.1.

The explicit methods are implemented in a trivial fashion, with the single unknown at the new time level $(n + 1)$ being given explicitly from the known values at the old time level n.

The implicit methods require the solution of a periodic tridiagonal system of linear algebraic equations each time step. A slightly modified form of the Thomas algorithm has been developed for the solution of periodic tridiagonal systems. This algorithm can be found in Noye (1984a) and in Hirsch (1990).

The programs were written in Fortran 90 and run on a Toshiba Tecra 8000.

4.1.5 Results

The results of applying the Gauss test with $J = 50$ and $J = 100$ to a final time of a quarter cycle are shown in Figures 4.1 and 4.2. The graphical results after one cycle are not presented because at this final time the pulse is too narrow to compare the methods. The main features that can be isolated from the diagrams are how well the scheme models the amplitude and the wave speed of the pulse.

Amplitude errors are seen as the difference between the true height of the pulse and the height of the numerical solution. For this numerical test, the exact height of the pulse is equal to one. Wave speed errors are seen as a displacement of the peak of the numerical solution relative to the true peak position. An error in the wave speed of the scheme can cause spurious oscillations and unrealistic negative values to be introduced into the numerical solution, which can also be detected in the graphical results.

There are two possible sources for any loss in peak amplitude seen in the numerical results. The first cause is the introduction of amplitude errors (such as artificial diffusion) into the numerical solution by the scheme. The second possibility is the result of the dispersing (or spreading) of the pulse into different waves due to different Fourier components in the initial condition being propagated with different speeds. Therefore, the amplitude of a method with a dominant wave speed error, may also be poor.

1.	Initial condition $\hat{\tau}(x,0) = \exp\{-400(x-0.1)^2\}$
2.	Periodic boundary conditions $\hat{\tau}(1+x,t) = \hat{\tau}(x,t)$
3.	Courant number $c = u\Delta t/\Delta x = uTJ/N$
4.	cmax $= 1$
5.	$u(x,t) = \kappa(0.5 + \sin^2 \pi x)\cos t$
6.	$T = \pi/2$, $\kappa = 4/3\pi$, $u$max $= 2/\pi$: quarter cycle
7.	$T = 2\pi$, $\kappa = 1/3\pi$, $u$max $= 1/2\pi$: one cycle
8.	$N = J$ with $J = 50, 100, 200, \ldots, 5000$
9.	exact solution when $T = \pi/2$ is the solution of mod2_O when $J = 10000$
10.	exact solution when $T = 2\pi$ is the initial condition

Table 4.1: Data for the Gauss Test.

Leith's method and the mod_L and mod2_L methods are compared for $J = 50$ in the top left diagram of Figure 4.1. The solutions exhibit spurious trailing oscillations and unrealistic negative values, indicative of poor wave speed modelling. Spurious oscillations can develop if the scheme transmits the short wavelength Fourier components in the initial condition at different speeds to the long wavelength components (Noye 1984b). These oscillations grow unless they are damped by the scheme.

It is apparent that the peak of the base method is better aligned to the exact solution and exhibits less damping than either modified method. Table 4.2 (see Section 4.1.6) shows that Leith is only more accurate than mod_L and mod2_L when $J = 50$. Because of their higher convergence, the accuracy of both modified methods improves more rapidly than that of the base method as the grid is refined.

Table 4.4 shows that the amplitude error introduced by the mod2_L method is smaller than that introduced by the mod_L method. That is, the solution of the mod2_L method is less damped than that of the mod_L method (top left diagram in Figure 4.1). This is explained in Table 3.2; the mod2_L method uses second-order centered space differencing in its modification procedure, which is more accurate than

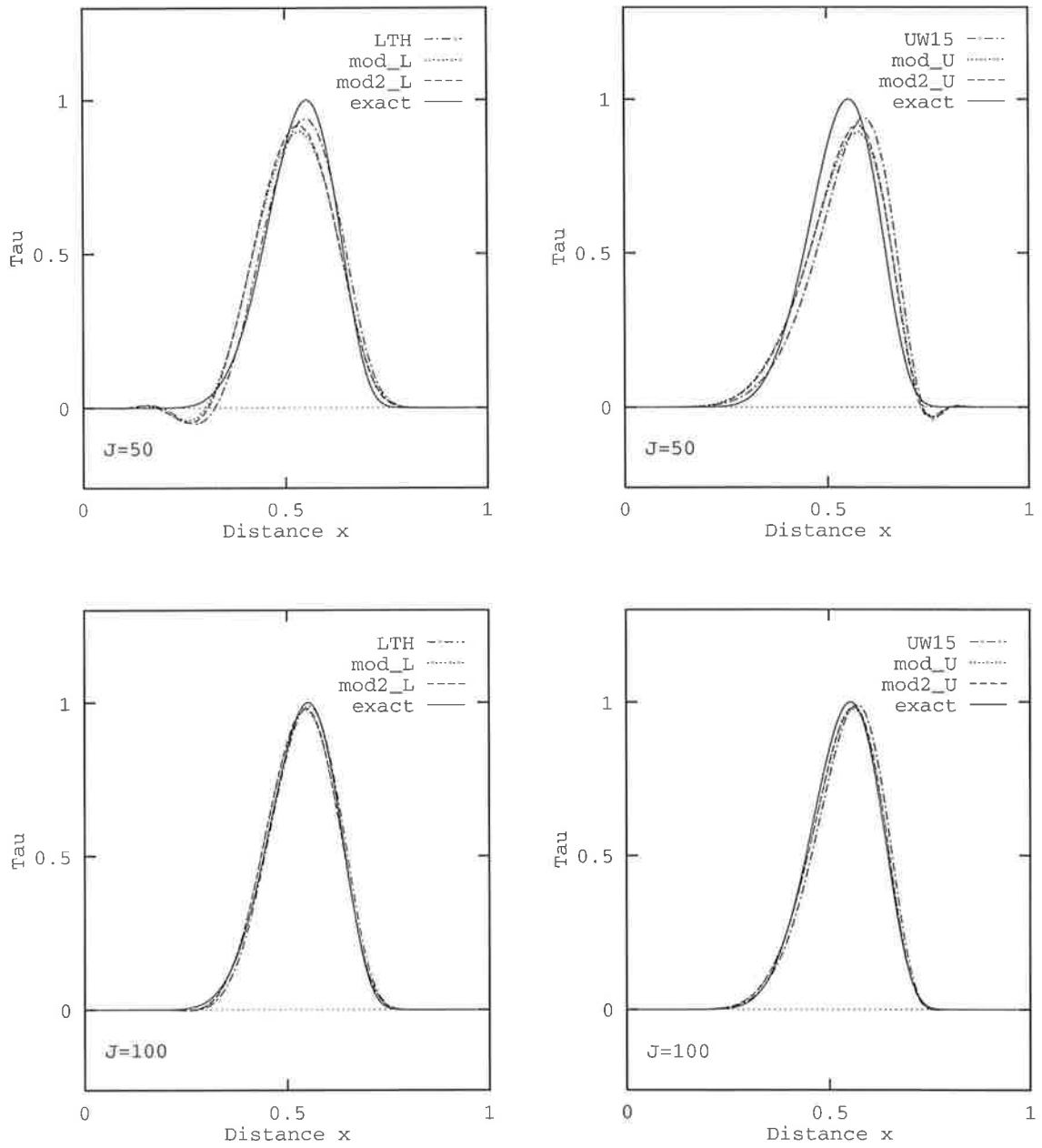

Figure 4.1: The top diagrams compare Leith's method and its modifications to the UW15 method and its modifications for $J = 50$. The bottom diagrams are for $J = 100$.

the first-order upwind differencing used for the mod_L method. The use of upwind differencing is known to damp the numerical solution (Noye 1984b). A comparison of the UW15, mod_U and mod2_U methods in the top right diagram of Figure 4.1, shows that the height of the pulse advected by the base method is closer to the exact value, but the pulses advected by the modified methods are better aligned to the true solution. All three methods advect the pulse too rapidly and a leading oscillation is generated. This oscillation is slightly worse for the base method (see Table 4.5 which gives the minimum value of the

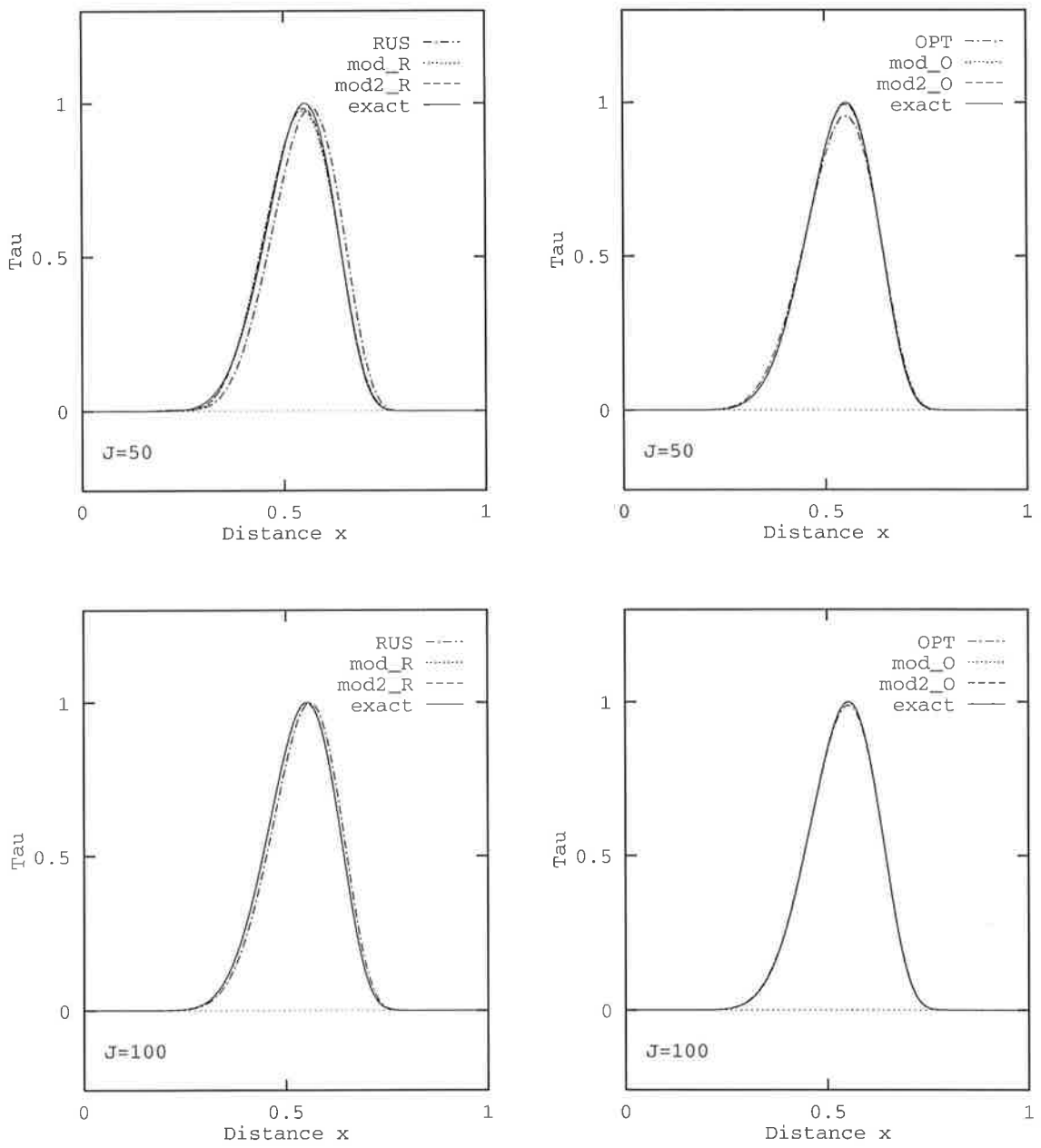

Figure 4.2: The top diagrams compare Rusanov's method and its modifications to the OPT method and its modifications for $J = 50$. The bottom diagrams are for $J = 100$.

numerical solution for each scheme). Because of the centered space differencing used in its modification procedure, the solution of the mod2_U method is less damped than that of the mod_U method (see top right diagram in Figure 4.1). Since the UW15 method was modified in the same way as Leith's method, any differences seen in the results of their respective modified methods are because of intrinsic differences between the base schemes, rather than as a consequence of the modification procedure. The top diagrams in Figure 4.1 show that the leading oscillation generated by the UW15 method is spread over less grid points than the trailing oscillation produced by Leith's method, but Leith's solution is

better aligned to the exact solution. When the grid is refined to $J = 100$, the results of all methods improve (see bottom diagrams in Figure 4.1); particularly noticeable is the improved alignment of the modified solutions to the exact solution. In all cases, the spurious oscillations are no longer visible, but the solutions of the modified methods are still more damped than those of the base methods.

The top left diagram in Figure 4.2 shows that the solution of Rusanov's method leads the exact solution with a reduced peak height. This is consistent with the fact that the dominant errors of this base method include first and second-order wave speed errors and second-order artificial diffusion (refer to Table 3.1). The spurious oscillations and unrealistic negative values introduced by the schemes in Figure 4.1 are not evident in these test results.

Table 3.2 indicates that the mod_R and mod2_R methods do not introduce any first or second-order errors into the numerical solution. The dominant errors affect their wave speed. The top left diagram in Figure 4.2 shows that, because these errors are small, the numerical solutions are closely aligned to the exact solution. The only visible difference between the two modified methods is that the mod2_R method models the height of the pulse slightly better than the mod_R method.

The implicit OPT method introduces second-order wave and amplitude errors into the numerical solution. Since it introduces no first-order wave speed errors into the numerical solution, its solution should be better aligned to the exact solution than that of the explicit base methods. This is verified by comparing the solutions of the base methods displayed in Figures 4.1 and 4.2. Although the OPT method gives smaller amplitude errors than Leith's method or the UW15 method for the solutions displayed, Rusanov's method models the height of the pulse even better (see also Table 4.4).

Table 3.2 shows that the dominant error of the mod_O method includes third-order numerical diffusion, and that the mod2_O method introduces fourth-order amplitude and wave speed errors into its solution. The top right diagram in Figure 4.2 shows that these errors are so small that only a slight loss in peak height is seen when $J = 50$. The bottom diagrams in Figure 4.2 show that when the grid is refined, the results of the third and fourth-order methods are indistinguishable from the exact solution, but the pulse advected by Rusanov's method still lies ahead of the exact solution, and a slight loss in amplitude is observed for the OPT method.

Peak clipping is a problem that often arises in the numerical simulation of advection. Unless the peak is advected in an integral number of grid points, information is lost, giving the appearance that the peak has been flattened (Steinle 1994). This effect is more prominent for coarse grids, where insufficient data points are available to represent the solution accurately. Most of the schemes display some form of peak clipping when $J = 50$. However, this is no longer a dominant feature of the results when $J = 100$.

It is possible to recover the peak by using polynomial interpolation near the maximum of the pulse, and the peak position can be found by inverse interpolation (Steinle 1994). The order of the polynomial is usually chosen to match that of the scheme. So that the performance of the schemes at hand can be assessed, interpolation to capture the peak is not considered.

Because the pure advection equation governs the way in which a passive quantity is carried along by a fluid in the absence of diffusion, sources or sinks, the maximum and minimum values of the finite-difference solution should be identical to those of the initial condition. For this numerical test, the maximum and minimum values of the solution should therefore be one and zero, respectively.

Table 4.4 gives the error in amplitude for the methods after a quarter cycle. Except for the UW15 method, for which the amplitude is greater than one for grid numbers exceeding 500, all methods have the property that the height of the pulse approaches one from below as the grid is refined. The modified implicit schemes give the smallest amplitude errors. The explicit mod2_R method also performs well, especially when the grid number is greater than 50. The modified second-order explicit formulae, however, are always outperformed by their base counterparts.

The schemes should not introduce negative values into the numerical solution, so the values $|\tau_{min}|$ are compared in Table 4.5. Values smaller than 1e-14 are taken to be zero, and are represented by a dash in the tables. The implicit methods usually outperform the explicit ones, giving results which are very close to being non-negative. As the grid is refined, it can be seen from Table 4.5 that $\tau_{min} \rightarrow 0$ most rapidly for the mod2_O method. The explicit scheme for which $\tau_{min} \rightarrow 0$ fastest is the mod_R method.

To summarize, of the three explicit base schemes considered, Rusanov's method gives the best amplitude, while Leith's method positions the pulse the best. The implicit OPT method aligns the pulse more accurately than Leith's method and, in addition, introduces no detectable negative values or oscillations into the numerical solution. Rusanov's method captures the height of the pulse better than the OPT method, but its numerical solution leads the true solution considerably. The second-order explicit modified methods do not capture the height of the pulse as well as their respective base methods.

The third and fourth-order modified methods advect the Gaussian pulse most accurately. When the grid is refined, there is an obvious improvement in the alignment and peak resolution of all methods. This is more noticeable for the solutions of the modified methods, which are usually better aligned than the solutions of their base counterparts, and which become indistinguishable from the exact solution in the case of the third and fourth-order methods.

4.1.6 Further Comparisons

The rms error in the finite-difference solution relative to the exact solution was calculated using all the grid points at the final time. The exact solution after a quarter cycle was taken to be the result given by the mod2_O method when $J = 10000$. The rms errors for all methods are given in Table 4.2 and the average cpu time over 20 consecutive runs of the algorithm is given in Table 4.3.

From the values in Table 4.2, the methods can be classified in terms of their overall accuracy. The results show that the implicit OPT method, which is second-order convergent, is the most accurate base method. Of the explicit base methods, all of which are first-order, Leith's method is the most accurate,

followed by Rusanov's method and the UW15 method. Although Leith's method is more accurate than Rusanov's method, the latter scheme may be preferred, since it doesn't appear to introduce oscillations or unrealistic negative values into the numerical solution (compare Figures 4.1 and 4.2.)

The OPT method is the most accurate second-order method. Unless $J = 50$, when the mod_L method is more accurate, the mod2_L method is the most accurate explicit second-order scheme. The mod_L method is more accurate than the mod2_U method unless $J = 5000$, and the mod_U method is the least accurate second-order method. Leith's method is more accurate than the mod_L and mod2_L methods when $J = 50$, and more accurate than the mod_U and mod2_U methods when $J = 50$ or 100.

Of the three third-order methods developed for variable coefficients, the implicit mod_O method is the most accurate, followed by the explicit mod2_R and mod_R methods (see Table 4.2). The fourth-order implicit mod2_O method is the most accurate method overall, except for when $J = 50$ and 100, when it is outperformed by the third-order mod_O method (see Table 4.2).

From the above discussion we may conclude that, generally speaking, implicit methods are more accurate than explicit methods of the same order, and the higher the order of convergence, the more accurate the method. Table 4.3 can be used to classify the methods in terms of the computational effort required to obtain the numerical solution.

For all methods, it can be seen that doubling the grid number, which also doubles the number of time steps, increases the computational time four-fold. Leith's method is the fastest method overall. This is because of the simplicity of the coefficients in its (1,3) computational stencil. The mod2_L method requires about 2.25 times longer to run than Leith's method for each grid number. The mod_L method takes a little longer to run than the mod2_L method (refer to Table 4.3).

The computational times for the UW15 method and its modifications are a little longer than for Leith's method and its modifications. This is because they have (1,5) stencils. Rusanov's method has more complicated coefficients than the UW15 method and so requires more computational time. The mod_R method takes about 3.71 times longer to run for each grid number than Rusanov's method. It has complicated coefficients and requires the evaluation of two correction factors at each point in the domain.

Although the implicit OPT method is the slowest base method, it requires less cpu time than the explicit modified schemes. The modified implicit methods require the longest time to run. The mod_O method requires about 3 times longer to run than the OPT method for each grid, while the mod2_O method is a little faster than the mod_O method (see Table 4.3).

It is recalled that a periodic variant of the Thomas algorithm for tridiagonal systems was used to solve the implicit schemes. This algorithm requires a total of $12J$ multiplication and division operations for a system of J equations in J unknowns (Noye 1984a). By contrast, the standard Thomas algorithm for tridiagonal systems requires only $5J$ such operations for a system of the same size (Hirsch 1990).

These results lead to the conclusion that the simpler the computational stencil and coefficients of the scheme, the smaller the cpu time. Specifically, the base schemes with (1,3) stencils were faster than those with (1,5) and (3,3) stencils. Furthermore, it is observed that the least accurate methods tend to be the fastest. There is clearly a trade-off between speed and accuracy. For this reason, it may be more appropriate to judge the methods in terms of their computational efficiency, rather than considering their speed and their accuracy separately. This will be discussed below.

Convergence can be illustrated by plotting the rms error against the grid number on logarithmic scales. The order of convergence can be found from the slope of the line of best fit to the data. The equation of the line of best fit is found by using the method of least squares. Accuracy in relation to computational efficiency is studied by graphs showing the rms error versus cpu time using logarithmic scales. Similarly, the cpu time is plotted against the grid number so that just the speed of the methods can be compared.

Graphs showing the convergence, efficiency and cpu times of the methods after a quarter cycle are given in Figures 4.3 and 4.4. The main information given by these plots is the comparative performance of each modified method to its base method, and an indication of how well the two modified methods of each base scheme perform with respect to each other. It is observed that, although the modified methods always required more time to run than their base counterpart, they were almost always more accurate and more efficient to use.

The top left diagram in Figure 4.3 shows that Leith's method is more accurate than the mod_L and mod2_L methods when $J = 50$. The middle left diagram in Figure 4.3 indicates that it is also more efficient than these methods when $J = 50$ and 100. However, if the grid is refined further, both modified methods outperform Leith's method quite considerably in terms of accuracy and efficiency. The second-order explicit methods cannot be separated from the diagrams in Figure 4.3, however Tables 4.2 and 4.3 have already been used to compare the speed and accuracy of these methods.

Although the implicit OPT method expends more time than the other base methods, this additional cost is more than compensated by its superior accuracy. Not only is the OPT method the most efficient base method, it is also the most efficient second-order method. This is because it is the fastest and most accurate second-order method (see Tables 4.2 and 4.3).

The superior performances of the third and fourth-order modified methods over the base methods is evident in Figure 4.4. The third-order mod2_R method is clearly the most accurate and efficient explicit scheme, while the fourth-order implicit mod2_O method is generally the most accurate and efficient method overall. The third-order mod_O method is marginally more accurate when $J = 50$ and 100, but is otherwise outperformed by the mod2_O method (see Table 4.2).

It is observed that for this numerical test, the third-order mod2_R method actually gave results which are accurate to fourth-order for grid numbers not exceeding 500. The mod_O method also shows fourth-order convergence, but only for grid numbers not exceeding 200. Otherwise, the lines of convergence for all methods are very close to being straight lines (compare top diagrams in Figures 4.3 and 4.4).

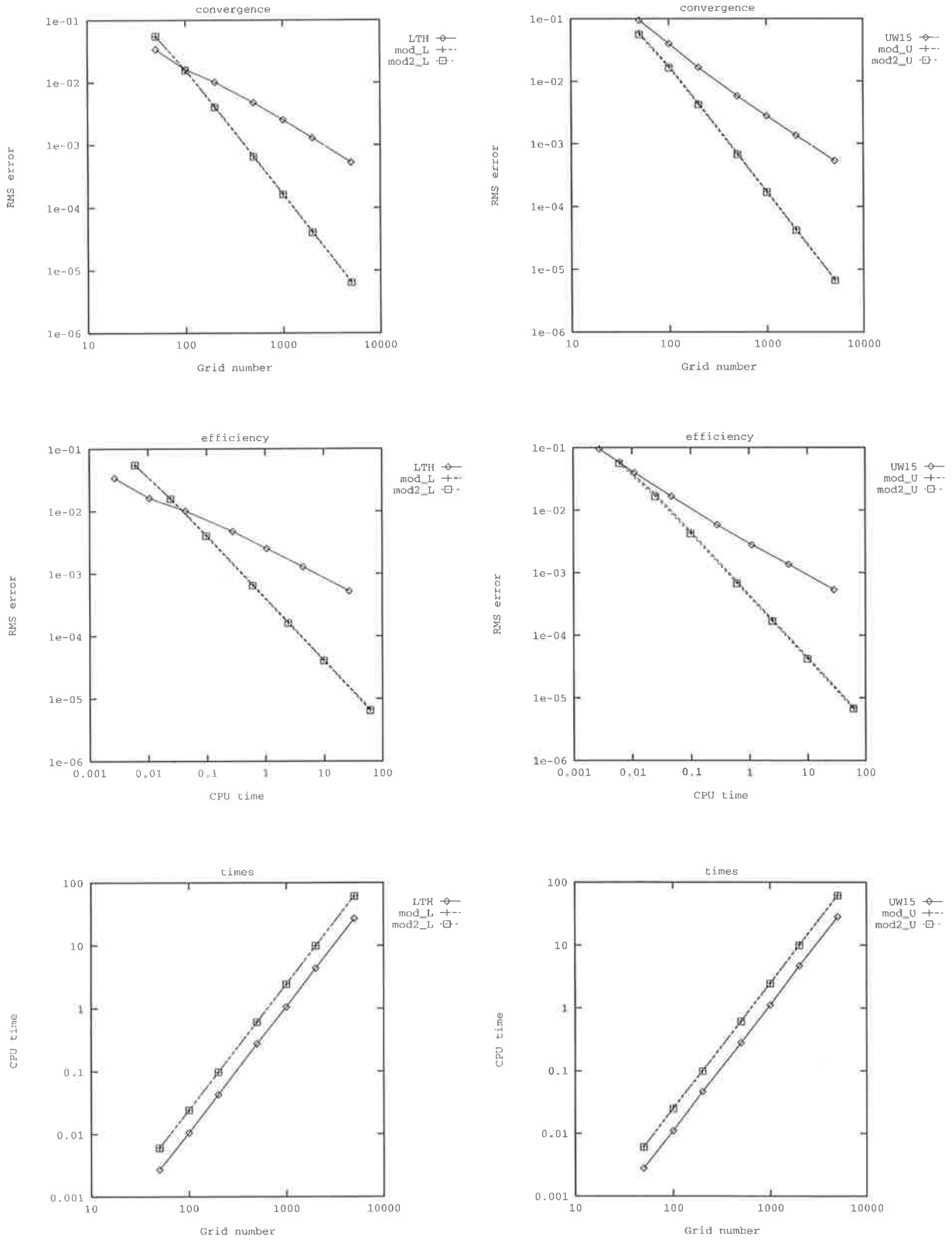

Fig 4.3: Convergence, efficiency and times shown for LTH and its modifications on the left and UW15 and its modifications on the right.

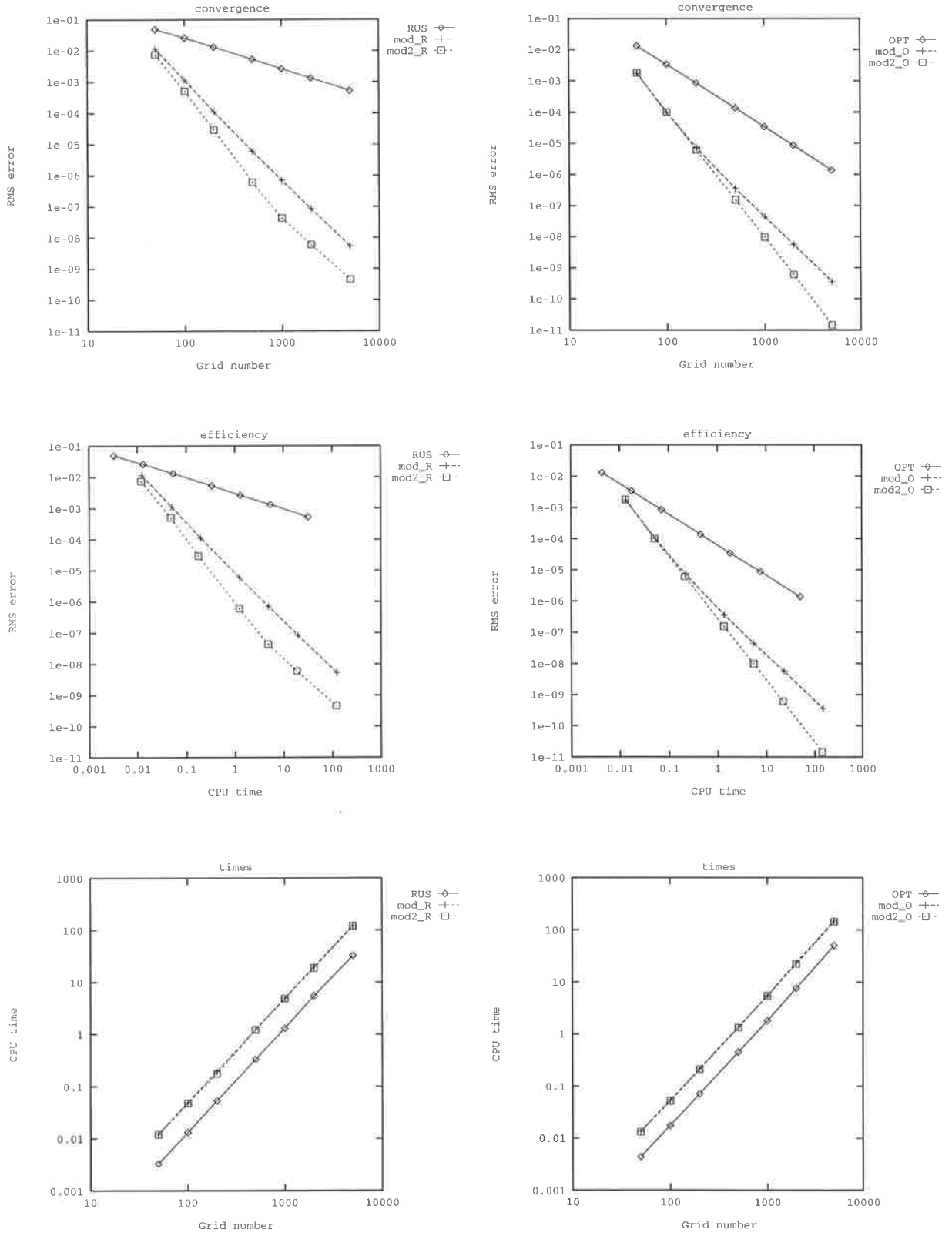

Fig 4.4: Convergence, efficiency and times shown for RUS and its modifications on the left and OPT and its modifications on the right.

J	LTH	UW15	RUS	OPT
50	3.41e−02	9.53e−02	4.88e−02	1.32e−02
100	1.63e−02	3.99e−02	2.62e−02	3.39e−03
200	1.02e−02	1.67e−02	1.32e−02	8.54e−04
500	4.78e−03	5.84e−03	5.29e−03	1.37e−04
1000	2.52e−03	2.78e−03	2.64e−03	3.43e−05
2000	1.29e−03	1.36e−03	1.32e−03	8.57e−06
5000	5.24e−04	5.34e−04	5.29e−04	1.37e−06
J	mod_L	mod_U	mod_R	mod_O
50	5.51e−02	5.98e−02	1.15e−02	1.74e−03
100	1.62e−02	1.75e−02	1.14e−03	9.85e−05
200	4.17e−03	4.44e−03	1.12e−04	7.18e−06
500	6.69e−04	7.02e−04	6.02e−06	3.55e−07
1000	1.67e−04	1.74e−04	7.05e−07	4.36e−08
2000	4.18e−05	4.34e−05	8.52e−08	5.51e−09
5000	6.69e−06	6.93e−06	5.34e−09	3.57e−10
J	mod2_L	mod2_U	mod2_R	mod2_O
50	5.53e−02	5.61e−02	7.46e−03	1.81e−03
100	1.58e−02	1.65e−02	5.06e−04	9.96e−05
200	4.02e−03	4.22e−03	2.93e−05	6.06e−06
500	6.44e−04	6.72e−04	6.05e−07	1.54e−07
1000	1.61e−04	1.68e−04	4.31e−08	9.62e−09
2000	4.02e−05	4.18e−05	5.96e−09	6.00e−10
5000	6.42e−06	6.68e−06	4.58e−10	1.44e−11

Table 4.2: RMS errors after a quarter time cycle when the Gauss test is applied to the one-dimensional non-conservative advection problem.

J	LTH	UW15	RUS	OPT
50	2.70e−03	2.80e−03	3.30e−03	4.40e−03
100	1.05e−02	1.10e−02	1.32e−02	1.76e−02
200	4.28e−02	4.67e−02	5.36e−02	7.17e−02
500	2.76e−01	2.80e−01	3.33e−01	4.49e−01
1000	1.06e+00	1.11e+00	1.30e+00	1.80e+00
2000	4.36e+00	4.70e+00	5.44e+00	7.59e+00
5000	2.69e+01	2.83e+01	3.23e+01	5.03e+01
J	mod_L	mod_U	mod_R	mod_O
50	6.05e−03	6.10e−03	1.24e−02	1.35e−02
100	2.42e−02	2.58e−02	4.97e−02	5.38e−02
200	9.78e−02	1.01e−01	1.96e−01	2.17e−01
500	6.12e−01	6.23e−01	1.23e+00	1.36e+00
1000	2.45e+00	2.50e+00	4.91e+00	5.44e+00
2000	9.83e+00	1.02e+01	1.96e+01	2.33e+01
5000	6.15e+01	6.27e+01	1.22e+02	1.49e+02
J	mod2_L	mod2_U	mod2_R	mod2_O
50	6.00e−03	6.05e−03	1.20e−02	1.32e−02
100	2.41e−02	2.48e−02	4.78e−02	5.22e−02
200	9.72e−02	9.83e−02	1.76e−01	2.11e−01
500	6.05e−01	6.09e−01	1.22e+00	1.33e+00
1000	2.42e+00	2.44e+00	4.82e+00	5.43e+00
2000	9.81e+00	9.89e+00	1.87e+01	2.20e+01
5000	6.09e+01	6.12e+01	1.20e+02	1.44e+02

Table 4.3: CPU times after a quarter time cycle when the Gauss test is applied to the one-dimensional non-conservative advection problem.

J	LTH	UW15	RUS	OPT
50	6.29e−02	6.21e−02	1.47e−02	4.36e−02
100	1.17e−02	7.95e−03	6.80e−04	1.09e−02
200	1.56e−03	6.54e−04	3.66e−04	2.88e−03
500	1.62e−04	3.83e−05	2.79e−05	4.67e−04
1000	3.79e−05	-1.22e−05	3.69e−06	1.05e−04
2000	9.62e−06	-2.85e−06	2.09e−06	2.70e−05
5000	1.24e−06	-5.89e−07	2.08e−07	4.58e−06
J	mod_L	mod_U	mod_R	mod_O
50	1.02e−01	1.06e−01	2.39e−02	5.60e−03
100	2.56e−02	2.27e−02	1.60e−03	6.37e−04
200	5.50e−03	4.48e−03	3.83e−04	3.11e−04
500	8.02e−04	6.47e−04	5.19e−05	5.15e−05
1000	1.90e−04	1.42e−04	1.06e−06	1.07e−06
2000	4.74e−05	3.51e−05	9.57e−07	9.64e−07
5000	7.68e−06	5.86e−06	4.14e−07	4.15e−07
J	mod2_L	mod2_U	mod2_R	mod2_O
50	8.32e−02	8.37e−02	1.69e−02	4.66e−03
100	1.85e−02	1.53e−02	9.33e−04	5.15e−04
200	3.55e−03	2.51e−03	3.17e−04	2.96e−04
500	4.85e−04	3.29e−04	5.04e−05	5.05e−05
1000	1.11e−04	6.26e−05	9.05e−07	9.49e−07
2000	2.75e−05	1.53e−05	9.42e−07	9.49e−07
5000	4.51e−06	5.58e−07	4.13e−07	4.14e−07

Table 4.4: Amplitude errors after a quarter time cycle when the Gauss test is applied to the one-dimensional non-conservative advection problem.

J	LTH	UW15	RUS	OPT
50	5.18e−02	4.08e−02	4.24e−04	3.37e−05
100	2.30e−03	1.36e−03	1.85e−06	1.55e−08
200	3.22e−06	1.14e−05	4.26e−10	2.10e−14
500	1.86e−13	2.87e−10	–	–
1000	–	–	–	–
2000	–	–	–	–
5000	–	–	–	–
J	mod_L	mod_U	mod_R	mod_O
50	4.23e−02	3.12e−02	5.38e−04	2.46e−05
100	2.10e−03	1.10e−03	1.44e−06	1.09e−07
200	3.02e−06	9.90e−06	3.45e−10	5.85e−09
500	5.65e−13	2.63e−10	–	3.41e−11
1000	–	–	–	1.56e−12
2000	–	–	–	5.26e−14
5000	–	–	–	2.87e−14
J	mod2_L	mod2_U	mod2_R	mod2_O
50	4.90e−02	3.33e−02	5.01e−04	2.59e−05
100	2.29e−03	1.13e−03	1.85e−06	1.07e−08
200	3.24e−06	1.03e−05	4.40e−10	–
500	1.97e−13	2.69e−10	–	–
1000	–	–	–	–
2000	–	–	–	–
5000	–	–	–	–

Table 4.5: Values $|\tau_{min}|$ after a quarter time cycle when the Gauss test is applied to the one-dimensional non-conservative advection problem.

Method	$T = \pi/2$	$T = 2\pi$	Theory
LTH	0.89	1.05	1
mod_L	1.97	2.05	2
mod2_L	1.98	2.92	2
UW15	1.12	1.21	1
mod_U	1.98	2.15	2
mod2_U	1.98	2.86	2
RUS	0.99	1.02	1
mod_R	3.17	3.39	3
mod2_R	3.69	3.15	3
OPT	1.99	–	2
mod_O	3.31	3.01	3
mod2_O	4.04	–	4

Table 4.6: The orders of convergence for the Gauss test are given by the slope of the line of best fit to the data.

After one time cycle, the exact solution is given by the initial condition, and the rms errors are presented in Table B.1 of Appendix B. Since the numerical test was constructed in such a way that no additional computational effort is required to run the tests to a full cycle, the cpu times for this final time are not presented.

Table B.1 shows that the solutions given by the implicit OPT and mod2_O methods after one cycle are virtually identical to the exact solution. The errors given by these methods are at the limits of machine accuracy and may be taken to be zero. A possible explanation for this is that errors in the solution which have built up over the first half-cycle are compensated for during the second half-cycle by errors opposite in sign but equal in magnitude.

To verify this, the errors that accumulate after half a cycle and three quarters of a cycle were calculated. The results, which are not presented, showed that the errors grow to a maximum after a quarter cycle, after which they decline in the second quarter, so that exact results are again obtained after half a cycle. This pattern continues, with errors increasing in the third quarter and decreasing in the fourth quarter.

Apart from these remarkable results and the fact that the second-order mod2_L and mod2_U methods gave almost third-order accurate results after one cycle (see Table B.1), similar results to those after a quarter cycle are obtained, so the convergence plots are omitted.

4.1.7 Summary

The methods developed in Chapter 3 were used to simulate the advection of a Gaussian pulse, subject to a variable velocity field and periodic boundary conditions. The methods were examined in terms of their accuracy, efficiency, time expended to obtain a result, and the severity of any amplitude and wave errors introduced into the numerical solution.

The convergence of the schemes was illustrated by plotting the rms error against the grid number on logarithmic scales. The orders of convergence in Table 4.6 were obtained by performing a least squares fit to the data. These indicate that for this numerical test all methods gave close to or better convergence rates than expected from the theory. Since the OPT and mod2_O methods gave exact solutions after one cycle, their convergence rates at this final time are not presented.

Based on the values given in Table 4.6, it can be concluded that we have been successful in the aim to modify low-order methods for the numerical simulation of advection into high-order methods. Moreover, the modified high-order methods were shown to be almost always more accurate and efficient than their low-order base counterparts; particularly for fine grids and small time-steps.

Although implicit schemes may require more time to run, they can produce significantly more accurate results than explicit methods of the same order. It was seen that artificial oscillations were generated in the results of all the first and second-order explicit methods except for Rusanov's method. The implicit schemes introduced no detectable oscillations into the numerical solution and gave results which were virtually non-negative.

Overall, the most accurate explicit and implicit schemes were the third-order mod2_R and fourth-order mod2_O methods. These were usually also the most efficient methods. Both schemes propagated a smooth steep front without introducing any obvious oscillations into the solution. Each approximated the speed of the numerically simulated pulse successfully.

Of all the schemes considered, the mod2_O method introduced the least artificial diffusion into the solution. Regarding this property, the mod2_R method was amongst the best performing explicit schemes. Both methods were modified from low to high-order, and each can be recommended to give excellent results at both coarse and fine grid resolutions.

4.1.8 Discrete Profile

The test was repeated for the second-order explicit schemes, using a discrete velocity field $u_j^n = u(x_j, t_n)$, where u and the other parameters are given in Table 4.1. The correction factor d is approximated by (3.51). After a quarter cycle, the percentage change in rms error for these methods, compared to those given in Table 4.2, is only approximately 0.5% (see Table 4.7). After two cycles, the errors increased by about 3% for the mod2_L method, but less than by 2% for the other second-order explicit schemes.

J	mod_L	mod2_L	mod_U	mod2_U
50	0.54	0.36	-0.33	-0.54
100	0.62	0.63	-0.57	-0.61
200	0.48	0.50	-0.45	-0.71
500	0.60	0.47	-0.57	-0.45
1000	0.60	0.62	-0.58	-0.60
2000	0.48	0.50	-0.46	-0.48
5000	0.45	0.62	-0.58	-0.60
50	1.71	3.35	0.99	1.32
100	1.18	3.41	0.71	1.16
200	0.93	3.17	0.34	0.78
500	0.64	3.47	0.28	1.05
1000	0.13	3.32	0.12	0.93
2000	0.00	3.10	0.52	1.49
5000	0.34	3.46	0.00	1.05

Table 4.7: Percentage change in RMS error when a discrete velocity profile is used. Values at top are for a quarter cycle and bottom values are for one cycle.

4.1.9 Non-negativity

It has already been mentioned that numerical schemes should preserve the physical properties of the quantity under consideration. In particular, if non-negative initial and boundary values are specified, then the exact solution is also non-negative (Vreugdenhil and Koren 1993). For this reason, numerical schemes should not introduce unwanted (non-physical) negative values into the numerical solution. Negative values are caused by spurious oscillations, which are in turn created by poor wave speed modelling of the short wavelength Fourier components (Noye 1986).

Godunov's theorem (Godunov 1959) tells us that there is no linear scheme, with an order higher than one, which is free from oscillations. The simplest and least diffusive linear scheme which gives non-negative and non-oscillatory results is the first-order upwind method (Godunov 1959). Hence, it is possible to use the first-order upwind method to remove the negative values from the results given by the methods described in Chapter 3.

A way to achieve this is by testing the sign of the value τ_j^{n+1} obtained using the numerical scheme. If it is negative, the value is recalculated, this time using the first-order upwind method. This technique has been applied to all the numerical schemes using the Gauss test (Table 4.1) with $T = \pi/2$. Since negative values could only be seen in the results of the methods depicted in Figure 4.1 for $J = 50$, only the new results for these methods are shown below.

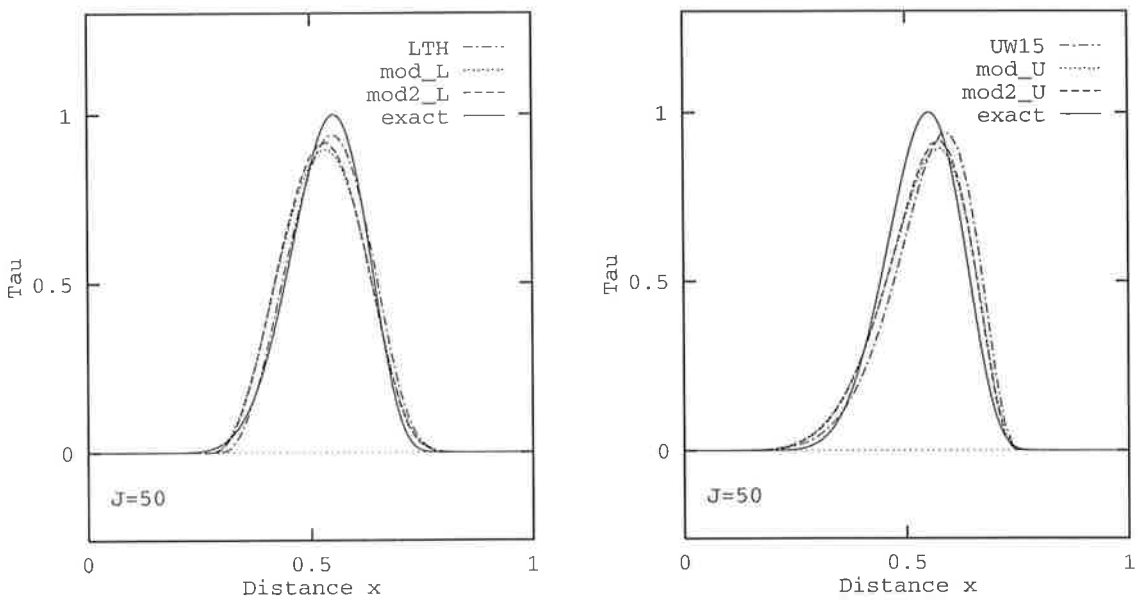

Figure 4.5: Upwind method is used to eliminate negative values in the Gauss test.

Other than the fact that the results are now non-negative and non-oscillatory, none of the other features of the solution change by using the first-order upwind method. Although the first-order upwind method is very diffusive, the height of the pulse is unaffected, because it is not used near the peak of the pulse. Even though the upwind method is only first-order, it is used only for a few grid points in the solution domain, and so the convergence rates given in Table 4.6 are unchanged.

Although using the first-order order upwind method to remove all spurious oscillations has been successful for this numerical test, this will not generally be the case. If instead of advecting a smooth front like the one considered in the Gauss test, the numerical methods are used to advect, for example, a square pulse, then oscillations may be generated along the entire top surface of the pulse (see for example Steinle and Morrow 1989). Such oscillations cannot be removed by the technique described above.

Research undertaken to develop non-oscillatory schemes includes using a linear filter (Shapiro 1970), or using polynomial approximation with the interpolation being done so as to avoid overshoots and undershoots as in the CIP method (Takewaki and Yabe 1987). The upwind method is used as the basis in the method of Smolarkiewicz (1983); some of the excess diffusion is removed by using a second corrective upwind step, which guarantees that the numerical solution remains positive. Other schemes include those of Van Leer (1974, 1977), and flux-corrected transport (Boris and Book 1973, 1976).

Most of these schemes have shortcomings. Linear filtering cannot remove all the negative values from the numerical results (Shapiro 1970), the CIP method is very diffusive (Steinle 1994), the method of Smolarkiewicz deforms and skews the solution (Vreugdenhil and Koren 1993), and the methods of Van Leer cannot differentiate between a spurious oscillation and a true maximum or minimum (Vreugdenhil and Koren 1993). The flux-corrected transport method has been used quite successfully to ensure positive, non-oscillatory solutions (see Steinle, Morrow and Roberts 1989) for a square wave test.

4.2 An Analytical Solution

In this section, an analytical solution is developed for the purpose of testing the schemes. Consider

$$\frac{\partial \hat{\tau}}{\partial t} + u(x,t)\frac{\partial \hat{\tau}}{\partial x} = 0, \tag{4.1}$$

with $u(x,t)$ in the separable form

$$u(x,t) = r(t)/s(x), \quad s(x) \neq 0 \tag{4.2}$$

then a solution of (4.1) may be sought in the separable form

$$\hat{\tau}(x,t) = X(x)T(t). \tag{4.3}$$

Substitution of (4.3) into (4.1) leads to

$$\frac{1}{r}\frac{T'}{T} = -\frac{1}{s}\frac{X'}{X}. \tag{4.4}$$

Since the expression on the left is a function of t, while the expression on the right is a function of x, both expressions must equal a constant K. This yields two separable first-order differential equations

$$\frac{T'}{T} = Kr \quad \text{and} \quad \frac{X'}{X} = -Ks, \tag{4.5}$$

which have solutions

$$T(t) = A\exp\{K\int r(t)dt\} \quad \text{and} \quad X(x) = B\exp\{-K\int s(x)dx\}, \tag{4.6}$$

in which A and B are constants. The general solution of (4.1) is thus given by

$$\hat{\tau}(x,t) = AB\exp\{K\int r(t)dt\}\exp\{-K\int s(x)dx\}. \tag{4.7}$$

The methods will be compared for the case $r = \cos(\pi t/2)$ and $s = e^x$, when (4.7) becomes

$$\hat{\tau}(x,t) = \exp\left\{\frac{2}{\pi}\sin\left(\frac{\pi t}{2}\right) - e^x\right\}, \tag{4.8}$$

where the constants have been set to $K = 1$ and $AB = 1$.

4.3 Parameter Analysis

The aim of this section is to determine how the accuracy and the efficiency of the schemes are affected if the size of the Courant number is varied. For constant velocity, an analytical relationship between a scheme's accuracy and the size of the Courant number can be established by examining the leading term in the truncation error of the MEPDE (cf. Section 3.3). Assuming that the high-order terms in the truncation error can be ignored, the leading error term gives an indication of the performance of the scheme with respect to the size of the Courant number. Several three-point two-level methods for the constant coefficient advection equation are compared in this way in Noye (1986).

For variable coefficients, the leading term in the truncation error of the MDPDE of the scheme depends on the derivatives of the velocity, rather than on the Courant number itself, as is the case for constant coefficients. Consequently, the scheme's accuracy with respect to the Courant number cannot be quantified in the way described above. Instead, the methods can be compared in a numerical test for which an exact solution is known.

All of the methods described in Chapter 3 (refer to synopsis Figure 3.1) are compared with the analytical solution given in Section 4.2. The initial condition is obtained by setting $t = 0$ in the analytical solution. As indicated in Section 3.10, it is assumed that boundary values are available at both ends of the computational domain, and these are also given by the analytical solution. The numerical solution is sought at time $T = 5$ for values of the grid number ranging from $J = 50$ to 1000. The number of time steps is chosen so that the maximum value the Courant number takes is one of the three values 1, 0.5 and 0.25. A summary of the data required to implement the numerical test is given in Table 4.8.

4.3.1 Implementation

The explicit schemes are implemented in a trivial fashion, with the single unknown at the new time level $(n + 1)$ being given explicitly from the known values at the old time level n.

Those methods which have (1,3) computational stencils are solved for $j = 1(1)J - 1$ with the values at the boundaries being given by the analytical solution.

The explicit methods which have (1,5) stencils are solved for $j = 2(1)J - 2$ and need to be supplemented near the boundaries. The values close to the boundaries are given by the analytical solution.

The implicit methods require the solution of a tridiagonal system of linear algebraic equations each time step. The Thomas algorithm is used to solve the tridiagonal system.

The coefficients of the explicit schemes are evaluated at time t_n, while those for the implicit schemes are evaluated at $t_{n+1/2}$. The programs were written in Fortran 90 and run on a Toshiba Tecra 8000.

1. Exact solution $\hat{\tau}(x,t) = \exp\left\{\frac{2}{\pi}\sin(\frac{\pi t}{2}) - e^x\right\}$

2. Initial condition: $\hat{\tau}(x,0)$ is given by the exact solution

3. Boundary conditions: $\hat{\tau}(0,t)$ and $\hat{\tau}(1,t)$ are given by the exact solution

4. Courant number $c = u\Delta t/\Delta x = uTJ/N$

5. $u(x,t) = e^{-x}\cos(\pi t/2)$

6. $u\text{max} = 1$, $T = 5$, $J = 50, 100, 200, 500, 1000$

7. setting $N = TJ$, $2TJ$, $4TJ$ gives $c\text{max} = 1, 0.5, 0.25$

Table 4.8: Data for the Numerical Test.

4.3.2 Results

To measure the accuracy of the schemes, the rms error in the numerical solution relative to the exact solution is calculated using all the grid points at the final time. The errors for all methods are given in Table 4.9 for the three values of the maximum Courant number, and the corresponding cpu times averaged over 20 consecutive runs of the algorithms are presented in Table 4.10. A comparison of the efficiency of the schemes for each Courant number is shown in Figures 4.6 and 4.7. These graphs were obtained by plotting the rms error against the cpu time on logarithmic scales.

For all methods, the cpu times approximately double when the maximum Courant number is halved. This is because doubling the number of time steps halves the Courant number. Table 4.9 shows that the first-order explicit base methods (LTH, UW15, RUS) all give similar accuracy. Furthermore, the errors for these methods are approximately halved when the maximum Courant number is halved. Their efficiency plots show that they are more efficient for $c\text{max} = 0.25$ than for the other two values of $c\text{max}$. This is because for any specified accuracy, using them with $c\text{max} = 0.25$ gives the result faster.

The OPT method is the most accurate and efficient second-order method when $c\text{max} = 1$. However, because its accuracy decreases as the maximum Courant number is halved (see Table 4.9), the OPT method is not always the most efficient second-order method to use. For example, it is more efficient to use the mod2_L method with $c\text{max} = 0.5$ than the OPT method with $c\text{max} = 0.5$ or 0.25 (compare bottom left diagram in Figure 4.6 with top right diagram in Figure 4.7).

The least efficient second-order results are given by using the mod_U method with $c\max = 0.25$. This is established from Figure 4.6 by observing that for 100 seconds of cpu time, the mod_U method would give the least accurate second-order results. It is however always more efficient to use the least efficient second-order scheme than the most efficient first-order scheme. For example, from their efficiency plots in Figure 4.6, it can be ascertained that it takes less than 0.1 seconds to achieve an accuracy of 1.0e-04 using the mod_L method, but more than 1 second using Leith's method for $c\max = 0.25$.

Table 4.9 shows that the implicit mod_O method is the most accurate third-order method when $c\max = 1$. Although it requires more cpu time than the explicit mod_R and mod2_R methods (see Table 4.10), it is nevertheless the most efficient third-order scheme for this Courant number. This can be seen by comparing the efficiency plots for the mod_O, mod_R and mod2_R methods pictured in Figure 4.7. Tables 4.9 and 4.10 show that the mod_O method is, however, less accurate and less efficient than the mod2_R method when the maximum Courant number is halved. Furthermore, if the maximum Courant number is halved again, the mod_O method is the least efficient third-order scheme overall.

Comparing the least efficient third-order results, given by the mod_O method with $c\max = 0.25$, with the most efficient second-order results, given by the OPT method with $c\max = 1$, shows that it is always more efficient to use the third-order scheme. For example, Tables 4.9 and 4.10 show that the OPT method used with $J = 1000$ gives an error of 3.69e-08 in 9.95 seconds, which is worse than the error of 2.78e-08 given by the mod_O method in 1.56 seconds when $J = 100$. Hence, the third-order method is more accurate and more efficient than the second-order scheme. The fourth-order mod2_O method yields the most accurate and efficient solutions to the test problem (see Table 4.9 and Figure 4.7).

Tables 4.11 and 4.12 summarize the results by ranking the accuracy and efficiency of each scheme for the three values of $c\max$ considered. The efficiency of the mod_U method for $c\max = 1$ and 0.5 overlaps (see middle right diagram in Figure 4.6), so the efficiencies for these two Courant numbers have the same rank. This is also the case for the efficiency of the mod_R method for these two values of $c\max$. The least accurate results for all schemes except for the OPT and mod_O methods are given when $c\max = 1$. The OPT and mod_O methods are on the other hand most accurate when they are used for this value. In terms of accuracy, only the mod2_L method performs at its best when the maximum Courant number takes the value 0.5. It can also be used most efficiently for this Courant number.

The base methods and the mod_O method all have the property that they are most efficient or least efficient for the same value of the maximum Courant number for which they give the most accurate or least accurate results. This is not the case for the other methods. For instance, although the mod2_U method is most accurate to use when $c\max = 0.25$ and least accurate when $c\max = 1$, the opposite can be said about its efficiency for these two Courant numbers (compare Tables 4.11 and 4.12).

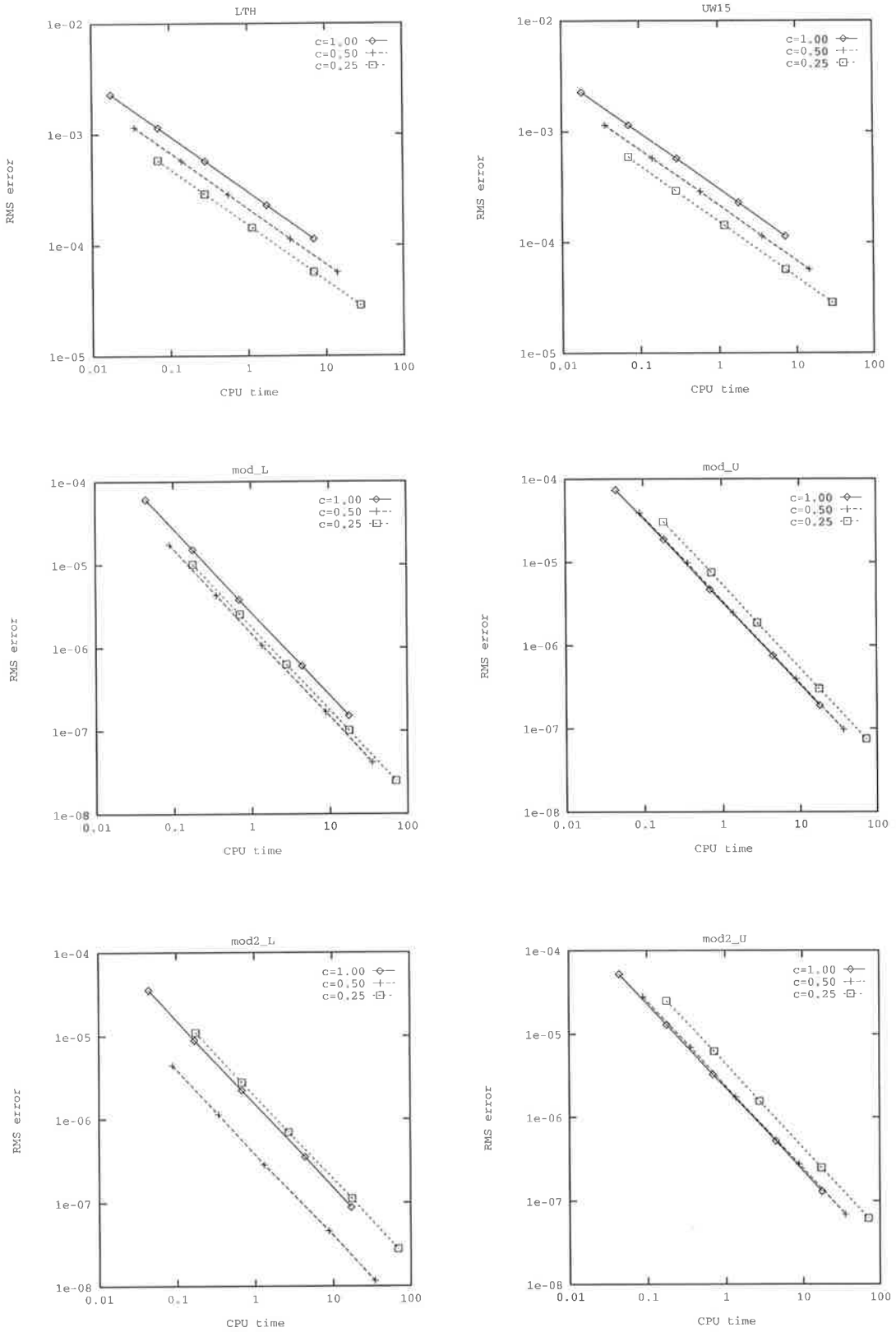

Fig 4.6: Efficiency shown for LTH, UW15 and their modifications for various $c$max.

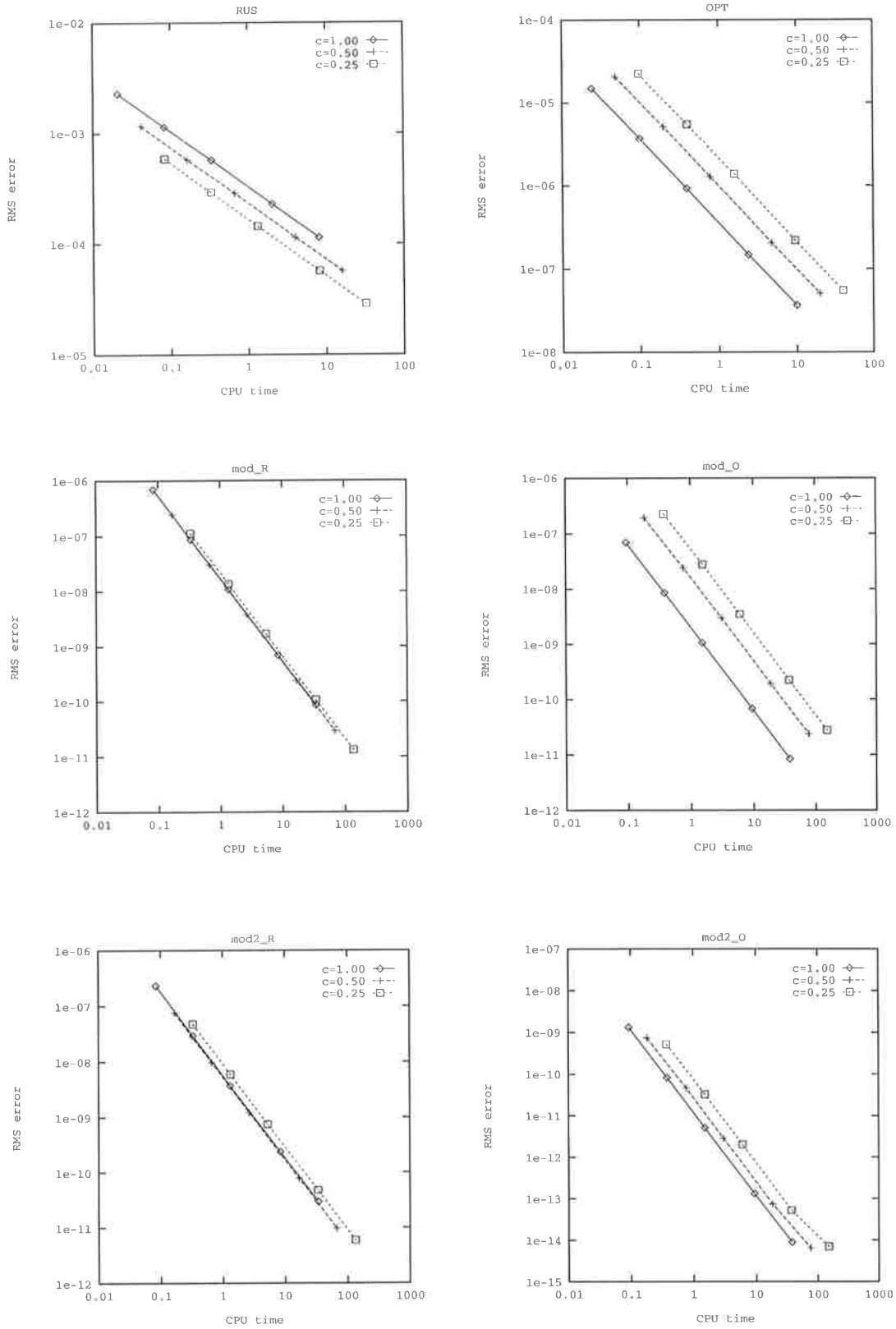

Fig 4.7: Efficiency shown for RUS, OPT and their modifications for various *c*max.

rms errors when cmax = 1				
J	LTH	UW15	RUS	OPT
50	2.29e−03	2.26e−03	2.28e−03	1.49e−05
100	1.14e−03	1.14e−03	1.14e−03	3.71e−06
200	5.72e−04	5.70e−04	5.71e−04	9.25e−07
500	2.29e−04	2.28e−04	2.28e−04	1.48e−07
1000	1.14e−04	1.14e−04	1.14e−04	3.69e−08
J	mod_L	mod_U	mod_R	mod_O
50	6.08e−05	7.39e−05	6.90e−07	6.94e−08
100	1.52e−05	1.87e−05	8.36e−08	8.55e−09
200	3.80e−06	4.72e−06	1.08e−08	1.06e−09
500	6.07e−07	7.58e−07	6.89e−10	6.76e−11
1000	1.52e−07	1.90e−07	8.60e−11	8.44e−12
J	mod2_L	mod2_U	mod2_R	mod2_O
50	3.56e−05	5.23e−05	2.30e−07	1.32e−09
100	8.88e−06	1.30e−05	2.92e−08	8.22e−11
200	2.21e−06	3.26e−06	3.68e−09	5.12e−12
500	3.54e−07	5.21e−07	2.37e−10	1.31e−13
1000	8.84e−08	1.30e−07	2.97e−11	8.84e−15
rms errors when cmax = 0.5				
J	LTH	UW15	RUS	OPT
50	1.15e−03	1.14e−03	1.16e−03	2.06e−05
100	5.75e−04	5.72e−04	5.74e−04	5.12e−06
200	2.86e−04	2.86e−04	2.86e−04	1.28e−06
500	1.14e−04	1.14e−04	1.14e−04	2.04e−07
1000	5.71e−05	5.71e−05	5.71e−05	5.10e−08
J	mod_L	mod_U	mod_R	mod_O
50	1.74e−05	3.89e−05	2.44e−07	1.93e−07
100	4.30e−06	9.79e−06	3.01e−08	2.40e−08
200	1.07e−06	2.46e−06	3.73e−09	2.99e−09
500	1.69e−07	3.93e−07	2.37e−10	1.91e−10
1000	4.19e−08	9.84e−08	2.95e−11	2.39e−11
J	mod2_L	mod2_U	mod2_R	mod2_O
50	4.42e−06	2.80e−05	7.58e−08	7.39e−10
100	1.13e−06	6.95e−06	9.65e−09	4.59e−11
200	2.85e−07	1.73e−06	1.21e−09	2.86e−12
500	4.60e−08	2.76e−07	7.79e−11	7.33e−14
1000	1.16e−08	6.91e−08	9.74e−12	6.46e−15
rms errors when cmax = 0.25				
J	LTH	UW15	RUS	OPT
50	5.81e−04	5.82e−04	5.87e−04	2.21e−05
100	2.89e−04	2.88e−04	2.89e−04	5.50e−06
200	1.44e−04	1.43e−04	1.44e−04	1.37e−06
500	5.73e−05	5.72e−05	5.72e−05	2.19e−07
1000	2.86e−05	2.86e−05	2.86e−05	5.47e−08
J	mod_L	mod_U	mod_R	mod_O
50	1.01e−05	3.03e−05	1.12e−07	2.24e−07
100	2.53e−06	7.56e−06	1.37e−08	2.78e−08
200	6.31e−07	1.89e−06	1.69e−09	3.47e−09
500	1.01e−07	3.02e−07	1.07e−10	2.22e−10
1000	2.51e−08	7.54e−08	1.33e−11	2.77e−11
J	mod2_L	mod2_U	mod2_R	mod2_O
50	1.10e−05	2.51e−05	4.72e−08	5.21e−10
100	2.77e−06	6.24e−06	5.95e−09	3.24e−11
200	6.95e−07	1.56e−06	7.45e−09	2.02e−12
500	1.12e−07	2.49e−07	4.76e−11	5.20e−14
1000	2.80e−08	6.21e−08	5.94e−12	6.93e−15

Table 4.9: Comparison of RMS errors for various values of the maximum Courant number for the one-dimensional non-conservative advection equation.

cpu times when cmax = 1				
J	LTH	UW15	RUS	OPT
50	1.74e−02	1.80e−02	2.08e−02	2.42e−02
100	7.03e−02	7.20e−02	8.24e−02	9.98e−02
200	2.83e−01	2.92e−01	3.34e−01	3.90e−01
500	1.74e+00	1.79e+00	2.03e+00	2.37e+00
1000	6.97e+00	7.15e+00	8.04e+00	9.95e+00
J	mod_L	mod_U	mod_R	mod_O
50	4.45e−02	4.47e−02	8.46e−02	9.38e−02
100	1.78e−01	1.82e−01	3.39e−01	3.84e−01
200	6.94e−01	6.97e−01	1.36e+00	1.52e+00
500	4.43e+00	4.46e+00	8.49e+00	9.42e+00
1000	1.76e+01	1.80e+01	3.41e+01	3.82e+01
J	mod2_L	mod2_U	mod2_R	mod2_O
50	4.39e−02	4.44e−02	8.41e−02	9.32e−02
100	1.70e−01	1.78e−01	3.34e−01	3.80e−01
200	6.83e−01	6.89e−01	1.32e+00	1.51e+00
500	4.40e+00	4.43e+00	8.46e+00	9.40e+00
1000	1.71e+01	1.76e+01	3.39e+01	3.80e+01
cpu times when cmax = 0.5				
J	LTH	UW15	RUS	OPT
50	3.49e−02	3.59e−02	4.14e−02	4.86e−02
100	1.41e−01	1.44e−01	1.64e−01	2.00e−01
200	5.60e−01	5.82e−01	6.68e−01	7.79e−01
500	3.47e+00	3.58e+00	4.04e+00	4.73e+00
1000	1.40e+01	1.46e+01	1.61e+01	1.98e+01
J	mod_L	mod_U	mod_R	mod_O
50	8.93e−02	8.94e−02	1.69e−01	1.85e−01
100	3.56e−01	3.62e−01	6.77e−01	7.70e−01
200	1.37e+00	1.39e+00	2.70e+00	3.08e+00
500	8.90e+00	8.92e+00	1.70e+01	1.88e+01
1000	3.52e+01	3.64e+01	6.80e+01	7.72e+01
J	mod2_L	mod2_U	mod2_R	mod2_O
50	8.80e−02	8.88e−02	1.67e−01	1.83e−01
100	3.42e−01	3.58e−01	6.64e−01	7.64e−01
200	1.32e+00	1.35e+00	2.68e+00	3.00e+00
500	8.86e+00	8.90e+00	1.68e+01	1.84e+01
1000	3.44e+01	3.54e+01	6.67e+01	7.61e+01
cpu times when cmax = 0.25				
J	LTH	UW15	RUS	OPT
50	6.95e−02	7.19e−02	8.30e−02	9.64e−02
100	2.80e−01	2.86e−01	3.30e−01	3.97e−01
200	1.13e+00	1.17e+00	1.32e+00	1.57e+00
500	6.97e+00	7.18e+00	8.19e+00	9.52e+00
1000	2.79e+01	2.88e+01	3.23e+01	3.97e+01
J	mod_L	mod_U	mod_R	mod_O
50	1.78e−01	1.80e−01	3.36e−01	3.78e−01
100	7.10e−01	7.30e−01	1.36e+00	1.56e+00
200	2.79e+00	2.81e+00	5.42e+00	6.06e+00
500	1.77e+01	1.78e+01	3.38e+01	3.74e+01
1000	7.06e+01	7.24e+01	1.36e+02	1.54e+02
J	mod2_L	mod2_U	mod2_R	mod2_O
50	1.76e−01	1.78e−01	3.34e−01	3.73e−01
100	6.87e−01	7.16e−01	1.33e+00	1.52e+00
200	2.71e+00	2.74e+00	5.30e+00	6.03e+00
500	1.76e+01	1.74e+01	3.36e+01	3.72e+01
1000	6.86e+01	7.10e+01	1.34e+02	1.50e+02

Table 4.10: Comparison of CPU times for various values of the maximum Courant number for the one-dimensional non-conservative advection equation.

Method	$c = 1$	$c = 0.5$	$c = 0.25$
LTII	3	2	1
mod_L	3	2	1
mod2_L	3	1	2
UW15	3	2	1
mod_U	3	2	1
mod2_U	3	2	1
RUS	3	2	1
mod_R	3	2	1
mod2_R	3	2	1
OPT	1	2	3
mod_O	1	2	3
mod2_O	3	2	1

Table 4.11: Accuracy from best 1 to worst 3 for the various cmax.

Method	$c = 1$	$c = 0.5$	$c = 0.25$
LTH	3	2	1
mod_L	3	1	2
mod2_L	2	1	3
UW15	3	2	1
mod_U	1,2	1,2	3
mod2_U	1	2	3
RUS	3	2	1
mod_R	1,2	1,2	3
mod2_R	2	1	3
OPT	1	2	3
mod_O	1	2	3
mod2_O	1	2	3

Table 4.12: Efficiency from best 1 to worst 3 for the various cmax.

4.3.3 Summary

An analytical solution was developed to compare the performance of the schemes in terms of accuracy, execution time and efficiency. The initial condition, boundary conditions, and any supplementary values required near the boundaries were all given by the analytical solution. The effect of changing the size of the maximum Courant number on the accuracy and efficiency of the schemes was investigated.

The values of the maximum Courant number for which each scheme was most or least accurate and most or least efficient, were summarized in Tables 4.11 and 4.12. Other than for the mod_O method and for the base methods, the schemes were most efficient for a different value of the maximum Courant number to that for which they were most accurate.

For this numerical test, the most accurate first-order results could be obtained by using either Leith's method or the UW15 method with a maximum Courant number of 0.25. The most accurate second, third and fourth-order results were yielded by the mod_L, mod2_R and mod2_O methods, respectively, all using $c\text{max} = 0.25$.

The most efficient first, second, third and fourth-order results were given by Leith's method, the OPT method, the mod_O method and the mod2_O method, respectively, when the size of their maximum Courant number took the value 0.25, 1, 1 and 1, respectively.

It was always more accurate and efficient to use a third-order scheme rather than a second-order one, and a second-order scheme instead of a first-order method. The fourth-order implicit mod2_O method was the most accurate and efficient method overall. In terms of both accuracy and efficiency, the new modified methods were seen to be superior to their base counterparts.

4.3.4 Stability

Because of stability constraints, Leith's method and Rusanov's method can only be used if their maximum Courant number remains less than or equal to one. Although the OPT method is unconditionally stable, its effective range of stability is reduced to a maximum Courant number of one because the coefficient matrix must remain diagonally dominant if the efficient Thomas algorithm is to be used.

Although the UW15 method is stable for a maximum Courant number of two, it might in practice need to be supplemented at the boundaries because of its (1,5) computational stencil. If Leith's method, for example, is used to obtain these values, the effective range of stability of the UW15 method is reduced to that of Leith's method. Hence, none of the methods benefit from having a larger stability region.

Chapter 5

Two-Dimensional Non-Conservative Advection

Geophysical flow problems like those found in meteorology (Holmgren 1994) and oceonography (Hubbard and Baines 1997) are often two-dimensional. Two-dimensional flow problems frequently arise in the aerospace industry (Sharif and Busnaina 1988) and in acoustic or elastic wave propagation (Leveque 1997). Consequently, much effort has gone into improving discretization schemes for the two-dimensional advection equation. In Chapter 3, we showed how finite-difference methods used to solve the one-dimensional non-conservative advection equation could be modified to give high-order convergence for variable advective velocities. In this chapter we demonstrate that these modified methods can be used to give high-order results to the two-dimensional non-conservative advection equation.

5.1 Locally One-Dimensional (LOD) Methods

Multi-dimensional equations may be divided into a number of simpler locally one-dimensional equations, each of which is solved sequentially to approximate the solution of the original equation. Difficulties associated with discretizing the full equation to obtain an FDE based on a three-dimensional computational stencil are thus avoided. Separate numerical algorithms can be used for each one-dimensional problem, and if an exact solution is known to any of the subprocesses it may be used with numerical schemes for the remaining subprocesses to yield a solution to the complete equation.

As boundary conditions are defined for the full two-dimensional equation only, appropriate intermediate boundary conditions must generally be determined for the split equations. Gourlay and Mitchell (1972) and more recently Leveque and Oliger (1983) have shown how intermediate boundary conditions for two-dimensional hyperbolic equations may be derived. Alternatively, the boundary conditions and the intermediate boundary conditions may be periodic.

A necessary condition for the stability of the LOD method is that the stability conditions for the methods used for each spatial dimension are satisfied simultaneously (Mitchell and Griffiths 1980). A necessary condition for the stability of the individual methods is that given by conducting a local von Neumann stability analysis (Hirsch 1990). This will be taken to be the case in the following.

5.2 Application

Consider the two-dimensional non-conservative form of the advection equation given by

$$\frac{\partial \hat{\tau}}{\partial t} + u(x,y,t)\frac{\partial \hat{\tau}}{\partial x} + v(x,y,t)\frac{\partial \hat{\tau}}{\partial y} = 0, \tag{5.1}$$

in which $\hat{\tau} = \hat{\tau}(x,y,t)$. The numerical solution of (5.1) over one time-step may be obtained by solving

$$\frac{\partial \hat{\tau}}{\partial t} + u(x,y,t)\frac{\partial \hat{\tau}}{\partial x} = 0, \tag{5.2}$$

commencing with the values $\tau^n_{j,k}$ to give the intermediate values $\tau^*_{j,k}$. Then using these as initial values

$$\frac{\partial \hat{\tau}}{\partial t} + v(x,y,t)\frac{\partial \hat{\tau}}{\partial y} = 0 \tag{5.3}$$

is solved, yielding an approximate solution $\tau^{n+1}_{j,k}$ to the full problem after one time-step. The values $\tau^*_{j,k}$ are fictitious and have no physical significance.

5.3 Leith's Method

Application of Leith's method to each one-dimensional equation leads to the explicit forms

$$\tau^*_{j,k} = \frac{1}{2}c_x(c_x+1)\tau^n_{j-1,k} + (1-c_x^2)\tau^n_{j,k} + \frac{1}{2}c_x(c_x-1)\tau^n_{j+1,k}, \tag{5.4}$$

which, by sweeping in the x direction for each y value, yields the intermediate values $\tau^*_{j,k}$, and

$$\tau^{n+1}_{j,k} = \frac{1}{2}c_y(c_y+1)\tau^*_{j,k-1} + (1-c_y^2)\tau^*_{j,k} + \frac{1}{2}c_y(c_y-1)\tau^*_{j,k+1}, \tag{5.5}$$

which, by sweeping in the y direction for each x value, yields an approximate solution to the full two-dimensional problem at time-level $(n+1)$. In the above, $c_x = u(x,y,t)\Delta t/\Delta x$ and $c_y = v(x,y,t)\Delta t/\Delta y$.

In a similar fashion, either the mod_L or mod2_L method may be applied to each one-dimensional equation. This is demonstrated here for the mod_L method, so that

$$\tau^*_{j,k} = \frac{1}{2}(c_x + c_x^2 + d_x + |d_x|)\tau^n_{j-1,k} + (1 - c_x^2 - |d_x|)\tau^n_{j,k} - \frac{1}{2}(c_x - c_x^2 + d_x - |d_x|)\tau^n_{j+1,k}, \tag{5.6}$$

is applied by sweeping in the x direction for each y value. Then using these as initial values

$$\tau^{n+1}_{j,k} = \frac{1}{2}(c_y + c_y^2 + d_y + |d_y|)\tau^*_{j,k-1} + (1 - c_y^2 - |d_y|)\tau^*_{j,k} - \frac{1}{2}(c_y - c_y^2 + d_y - |d_y|)\tau^*_{j,k+1}, \tag{5.7}$$

is applied by sweeping in the y direction for each x value. The parameter d_x is defined by (3.18), which on replacing Δx with Δy and u by v defines the parameter d_y. The constituent formulae are locally stable provided $|c_x| \leq 1$ and $|c_y| \leq 1$, so the LOD method is locally stable when both these criteria hold.

5.4 The UW15 Method

The UW15 method may be applied to each one-dimensional problem as follows. To obtain values at the intermediate level over the solution domain, first

$$\begin{aligned}\tau^*_{j,k} = &-\frac{1}{4}(1 - c_x)(c_x + |c_x|)\tau^n_{j-2,k} + \frac{1}{2}(2 - c_x)(c_x + |c_x|)\tau^n_{j-1,k} + \frac{1}{2}(2 - 3|c_x| + c_x^2)\tau^n_{j,k} \\ &+\frac{1}{2}(2 + c_x)(|c_x| - c_x)\tau^n_{j+1,k} - \frac{1}{4}(1 + c_x)(|c_x| - c_x)\tau^n_{j+2,k}\end{aligned} \tag{5.8}$$

is solved as it would be for the one-dimensional equation, but repeated for each y value. Then

$$\begin{aligned}\tau^{n+1}_{j,k} = &-\frac{1}{4}(1 - c_y)(c_y + |c_y|)\tau^*_{j,k-2} + \frac{1}{2}(2 - c_y)(c_y + |c_y|)\tau^*_{j,k-1} + \frac{1}{2}(2 - 3|c_y| + c_y^2)\tau^*_{j,k} \\ &+\frac{1}{2}(2 + c_y)(|c_y| - c_y)\tau^*_{j,k+1} - \frac{1}{4}(1 + c_y)(|c_y| - c_y)\tau^*_{j,k+2}\end{aligned} \tag{5.9}$$

is solved by sweeping in the y direction for each x value, giving an approximate solution to the two-dimensional problem. Similarly, either the mod_U or mod2_U method could be used for each one-dimensional advection equation. The correction factors d_x and d_y are exactly the same as for the mod_L and mod2_L methods. The constituent formulae are locally stable if $|c_x| \leq 2$ and $|c_y| \leq 2$, so the LOD method is locally stable when both of these conditions hold.

5.5 Rusanov's Method

Consider the application of Rusanov's method to each of the one-dimensional problems, so that

$$\begin{aligned}\tau^*_{j,k} = -\frac{1}{24}c_x(1-c_x^2)(2+c_x)\tau^n_{j-2,k} + \frac{1}{6}c_x(1+c_x)(4-c_x^2)\tau^n_{j-1,k} + \frac{1}{4}(1-c_x^2)(4-c_x^2)\tau^n_{j,k} \\ -\frac{1}{6}c_x(1-c_x)(4-c_x^2)\tau^n_{j+1,k} + \frac{1}{24}c_x(1-c_x^2)(2-c_x)\tau^n_{j+2,k}\end{aligned} \tag{5.10}$$

is first solved by sweeping in the x direction for each y value, followed by the application of

$$\begin{aligned}\tau^{n+1}_{j,k} = -\frac{1}{24}c_y(1-c_y^2)(2+c_y)\tau^*_{j,k-2} + \frac{1}{6}c_y(1+c_y)(4-c_y^2)\tau^*_{j,k-1} + \frac{1}{4}(1-c_y^2)(4-c_y^2)\tau^*_{j,k} \\ -\frac{1}{6}c_y(1-c_y)(4-c_y^2)\tau^*_{j,k+1} + \frac{1}{24}c_y(1-c_y^2)(2-c_y)\tau^*_{j,k+2}\end{aligned} \tag{5.11}$$

as it would be for the one-dimensional equation (5.3), but repeated for each x value. This gives an approximate solution to the two-dimensional advection equation after one time-step. In an identical fashion the mod_R and mod2_R methods could be used in a LOD fashion. The parameters d_x and h_x are then defined by (3.36); d_y and h_y are obtained by replacing Δx by Δy and u by v. A necessary condition for the stability of the LOD method is that $|c_x| \leq 1$ and $|c_y| \leq 1$.

5.6 The Optimal Method

Application of the optimal method to each one-dimensional equation leads to the implicit forms:

$$\begin{aligned}&(1-c_x)(2-c_x)\tau^*_{j-1,k} + 2(2-c_x)(2+c_x)\tau^*_{j,k} + (1+c_x)(2+c_x)\tau^*_{j+1,k} \\ &\quad = (1+c_x)(2+c_x)\tau^n_{j-1,k} + 2(2-c_x)(2+c_x)\tau^n_{j,k} + (1-c_x)(2-c_x)\tau^n_{j+1,k},\end{aligned} \tag{5.12}$$

which is solved by sweeping in the x direction for each value of y, and

$$\begin{aligned}&(1-c_y)(2-c_y)\tau^{n+1}_{j,k-1} + 2(2-c_y)(2+c_y)\tau^{n+1}_{j,k} + (1+c_y)(2+c_y)\tau^{n+1}_{j,k+1} \\ &\quad = (1+c_y)(2+c_y)\tau^*_{j,k-1} + 2(2-c_y)(2+c_y)\tau^*_{j,k} + (1-c_y)(2-c_y)\tau^*_{j,k+1},\end{aligned} \tag{5.13}$$

applied by sweeping in the y direction for each x value. The coefficients of the implicit schemes are evaluated at $(x_j, y_k, t_{n+1/2})$. The mod_O and mod2_O methods are implemented in the same way, with the correction factors d_x and h_x defined by (3.49); and d_y and h_y obtained by replacing Δx by Δy and u by v. Although the individual equations are unconditionally locally stable, they are only diagonally dominant and hence solvable using the very efficient Thomas algorithm if $|c_x| \leq 1$ and $|c_y| \leq 1$.

5.7 Numerical Test

The initial condition for this numerical test is a two-dimensional Gaussian distribution of unit height given by $\hat{\tau}(x, y, 0) = \exp\{-400(x - 0.5)^2 - 400(y - 0.5)^2\}$. The velocity profiles are chosen to take the form $u(x, t)$ and $v(y, t)$, where u is the same as given in Table 5.1, and v is obtained by replacing x by y. A uniform grid with grid spacings of Δx and Δy in the x and y directions is defined, and the advection equation is solved on the domain $[0, 1]$ in each spatial direction. The boundary conditions and the intermediate boundary conditions are periodic. The numerical solution is sought at time $T = \pi$, with $J = K = N$, and $\kappa = 2/3\pi$, so the maximum value of the Courant numbers c_x and c_y is one, and the exact solution is given by the initial condition. The numerical test is summarized in Table 5.1.

1.	Initial condition $\hat{\tau}(x, y, 0) = \exp\{-400(x - 0.5)^2 - 400(y - 0.5)^2\}$
2.	Periodic boundary and intermediate boundary conditions
3.	Courant numbers $c_x = u\Delta t/\Delta x = uTJ/N$, $c_y = v\Delta t/\Delta y = vTK/N$
4.	maximum value of c_x and c_y is one
5.	$u(x, t) = \kappa(0.5 + \sin^2 \pi x) \cos t$, $v(y, t) = \kappa(0.5 + \sin^2 \pi y) \cos t$
6.	$T = \pi$, $\kappa = 2/3\pi$: half cycle
7.	maximum value of u and v is $1/\pi$
8.	$N = J = K$ with $J = 50, 80, 100, 150, 200$
9.	exact solution is the initial condition

Table 5.1: Data for the locally one-dimensional Gauss Test.

All of the methods described in Chapter 3 are tested. Although different schemes could be used for each spatial direction, the same FDE is applied in both directions, so that the convergence of the LOD method can be examined. For each time-step the LOD method is implemented as follows:

1. The FDE is first used in the x direction as it would be for the one-dimensional equation (5.2).

2. The application is repeated for each y value, yielding the intermediate values $\tau^*_{j,k}$ over the spatial domain.

3. Then using the $\tau^*_{j,k}$ as the initial condition, the same FDE is used in the y direction as it would normally be used to solve the one-dimensional equation (5.3).

4. This is repeated for each x value, yielding the values $\tau^{n+1}_{j,k}$ at the end of the time-level $(n+1)$.

Table 5.2: Implementation of the LOD technique.

The explicit schemes are solved by calculating the single unknown directly from the known values at the old time level. The periodic tridiagonal system of linear algebraic equations resulting from the implementation of the implicit schemes is solved using the cyclic variant of the Thomas algorithm.

5.7.1 Results

To allow easy comparison of the results, the figures and tables referred to in this section have been placed in a group from page 68 to 72. Table 5.3 shows that the implicit OPT and mod2_O results are virtually identical to the exact solution. This is because for these methods, the errors that accumulate over the first quarter cycle are eliminated in the second quarter cycle by errors of equal magnitude, but of opposite sign. Of the other methods, the third-order implicit mod_O method is the most accurate method overall, while the third-order mod2_R method is the most accurate explicit scheme.

The mod_O method is more accurate, but also more time consuming than the mod2_R method (compare Tables 5.3 and 5.4). The efficiency plots on the bottom right of Figure 5.2, however, reveal that the extra time spent implementing the mod_O method is worthwhile, since it outperforms the explicit mod2_R method considerably in terms of efficiency. These plots show that to achieve an error of 1e-04 takes less than 10 seconds using the mod_O method, but more than 50 seconds using the mod2_R method.

Table 5.3 shows that all the first-order explicit base methods are of similar accuracy, while the most accurate explicit second-order scheme is the mod2_L method. Because it expends less time than the other explicit second-order methods (see Table 5.4), the mod2_L method is also the most efficient explicit second-order scheme. This can also be concluded from Figure 5.2.

From Table 5.4, it is seen that doubling the size of the spatial grid leads approximately to an eightfold increase in the cpu times for all methods. This is to be expected, since for this numerical test, doubling the grid size in both spatial directions, also doubles the number of time steps.

The mod_L and mod_U methods require approximately twice as long to run for each grid than their respective base counterparts. The mod_R method needs about 3.3 times longer to run than Rusanov's method, while the mod_O method takes about 2.3 times longer than its base method for each grid.

In addition to overall accuracy and efficiency, it is important to determine how well the schemes represent the exact solution in terms of the introduction of artificial diffusion and spurious oscillations. To examine these aspects, profiles of the numerical solution and the exact solution are taken as follows.

It is recalled that the initial condition is a two-dimensional Gaussian distribution of unit height centered at $(0.5, 0.5)$. Furthermore, because the pulse returns to its initial location with its original shape every half time cycle, the exact solution (pictured in Figure 5.1) is identical to the initial condition.

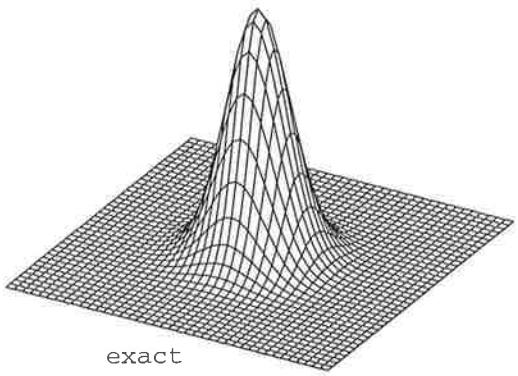

Figure 5.1: Initial condition and exact solution for two-dimensional Gauss test.

For all schemes, the numerical solution is symmetrical about $y = x$. Although this will not generally be the case, it occurs for this test problem, because the same FDE is being used in both spatial dimensions, and because the exact solution itself is symmetrical about $y = x$.

Because they introduce wave speed errors into the numerical solution, the explicit base methods position the pulse one grid point to the right of the true location. That is, for the 50*50 grid, the maximum is found at grid point $(j, k) = (26, 26)$ instead of at grid point $(25, 25)$. Likewise, for the 80*80 grid, the maximum occurs at grid point $(41, 41)$. The modified methods position the pulse correctly (see below).

Therefore, only the slice at the y location where the maximum occurred is displayed. That is, the profiles for the exact solution and modified methods are taken at $y = 0.5$, while those for the base methods are taken at $y = 0.5 + \Delta y$. Figures 5.3 to 5.5 verify that the solutions of the explicit base methods lead the exact solution. It is apparent that leading and trailing oscillations and negative values (see Table 5.6)

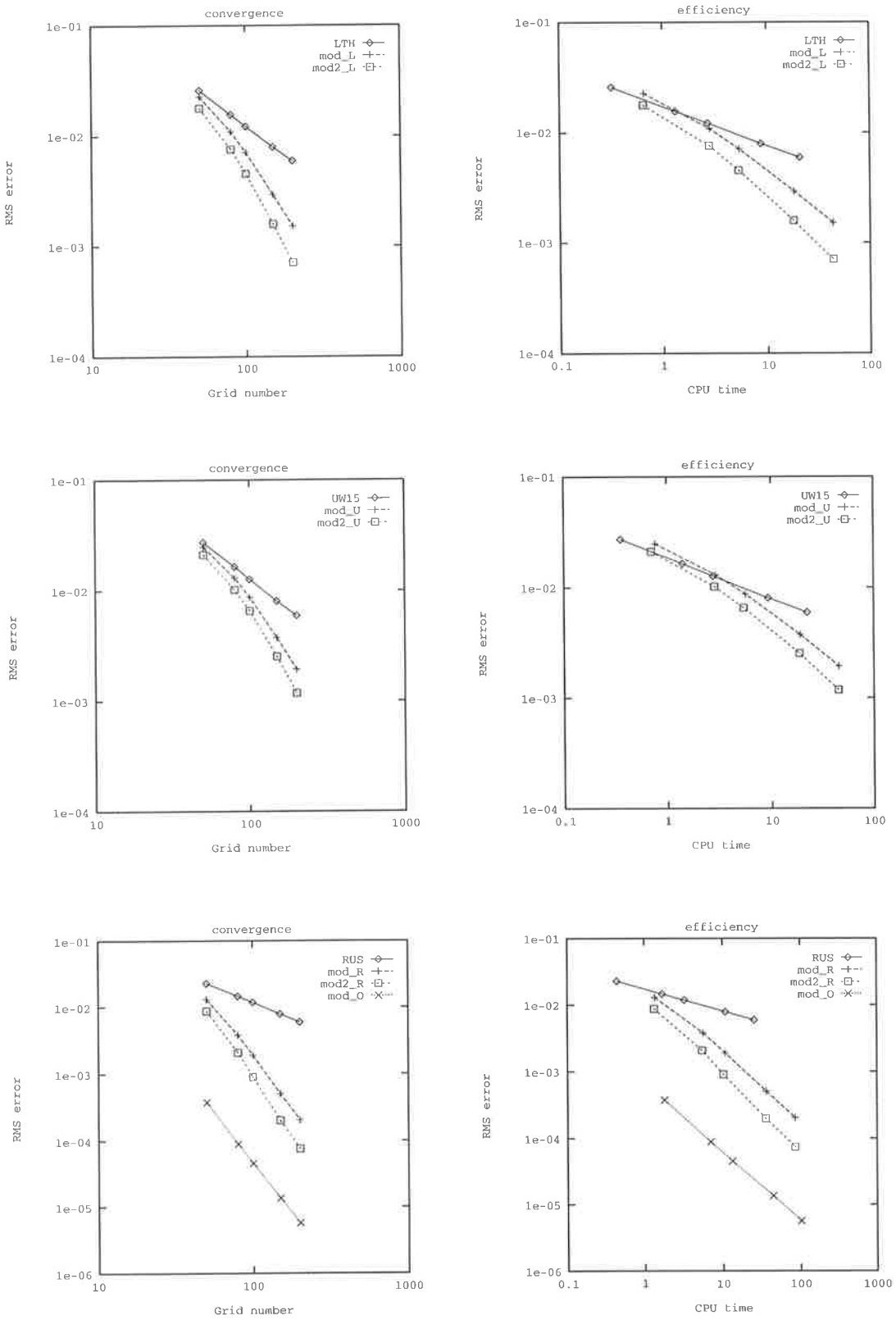

Fig 5.2: Convergence on left and efficiency on right shown for the methods. OPT, mod2_O give exact solution and are not displayed, mod_O results are attached to those of RUS.

J	LTH	UW15	RUS	OPT
50	2.59e−02	2.73e−02	2.28e−02	1.61e−16
80	1.56e−02	1.65e−02	1.46e−02	1.70e−16
100	1.22e−02	1.27e−02	1.18e−02	2.01e−16
150	7.94e−03	8.05e−03	7.91e−03	2.97e−16
200	5.95e−03	5.96e−03	5.95e−03	2.73e−16
J	mod_L	mod_U	mod_R	mod_O
50	2.27e−02	2.49e−02	1.30e−02	3.74e−04
80	1.09e−02	1.30e−02	3.80e−03	8.88e−05
100	7.10e−03	8.71e−03	1.89e−03	4.54e−05
150	2.94e−03	3.75e−03	5.09e−04	1.35e−05
200	1.52e−03	1.93e−03	2.05e−04	5.69e−06
J	mod2_L	mod2_U	mod2_R	mod2_O
50	1.78e−02	2.12e−02	8.74e−03	1.93e−16
80	7.56e−03	1.02e−02	2.06e−03	2.66e−16
100	4.55e−03	6.56e−03	9.06e−04	2.65e−16
150	1.59e−03	2.53e−03	2.02e−04	3.75e−16
200	7.05e−04	1.18e−03	7.55e−05	4.00e−16

Table 5.3: RMS errors for the two-dimensional Gauss test.

J	LTH	UW15	RUS	OPT
50	3.20e−01	3.57e−01	4.40e−01	7.95e−01
80	1.32e+00	1.41e+00	1.70e+00	3.05e+00
100	2.69e+00	2.76e+00	3.25e+00	5.74e+00
150	8.68e+00	9.34e+00	1.09e+01	1.83e+01
200	2.06e+01	2.23e+01	2.54e+01	4.23e+01
J	mod_L	mod_U	mod_R	mod_O
50	6.60e−01	7.75e−01	1.37e+00	1.81e+00
80	2.80e+00	2.93e+00	5.61e+00	6.98e+00
100	5.38e+00	5.66e+00	1.08e+01	1.33e+01
150	1.81e+01	1.90e+01	3.60e+01	4.33e+01
200	4.30e+01	4.51e+01	8.53e+01	1.02e+02
J	mod2_L	mod2_U	mod2_R	mod2_O
50	6.50e−01	7.10e−01	1.34e+00	1.71e+00
80	2.78e+00	2.84e+00	5.46e+00	6.94e+00
100	5.32e+00	5.42e+00	1.02e+01	1.31e+01
150	1.79e+01	1.87e+01	3.54e+01	4.28e+01
200	4.28e+01	4.45e+01	8.49e+01	1.01e+02

Table 5.4: CPU times for the two-dimensional Gauss test.

J	LTH	UW15	RUS	OPT
50	3.95e-01	4.54e-01	2.14e-01	–
80	1.90e-01	2.44e-01	5.94e-02	–
100	1.20e-01	1.63e-01	2.79e-02	–
150	4.48e-02	6.67e-02	7.00e-03	–
200	2.06e-02	3.18e-02	3.00e-03	–
J	mod_L	mod_U	mod_R	mod_O
50	4.50e-01	4.84e-01	2.52e-01	9.00e-03
80	2.39e-01	2.76e-01	7.22e-02	2.03e-03
100	1.61e-01	1.93e-01	3.25e-02	1.04e-03
150	6.96e-02	8.73e-02	6.02e-03	3.06e-04
200	3.63e-02	4.59e-02	1.65e-03	1.29e-04
J	mod2_L	mod2_U	mod2_R	mod2_O
50	3.58e-01	4.15e-01	1.77e-01	–
80	1.67e-01	2.18e-01	4.43e-02	–
100	1.05e-01	1.45e-01	1.92e-02	–
150	3.82e-02	5.87e-02	3.66e-03	–
200	1.72e-02	2.79e-02	1.13e-03	–

Table 5.5: Amplitude errors for the two-dimensional Gauss test.

J	LTH	UW15	RUS	OPT
50	2.56e-02	2.16e-02	2.31e-02	–
80	1.27e-02	1.65e-02	1.78e-04	–
100	6.56e-03	1.09e-02	1.36e-05	–
150	5.03e-04	2.62e-03	9.70e-13	–
200	8.16e-06	4.18e-04	–	–
J	mod_L	mod_U	mod_R	mod_O
50	1.66e-02	1.80e-02	3.17e-02	1.37e-04
80	8.87e-03	1.40e-02	4.95e-04	4.11e-06
100	4.67e-03	9.38e-03	6.70e-06	1.57e-06
150	3.61e-04	2.27e-03	3.69e-13	3.34e-07
200	5.90e-06	3.69e-04	–	4.52e-08
J	mod2_L	mod2_U	mod2_R	mod2_O
50	2.75e-02	2.73e-02	3.13e-02	–
80	1.39e-02	1.83e-02	1.41e-03	–
100	6.59e-03	1.20e-02	1.96e-05	–
150	5.23e-04	2.92e-03	1.48e-12	–
200	8.47e-06	4.56e-04	–	–

Table 5.6: Values $|\tau_{min}|$ for the two-dimensional Gauss test.

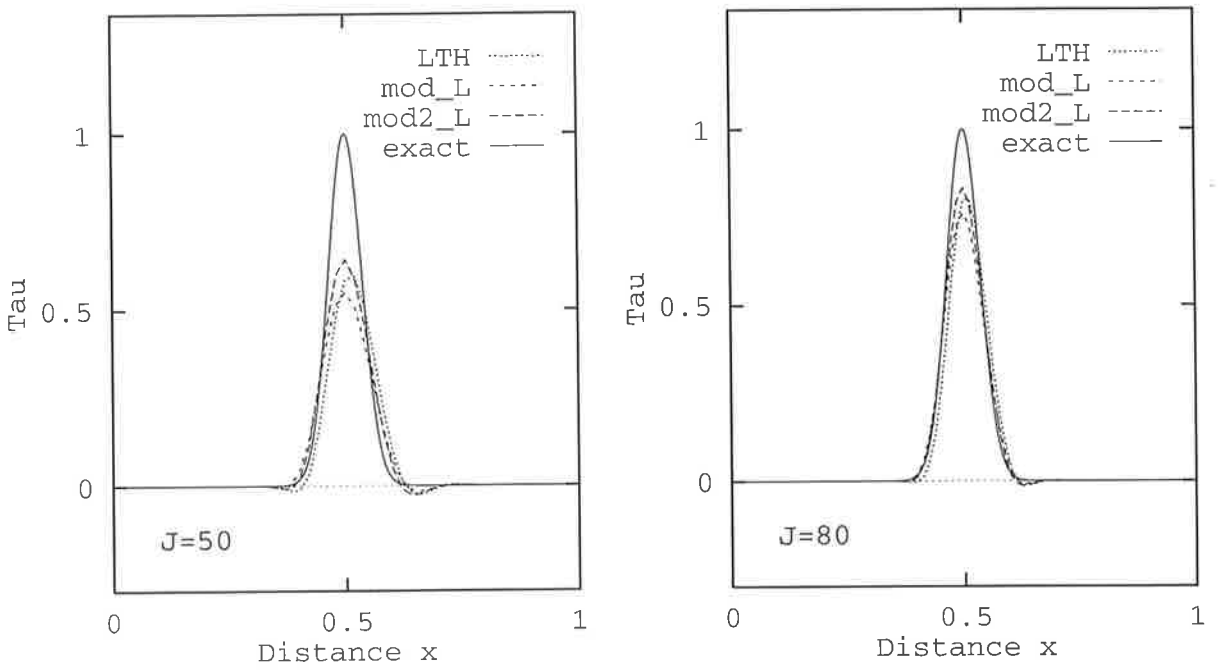

Figure 5.3: Profile at peak position for LTH, mod_L and mod2_L when $J = 50$ and $J = 80$.

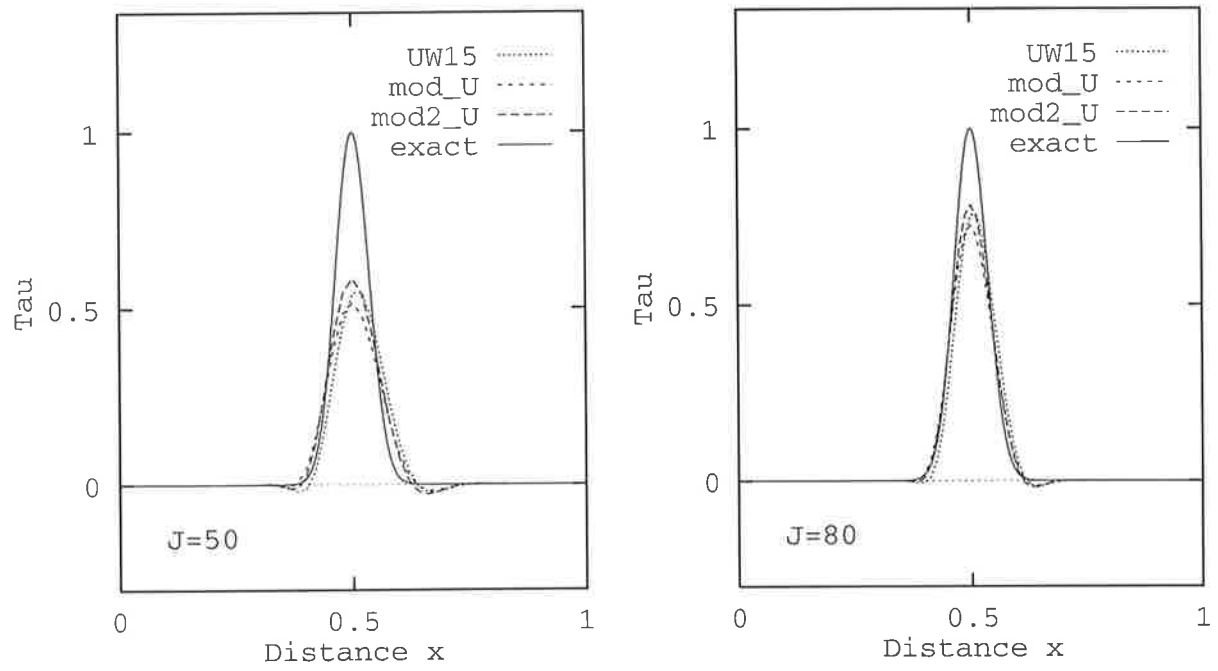

Figure 5.4: Profile at peak position for UW15, mod_U and mod2_U when $J = 50$ and $J = 80$.

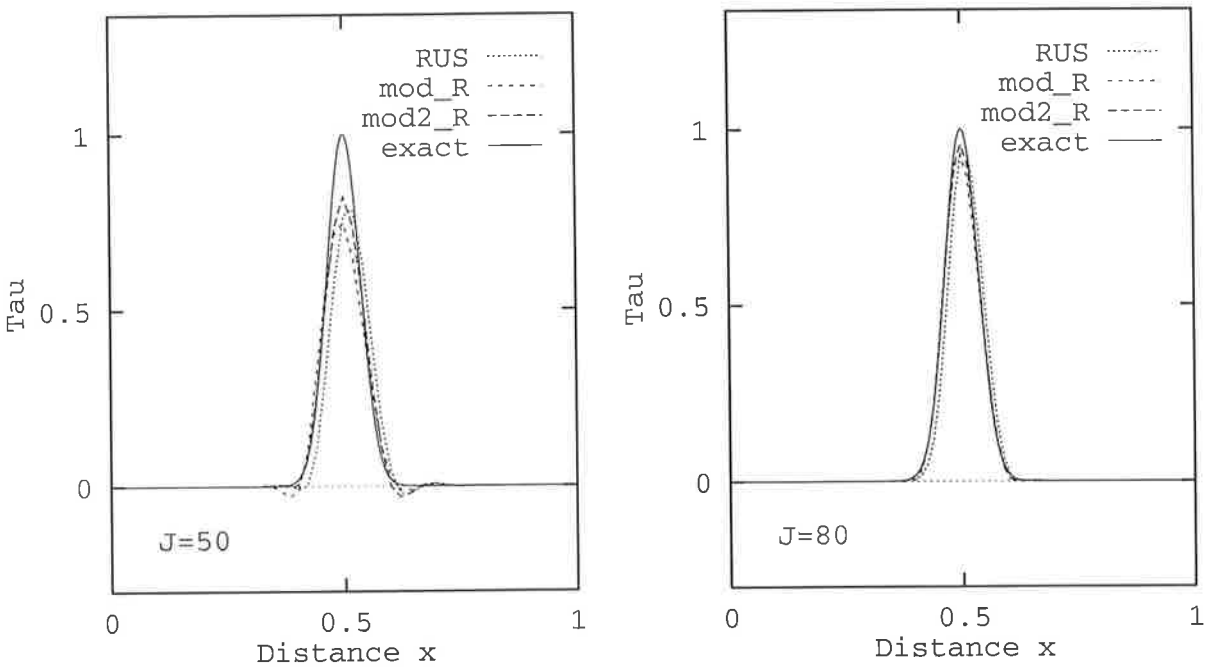

Figure 5.5: Profile at peak position for RUS, mod_R and mod2_R when $J = 50$ and $J = 80$.

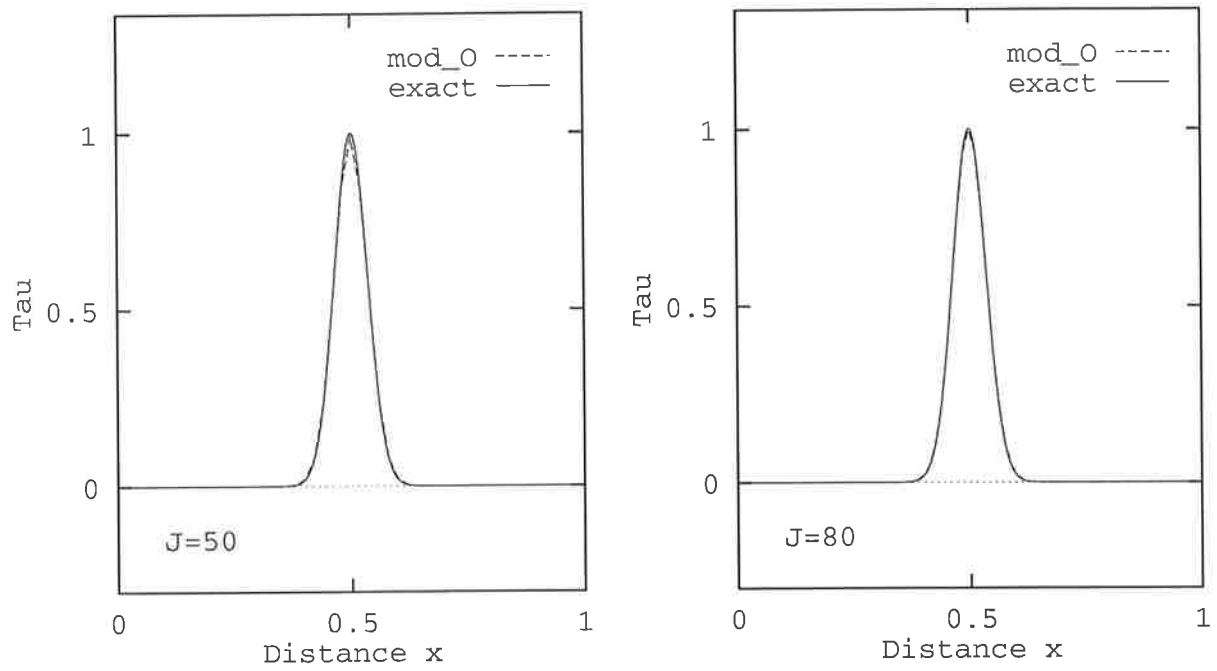

Figure 5.6: Profile at peak position for mod_O when $J = 50$ and $J = 80$.

Method	$T = \pi$	Theory
LTH	1.06	1
mod_L	1.97	2
mod2_L	2.34	2
UW15	1.11	1
mod_U	1.86	2
mod2_U	2.10	2
RUS	0.97	1
mod_R	3.02	3
mod2_R	3.47	3
OPT	–	2
mod_O	3.02	3
mod2_O	–	4

Table 5.7: The orders of convergence for the Gauss test are given by the slope of the line of best fit to the data.

are introduced into the numerical solution by the explicit schemes. Additionally, considerable artificial diffusion is introduced into the solution by most of the first and second-order explicit schemes. For example, for the mod_U method, the amplitude is only 52% of the true height when $J = 50$, and 72% of the true height when $J = 80$ (see Table 5.5).

The benefits of using the modified methods based on centered space differencing instead of those based on upwind differencing are evident in the results shown in Figures 5.3 to 5.5. For example, when $J = 50$, the amplitude of the solution given by the mod2_L method is 0.64, which is somewhat closer to one, than the peak height of 0.55 given by the mod_L method. Likewise, the mod2_U and mod2_R methods capture the height of the pulse better than the mod_U and mod_R methods (see Figures 5.4 and 5.5).

Figure 5.5 shows that Rusanov's solution is less spread out and the peak is better resolved than for the other first-order explicit base methods. Table 5.5 shows that its amplitude is 79% of the true value when $J = 50$, and 94% of the true value when $J = 80$. Although the mod_R and mod2_R methods introduce pronounced oscillations and negative values (see Table 5.6) into their solutions when $J = 50$, no oscillations are detectable when $J = 80$. Table 5.3 shows that the third-order mod2_R method is the most accurate explicit method. It is outperformed only by the implicit third-order mod_O method.

Of the methods pictured in Figures 5.3 to 5.6, the mod_O method gives the most accurate results. It aligns the solution correctly, doesn't introduce any significant negative values or spurious oscillations into the solution, and captures the height of the pulse to within 1% for all grids considered. Although it uses more cpu time than the other methods, it is the most efficient method (see Figure 5.2). In this numerical test, the OPT and mod2_O methods gave the exact solution. However, all methods will be retested in Section 5.8, so that an indication of their performances can be gained.

5.7.2 Non-negativity

The negative values appearing in the numerical solution of the schemes can be eliminated by using the first-order upwind method, whenever the value $\tau^*_{j,k}$ or $\tau^{n+1}_{j,k}$ becomes negative. This has been applied to all of the LOD schemes, and the solution profiles at the location of the maximum for $J = 50$ are presented in Figure C.1 in Appendix C. Since the first-order upwind method is only used for a few points in the solution domain, the rms errors don't change enough to alter the convergence rates of the schemes. Additionally, even though the upwind method is usually very diffusive, because it is not used near the peak of the pulse, no loss in the height of the numerical solution is observed. Other than the fact that the results are non-negative, none of the other features of the solution change significantly.

5.7.3 Summary

The performance of the schemes was examined in a numerical test in which a two-dimensional Gaussian pulse was advected subject to periodic initial and boundary conditions. The two-dimensional problem was solved using the LOD technique, which essentially separates the advection into the two spatial directions (Noye 1984a). The same scheme was applied for each spatial dimension. A necessary condition for the stability of the individual methods is that given by conducting a local von Neumann stability analysis (Hirsch 1990). A necessary condition for the stability of the LOD method is then, that the conditions for the individual equations are both satisfied (Mitchell and Griffiths 1980).

For the given test problem, the modification procedure described in Chapter 3, to improve the convergence rates of the base methods to (or close to) their constant coefficient rate, but for use in variable coefficient problems, has been successful. In Figure 5.2, the superior convergence of the modified methods, compared to the base schemes, was pictured. A least squares fit to the data gives the slope of the line of best fit, which then gives the order of convergence of the method. The convergence rates tabulated in Table 5.7, verify the improved convergence of the modified methods over the base schemes.

5.8 An Analytical Solution

An exact solution to (5.2) when $u(x,t) = r(t)/s(x)$ is given by

$$\hat{\tau}_1(x,t) = L\exp\{K\int r(t)dt\}\exp\{-K\int s(x)dx\}, \tag{5.14}$$

where K and L are constants. Similarly, when $v(y,t) = p(t)/q(y)$, then a solution of (5.3) is given by

$$\hat{\tau}_2(y,t) = M\exp\{C\int p(t)dt\}\exp\{-C\int q(y)dy\}, \tag{5.15}$$

where M and C are constants. An exact solution of (5.1) is then given by the product

$$\hat{\tau}(x,y,t) = \hat{\tau}_1(x,t)\hat{\tau}_2(y,t), \tag{5.16}$$

which is also used to supply the initial and boundary values for the test problem. However, it is important that the boundary values for the intermediate level are not taken to be those given for the full two-dimensional problem. This is because, at the intermediate stage, only the advection in the x direction has been evaluated, but the boundary values given for the full problem incorporate the effects of the advection in both spatial dimensions. At the intermediate level, an exact solution is given by

$$\hat{\tau}(x,y,t_*) = \hat{\tau}_1(x,t_{n+1})\hat{\tau}_2(y,t_n), \tag{5.17}$$

which may be used to supply the intermediate boundary conditions. In this way, at the intermediate level, only the effect of the advection in the x direction at the new time level t_{n+1} is incorporated. The methods will be compared for the case $r(t) = p(t) = \sin t$, $s(x) = 20(x+0.1)$, $q(y) = 20(y+0.1)$, so that from (5.16), an analytical solution to (5.1) is given by

$$\hat{\tau}(x,y,t) = \exp\left\{-10(x+0.1)^2 - 10(y+0.1)^2 - 2\cos t\right\}, \tag{5.18}$$

where the constants have been set to 1.

5.8.1 Numerical Test

The two-dimensional advection equation is solved on the domain $[0,1]\times[0,1]$ to a final time of $T = \pi/2$. A uniform grid with $J = K = N$ is defined, so that the maximum value of the Courant numbers c_x and c_y is $\pi/4 \approx 0.785$, for all grids considered. The initial and boundary conditions are given by the analytical solution, and the intermediate boundary conditions are given by (5.17). A summary of the data required to implement the numerical test is given in Table 5.8.

1.	Exact solution $\hat{\tau}(x,y,t) = \exp\{-10(x+0.1)^2 - 10(y+0.1)^2 - 2\cos t\}$
2.	Initial condition: $\hat{\tau}(x,y,0)$ given by the exact solution
3.	Boundary conditions: $\hat{\tau}(0,y,t)$, $\hat{\tau}(1,y,t)$, $\hat{\tau}(x,0,t)$, $\hat{\tau}(x,1,t)$ given by exact solution
4.	Intermediate BCs: $\hat{\tau}(0,y,t_*)$, $\hat{\tau}(1,y,t_*)$, $\hat{\tau}(x,0,t_*)$, $\hat{\tau}(x,1,t_*)$ given by (5.17)
5.	Courant numbers $c_x = u\Delta t/\Delta x = uTJ/N$, $c_y = v\Delta t/\Delta y = vTK/N$
6.	maximum value of c_x and c_y is $\pi/4 \approx 0.785$
7.	$u(x,t) = \sin t/20(x+0.1)$, $v(y,t) = \sin t/20(y+0.1)$
8.	$T = \pi/2$: quarter cycle
9.	maximum value of u and v is 0.5
10.	$N = J = K$ with $J = 50, 80, 100, 150, 200$

Table 5.8: Data for the locally one-dimensional test problem.

For each time-step, the LOD method is implemented by first applying the FDE in the x direction for each y value, yielding intermediate values over the spatial domain. Then, using these as the initial conditions, the same FDE is solved in the y direction, but this time, for each x value. This yields an approximate solution to the full two-dimensional problem after one time-step.

The explicit schemes are implemented in a trivial fashion, with the single unknown being given directly from the values at the old level. Those methods which have (1,5) computational stencils (recall Figure 3.1), need to be supplemented near the boundaries (cf. Section 4.3.1), and these values are given by the analytical solution (5.17) for the intermediate stage, or from (5.16) for the second stage. The implicit methods are solved using the Thomas algorithm for tridiagonal systems.

As a measure of the accuracy of each LOD scheme, the rms errors at the final time are calculated and given in Table 5.9. The cpu times required are presented in Table 5.10. In Figures 5.7 and 5.8 the data are plotted using logarithmic scales, giving a comparison of the convergence and efficiency of each modified method in relation to its base method, as well as a comparison of their run times.

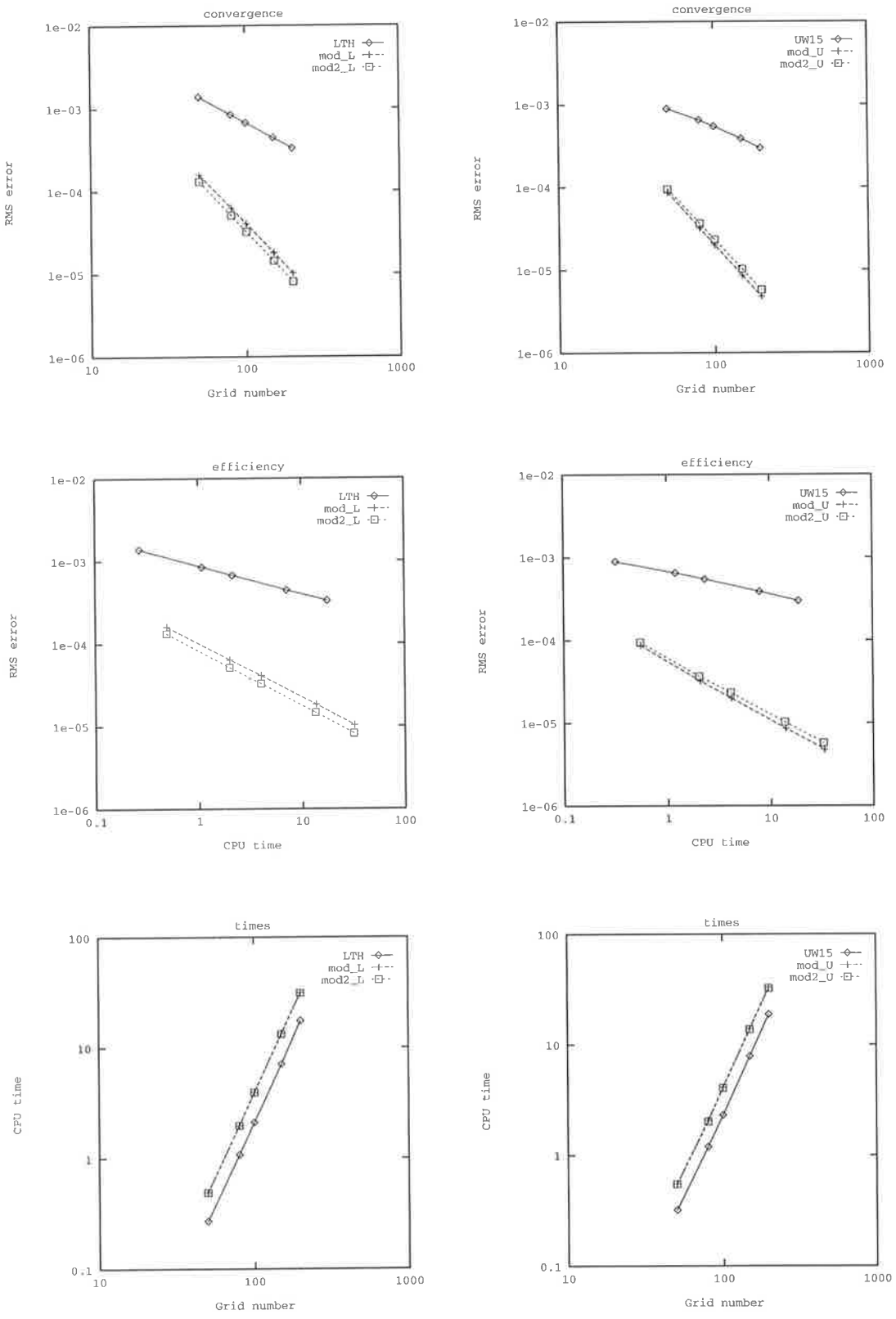

Fig 5.7: Convergence, efficiency and times shown for LTH and its modifications on the left and UW15 and its modifications on the right.

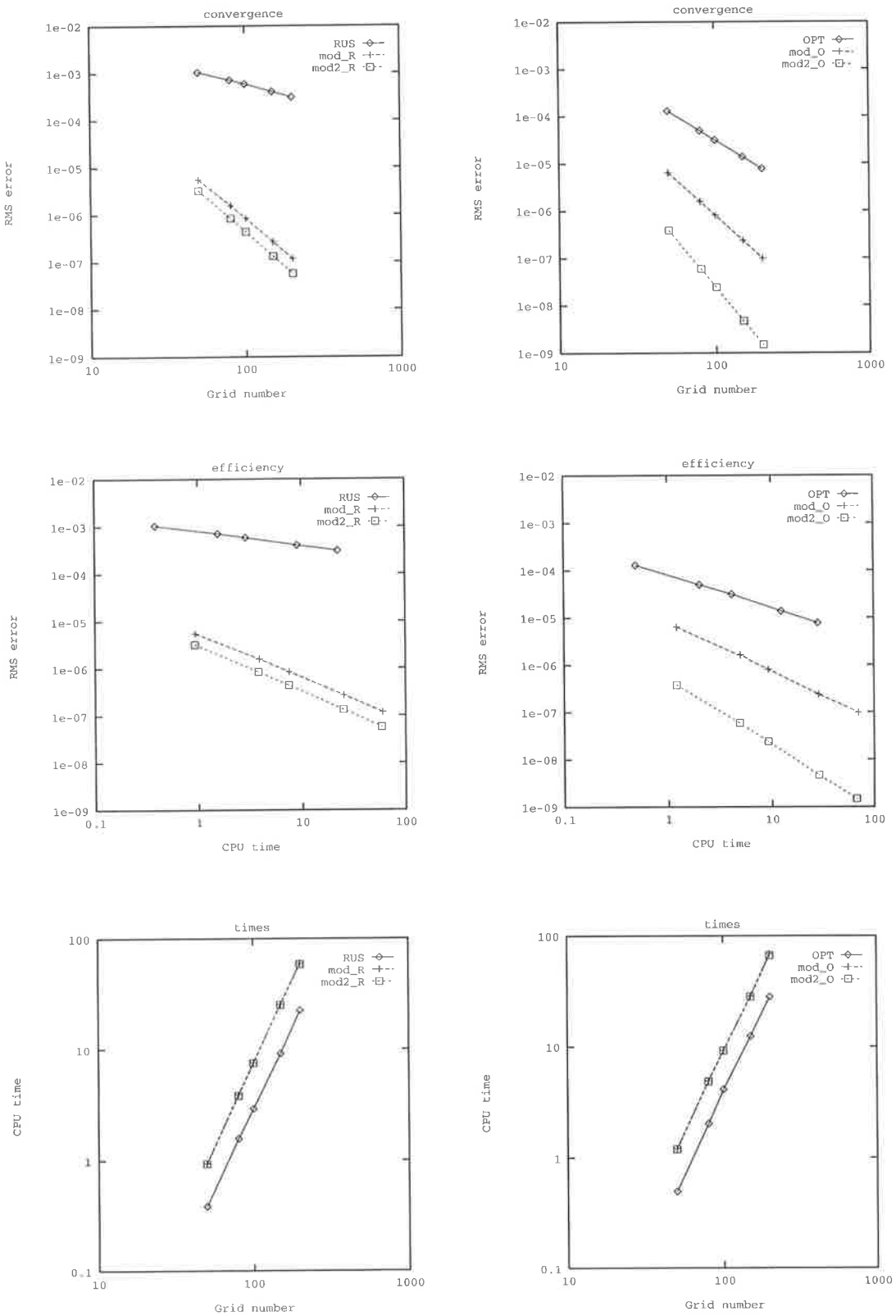

Fig 5.8: Convergence, efficiency and times shown for RUS and its modifications on the left and OPT and its modifications on the right.

J	LTH	UW15	RUS	OPT
50	1.38e−03	8.91e−04	1.04e−03	1.29e−04
80	8.46e−04	6.50e−04	7.07e−04	4.93e−05
100	6.72e−04	5.46e−04	5.82e−04	3.13e−05
150	4.43e−04	3.87e−04	4.03e−04	1.38e−05
200	3.31e−04	2.99e−04	3.08e−04	7.75e−06
J	mod_L	mod_U	mod_R	mod_O
50	1.57e−04	8.68e−05	5.51e−06	6.36e−06
80	6.27e−05	3.20e−05	1.57e−06	1.57e−06
100	4.04e−05	2.00e−05	8.44e−07	8.00e−07
150	1.81e−05	8.61e−06	2.68e−07	2.36e−07
200	1.02e−05	4.77e−06	1.17e−07	9.91e−08
J	mod2_L	mod2_U	mod2_R	mod2_O
50	1.30e−04	9.42e−05	3.23e−06	3.83e−07
80	5.06e−05	3.66e−05	8.40e−07	5.83e−08
100	3.23e−05	2.33e−05	4.38e−07	2.38e−08
150	1.44e−05	1.03e−05	1.33e−07	4.69e−09
200	8.07e−06	5.76e−06	5.69e−08	1.48e−09

Table 5.9: RMS errors for test problem when $N = K = J$.

J	LTH	UW15	RUS	OPT
50	2.69e−01	3.19e−01	3.85e−01	4.95e−01
80	1.08e+00	1.20e+00	1.57e+00	2.02e+00
100	2.12e+00	2.32e+00	2.91e+00	4.12e+00
150	7.10e+00	7.89e+00	9.15e+00	1.24e+01
200	1.75e+01	1.88e+01	2.24e+01	2.82e+01
J	mod_L	mod_U	mod_R	mod_O
50	4.90e−01	5.50e−01	9.40e−01	1.21e+00
80	1.98e+00	2.08e+00	3.90e+00	4.89e+00
100	3.96e+00	4.12e+00	7.52e+00	9.23e+00
150	1.34e+01	1.40e+01	2.53e+01	2.85e+01
200	3.15e+01	3.30e+01	6.00e+01	6.88e+01
J	mod2_L	mod2_U	mod2_R	mod2_O
50	4.87e−01	5.46e−01	9.30e−01	1.19e+00
80	1.97e+00	2.02e+00	3.81e+00	4.85e+00
100	3.94e+00	4.07e+00	7.48e+00	9.18e+00
150	1.32e+01	1.37e+01	2.51e+01	2.82e+01
200	3.11e+01	3.24e+01	5.87e+01	6.67e+01

Table 5.10: CPU times for test problem when $N = K = J$.

Method	$T = \pi$	Theory
LTH	1.03	1
mod_L	1.97	2
mod2_L	2.00	2
UW15	0.79	1
mod_U	2.09	2
mod2_U	2.02	2
RUS	0.88	1
mod_R	2.78	3
mod2_R	2.92	3
OPT	2.03	2
mod_O	3.00	3
mod2_O	4.01	4

Table 5.11: The orders of convergence for the test problem are given by the slope of the line of best fit to the data.

Examining Table 5.9 shows that, for this numerical test, the first-order explicit UW15 method is the most accurate explicit base method, followed by Rusanov's method and Leith's method, respectively. The implicit second-order OPT method is, however, the most accurate base method. It is also more accurate than the second-order mod_L and mod2_L schemes, but less accurate than both of the modifications of the UW15 method.

The explicit third-order mod2_R method is more accurate than the implicit third-order mod_O method in this test, and since the implicit method also requires more computational time (see Table 5.10), it is less efficient to use than the mod2_R method. The implicit fourth-order mod2_O method is the most accurate method overall.

The efficiency of the schemes can also be examined by comparing the middle diagrams in Figures 5.7 and 5.8. It can be seen from Figure 5.7 that the UW15 method is a little more efficient than Leith's method when $J = 50$, but otherwise these two base methods are of very similar efficiency. Additionally, it is seen that, even though they require more cpu time (see Table 5.10), the modifications of the UW15 method are nevertheless more efficient than those of Leith's method.

The second-order implicit OPT method is clearly the most efficient base scheme (see middle right diagram in Figure 5.8). The explicit third-order mod2_R method is, as previously mentioned, more efficient than the implicit third-order mod_O method. Furthermore, it is seen that the mod2_R method is the most efficient third-order scheme overall. Figure 5.8 shows that the fourth-order implicit mod2_O method is not only the most accurate method, but also the most efficient method overall.

The benefits gained by using the modified methods, instead of the base schemes, are evident from the results presented in the diagrams. As a specific example, from Tables 5.9 and 5.10, it can be seen that using the mod2_R method instead of Rusanov's method with $J = 80$, takes 2.24 seconds longer, but the error is reduced by a factor of 842. For all methods, doubling the grid number leads approximately to an eightfold increase in the cpu times.

In Table 5.11, the orders of convergence of the LOD schemes are tabulated. These were obtained from a least squares fit to the data, and verify the improved convergence of the modified methods over the base schemes. For this numerical test, the UW15 method and Rusanov's methods are a little less than first-order convergent, and the mod_R method gives results which are a little less than third-order. For the other schemes, the convergence rates are very close to being those expected from the theory.

5.8.2 Comment

It has already been mentioned that very few analytical solutions are known for variable coefficient problems, and numerical schemes must usually be tested on simplified equations. Because of the lack of available analytical solutions for more general problems, the numerical tests considered in this chapter examined the special case u independent of y and v independent of x.

However, if we carefully examine the way in which LOD techniques are implemented, it is seen that for each sweep in the x direction, the value of y is actually held constant. Likewise, for each sweep in the y direction, the value of x is fixed. In other words, even if, in the most general case, the velocity u is also dependent on y, for each sweep, the dependence on y would be treated as a constant. Hence, although a special case has been considered here, the LOD technique should also be applicable for the more general case.

It should be noted that the Molenkamp test, which is commonly used in the literature to test schemes for two-dimensional unsteady advection (see for example Vreugdenhil and Koren 1993), is also a special case. In fact, in the Molenkamp test, the velocity u is independent of x and t, and the velocity v is independent of y and t. For the purpose of testing the accuracy of the methods considered in this work, this test is inapplicable, since for each sweep, the velocity is essentially a constant.

Chapter 6

One-Dimensional Conservative Advection

Although the non-conservative form of the advection equation is a valid equation, describing many physical processes (see Zoppou and Knight 1997a, Leveque 1997), it is often important that the conservative form of the advection equation is considered. This is particularly the case for non-linear problems, such as in the propagation of a shock (Vreugdenhil and Koren 1993). The one-dimensional advection equation, written in its most general conservative form

$$\frac{\partial \hat{\tau}}{\partial t} + \frac{\partial (u\hat{\tau})}{\partial x} = 0, \tag{6.1}$$

in which $u = u(x,t)$, or in the advective form

$$\frac{\partial \hat{\tau}}{\partial t} + u\frac{\partial \hat{\tau}}{\partial x} + \frac{\partial u}{\partial x}\hat{\tau} = 0, \tag{6.2}$$

describes the processes of advection and decay (or growth) taking place simultaneously. It is the advection term $u\partial\hat{\tau}/\partial x$ in (6.2) which is particularly difficult to approximate accurately. It has already been seen how the discretization of this term can lead to the introduction of artificial diffusion, wave-speed errors, numerical oscillations, and a general reduction in the order of convergence of the scheme, in situations where the velocity is variable. We were able to improve the accuracy of a variety of FDMs, thereby minimizing the negative effects of using low-order schemes. However, the term $\hat{\tau}\partial u/\partial x$ was neglected, so that the integral of $\hat{\tau}$ was generally not conserved. Attention will now be focussed on ways in which this term can be incorporated into the discretization process.

6.1 Conventional Methods

The purpose of this section is to present a few schemes which have been used to approximate (6.1) in the past. These particular schemes have been chosen because they are generalized versions of some of the base methods discussed in Chapter 3 for the non-conservative form of the advection equation. Other schemes can be found in Roberts and Weiss (1966), Crowley (1968), Boris and Book (1973, 1975, 1976) and Leonard (1991).

6.1.1 Crowley's Scheme

Crowley's approach was to consider the second term in (6.1) as the divergence of a flux and apply Green's theorem: "In any zone the decrease in ψ with time is proportional to the net flux out of the zone" (Crowley 1968). Here $\tau \equiv \psi$. The decrease in τ with time corresponds to forward-time differencing of the time derivative in (6.1), resulting in

$$\tau_j^{n+1} = \tau_j^n - \frac{\Delta t}{\Delta x}(F_{j+1/2}^n - F_{j-1/2}^n), \tag{6.3}$$

where $F_{j+1/2}$ is the flux across the boundary at $x_{j+1/2}$, defined as

$$F_{j+1/2} = (u\tau)_{j+1/2} = \frac{1}{\Delta t}\int_{x_{j+1/2}-u\Delta t}^{x_{j+1/2}} \tau(x',t)dx'. \tag{6.4}$$

The quantity τ was assumed to vary linearly between τ_j and τ_{j+1} and (6.4) integrated, yielding

$$F_{j+1/2}\frac{\Delta t}{\Delta x} = \frac{1}{2}c_{j+1/2}(\tau_{j+1} + \tau_j) - \frac{1}{2}c_{j+1/2}^2(\tau_{j+1} - \tau_j). \tag{6.5}$$

A corresponding expression for $F_{j-1/2}$ can be obtained by setting $-\Delta x$ for Δx, so $-c$ replaces c, and $j-1$ replaces $j+1$. Upon substitution in (6.3) Crowley's "second-order" flux-divergence scheme results:

$$\begin{aligned}\tau_j^{n+1} = {} & \frac{1}{2}c_{j-1/2}(1 + c_{j-1/2})\tau_{j-1}^n + \frac{1}{2}(2 - c_{j+1/2} - c_{j+1/2}^2 + c_{j-1/2} - c_{j-1/2}^2)\tau_j^n \\ & + \frac{1}{2}c_{j+1/2}(c_{j+1/2} - 1)\tau_{j-1}^n,\end{aligned} \tag{6.6}$$

where c is evaluated at time t_n. For constant velocity, Crowley's scheme, denoted CROW, corresponds to Leith's method. This advection scheme is first-order for non-uniform velocity fields, but can be constructed to have second-order convergence for arbitrary flow fields (Smolarkiewicz 1985).

6.1.2 Rusanov's Method

A generalized version of Rusanov's explicit fourth-order method for constant velocity, that can be used to solve (6.1), written in flux form (Steinle 1994), is given by

$$\begin{aligned}\tau_j^{n+1} = \tau_j^n - c_{\rm p}\tau_j^n + c_{\rm m}\tau_j^n - \frac{1}{2}c_{\rm p}(1-c_{\rm p})(\tau_{j+1}^n - \tau_j^n) + \frac{1}{2}c_{\rm m}(1-c_{\rm m})(\tau_j^n - \tau_{j-1}^n) \\ +\frac{1}{6}c_{\rm p}(1-c_{\rm p}^2)(\tau_{j+1}^n - 2\tau_j^n + \tau_{j-1}^n) - \frac{1}{6}c_{\rm m}(1-c_{\rm m}^2)(\tau_j^n - 2\tau_{j-1}^n + \tau_{j-2}^n) \\ +\frac{1}{24}c_{\rm p}(1-c_{\rm p}^2)(2-c_{\rm p})(\tau_{j+2}^n - 3\tau_{j+1}^n + 3\tau_j^n - \tau_{j-1}^n) \\ -\frac{1}{24}c_{\rm m}(1-c_{\rm m}^2)(2-c_{\rm m})(\tau_{j+1}^n - 3\tau_j^n + 3\tau_{j-1}^n - \tau_{j-2}^n),\end{aligned} \tag{6.7}$$

in which $c_{\rm m} = c_{j-1/2}^n$ and $c_{\rm p} = c_{j+1/2}^n$. This method will be denoted RUS2 to distinguish it from the scheme described in Chapter 3.

6.1.3 An Implicit Algorithm

An implicit scheme, denoted IMP, for non-uniform velocity fields is given by Steinle et al. (1989). This method takes the form

$$\begin{aligned}&(2 - 3c_{j-1/2} + c_{j-1/2}^2)\tau_{j-1}^{n+1} + (8 + 3c_{j+1/2} - 3c_{j-1/2} - c_{j+1/2}^2 - c_{j-1/2}^2)\tau_j^{n+1} \\ &\qquad +(2 + 3c_{j+1/2} + c_{j+1/2}^2)\tau_{j+1}^{n+1} \\ &= (2 + 3c_{j-1/2} + c_{j-1/2}^2)\tau_{j-1}^n + (8 - 3c_{j+1/2} + 3c_{j-1/2} - c_{j+1/2}^2 - c_{j-1/2}^2)\tau_j^n \\ &\qquad +(2 - 3c_{j+1/2} + c_{j+1/2}^2)\tau_{j+1}^n,\end{aligned} \tag{6.8}$$

which is the fourth-order OPT FDE if the velocity is constant. The authors use (6.8) as their high-order solution as part of a flux-corrected transport (FCT) algorithm. It should be noted that for variable velocities, (6.8) is only first-order when c is evaluated at t_n, or second-order if it is evaluated at $t_{n+1/2}$.

6.1.4 Numerical Test

The accuracy of the conventional methods is determined using the Gauss test. The data required to implement the Gauss test are summarized in Table 6.1. Since no exact solution is known when $T = \pi/2$, the result given by the second-order IMP method with $J = 10000$, and c evaluated at $t_{n+1/2}$, is assumed to represent the exact solution. The Thomas algorithm for periodic tridiagonal systems is used to solve the IMP method. The coefficients of the explicit methods are evaluated at time t_n, and the single unknown at the new time-level is given directly from the known values at the old time-level.

1. Initial condition $\hat{\tau}(x,0) = \exp\{-400(x-0.1)^2\}$

2. Periodic boundary conditions $\hat{\tau}(1+x,t) = \hat{\tau}(x,t)$

3. Courant number $c = u\Delta t/\Delta x = uTJ/N$

4. cmax $= 1$

5. $u(x,t) = \kappa(0.5 + \sin^2 \pi x)\cos t$

6. $T = \pi/2$, $\kappa = 4/3\pi$, $u\max = 2/\pi$: quarter cycle

7. $N = J$ with $J = 50, 100, 200, \ldots, 5000$

8. exact solution when $T = \pi/2$ is the solution of IMP when $J = 10000$

Table 6.1: Data for the Gauss Test applied to the conventional methods.

The rms errors in Table 6.2 show the first-order convergence of both Crowley's advection scheme and the generalized version of Rusanov's method. The second-order IMP method is the most accurate method. Because Crowley's scheme has the simplest coefficients and a (1,3) computational stencil, it requires the least time to run (see Table 6.2), whereas the implicit scheme is the most time consuming method.

Crowley's method is more efficient than the generalized Rusanov method. This is because it is faster, and gives more accurate results for all grids considered. The superior performance of the IMP method over the first-order schemes can be seen by the following example. Crowley's method used with $J = 1000$ takes 1.79 seconds and yields an error of 1.17e-03, whereas the IMP method used with $J = 100$ yields a smaller error, and takes only a fraction of the computational effort.

The cpu times in Table 6.2 indicate that doubling the grid number leads approximately to a fourfold increase in the computational times for all methods. Additionally, the generalized form of Rusanov's method requires approximately 1.37 times longer to run for each grid than Crowley's method, while the implicit method takes about 1.85 times longer than Crowley's method for each grid.

	rms errors		
J	CROW	RUS2	IMP
50	1.74e−02	2.32e−02	4.47e−03
100	8.62e−03	1.21e−02	1.12e−03
200	5.00e−03	6.07e−03	2.80e−04
500	2.24e−03	2.43e−03	4.47e−05
1000	1.17e−03	1.21e−03	1.11e−05
2000	5.95e−04	6.07e−04	2.69e−06
5000	2.41e−04	2.43e−04	3.36e−07
	cpu times		
J	CROW	RUS2	IMP
50	4.40e−03	6.12e−03	8.20e−03
100	1.88e−02	2.47e−02	3.36e−02
200	7.14e−02	9.80e−02	1.32e−01
500	4.42e−01	6.15e−01	8.09e−01
1000	1.79e+00	2.46e+00	3.25e+00
2000	7.47e+00	9.84e+00	1.34e+01
5000	4.42e+01	6.15e+01	8.12e+01

Table 6.2: Results of the conventional methods in the Gauss test.

6.2 A Discretization Procedure

In Chapter 3, a technique was outlined which could modify any FDM for the non-conservative advection equation, so that it could retain its constant-coefficient order in the variable-coefficient situation. This idea may be extended to incorporate the decay term present in (6.2). Consider (6.2) with the decay term interpreted, for the moment, as a sink term

$$F = -\frac{\partial u}{\partial x}\hat{\tau}, \tag{6.9}$$

then a FDM to approximate the solution of (6.2) may be based on any scheme for the non-conservative advection equation plus an additional term $\Delta t F_j^n$, to take into account the sink term. Take for example Leith's method: a solution to (6.2) can then be approximated from

$$\tau_j^{n+1} = \frac{1}{2}c(c+1)\tau_{j-1}^n + (1-c^2)\tau_j^n + \frac{1}{2}c(c-1)\tau_{j+1}^n + \Delta t F_j^n, \tag{6.10}$$

where the superscript n and subscript j have been omitted from the non-constant Courant number c_j^n.

The order of (6.10) is assessed by taking a Taylor expansion of each term about (x_j, t_n), giving

$$\mathcal{F}\{\hat{\tau}\} = -\Delta t E_{\mathrm{T}}, \tag{6.11}$$

where the truncation error is given by

$$E_{\mathrm{T}} = -\frac{1}{2}\Delta t\left(\frac{\partial^2\hat{\tau}}{\partial t^2} - u^2\frac{\partial^2\hat{\tau}}{\partial x^2}\right) + O\{2\}. \tag{6.12}$$

The time derivative is converted to space derivatives by differentiating (6.2) with respect to t, and then eliminating the cross derivative by differentiating (6.2) with respect to x, yielding

$$\frac{\partial^2\hat{\tau}}{\partial t^2} = -\left(\frac{\partial u}{\partial t} - u\frac{\partial u}{\partial x}\right)\frac{\partial\hat{\tau}}{\partial x} + u^2\frac{\partial^2\hat{\tau}}{\partial x^2} + \left(\frac{\partial F}{\partial t} - u\frac{\partial F}{\partial x}\right). \tag{6.13}$$

Substituting (6.13) into (6.11) produces

$$\mathcal{F}\{\hat{\tau}\} = -d\left[\Delta x\frac{\partial\hat{\tau}}{\partial x}\right] + \Delta t G + O\{3\}, \tag{6.14}$$

in which

$$d = \frac{(\Delta t)^2}{2\Delta x}\left(\frac{\partial u}{\partial t} - u\frac{\partial u}{\partial x}\right), \tag{6.15}$$

and

$$G = \frac{\Delta t}{2}\left(\frac{\partial F}{\partial t} - u\frac{\partial F}{\partial x}\right). \tag{6.16}$$

Equation (6.10) is clearly accurate to first-order. To obtain a second-order truncation error, the spatial derivative in (6.14) must be discretized to at least first-order; it can be approximated using either upwind or centered-space difference forms as before. This yields

$$\hat{\tau}_j^{n+1} = \mathrm{LTH} + \Delta t\left(F_j^n + G_j^n\right) + O\{3\}, \tag{6.17}$$

where

$$\mathrm{LTH} = \frac{1}{2}\left(c + c^2 + d + |d|\right)\hat{\tau}_{j-1}^n + \left(1 - c^2 - |d|\right)\hat{\tau}_j^n - \frac{1}{2}\left(c - c^2 + d - |d|\right)\hat{\tau}_{j+1}^n, \tag{6.18}$$

in the former case, or

$$\mathrm{LTH} = \frac{1}{2}\left(c + c^2 + d\right)\hat{\tau}_{j-1}^n + \left(1 - c^2\right)\hat{\tau}_j^n - \frac{1}{2}\left(c - c^2 + d\right)\hat{\tau}_{j+1}^n, \tag{6.19}$$

in the latter case. The new term, involving G, must be accurate to third-order, so G must be approximated to second-order. This may be achieved by computing the derivatives of F to at least first-order. If first-order forward-time and second-order centered-space forms are used, then

$$G \approx \frac{1}{4}\left\{2(F_j^{n+1} - F_j^n) - c(F_{j+1}^n - F_{j-1}^n)\right\}. \tag{6.20}$$

Finally, substituting (6.20) into (6.17) and omitting all error terms, yields a second-order method for approximating the conservative form of the advection equation, namely

$$\tau_j^{n+1} = \text{LTH} + \Delta t F_j^n + \frac{1}{4}\Delta t\left\{2(F_j^{n+1} - F_j^n) - c(F_{j+1}^n - F_{j-1}^n)\right\}, \tag{6.21}$$

where $F = -\tau \partial u/\partial x$. Rearranging (6.21) gives an explicit three-point formula, that is

$$\tau_j^{n+1} = \frac{1}{g}(\text{LTH} + \frac{1}{4}\Delta t\{cF_{j-1}^n + 2F_j^n - cF_{j+1}^n\}), \tag{6.22}$$

in which

$$g = 1 - 0.5\Delta t f_j^{n+1} \quad \text{and} \quad f = -\partial u/\partial x. \tag{6.23}$$

Because the dominant term in the truncation error of the UW15 method is the same as that for Leith's method, (6.22) can readily be used with UW15 replacing LTH. Note that since the equations involved are linear, only the stability of the homogeneous part of the FDE, obtained by setting $F = 0$, needs to be determined. A local stability analysis then yields necessary conditions for the stability of the variable coefficient problem (cf. Section 3.4.3).

If higher than second-order convergence is desired, for instance, by using Rusanov's method in place of either the Leith or UW15 schemes, then the conversion of the temporal derivatives in the truncation error to space derivatives implies the inclusion of many more terms like (6.16). Each of these must then be approximated to the appropriate order. Three levels in time will necessarily be involved. Clearly this modification procedure becomes very complicated and is suitable for only the simpler base schemes.

6.2.1 Numerical Test

The Gauss test, summarized in Table 6.3, is used to examine the performance of the schemes developed in this section. Since no exact solution is available when $T = \pi/2$, the numerical solution given by the second-order mod2_L method when $J = 20000$, is assumed to represent the exact solution. If one of the other second-order methods is used to give the exact solution with this grid number, the rms errors do not change, indicating that this numerical solution is accurate enough to represent the exact solution.

1. Initial condition $\hat{\tau}(x,0) = \exp\{-400(x-0.1)^2\}$

2. Periodic boundary conditions $\hat{\tau}(1+x,t) = \hat{\tau}(x,t)$

3. Courant number $c = u\Delta t/\Delta x = uTJ/N$

4. $c\max = 1$

5. $u(x,t) = \kappa(0.5 + \sin^2 \pi x)\cos t$

6. $T = \pi/2$, $\kappa = 4/3\pi$, $u\max = 2/\pi$: quarter cycle

7. $N = J$ with $J = 50, 100, 200, \ldots, 5000$

8. exact solution when $T = \pi/2$ is the solution of mod2_L when $J = 20000$

Table 6.3: Data for the Gauss Test applied to the discretized methods.

Figure 6.1 shows that Leith's method captures the height of the pulse better than the mod_L and mod2_L methods. The mod2_L method is less diffusive than the mod_L method because of the centered space differencing used to modify it, as opposed to the diffusive upwind differencing used for the mod_L method. For similar reasons, the mod2_U solution in Figure 6.1 is less diffuse than the mod_U solution. The spurious oscillations present in the solutions for $J = 50$, are not visible when $J = 100$.

The solution of the UW15 method in Figure 6.1 leads the exact solution with an amplitude that has apparently not decayed sufficiently. This is explained by examining the term $\partial u/\partial x$. For the velocity considered, $\partial u/\partial x$ is proportional to $\sin 2\pi x$, so when $x \in (0, 1/2)$ this term is positive and decay occurs, but when $x \in (1/2, 1)$ this term is negative and growth occurs. Because the pulse travels too rapidly, the growth which should occur later is already taking effect.

Figures 6.2 and 6.3 show that Leith's method is more accurate and more efficient than the UW15 method. The accuracy and efficiency of the second-order schemes cannot be separated from these figures, however, Tables 6.4 and 6.5 indicate that the mod2_L method is the most accurate and most efficient second-order scheme. This is because it gives the smallest errors in the least time. Comparing Leith's method and Crowley's scheme (cf. Section 6.1.1), shows that, although Leith's method is a little more accurate, it takes more than twice the cpu time for each grid (compare Tables 6.4 and 6.5 with Table 6.2).

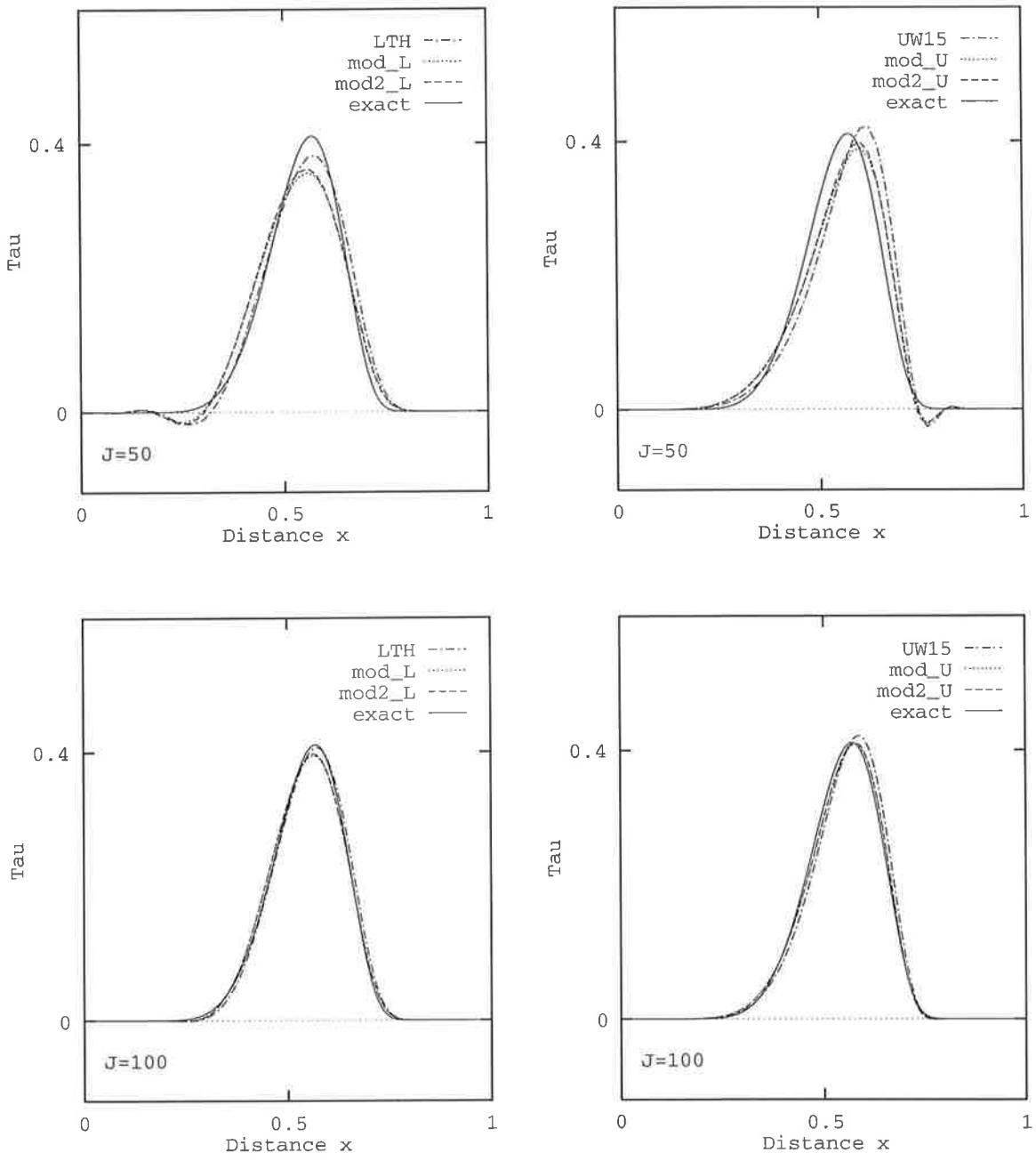

Fig 6.1: The discretised methods are used in the Gauss test for the one-dimensional conservative advection equation. The top diagrams are for $J = 50$ and those on the bottom are for $J = 100$.

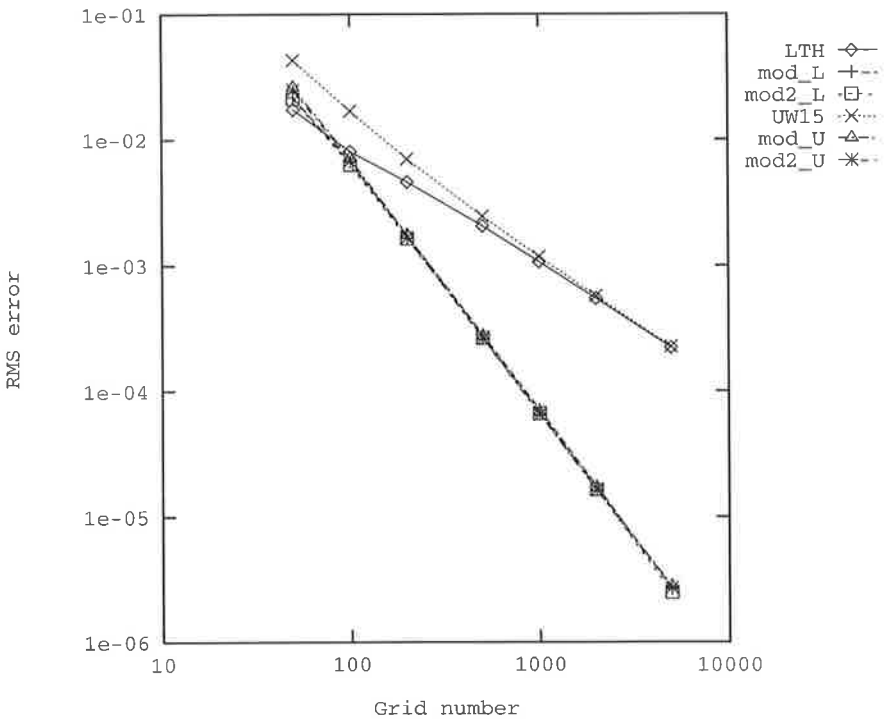

Figure 6.2: Convergence shown for the discretized methods.

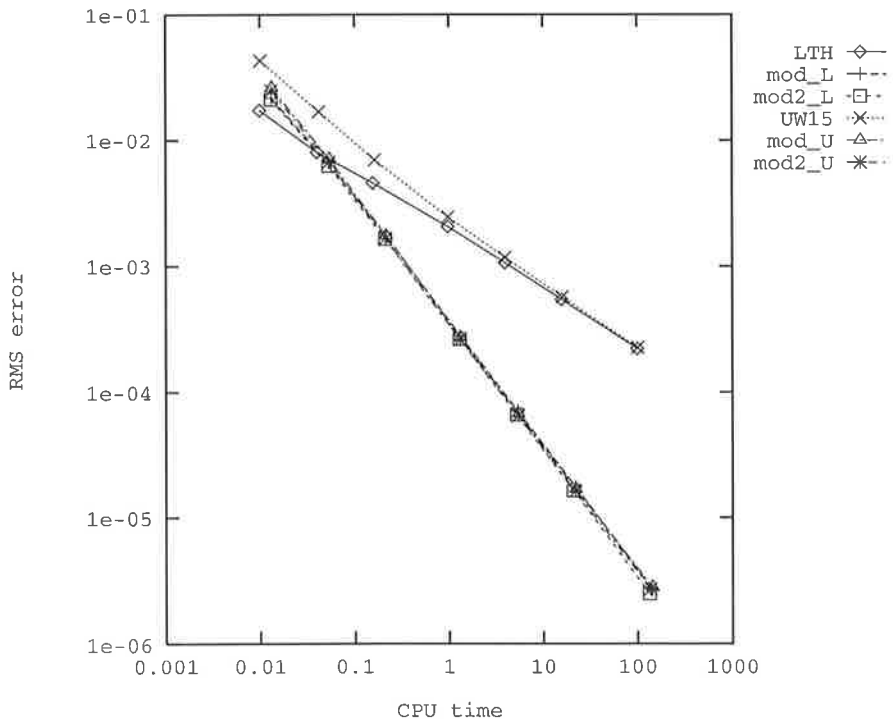

Figure 6.3: Efficiency shown for the discretized methods.

J	LTH	UW15
50	1.73e−02	4.30e−02
100	8.04e−03	1.69e−02
200	4.58e−03	6.98e−03
500	2.06e−03	2.45e−03
1000	1.07e−03	1.17e−03
2000	5.47e−04	5.71e−04
5000	2.22e−04	2.26e−04
J	mod_L	mod_U
50	2.17e−02	2.61e−02
100	6.60e−03	7.10e−03
200	1.74e−03	1.74e−03
500	2.82e−04	2.71e−04
1000	7.06e−05	6.71e−05
2000	1.76e−05	1.68e−05
5000	2.67e−06	2.79e−06
J	mod2_L	mod2_U
50	2.11e−02	2.48e−02
100	6.28e−03	6.74e−03
200	1.64e−03	1.66e−03
500	2.65e−04	2.61e−04
1000	6.61e−05	6.47e−05
2000	1.64e−05	1.62e−05
5000	2.49e−06	2.71e−06

Table 6.4: RMS errors for discretized methods in Gauss test.

J	LTH	UW15
50	9.90e−03	1.01e−03
100	4.00e−02	4.18e−02
200	1.56e−01	1.63e−01
500	9.70e−01	9.77e−01
1000	3.91e+00	3.93e+00
2000	1.57e+01	1.60e+01
5000	1.00e+02	1.01e+02
J	mod_L	mod_U
50	1.32e−02	1.33e−02
100	5.30e−02	5.39e−02
200	2.14e−01	2.15e−01
500	1.31e+00	1.32e+00
1000	5.34e+00	5.43e+00
2000	2.20e+01	2.21e+01
5000	1.42e+02	1.43e+02
J	mod2_L	mod2_U
50	1.31e−02	1.32e−02
100	5.29e−02	5.37e−02
200	2.10e−01	2.12e−01
500	1.30e+00	1.31e+00
1000	5.23e+00	5.41e+00
2000	2.09e+01	2.19e+01
5000	1.33e+02	1.37e+02

Table 6.5: CPU times for discretized methods in Gauss test.

6.3 Process Splitting

Rather than developing FDMs which approximate solutions to (6.2) directly in the manner just described, it is possible to divide the governing equation, solving the processes of advection and decay separately during each time-step. Consider the component equations

$$\frac{\partial \hat{\tau}}{\partial t} + u\frac{\partial \hat{\tau}}{\partial x} = 0, \tag{6.24}$$

$$\frac{\partial \hat{\tau}}{\partial t} + \frac{\partial u}{\partial x}\hat{\tau} = 0, \tag{6.25}$$

the first of which can be solved using any of the methods described in Chapter 3. The second equation may be solved either by using an exact solution, or by using any sufficiently accurate ODE solver, such as Heun's method for second-order accuracy or the fourth-order Runge-Kutta (RK4) method for higher accuracy. Assuming that both $\hat{\tau}(x,t)$ and the advective velocity are variable separable with $u(x,t) = f(x)g(t)$, then an exact solution to (6.25), written in discrete form, is given by

$$\hat{\tau}_j^{n+1} = \hat{\tau}_j^n \exp\{-f'(x_j)\int_{t_n}^{t_{n+1}} g(s)ds\}, \tag{6.26}$$

where $f' = df/dx$. Because the coefficients of the component equations are not constant, there is an error in splitting even if the subprocesses are treated exactly (Vreugdenhil and Koren 1993). This error can be avoided by making the splitting process symmetric (Strang 1968), which involves reversing the order of the subprocesses each time step (Vreugdenhil and Koren 1993).

6.3.1 Numerical Test

The Gauss test was applied to all of the methods described in Chapter 3. The decay process was approximated by applying Heun's method, where first or second-order convergence was expected (recall the synopsis given in Figure 3.1), or by using the RK4 method for higher convergence. A summary of the data required to implement the numerical test is given in Table 6.6.

The coefficients of the explicit methods are evaluated at time t_n, and the single unknown at the new time-level is given directly from the known values at the old time-level. The coefficients of the implicit methods are evaluated at $t_{n+1/2}$, and the Thomas algorithm for periodic tridiagonal systems is used to solve the system of linear algebraic equations arising.

1. Initial condition $\hat{\tau}(x,0) = \exp\{-400(x-0.1)^2\}$

2. Periodic boundary conditions $\hat{\tau}(1+x,t) = \hat{\tau}(x,t)$

3. Courant number $c = u\Delta t/\Delta x = uTJ/N$

4. $c\max = 1$

5. $u(x,t) = \kappa(0.5 + \sin^2 \pi x)\cos t$

6. $T = \pi/2$, $\kappa = 4/3\pi$, $u\max = 2/\pi$: quarter cycle

7. $T = 2\pi$, $\kappa = 1/3\pi$, $u\max = 1/2\pi$: one cycle

8. $N = J$ with $J = 50, 100, 200, \ldots, 5000$

9. exact solution when $T = \pi/2$ is the solution of mod2_O when $J = 10000$

10. exact solution when $T = 2\pi$ is the initial condition

Table 6.6: Data for the Gauss Test applied to the process splitting methods.

The diagrams in Figure 6.4 show the results for $J = 50$ after a quarter cycle. A comparison of the top two diagrams in Figures 6.1 and 6.4 indicates that in terms of accuracy there is little difference in performance between the process splitting methods and those used in the discretization procedure. This is verified by comparing their rms errors in Tables 6.4 and 6.7. However, Tables 6.5 and 6.8 show that the process splitting technique is considerably faster to execute than the discretization procedure.

The bottom left diagram in Figure 6.4 shows that the solution of Rusanov's method lies well ahead of the exact solution, and the height of the pulse is too large. This is due to the term $\partial u/\partial x$; growth occurs when $x \in (1/2, 1)$. The explicit third-order mod_R and mod2_R methods do not introduce any visible oscillations into the numerical solution, and are closely aligned to the exact solution. However, they introduce more diffusion into the solution than the implicit mod_O and mod2_O methods (compare bottom diagrams in Figure 6.4), causing a reduction in the height of the pulse.

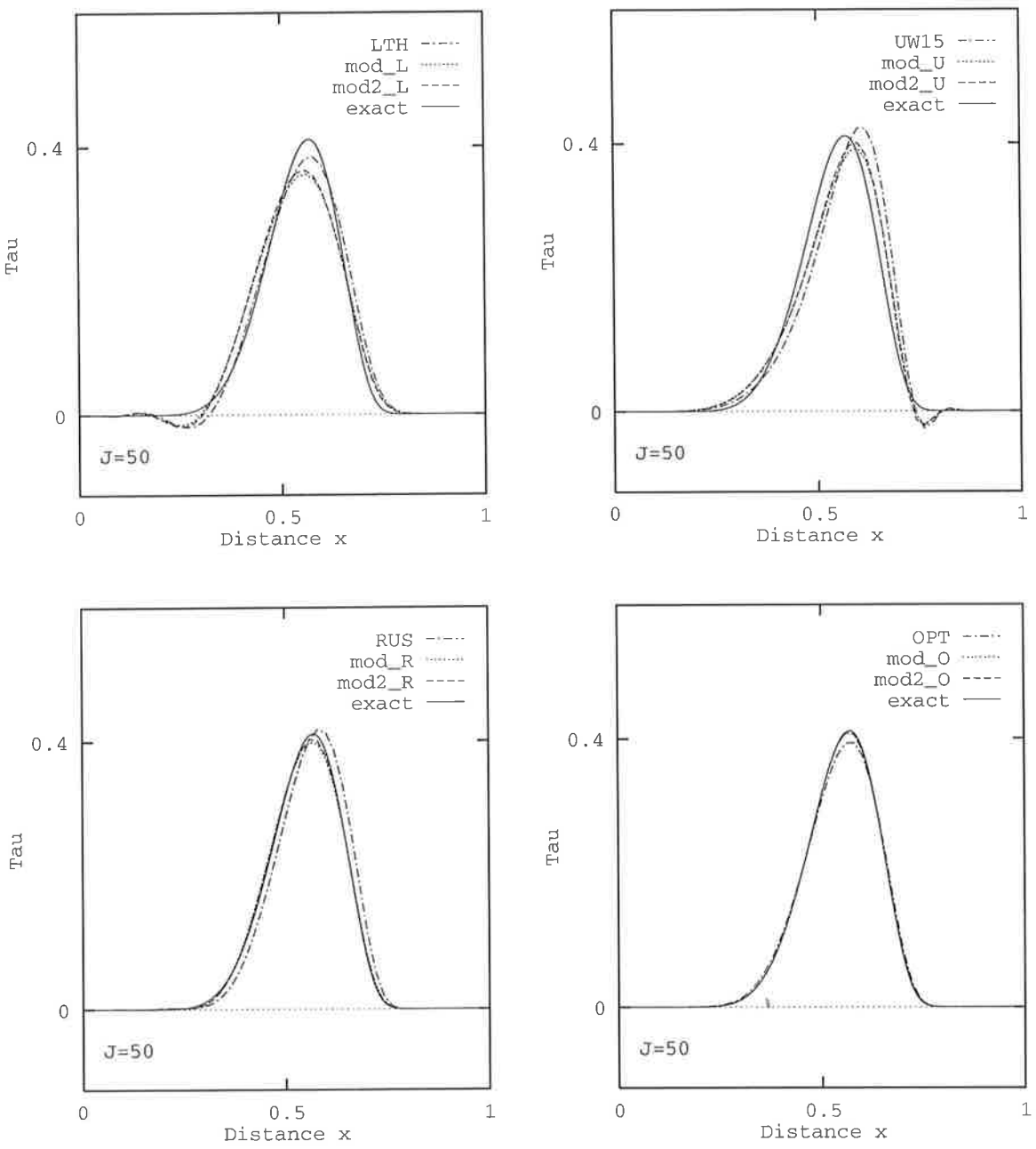

Figure 6.4: The Gauss test is applied for $J = 50$ to the process splitting methods to approximate the one-dimensional conservative advection equation.

The solution of the OPT method (bottom right diagram of Figure 6.4) is better aligned to the exact solution than the other base methods (LTH, UW15, RUS), and captures the height of the pulse better than LTH and UW15. Its solution is also better aligned to the exact solution than those of the explicit second-order schemes. Additionally, it introduces no visible oscillations into the solution. The implicit mod_O and mod2_O methods (whose performance cannot be separated in Figure 6.4) give the most accurate solutions, with only a small loss in the height of the pulse observed.

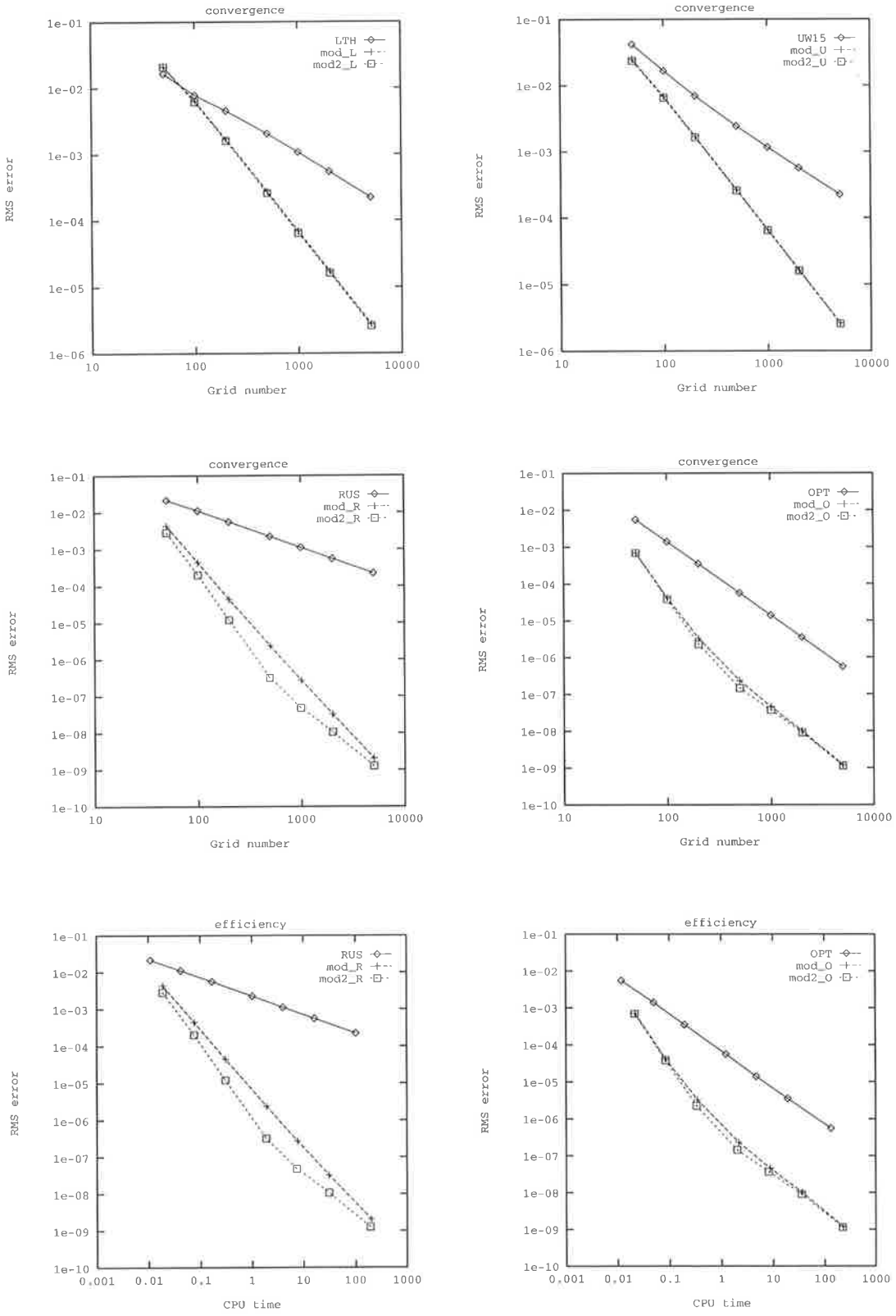

Figure 6.5: The top four diagrams show the convergence of the process splitting methods. The bottom two diagrams show the efficiency of RUS, OPT and their modifications.

J	LTH	UW15	RUS	OPT
50	1.65e−02	4.20e−02	2.15e−02	5.49e−03
100	7.82e−03	1.67e−02	1.12e−02	1.39e−03
200	4.54e−03	6.94e−03	5.60e−03	3.50e−04
500	2.05e−03	2.44e−03	2.24e−03	5.61e−05
1000	1.07e−03	1.17e−03	1.12e−03	1.40e−05
2000	5.47e−04	5.71e−04	5.59e−04	3.51e−06
5000	2.22e−04	2.25e−04	2.23e−04	5.61e−07
J	mod_L	mod_U	mod_R	mod_O
50	2.12e−02	2.49e−02	4.32e−03	6.92e−04
100	6.44e−03	6.86e−03	4.41e−04	4.15e−05
200	1.69e−03	1.70e−03	4.44e−05	3.28e−06
500	2.75e−04	2.67e−04	2.39e−06	2.31e−07
1000	6.88e−05	6.61e−05	2.76e−07	4.54e−08
2000	1.72e−05	1.64e−05	3.25e−08	9.99e−09
5000	2.75e−06	2.62e−06	2.03e−09	1.19e−09
J	mod2_L	mod2_U	mod2_R	mod2_O
50	2.08e−02	2.37e−02	2.86e−03	6.90e−04
100	6.16e−03	6.54e−03	2.01e−04	3.79e−05
200	1.60e−03	1.64e−03	1.20e−05	2.25e−06
500	2.59e−04	2.58e−04	3.20e−07	1.44e−07
1000	6.47e−05	6.43e−05	4.85e−08	3.67e−08
2000	1.62e−05	1.60e−05	1.06e−08	9.08e−09
5000	2.59e−06	2.56e−06	1.25e−09	1.14e−09

Table 6.7: RMS errors after a quarter time cycle when the Gauss test is applied to the one-dimensional conservative advection problem.

J	LTH	UW15	RUS	OPT
50	6.18e−03	6.60e−03	1.10e−02	1.19e−02
100	2.69e−02	2.76e−02	4.23e−02	4.99e−02
200	1.05e−01	1.07e−01	1.70e−01	1.95e−01
500	6.32e−01	6.70e−01	1.02e+00	1.23e+00
1000	2.49e+00	2.69e+00	4.07e+00	4.77e+00
2000	1.06e+01	1.08e+01	1.64e+01	1.94e+01
5000	6.21e+01	6.38e+01	1.02e+02	1.33e+02
J	mod_L	mod_U	mod_R	mod_O
50	9.90e−03	1.04e−02	1.92e−02	2.20e−02
100	4.12e−02	4.17e−02	7.68e−02	8.41e−02
200	1.56e−01	1.72e−01	3.08e−01	3.43e−01
500	9.73e−01	1.06e+00	1.92e+00	2.17e+00
1000	4.01e+00	4.19e+00	7.71e+00	8.77e+00
2000	1.60e+01	1.63e+01	3.09e+01	3.67e+01
5000	9.61e+01	1.02e+02	2.01e+02	2.32e+02
J	mod2_L	mod2_U	mod2_R	mod2_O
50	9.59e−03	1.02e−02	1.90e−02	2.15e−02
100	3.93e−02	3.95e−02	7.65e−02	8.38e−02
200	1.54e−01	1.56e−02	3.07e−01	3.22e−01
500	9.55e−01	9.88e−01	1.89e+00	2.04e+00
1000	3.80e+00	3.82e+00	7.60e+00	8.21e+00
2000	1.54e+01	1.55e+01	3.08e+01	3.56e+01
5000	9.60e+01	1.00e+02	1.92e+02	2.24e+02

Table 6.8: CPU times after a quarter time cycle when the Gauss test is applied to the one-dimensional conservative advection problem.

Method	$T = \pi/2$	$T = 2\pi$	Theory
LTH	0.92	1.07	1
mod_L	1.96	2.05	2
mod2_L	1.96	2.92	2
UW15	1.13	1.17	1
mod_U	2.00	2.15	2
mod2_U	1.99	2.87	2
RUS	1.00	1.02	1
mod_R	3.17	3.30	3
mod2_R	3.22	3.11	3
OPT	2.00	–	2
mod_O	2.83	3.01	3
mod2_O	2.83	–	4

Table 6.9: The orders of convergence for the Gauss test are given by the slope of the line of best fit to the data.

The top four diagrams in Figure 6.5 show the convergence of the schemes after a quarter time cycle. For most methods the results are as expected, but for the mod2_R, mod_O and mod2_O schemes, the convergence declines as the grid is refined. This decrease in accuracy causes a corresponding decrease in the efficiency of these schemes, as seen in the bottom two diagrams of Figures 6.5. Apparently, it is only slightly more profitable to use the fourth-order mod2_O method than the third-order mod_O method.

The convergence rates at both final times are given in Table 6.9. These were determined by performing a least squares fit to the data given in Tables 6.7 and B.2 (Appendix B). The convergence rate is given by the slope of the line of best fit to the data. After one cycle, both the mod2_R and mod_O methods gave results which have the expected convergence rates, and the second-order mod2_L and mod2_U methods performed better than expected, giving nearly third-order convergence.

The phenomenon called order reduction (see Sanz-Serna et al. 1987) may be responsible for the decrease in accuracy of the third and fourth-order methods when the RK4 method is used to approximate the decay term. It has been observed (Sanz-Serna et al. 1987) that in many problems involving the time integration of hyperbolic equations using Runge-Kutta methods, the convergence rate is less than the theoretical order. According to Sanz-Serna et al. (1987), order reduction occurs unless certain boundary conditions are fulfilled. These conditions are not natural to the problem, but arise as constraints when the Runge-Kutta method is used (Sanz-Serna et al. 1987). Such constraints will not be considered here.

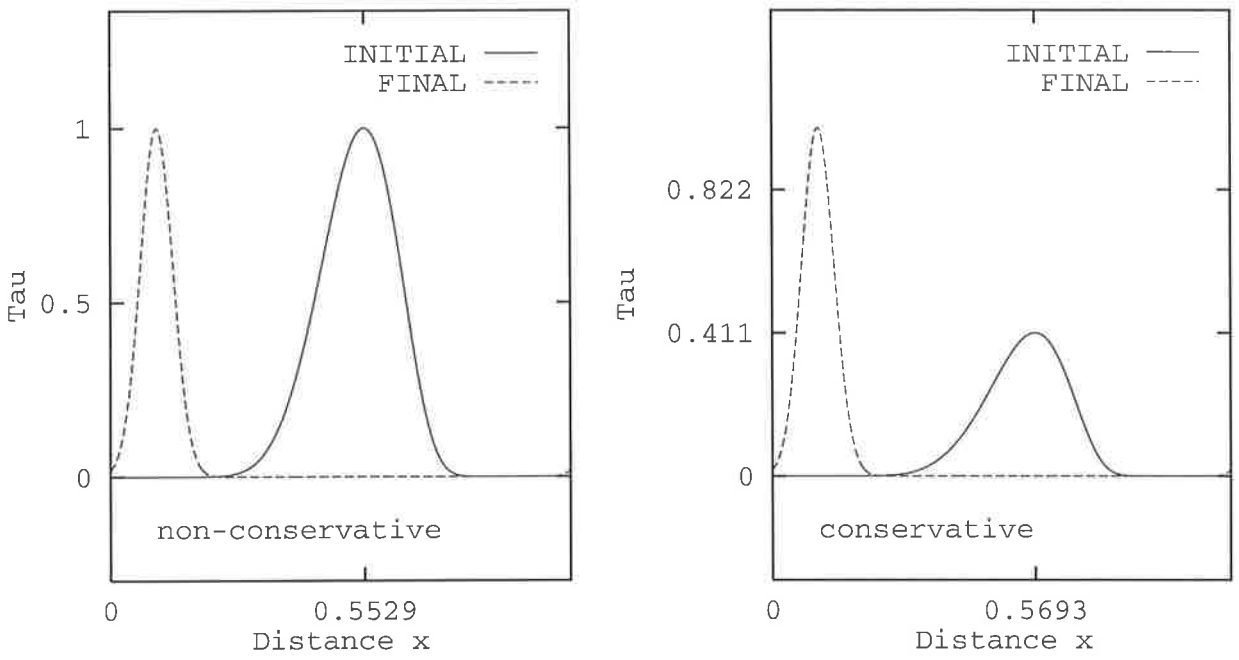

Figure 6.6: Solution of the non-conservative and conservative advection equation when $T = \pi/2$.

It is observed that the exact solution of the non-conservative and conservative forms of the advection equation to the same problem may be different. In Figure 6.6, the initial condition and exact solution at $T = \pi/2$ are illustrated for the Gauss test problem. In both cases, the exact solution is given by solving the advection equation using the implicit fourth-order mod2_O method with $J = 10000$. As shown in the diagrams, the peak of the non-conservative solution is located at $x_p = 5.529\text{e-}01$ with $\hat{\tau}_{max} = 1$, while the peak of the conservative solution is located 164 grid points to the right at $x_p = 5.693\text{e-}01$ with $\hat{\tau}_{max} \approx 0.411$. In other words, the concentration front travels slower in the non-conservative case and the peak is unattenuated. By contrast, the solution profile of the conservative equation decays exponentially in time. Similar observations have been made by Zoppou and Knight (1997a), who compared analytical solutions for the conservative and non-conservative forms of the advection equation with spatially variable coefficients. An approximation of the area under the profiles of the conservative and non-conservative solutions is given by Vreugdenhil (1993) as

$$A^n = \frac{1}{J}\sum_{j=0}^{J} \hat{\tau}_j^n,$$

giving an initial and final area of 8.86e-02 in the conservative case, but the area has increased to 2.10e-01 in the non-conservative case. For many problems, it is important that the conservative form is used, and this has been addressed by incorporating the decay term into the approximation. Note that after one cycle, the exact solution for both the conservative and non-conservative forms of the advection equation, is given by the initial condition. Hence, for this final time, both equations are equivalent.

6.4 Summary

Various techniques to incorporate the decay term into the non-conservative advection equation, so that the integral of $\hat{\tau}$ is conserved, were considered. Conventional methods based on the integral formulation of (6.1) and the flux of matter across the boundaries of a computational cell, were compared to a new discretization procedure and a process splitting method. Although the conventional methods were of low-order, they performed quite well compared to the base methods, described in Sections 6.2 and 6.3. The discretization procedure and the process splitting method gave similar accuracy, though the former technique was too complicated to carry out for all but the simplest base schemes. The latter technique was easy to implement, required less cpu time, and could be readily applied to all the schemes considered for the non-conservative advection equation, without alteration. Although some of the third and fourth-order methods did not show the expected convergence rates for finer grid resolutions, these methods were nevertheless still superior to all others.

Chapter 7

One-Dimensional Diffusion: A Special Case

The one-dimensional diffusion equation

$$\frac{\partial \hat{\tau}}{\partial t} - \frac{\partial}{\partial x}\left(\alpha \frac{\partial \hat{\tau}}{\partial x}\right) = 0, \tag{7.1}$$

in which the diffusion coefficient $\alpha(x,t) > 0$ is space and time dependent, quantifies the process of diffusion, which is involved in many physical situations, such as solute flow through heterogeneous porous media, and conduction phenomena involving transfer of heat in nonhomogeneous solids. Closure of the initial value problem within the domain $0 \leq x \leq 1$, $0 \leq t \leq T$ requires the prescription of an initial condition

$$\hat{\tau}(x,0) = f(x), \quad 0 \leq x \leq 1 \tag{7.2}$$

and boundary conditions

$$\begin{aligned} \hat{\tau}(0,t) &= g_{\mathrm{L}}(t), \quad 0 \leq t \leq T \\ \hat{\tau}(1,t) &= g_{\mathrm{R}}(t), \quad 0 \leq t \leq T \end{aligned} \tag{7.3}$$

where $f(x)$, $g_{\mathrm{L}}(t)$ and $g_{\mathrm{R}}(t)$ are known.

Considerable research and experimental studies have been undertaken to determine the form of the variable diffusion coefficient for a wide range of practical applications (see for example Crank 1956, Corey et al. 1970, Pickens and Grisak 1981a, Yates 1992).

In many applications, the diffusion coefficient is a function of travel time only (Barry and Sposito 1989, Basha and El-Habel 1993), so the governing equation simplifies to

$$\frac{\partial \hat{\tau}}{\partial t} - \alpha(t)\frac{\partial^2 \hat{\tau}}{\partial x^2} = 0. \tag{7.4}$$

In this chapter, the accuracy of various base methods will be analyzed, when they are applied to (7.4), with α in its most general form, namely

$$\frac{\partial \hat{\tau}}{\partial t} - \alpha(x,t)\frac{\partial^2 \hat{\tau}}{\partial x^2} = 0. \tag{7.5}$$

The relevance of considering the more general form (7.5), rather than (7.4), will become apparent in Chapter 9. Any methods developed to solve (7.5), will perform at least as well, when applied to (7.4).

As was the case for FDMs used to solve the variable-coefficient advection equation, high-order methods for constant diffusion usually become low-order when applied to variable diffusion problems. It will be shown how these methods can be modified, so as to retain their constant coefficient order.

7.1 The MEPDE

The MEPDE of a FDE which is consistent with the constant coefficient diffusion equation is given by

$$\frac{\partial \tau}{\partial t} - \alpha\frac{\partial^2 \tau}{\partial x^2} + \sum_{q=3}^{\infty} \frac{2\alpha(\Delta x)^{q-2}}{q!}\Gamma_q(s)\frac{\partial^q \tau}{\partial x^q} = 0. \tag{7.6}$$

If $\Gamma_q(s) = 0$ for $q = 3(1)Q+1$ and $\Gamma_{Q+2}(s) \neq 0$, then the FDE is of order Q, since the truncation error

$$E_{\mathrm{T}} = \sum_{q=Q+2}^{\infty} \frac{2\alpha(\Delta x)^{q-2}}{q!}\Gamma_q(s)\frac{\partial^q \tau}{\partial x^q}, \tag{7.7}$$

is $O\{(\Delta x)^Q\}$. A shift in the peak of the numerical solution occurs unless the odd coefficients are all zero. The even coefficients indicate that there is an error in the damping response (Noye and Hayman 1986a).

For variable coefficients, an analysis must be based on the MDPDE, obtained by replacing τ by $\hat{\tau}$ in the FDE, expanding each term about a fixed point in the solution domain, and then eliminating all the temporal derivatives using the original PDE. The analysis based on either the MEPDE or the MDPDE requires the solution to be continuously differentiable on the domain.

7.2 The FTCS Method

For constant diffusion, the FTCS method (Richtmyer and Morton 1967) is given by

$$\tau_j^{n+1} = s\tau_{j-1}^n + (1-2s)\tau_j^n + s\tau_{j+1}^n. \tag{7.8}$$

For variable coefficients, s is replaced by $s_j^n = \alpha_j^n \Delta t/(\Delta x)^2$. The superscript n and subscript j will be omitted from the following for convenience. A formal consistency analysis of (7.8) about the $(j,n)^{th}$ grid point gives the truncation error

$$E_{\mathrm{T}} = -\frac{\Delta t}{2}\left(\frac{\partial^2 \hat{\tau}}{\partial t^2} - \frac{1}{6}\alpha^2 s^{-1}\frac{\partial^4 \hat{\tau}}{\partial x^4}\right) + O\{4\}, \tag{7.9}$$

where

$$O\{p\} = O\{(\Delta x)^{p-2q}(\Delta t)^q,\ q \text{ an integer}\}, \tag{7.10}$$

since Δt is proportional to $(\Delta x)^2$. To determine whether terms in (7.9) cancel, the time derivative is converted to space derivatives as follows. Differentiating (7.5) with respect to t gives

$$\frac{\partial^2 \hat{\tau}}{\partial t^2} = \frac{\partial \alpha}{\partial t}\frac{\partial^2 \hat{\tau}}{\partial x^2} + \alpha\frac{\partial^3 \hat{\tau}}{\partial t \partial x^2}, \tag{7.11}$$

while differentiating (7.5) with respect to x, twice, yields

$$\frac{\partial^3 \hat{\tau}}{\partial t \partial x^2} = \frac{\partial^2 \alpha}{\partial x^2}\frac{\partial^2 \hat{\tau}}{\partial x^2} + 2\frac{\partial \alpha}{\partial x}\frac{\partial^3 \hat{\tau}}{\partial x^3} + \alpha\frac{\partial^4 \hat{\tau}}{\partial x^4}. \tag{7.12}$$

Substituting (7.12) into (7.11) produces

$$\frac{\partial^2 \hat{\tau}}{\partial t^2} = \left(\frac{\partial \alpha}{\partial t} + \alpha\frac{\partial^2 \alpha}{\partial x^2}\right)\frac{\partial^2 \hat{\tau}}{\partial x^2} + 2\alpha\frac{\partial \alpha}{\partial x}\frac{\partial^3 \hat{\tau}}{\partial x^3} + \alpha^2\frac{\partial^4 \hat{\tau}}{\partial x^4}, \tag{7.13}$$

which contains no time derivatives of $\hat{\tau}$. The truncation error then becomes

$$E_{\mathrm{T}} = -\frac{\Delta t}{2}\left(E\frac{\partial^2 \hat{\tau}}{\partial x^2} + K\frac{\partial^3 \hat{\tau}}{\partial x^3} + \alpha^2(1 - \frac{1}{6}s^{-1})\frac{\partial^4 \hat{\tau}}{\partial x^4}\right) + O\{4\}, \tag{7.14}$$

where

$$E = \frac{\partial \alpha}{\partial t} + \alpha\frac{\partial^2 \alpha}{\partial x^2}, \quad K = 2\alpha\frac{\partial \alpha}{\partial x}. \tag{7.15}$$

The FTCS method is therefore second-order convergent, independent of whether it is used for variable or constant coefficients, except in the case $s = 1/6$, when it becomes fourth-order for constant diffusion. It is locally stable if $0 < s \leq 1/2$ (Richtmyer and Morton 1967).

7.3 The Noye-Hayman Method

Consider the Noye-Hayman method (Noye and Hayman 1986b), written in the form (2.12), where

$$\mathcal{L}\{\tau\} = \tau_j^{n+1}$$

$$\mathcal{R}\{\tau\} = \frac{1}{12}(6s^2 - s)\tau_{j-2}^n + \frac{1}{3}(-6s^2 + 4s)\tau_{j-1}^n + \frac{1}{2}(2 + 6s^2 - 5s)\tau_j^n + \frac{1}{3}(-6s^2 + 4s)\tau_{j+1}^n + \frac{1}{12}(6s^2 - s)\tau_{j+2}^n. \tag{7.16}$$

For variable diffusion, s is replaced by s_j^n, but the superscript n and the subscript j are omitted for convenience. The convergence of the method is determined by a formal consistency analysis, achieved by replacing the finite-difference approximation τ by the exact solution $\hat{\tau}$, and taking the Taylor series expansion of each term in (7.16) about (x_j, t_n), yielding

$$\mathcal{F}\{\hat{\tau}\} = -\Delta t E_{\mathrm{T}}, \tag{7.17}$$

where the truncation error is given by

$$E_{\mathrm{T}} = -\frac{1}{2}\Delta t\left(\frac{\partial^2\hat{\tau}}{\partial t^2} - \alpha^2\frac{\partial^4\hat{\tau}}{\partial x^4}\right) + O\{4\}. \tag{7.18}$$

The time derivative is converted to space derivatives using (7.13), so that the truncation error becomes

$$E_{\mathrm{T}} = -\frac{1}{2}\Delta t\left(E\frac{\partial^2\hat{\tau}}{\partial x^2} + K\frac{\partial^3\hat{\tau}}{\partial x^3}\right) + O\{4\}, \tag{7.19}$$

where E and K are defined by (7.15). The truncation error indicates that the base method, denoted NH2, is fourth-order for constant diffusion, but only second-order for variable coefficients.

7.3.1 Modification

The following modification is given by Noye (1998). Substitution of (7.19) in (7.17) yields

$$\mathcal{F}\{\hat{\tau}\} = P\left[(\Delta x)^2\frac{\partial^2\hat{\tau}}{\partial x^2}\right] + Q\left[(\Delta x)^3\frac{\partial^3\hat{\tau}}{\partial x^3}\right] + O\{6\}, \tag{7.20}$$

where the correction factors P and Q are defined by

$$P = E(\Delta t)^2/2(\Delta x)^2, \quad Q = K(\Delta t)^2/4(\Delta x)^3. \tag{7.21}$$

The presence of the term involving $\partial^2\hat{\tau}/\partial x^2$ indicates that artificial diffusion has been introduced into the numerical solution. The third-order spatial derivative shows that there is a shift in the peak of the numerical solution. The exact solution will be better modelled if these terms are eliminated by using, for example, second-order centered-space difference forms, whereupon (7.20) becomes

$$\mathcal{F}\{\hat{\tau}\} = P\left(\hat{\tau}^n_{j-1} - 2\hat{\tau}^n_j + \hat{\tau}^n_{j+1}\right) + Q\left(\hat{\tau}^n_{j+2} - 2\hat{\tau}^n_{j+1} + 2\hat{\tau}^n_{j-1} - \hat{\tau}^n_{j-2}\right) + O\{6\}. \tag{7.22}$$

Omitting the error terms of order six and higher, replacing $\hat{\tau}$ by τ, and substituting the base method for $\mathcal{F}$, then yields a fourth-order method, denoted NH4, and given by

$$\begin{aligned}\tau^{n+1}_j = {} & \frac{1}{12}\left(6s^2 - s - 12Q\right)\tau^n_{j-2} + \frac{1}{3}\left(-6s^2 + 4s + 6Q + 3P\right)\tau^n_{j-1} + \frac{1}{2}\left(2 + 6s^2 - 5s - 4P\right)\tau^n_j \\ & + \frac{1}{3}\left(-6s^2 + 4s - 6Q + 3P\right)\tau^n_{j+1} + \frac{1}{12}\left(6s^2 - s + 12Q\right)\tau^n_{j+2}.\end{aligned} \tag{7.23}$$

The base method is stable if $0 < s \leq 2/3$ (Noye and Hayman 1986b), giving a necessary condition for the stability of the variable coefficient problem (Richtmyer and Morton 1967, Hirsch 1990).

7.4 The Inverted (5,1) Method

Since the Noye-Hayman method has a (1,5) computational stencil, it can only be used for $j = 2(1)J-2$, so a supplementary method must be used to find the values of τ^{n+1}_1 and τ^{n+1}_{J-1}. Such a method may be developed by inverting the NH2 method to give a formula with a (5,1) computational stencil. This is achieved as follows. Consider the base method, rewritten here for convenience:

$$\begin{aligned}\tau^{n+1}_j = {} & \frac{1}{12}(6s^2 - s)\tau^n_{j-2} + \frac{1}{3}(-6s^2 + 4s)\tau^n_{j-1} + \frac{1}{2}(2 + 6s^2 - 5s)\tau^n_j \\ & + \frac{1}{3}(-6s^2 + 4s)\tau^n_{j+1} + \frac{1}{12}(6s^2 - s)\tau^n_{j+2},\end{aligned} \tag{7.24}$$

in which s is evaluated at (j, n). By setting $-s$ for s, $(n-1)$ replaces $(n+1)$, so that

$$\begin{aligned}\tau^{n-1}_j = {} & \frac{1}{12}(6s^2 + s)\tau^n_{j-2} + \frac{1}{3}(-6s^2 - 4s)\tau^n_{j-1} + \frac{1}{2}(2 + 6s^2 + 5s)\tau^n_j \\ & + \frac{1}{3}(-6s^2 - 4s)\tau^n_{j+1} + \frac{1}{12}(6s^2 + s)\tau^n_{j+2}.\end{aligned} \tag{7.25}$$

Now, shifting the time index, so that $(n+1)$ replaces n, gives the inverted NH2 method, namely

$$\begin{aligned}\tau^n_j = {} & \frac{1}{12}(6s^2 + s)\tau^{n+1}_{j-2} + \frac{1}{3}(-6s^2 - 4s)\tau^{n+1}_{j-1} + \frac{1}{2}(2 + 6s^2 + 5s)\tau^{n+1}_j \\ & + \frac{1}{3}(-6s^2 - 4s)\tau^{n+1}_{j+1} + \frac{1}{12}(6s^2 + s)\tau^{n+1}_{j+2},\end{aligned} \tag{7.26}$$

in which s is evaluated at $(j, n+1)$. By setting $j = 2$ in (7.26) and rearranging, the value of τ^{n+1}_1 may be obtained, where the NH2 method has already been used to calculate τ at the new time-level

for $j = 2(1)J-2$, and τ_0^{n+1} is given by the left-hand boundary value. Similarly, by setting $j = J-2$ in (7.26) and rearranging, the value of τ_{J-1}^{n+1} is found, where τ_J^{n+1} is given by the right-hand boundary value. The NH4 method can also be inverted, giving a fourth-order method with a (5,1) stencil, namely

$$\begin{aligned}\tau_j^n = \frac{1}{12}\left(6s^2+s-12Q\right)\tau_{j-2}^{n+1} + \frac{1}{3}\left(-6s^2-4s+6Q+3P\right)\tau_{j-1}^{n+1} + \frac{1}{2}\left(2+6s^2+5s-4P\right)\tau_j^{n+1} \\ +\frac{1}{3}\left(-6s^2-4s-6Q+3P\right)\tau_{j+1}^{n+1} + \frac{1}{12}\left(6s^2+s+12Q\right)\tau_{j+2}^{n+1},\end{aligned} \tag{7.27}$$

in which s and the correction factors P and Q are all evaluated at $(j, n+1)$. The signs of P and Q do not change, because they are proportional to $(\Delta t)^2$, whereas the sign of s changes, because it is proportional to Δt. The inverted NH4 method can then be used to find the values of τ_1^{n+1} and τ_{J-1}^{n+1}.

7.5 Mitchell's Method

An implicit fourth-order method to solve (7.5) has been outlined by Mitchell (1969) as follows. If (7.5) is divided through by $\alpha(x,t) \neq 0$ and evaluated at $(x_j, t_{n+1/2})$, then

$$\left[\frac{1}{\alpha}\frac{\partial\hat{\tau}}{\partial t}\right]_j^{n+1/2} - \left.\frac{\partial^2\hat{\tau}}{\partial x^2}\right|_j^{n+1/2} = 0. \tag{7.28}$$

Replacing the space derivative by the average of its values at the n and $(n+1)^{th}$ time-levels, and then replacing these by second-order centered-difference forms, gives

$$\left.\frac{\partial^2\hat{\tau}}{\partial x^2}\right|_j^{n+1/2} = \frac{\hat{\tau}_{j+1}^{n+1} - 2\hat{\tau}_j^{n+1} + \hat{\tau}_{j-1}^{n+1}}{2(\Delta x)^2} + \frac{\hat{\tau}_{j+1}^{n} - 2\hat{\tau}_j^{n} + \hat{\tau}_{j-1}^{n}}{2(\Delta x)^2} - \frac{(\Delta x)^2}{12}\left.\frac{\partial^4\hat{\tau}}{\partial x^4}\right|_j^{n+1/2} + O\{4\}. \tag{7.29}$$

The spatial derivative in (7.29) may be written as

$$\frac{\partial^4\hat{\tau}}{\partial x^4} = \frac{\partial^2}{\partial x^2}\left(\frac{\partial^2\hat{\tau}}{\partial x^2}\right) = \frac{\partial^2}{\partial x^2}\left(\frac{1}{\alpha}\frac{\partial\hat{\tau}}{\partial t}\right), \tag{7.30}$$

from which follows, that

$$\left.\frac{\partial^4\hat{\tau}}{\partial x^4}\right|_j^{n+1/2} = \frac{1}{(\Delta x)^2}\left(\left[\frac{1}{\alpha}\frac{\partial\hat{\tau}}{\partial t}\right]_{j+1}^{n+1/2} - 2\left[\frac{1}{\alpha}\frac{\partial\hat{\tau}}{\partial t}\right]_j^{n+1/2} + \left[\frac{1}{\alpha}\frac{\partial\hat{\tau}}{\partial t}\right]_{j-1}^{n+1/2}\right) + O\{2\}. \tag{7.31}$$

Substituting this in (7.29), which is then substituted in (7.28), yields

$$\begin{aligned}&\frac{1}{12\alpha_{j-1}^{n+1/2}}\left.\frac{\partial\hat{\tau}}{\partial t}\right|_{j-1}^{n+1/2} + \frac{5}{6\alpha_j^{n+1/2}}\left.\frac{\partial\hat{\tau}}{\partial t}\right|_j^{n+1/2} + \frac{1}{12\alpha_{j+1}^{n+1/2}}\left.\frac{\partial\hat{\tau}}{\partial t}\right|_{j+1}^{n+1/2} \\ &\quad - \left(\frac{\hat{\tau}_{j+1}^{n+1} - 2\hat{\tau}_j^{n+1} + \hat{\tau}_{j-1}^{n+1}}{2(\Delta x)^2} + \frac{\hat{\tau}_{j+1}^{n} - 2\hat{\tau}_j^{n} + \hat{\tau}_{j-1}^{n}}{2(\Delta x)^2}\right) + O\{4\} = 0.\end{aligned} \tag{7.32}$$

If the time derivatives are replaced by second-order centered forms and the error terms are omitted, then Mitchell's fourth-order method is obtained, namely

$$\begin{aligned}\left(\frac{1}{s_{j-1}^{n+1/2}}-6\right)\tau_{j-1}^{n+1}+2\left(\frac{5}{s_j^{n+1/2}}+6\right)\tau_j^{n+1}+\left(\frac{1}{s_{j+1}^{n+1/2}}-6\right)\tau_{j+1}^{n+1}\\ =\left(\frac{1}{s_{j-1}^{n+1/2}}+6\right)\tau_{j-1}^{n}+2\left(\frac{5}{s_j^{n+1/2}}-6\right)\tau_j^{n}+\left(\frac{1}{s_{j+1}^{n+1/2}}+6\right)\tau_{j+1}^{n}.\end{aligned} \tag{7.33}$$

7.5.1 Stability and Solvability

The local stability of (7.33) can be determined by considering the case when $\alpha(x,t)=\alpha$, a constant. Then $s_j^{n+1/2}=s$ for all j and n and Mitchell's method can be written as

$$(1-6s)\tau_{j-1}^{n+1}+2(5+6s)\tau_j^{n+1}+(1-6s)\tau_{j+1}^{n+1}=(1+6s)\tau_{j-1}^{n}+2(5-6s)\tau_j^{n}+(1+6s)\tau_{j+1}^{n}, \tag{7.34}$$

which is Crandall's method (Crandall 1955). It is well known that Crandall's method is unconditionally stable and solvable. Therefore, Mitchell's method may be considered locally unconditionally stable and solvable, and can be used so long as $\alpha(x,t)$ is not zero for any values of x and t in the solution domain.

7.6 Three-Level Methods

Consider the fourth-order N131 method (Noye 1998) for the constant diffusion equation, which when written in the form (2.12), is given by

$$\mathcal{F}\{\tau\}=(1+6s)\tau_j^{n+1}-12s^2(\tau_{j-1}^{n}+\tau_{j+1}^{n})-2(1-12s^2)\tau_j^{n}+(1-6s)\tau_j^{n-1}, \tag{7.35}$$

which involves three-levels in time. Applying (7.35) to the variable-coefficient diffusion equation, results in only second-order convergence. This may be seen by replacing τ by $\hat{\tau}$ in (7.35) and expanding each term as a Taylor series about (x_j,t_n), yielding

$$\mathcal{F}\{\hat{\tau}\}=-12s\Delta t E_{\mathrm{T}}, \tag{7.36}$$

in which

$$E_{\mathrm{T}}=-\frac{\Delta t}{12s}\left(\frac{\partial^2\hat{\tau}}{\partial t^2}-\alpha^2\frac{\partial^4\hat{\tau}}{\partial x^4}\right)+O\{4\}. \tag{7.37}$$

After replacing the temporal derivative by (7.13), we obtain

$$E_{\mathrm{T}} = -\frac{\Delta t}{12s}\left(E\frac{\partial^2\hat{\tau}}{\partial x^2} + K\frac{\partial^3\hat{\tau}}{\partial x^3}\right) + O\{4\}, \tag{7.38}$$

in which E and K are given by (7.15). The truncation error is therefore second-order for variable coefficients, but fourth-order for constant diffusion. Note that if $s = 1/6$, then the N131, Noye-Hayman and FTCS methods are identical. Also, comparing the leading error term (7.38) with that of the NH2 method (7.19), indicates that for values of $s > 1/6$, the N131 method should be more accurate.

7.6.1 Modification

The following modification is suggested by Noye (1998). Substituting (7.38) into (7.36) yields

$$\mathcal{F}\{\hat{\tau}\} = P\left[(\Delta x)^2\frac{\partial^2\hat{\tau}}{\partial x^2}\right] + Q\left[2(\Delta x)^3\frac{\partial^3\hat{\tau}}{\partial x^3}\right] + O\{6\}, \tag{7.39}$$

where the correction factors P and Q are defined as

$$P = E(\Delta t)^2/(\Delta x)^2, \quad Q = K(\Delta t)^2/2(\Delta x)^3. \tag{7.40}$$

Replacing the space derivatives by second-order centered-space difference forms then gives

$$\mathcal{F}\{\hat{\tau}\} = P(\hat{\tau}^n_{j+1} - 2\hat{\tau}^n_j + \hat{\tau}^n_{j-1}) + Q(\hat{\tau}^n_{j+2} - 2\hat{\tau}^n_{j+1} + 2\hat{\tau}^n_{j-1} - \hat{\tau}^n_{j-2}) + O\{6\}. \tag{7.41}$$

Omitting the errors terms and equating with (7.35), produces the following fourth-order method

$$\tau^{n+1}_j = a^n_j\tau^n_{j-2} + b^n_j\tau^n_{j-1} + d^n_j\tau^n_j + e^n_j\tau^n_{j+1} + f^n_j\tau^n_{j+2} + g^n_j\tau^{n-1}_j, \tag{7.42}$$

which will be denoted as the N151 method when

$$\begin{aligned} &a = -f = -Q\xi, \;\; b = (12s^2 + P + 2Q)\xi, \;\; d = 2(1 - 12s^2 - P)\xi \\ &e = (12s^2 + P - 2Q)\xi, \;\; g = -(1 - 6s)\xi, \;\; \xi = 1/(1 + 6s). \end{aligned} \tag{7.43}$$

The base method is stable in the range $0 < s \leq 1/\sqrt{12}$ (Noye 1998), which gives a necessary condition for the stability of the variable coefficient problem.

7.6.2 Starting Procedure

Because they are three-level methods, the N131 and N151 methods require a two-level starter of appropriate order to compute the values at the first time level. A suitable second-order starting procedure for

the N131 method is the NH2 method, which can be applied for $j = 2(1)J-2$. The inverted NH2 method can then be used to provide the values τ_1^1 and τ_{J-1}^1. Likewise, the NH4 and inverted NH4 methods can provide the required values at the first time-level for the N151 method. At each time level, the N151 method must also be supplemented near the boundaries; again the inverted NH4 method can provide these auxiliary values, once the values for $j = 2(1)J-2$ have been evaluated using (7.42).

7.7 An Analytical Solution

An analytical solution of (7.5) will be developed for the purpose of testing the accuracy of the methods described in this chapter. If the diffusion coefficient is in the separable form $\alpha(x,t) = p(x)q(t)$, then a solution of (7.5) in the separable form

$$\hat{\tau}(x,t) = X(x)M(t), \tag{7.44}$$

leads to

$$\frac{1}{q}\frac{M'}{M} = p\frac{X''}{X} = K, \qquad K \text{ a constant.} \tag{7.45}$$

This yields two ordinary differential equations, the first of which has the solution

$$M(t) = A\exp\{K\int q(t)dt\}, \qquad A \text{ a constant.} \tag{7.46}$$

A solution of the second equation, when $p(x)$ is of the form $p(x) = 0.5[1-2(x-0.5)^2]^{-1}$, is given by

$$X(x) = B\exp\{-(x-0.5)^2\}, \qquad B \text{ a constant,} \tag{7.47}$$

where $K = -1$. The exact solution of (7.5) is then given by

$$\hat{\tau}(x,t) = L\exp\{-(x-0.5)^2\}\exp\{-\int q(t)dt\}, \tag{7.48}$$

where L is a constant, and $q(t)$ is the time varying component of the diffusion coefficient.

7.8 The Diffusion Field

For most problems, it is not adequate to assume that the diffusion coefficient is uniformly constant. Field evidence and experimental studies have suggested that, in describing, for example, solute transport in hydrogeologic systems, the diffusion coefficient increases as a function of travel time or travel distance

until it reaches an asymptotic value (Pickens and Grisak 1981b). The variability of the diffusion is usually attributed to the heterogeneity of the porous medium (Sauty 1980, Basha and El-Habel 1993). In the numerical tests to follow, the diffusion field will be assumed to be in separable form, where the space-dependent component is given by the function

$$p(x) = 0.5\left[1 - 2(x - 0.5)^2\right]^{-1}, \tag{7.49}$$

and the time-dependent component will take on one of the three functional forms given by Basha and El-Habel (1993). These are

Linear:

$$q(t) = D_0 \frac{t}{k} + D_m, \tag{7.50}$$

Asymptotic:

$$q(t) = D_0 \frac{t}{t + k} + D_m, \tag{7.51}$$

or Exponential:

$$q(t) = D_0\left[1 - \exp(-t/k)\right] + D_m, \tag{7.52}$$

where D_0 is the maximum diffusion, D_m is the molecular diffusion and k is equal to the mean travel time. In the following, $D_0 = D_m = 1/20$ and $k = 10$, in which case analytical solutions for the three diffusion profiles considered, are

Linear:

$$\hat{\tau}(x,t) = \exp\left\{-(x - 0.5)^2\right\} \exp\left\{-t^2/400 - t/20\right\}, \quad L = 1 \tag{7.53}$$

Asymptotic:

$$\hat{\tau}(x,t) = \sqrt{t/10 + 1} \exp\left\{-(x - 0.5)^2 - 0.1t\right\}, \quad L = 1/\sqrt{10} \tag{7.54}$$

or Exponential:

$$\hat{\tau}(x,t) = \exp\left\{-(x - 0.5)^2\right\} \exp\left\{0.5 - 0.1t - 0.5\exp(-0.1t)\right\}, \quad L = \exp(0.5). \tag{7.55}$$

These are depicted in Figure 7.1 for the case $t = 4$, along with the initial condition $t = 0$.

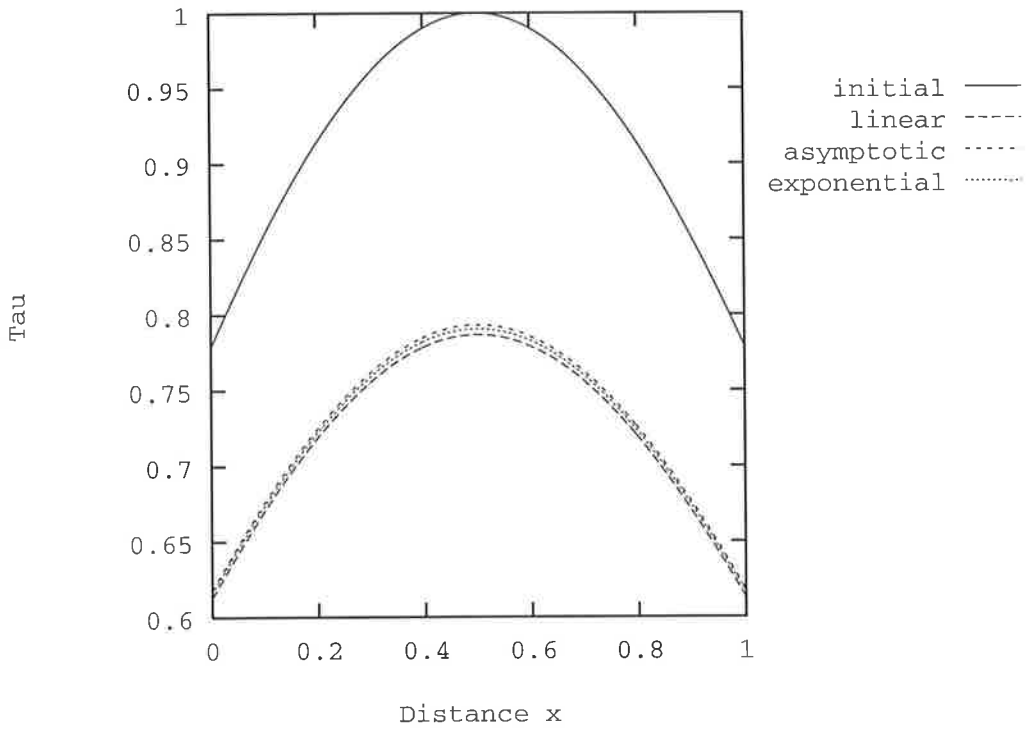

Figure 7.1: Initial condition and solution at $T = 4$ for diffusion profiles.

7.9 Numerical Tests

The one-dimensional diffusion equation, with the diffusion coefficient given by one of the three functional forms described, is solved on the domain $0 < x < 1, 0 < t \leq 4$. The initial and boundary values are given by the analytical solution. The grid number J ranges from 40 to 200. The number of time steps $N = J^2$, so that Δt is proportional to $(\Delta x)^2$, and the maximum value of the diffusion number remains fixed. For the three cases considered: linear, asymptotic and exponential, $s_{max} = \alpha_{max} T \approx 0.28, 0.26, 0.27$, so all schemes are locally stable. A summary of the data required to implement the numerical test for the asymptotic diffusion profile is given in Table 7.3 in Section 7.10.

Mitchell's method is implicit, so the Thomas algorithm is used to solve the resulting tridiagonal system. The other methods are explicit, so the single unknown is given directly from the known values at the old time-level, or from the previous two levels for the three-level methods. The rms errors and cpu times for the three diffusion profiles are given in Tables 7.1 and 7.2. The convergence and efficiency of the schemes for the asymptotic profile are shown in Figures 7.2 and 7.3. Table 7.1 shows that the results for the linear and exponential profiles are qualitatively similar to those of the asymptotic profile, so the plots are omitted. The second and fourth-order Noye-Hayman methods are denoted NH2 and NH4 respectively, and Mitchell's method is denoted M4.

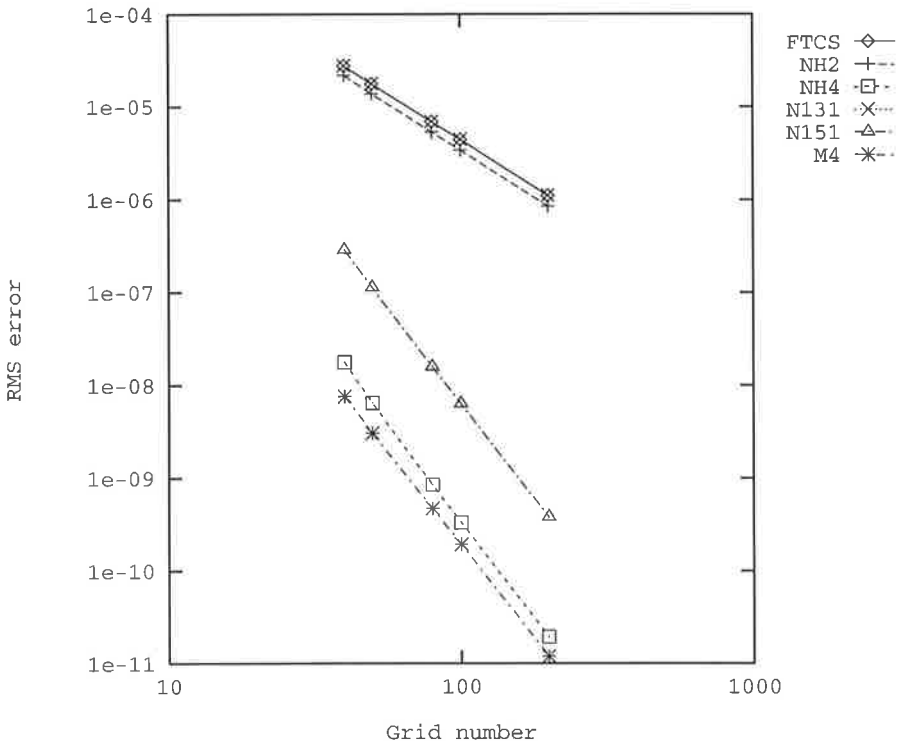

Figure 7.2: Convergence for asymptotic diffusion profile.

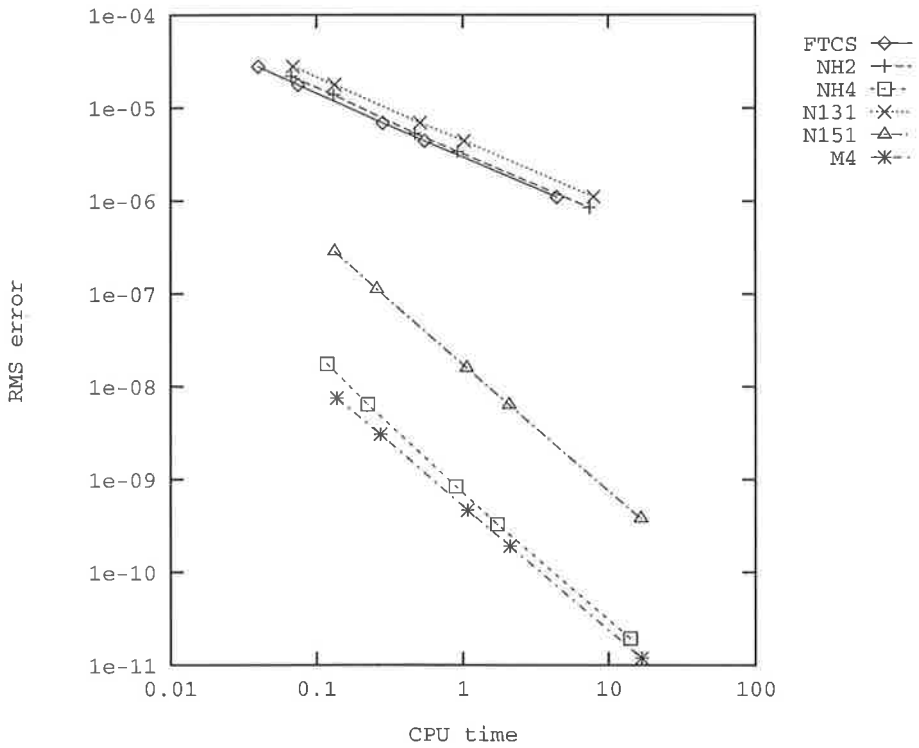

Figure 7.3: Efficiency for asymptotic diffusion profile.

J	Linear Time Variation					
	FTCS	NH2	NH4	N131	N151	M4
40	3.08e−05	2.59e−05	1.78e−08	3.15e−05	4.00e−07	7.84e−09
50	1.96e−05	1.64e−05	6.56e−09	2.01e−05	1.56e−07	3.20e−09
80	7.64e−06	6.31e−06	8.68e−10	7.83e−06	2.23e−08	4.87e−10
100	4.88e−06	4.02e−06	3.42e−10	5.01e−06	8.90e−09	1.99e−10
200	1.22e−06	1.00e−06	2.01e−11	1.25e−06	5.31e−10	1.24e−11
J	Asymptotic Time Variation					
	FTCS	NH2	NH4	N131	N151	M4
40	2.78e−05	2.19e−05	1.75e−08	2.79e−05	2.85e−07	7.52e−09
50	1.77e−05	1.38e−05	6.41e−09	1.78e−05	1.12e−07	3.07e−09
80	6.90e−06	5.32e−06	8.41e−10	6.94e−06	1.58e−08	4.67e−10
100	4.41e−06	3.40e−06	3.30e−10	4.43e−06	6.34e−09	1.91e−10
200	1.10e−06	8.45e−07	1.94e−11	1.11e−06	3.78e−10	1.19e−11
J	Exponential Time Variation					
	FTCS	NH2	NH4	N131	N151	M4
40	2.89e−05	2.34e−05	1.77e−08	2.93e−05	3.28e−07	7.62e−09
50	1.85e−05	1.48e−05	6.51e−09	1.87e−05	1.28e−07	3.11e−09
80	7.18e−06	5.70e−06	8.57e−10	7.29e−06	1.83e−08	4.73e−10
100	4.59e−06	3.64e−06	3.37e−10	4.66e−06	7.30e−09	1.94e−10
200	1.15e−06	9.05e−07	1.98e−11	1.16e−06	4.35e−10	1.21e−11

Table 7.1: RMS errors for one-dimensional diffusion problem.

J	Linear Time Variation					
	FTCS	NH2	NH4	N131	N151	M4
40	3.41e−02	5.50e−02	9.30e−02	6.04e−02	1.23e−01	1.32e−01
50	6.70e−02	1.10e−01	1.93e−01	1.19e−01	2.38e−01	2.52e−01
80	2.38e−01	4.29e−01	7.80e−01	4.65e−01	9.72e−01	9.97e−01
100	4.63e−01	8.16e−01	1.51e+00	8.86e−01	1.92e+00	1.95e+00
200	3.76e+00	6.71e+00	1.19e+01	7.19e+00	1.54e+01	1.59e+01
J	Asymptotic Time Variation					
	FTCS	NH2	NH4	N131	N151	M4
40	4.01e−02	6.70e−02	1.18e−01	6.88e−02	1.33e−01	1.38e−01
50	7.47e−02	1.31e−01	2.25e−01	1.33e−01	2.59e−01	2.74e−01
80	2.86e−01	4.71e−01	8.94e−01	5.11e−01	1.06e+00	1.07e+00
100	5.49e−01	9.18e−01	1.72e+00	1.02e+00	2.08e+00	2.10e+00
200	4.45e+00	7.44e+00	1.41e+01	7.94e+00	1.66e+01	1.68e+01
J	Exponential Time Variation					
	FTCS	NH2	NH4	N131	N151	M4
40	7.25e−02	9.23e−02	2.42e−01	1.05e−01	2.64e−01	2.66e−01
50	1.37e−01	1.74e−01	4.87e−01	2.00e−01	5.14e−01	5.18e−01
80	5.43e−01	6.95e−01	1.88e+00	7.47e−01	2.11e+00	2.15e+00
100	1.06e+00	1.35e+00	3.67e+00	1.48e+00	4.10e+00	4.13e+00
200	8.65e+00	1.10e+01	2.95e+01	1.18e+01	3.29e+01	3.30e+01

Table 7.2: CPU times for one-dimensional diffusion problem.

A comparison of the data given in Table 7.1 shows that all diffusion profiles give very similar results. This observation was also made by Yates (1992), who investigated analytical solutions of the transport equation for a linear and exponential diffusion coefficient. He states that the "solutions will produce essentially the same results at early times, when their respective dispersion functions are approximately the same, but differences occur at intermediate and large times."

Yates (1992) also observed that significant differences in the concentration curves are only found when the diffusion coefficient differs by a factor of 10. Consequently, he concluded that it would be difficult and expensive to determine whether the diffusion is linear or asymptotic at early times, and a clear distinction may only be possible at very large times.

Pickens and Grisak (1981b) found that "the effect of early time scale-dependent dispersion may be, in some cases, of little consequence in predictions at large mean travel distances." For large travel distances, Pickens and Grisak recommended using the asymptotic diffusion coefficient.

Figure 7.2 shows that all methods give the predicted convergence rates. The orders of convergence of the second-order FTCS, NH2 and N131 methods, given by the slope of the line of best fit to the data, are 2.01, 2.02 and 2.00 respectively. The orders of convergence of the fourth-order NH4, N151 and M4 methods are 4.22, 4.12 and 4.01 respectively.

Although the NH2 method is the most accurate second-order scheme, the FTCS method is more efficient to use. This may be seen by examiming Figure 7.3, and is because the FTCS method has a simpler stencil, and so requires less time to run than the NH2 method.

The computational times presented in Table 7.2 show that the schemes require approximately twice as long to run for the exponential diffusion profile than for the linear profile. For all methods and profiles, doubling the grid number leads to an eightfold increase in the run times.

It was mentioned in Section 7.6 that, based on the leading term of their truncation errors, the N131 method should be more accurate than the NH2 method if $s > 1/6$. However, it is observed that the latter method is actually more accurate in these tests. This can be explained by calculating the percentage of grid points in the computational domain for which $s > 1/6$. For the asymptotic diffusion profile, this was so for less than 25% of the grid points for each grid resolution. Consequently, the NH2 method is more accurate.

The most accurate scheme is Mitchell's implicit fourth-order method, and although it expends more time than the explicit NH4 method, it can be seen by comparing their efficiency plots that it is nevertheless the most efficient scheme. Another advantage of Mitchell's method is that it is unconditionally stable and solvable; the other methods are restricted to diffusion numbers as small as 0.288 in the case of the N131 and N151 schemes, and at most 0.667 for the NH2 and NH4 methods.

The implicit Crank-Nicolson method (Crank and Nicolson 1947), which has a second-order truncation error for constant diffusion, is another unconditionally stable and solvable method. A formal consistency analysis of the Crank-Nicolson method about $(x_j, t_{n+1/2})$, reveals its truncation error

$$E_{\mathrm{T}} = \frac{1}{12}\alpha(\Delta x)^2 \frac{\partial^4 \hat{\tau}}{\partial x^4} + O\{4\}.$$

An examination of the truncation error indicates that the leading error term contains no derivatives of the diffusion coefficient. Hence, the Crank-Nicolson method retains its constant-coefficient order in variable-coefficient situations. Moreover, since the coefficient of the leading error term is positive, insufficient damping of the high frequency components of the solution may occur. Examples of problems in which high frequency components occur in the solution are: problems where the boundary conditions are discontinuous, and problems where the solution decays very rapidly (Cash 1984). The Crank-Nicolson method is known to perform poorly on such problems.

To summarize, because Mitchell's method is the most accurate and efficient scheme, and has the largest stability region, it would be most recommended. In terms of cpu time, the second-order explicit FTCS method is, however, the fastest scheme, requiring only a quarter the time of Mitchell's method to run for each grid. Generally, no benefit could be found by employing the three-level schemes; they are neither more accurate and efficient, nor more stable than the other methods, and are the most difficult of all schemes to implement. Because the accuracy and the efficiency of the schemes depends on the size of the diffusion number, they will now be compared for different values of the maximum diffusion number.

7.10 Parameter Analysis

For constant coefficients, the leading term in the truncation error of the MEPDE can be used to give an indication of the accuracy of the methods for different values of the diffusion number. The leading error terms for the methods considered in this chapter are given by (see Noye and Hayman 1986b)

$$\Gamma_4^{\mathrm{FTCS}}(s) = 6s - 1, \;\; \Gamma_6^{\mathrm{NH}}(s) = 2(2 - 15s + 30s^2), \;\; \Gamma_6^{\mathrm{M4}}(s) = 3(20s^2 - 1)/2, \;\; \Gamma_6^{\mathrm{N131}}(s) = 60s^2 - 1,$$

where Mitchell's method reduces to Crandall's method for constant diffusion (cf. Section 7.5.1). These relations give the functional dependence of the leading term on s, and indicate that the FTCS method is fourth-order if $s = 1/6$, Mitchell's method is sixth-order if $s^2 = 1/20$, and the N131 method is sixth-order if $s^2 = 1/60$. A relative error measure, which may be used to compare the accuracy of different methods of the same order, is given by the modulus of the leading term (see Noye and Hayman 1986b).

For variable coefficients, because the leading term of the truncation error of the MDPDE contains derivatives of the diffusion coefficient, such a form of parameter analysis is generally not possible. Instead, the accuracy and efficiency of the methods for different values of the maximum diffusion number

1. Exact solution: $\hat{\tau}(x,t) = \sqrt{1+t/10}\exp\{-(x-0.5)^2 - 0.1t\}$

2. Initial condition: $\hat{\tau}(x,0)$ is given by the exact solution

3. Boundary conditions: $\hat{\tau}(0,t)$ and $\hat{\tau}(1,t)$ are given by the exact solution

4. Diffusion number: $s = \alpha\Delta t/(\Delta x)^2 = \alpha T J^2/N$

5. Asymptotic profile: $\alpha(x,t) = 0.05(t+5)(t+10)^{-1}[1-2(x-0.5)^2]^{-1}$

6. $\alpha\max = 1/20,\ T = 4,\ J = 40, 50, 80, 100, 200$

7. setting $N = J^2/4,\ J^2/2,\ J^2, 2J^2, 4J^2$ gives $s\max \approx 1.04,\ 0.52,\ 0.26,\ 0.13,\ 0.07$

Table 7.3: Data for Parameter Analysis.

are compared in a numerical test. To this effect, the methods are tested using the asymptotic diffusion profile for various values of the maximum diffusion number, as given in Table 7.3. Only Mitchell's method is stable when $s\max = 1.04$, the three-level methods are unstable if $s\max = 0.52$, and all methods are stable for the other values of the maximum diffusion number.

Note that the FTCS method, which is locally stable only if $s < 0.5$, is stable when $s\max = 0.52$. This is because the values of $s\max$ given in the table are the maximum values on the *solution* domain, in the limit as the grid spacings tend to zero, whereas in the numerical test, the values are computed at discrete points in the *computational* domain, so the maximum diffusion number is not quite attained.

7.10.1 Results

Graphs showing the convergence and the efficiency of the methods for the various values of the maximum diffusion number for which they are stable are presented in Figures 7.4 and 7.5. Additionally the rms errors for the test problem are given in Table 7.4. The cpu times are not presented, since they follow from those given in Table 7.2 for the asymptotic profile, in the sense that doubling the number of time steps doubles the times, quartering the number of time steps quarters the times, and so on.

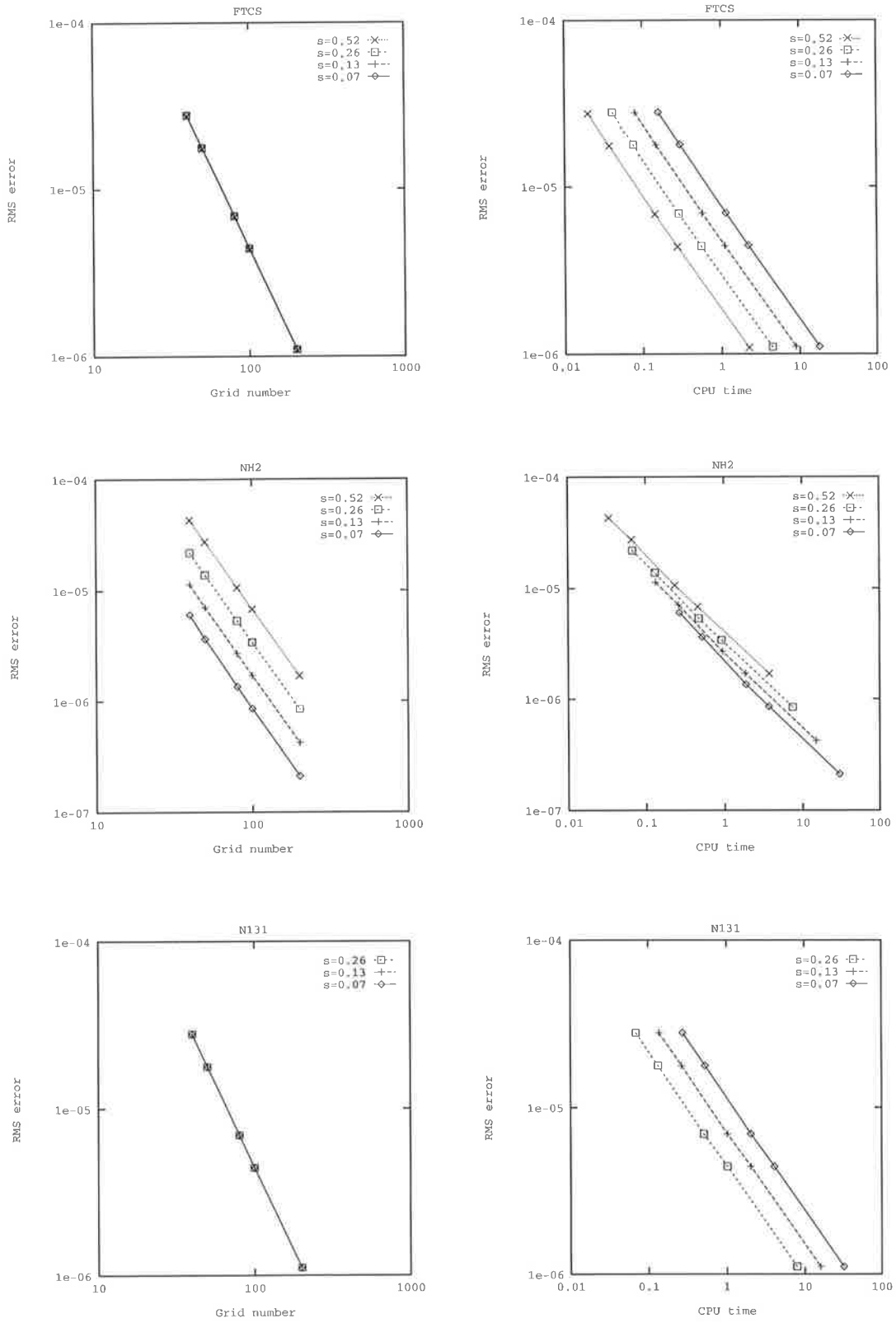

Fig 7.4: Convergence shown on left and efficiency shown on right for the second-order FTCS, NH2 and N131 methods for various values of $s$max.

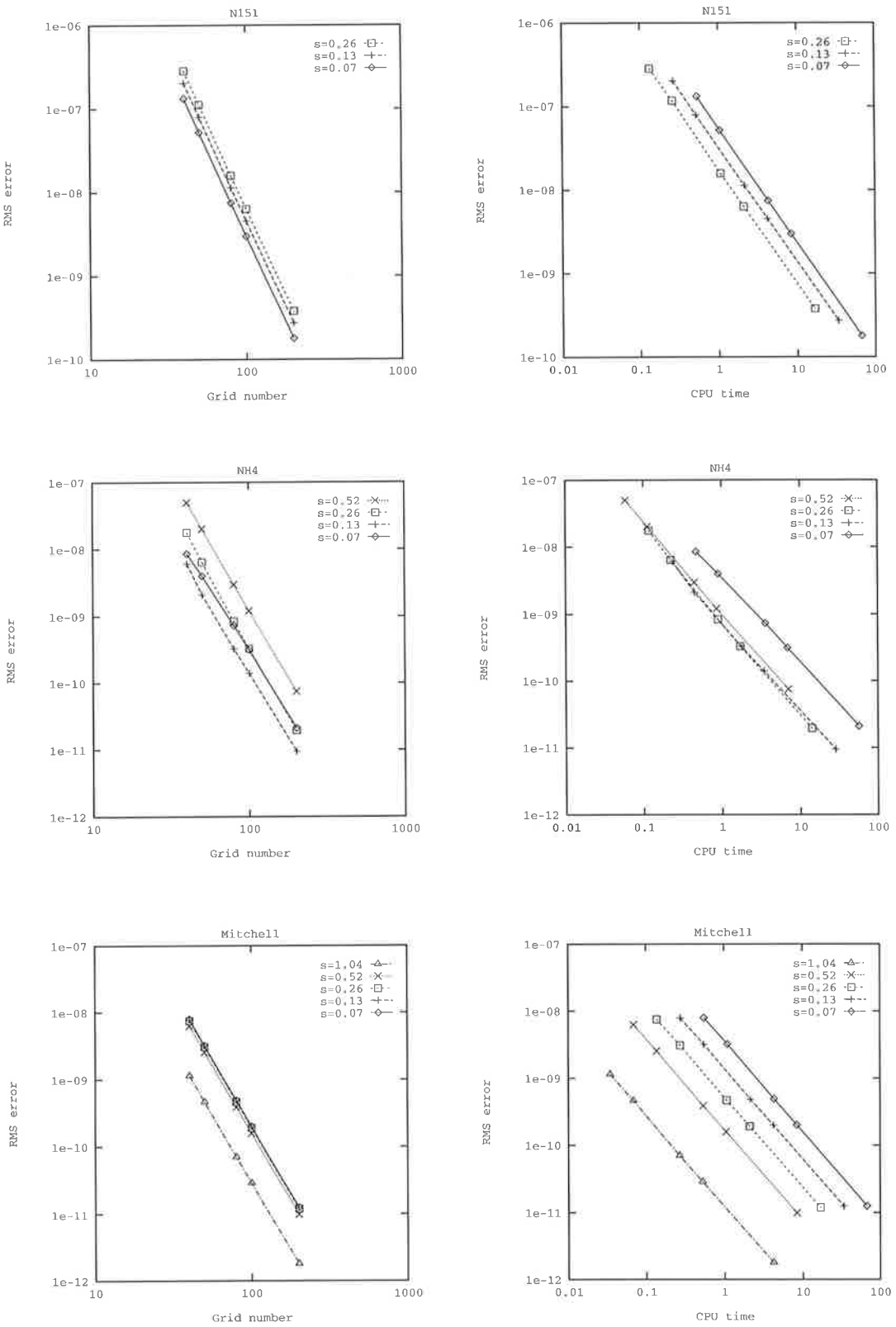

Fig 7.5: Convergence shown on left and efficiency shown on right for the fourth-order N151, NH4 and Mitchell methods for various values of $s$max.

	$s$max = 0.07					
J	FTCS	NH2	NH4	N131	N151	M4
40	2.80e−05	6.01e−06	8.54e−09	2.79e−05	1.32e−07	7.92e−09
50	1.79e−05	3.64e−06	3.99e−09	1.78e−05	5.18e−08	3.23e−09
80	6.96e−06	1.35e−06	7.32e−10	6.94e−06	7.43e−09	4.92e−10
100	4.45e−06	8.57e−07	3.14e−10	4.43e−06	2.98e−09	2.01e−10
200	1.11e−06	2.12e−07	2.10e−11	1.11e−06	1.79e−10	1.25e−11
	$s$max = 0.13					
J	FTCS	NH2	NH4	N131	N151	M4
40	2.79e−05	1.13e−05	6.06e−09	2.79e−05	2.02e−07	7.84e−09
50	1.78e−05	7.03e−06	2.11e−09	1.78e−05	7.91e−08	3.20e−09
80	6.94e−06	2.68e−06	3.27e−10	6.94e−06	1.13e−08	4.87e−10
100	4.43e−06	1.70e−06	1.40e−10	4.43e−06	4.53e−09	1.99e−10
200	1.11e−06	4.23e−07	9.50e−12	1.11e−06	2.72e−10	1.24e−11
	$s$max = 0.26					
J	FTCS	NH2	NH4	N131	N151	M4
40	2.78e−05	2.19e−05	1.75e−08	2.79e−05	2.85e−07	7.52e−09
50	1.77e−05	1.38e−05	6.41e−09	1.78e−05	1.12e−07	3.07e−09
80	6.90e−06	5.32e−06	8.41e−10	6.94e−06	1.58e−08	4.67e−10
100	4.41e−06	3.40e−06	3.30e−10	4.43e−06	6.34e−09	1.91e−10
200	1.10e−06	8.45e−07	1.94e−11	1.11e−06	3.78e−10	1.19e−11
	$s$max = 0.52					
J	FTCS	NH2	NH4	N131	N151	M4
40	2.75e−05	4.30e−05	4.98e−08			6.24e−09
50	1.76e−05	2.74e−05	2.00e−08			2.55e−09
80	6.84e−06	1.06e−05	2.97e−09	unstable	unstable	3.87e−10
100	4.37e−06	6.78e−06	1.21e−09			1.58e−10
200	1.09e−06	1.69e−06	7.49e−11			9.87e−12
	$s$max = 1.04					
J	FTCS	NH2	NH4	N131	N151	M4
40						1.14e−09
50						4.63e−10
80	unstable	unstable	unstable	unstable	unstable	7.04e−11
100						2.88e−11
200						1.80e−12

Table 7.4: RMS errors for asymptotic profile for various $s$max.

The first observation is that the accuracy of the second-order FTCS and N131 methods is virtually independent of the size of the maximum diffusion number. This is seen by examining the left top and bottom diagrams in Figure 7.4, or the values tabulated in Table 7.4. In fact, Table 7.4 shows that the results of the N131 method are so similar for the three values of the maximum diffusion number for which it is stable, that the errors are identical for all three values.

The best choice of the diffusion number is then the one that gives the results most quickly which, from the efficiency plots given in Figure 7.4, is the largest value of the diffusion number for which these methods are stable. That is, the FTCS method is most efficient when it is used for $s\max = 0.52$, while the N131 method is most efficient for $s\max = 0.26$.

The errors given by the NH2 method are approximately doubled as the maximum value of the diffusion number is doubled (see Table 7.4), whereas the cpu times are halved because the number of time steps is halved. For this reason, the graphs showing the convergence of the NH2 method are more spread out than those showing its efficiency (compare the middle diagrams in Figure 7.4).

Whereas the other second-order methods were most efficient for the largest value of the maximum diffusion number for which they were tested, the NH2 method is most efficient for the smallest value of the maximum diffusion number. The most accurate and efficient second-order results overall are achieved by using the NH2 method with $s\max = 0.07$.

As can be seen by examining Figure 7.5, the fourth-order N151 method is least efficient for the same value of the maximum diffusion number for which it is most accurate. This is because it takes four times longer to achieve the results with $s\max = 0.07$ than with $s\max = 0.26$, but the errors are only approximately halved.

The fourth-order NH4 method gives the most accurate results when it is used with $s\max = 0.13$. It gives similar efficiency for all values of the maximum diffusion number except for $s\max = 0.07$, for which it performs worst. The most accurate and efficient results given by any method, are those given by Mitchell's method for $s\max = 1.04$. Furthermore, Mitchell's method is the only stable scheme for this value of the maximum diffusion number.

However, using the NH4 method with $s\max = 0.52, 0.26$ or 0.13, is more efficient than using Mitchell's method with $s\max = 0.13$ or 0.07. And in fact, from Table 7.4, it can be seen that the NH4 method is actually more accurate than Mitchell's method when $s\max = 0.13$.

Note that the least efficient fourth-order results, given by the N151 method with $s\max = 0.07$, are still much more accurate and efficient than the most accurate and efficient second-order results, given by the NH2 method with $s\max = 0.07$. For example, from the tabulated values, it can be seen that the NH2 method used with $J = 200$ is less accurate than the N151 method used with $J = 40$.

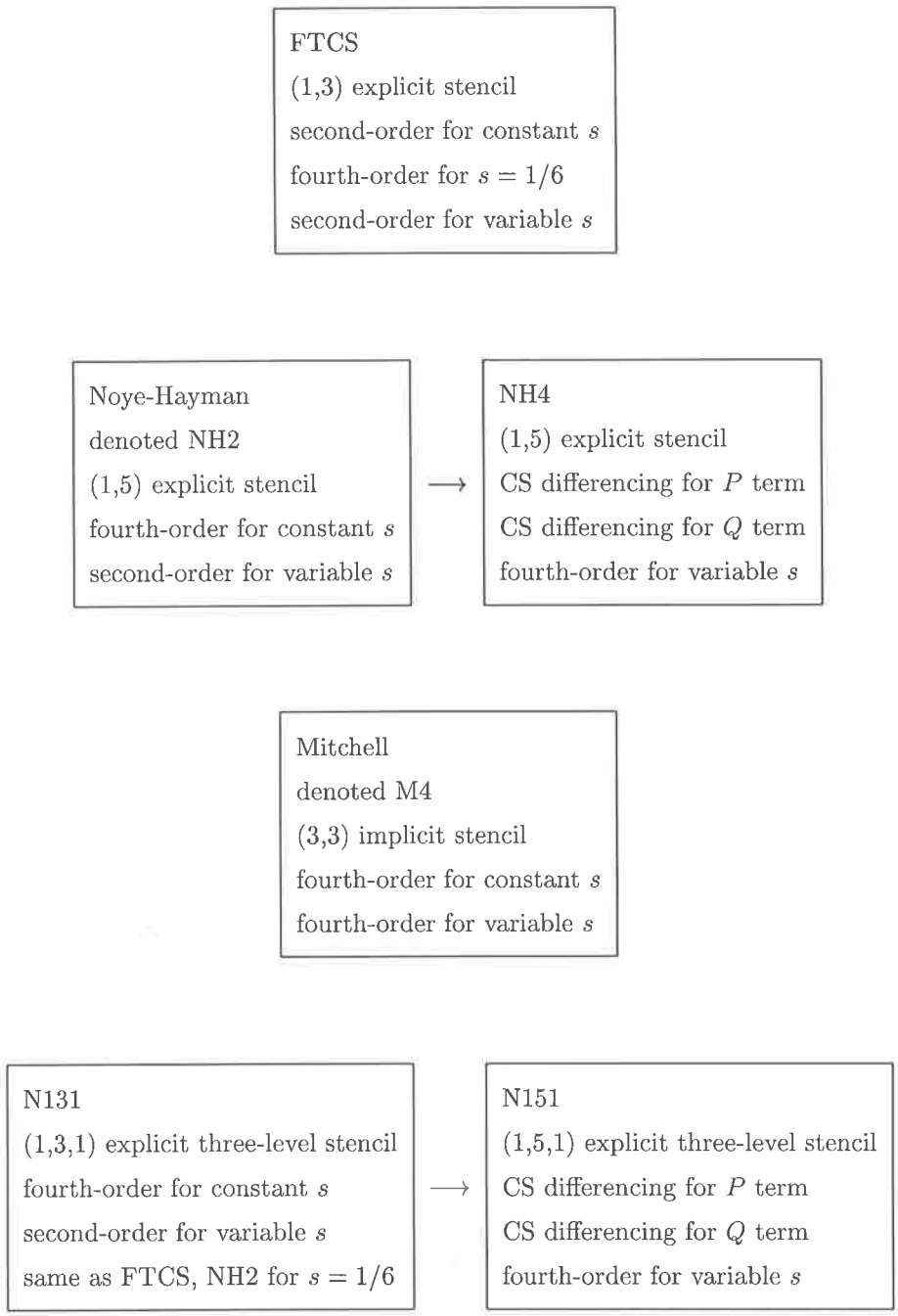

Figure 7.6: Synopsis of the properties of the methods.

Note that the variable coefficients are evaluated at t_n for all the explicit methods, whereas for Mitchell's method they are evaluated at $t_{n+1/2}$.

7.11 Summary

In this chapter, the accuracy of several finite-difference methods used to approximate the solution of the variable coefficient diffusion equation were examined. This is of considerable importance because in many applications, it is inappropriate to assume that the diffusion is constant; it is far more likely to depend on the travel time or travel distance.

It was seen that high-order methods for the constant coefficient equation reduce to low-order when used for variable coefficients. A technique to modify these methods so as to increase their order back to the constant coefficient rate was implemented.

The fourth-order Noye-Hayman method, which reverts to second-order for variable diffusion, was modified to give a new fourth-order method. The second-order FTCS method and Mitchell's implicit fourth-order method were investigated, and a three-level second-order method was modified to fourth-order, thereby retaining its constant-coefficient order. A synopsis of the methods is given in Figure 7.6.

The schemes were tested for the three diffusion profiles suggested by Basha and El-Habel (1993). It was found that the methods gave similar performances for all three profiles, a result also seen in the literature. The modified methods outperformed their base counterparts considerably, in terms of both accuracy and efficiency. Mitchell's method gave the most accurate and efficient results overall.

Since accuracy depends on the size of the diffusion number, the methods were also tested for various values of the maximum diffusion number using the asymptotic profile. The accuracy of the second-order FTCS and N131 methods was seen to be virtually independent of the size of the maximum diffusion number. The NH2 method was the most accurate and efficient second-order scheme, and performed best with $s\max = 0.07$.

Mitchell's fourth-order method used with $s\max = 1.04$ was seen to be the most accurate and efficient method overall. Mitchell's method also has the advantage of being unconditionally stable and solvable; the other schemes are rather restricted in their stability range.

To determine a necessary condition for the stability of the schemes for variable coefficient problems, a local stability analysis was performed, achieved by freezing the coefficients, so that the diffusion may be considered constant within the computational stencil (Richtmyer and Morton 1967, Hirsch 1990).

Chapter 8

Two-Dimensional Diffusion: A Special Case

The two-dimensional form of (7.5), namely

$$\frac{\partial \hat{\tau}}{\partial t} - \alpha_x(x,y,t)\frac{\partial^2 \hat{\tau}}{\partial x^2} - \alpha_y(x,y,t)\frac{\partial^2 \hat{\tau}}{\partial y^2} = 0, \tag{8.1}$$

in which $\hat{\tau} = \hat{\tau}(x,y,t)$, may be solved using the LOD technique described in Chapter 5, so that each of the one-dimensional equations

$$\frac{\partial \hat{\tau}}{\partial t} - \alpha_x(x,y,t)\frac{\partial^2 \hat{\tau}}{\partial x^2} = 0, \tag{8.2}$$

$$\frac{\partial \hat{\tau}}{\partial t} - \alpha_y(x,y,t)\frac{\partial^2 \hat{\tau}}{\partial y^2} = 0, \tag{8.3}$$

is solved consecutively over a time-step. Appropriate intermediate boundary conditions must be derived if the LOD method is to retain the accuracy of the component equations. Morris and Gourlay (1973), Mitchell and Griffiths (1980), and Noye and Hayman (1994) discuss the implementation of intermediate boundary conditions for LOD methods applied to parabolic equations.

The two-dimensional diffusion equation is used to model contaminant dispersion in rivers and estuaries. In many applications, it can be assumed that the flow is well mixed over the water depth, so that the contaminant is approximately uniform in the vertical direction (Croucher and O'Sullivan 1998). A two-dimensional depth averaged form of the three-dimensional equation can then be used to approximate the movement due to diffusion.

In this chapter, the methods described in Chapter 7 to solve the one-dimensional equation will be used in a locally one-dimensional fashion to approximate solutions to the two-dimensional diffusion equation. Since three-level methods cannot be used in a LOD fashion, they will not be considered here.

8.1 The FTCS Method

To solve (8.1), the FTCS method can be applied to (8.2) followed by (8.3), so that

$$\tau^*_{j,k} = s_x\tau^n_{j-1,k} + (1-2s_x)\tau^n_{j,k} + s_x\tau^n_{j+1,k} \tag{8.4}$$

is applied by sweeping in the x direction for each y value, and

$$\tau^{n+1}_{j,k} = s_y\tau^*_{j,k-1} + (1-2s_y)\tau^*_{j,k} + s_y\tau^*_{j,k+1} \tag{8.5}$$

is applied by sweeping in the y direction for each x value.

In the above, $s_x = \alpha_x(x,y,t)\Delta t/(\Delta x)^2$ and $s_y = \alpha_y(x,y,t)\Delta t/(\Delta y)^2$. The constituent formulae are locally stable if $s_x \leq 1/2$ and $s_y \leq 1/2$, so the LOD method is locally stable when both conditions hold.

8.2 The Noye-Hayman Method

The two-dimensional diffusion equation may also be solved by applying the Noye-Hayman (NH2) method to each of the one-dimensional diffusion equations as follows. First solve

$$\begin{aligned}\tau^*_{j,k} = \frac{1}{12}(6s_x^2 - s_x)\tau^n_{j-2,k} + \frac{1}{3}(-6s_x^2 + 4s_x)\tau^n_{j-1,k} + \frac{1}{2}(2 + 6s_x^2 - 5s_x)\tau^n_{j,k} \\ + \frac{1}{3}(-6s_x^2 + 4s_x)\tau^n_{j+1,k} + \frac{1}{12}(6s_x^2 - s_x)\tau^n_{j+2,k}\end{aligned} \tag{8.6}$$

by sweeping in the x direction for each y value, then solve

$$\begin{aligned}\tau^{n+1}_{j,k} = \frac{1}{12}(6s_y^2 - s_y)\tau^*_{j,k-2} + \frac{1}{3}(-6s_y^2 + 4s_y)\tau^*_{j,k-1} + \frac{1}{2}(2 + 6s_y^2 - 5s_y)\tau^*_{j,k} \\ + \frac{1}{3}(-6s_y^2 + 4s_y)\tau^*_{j,k+1} + \frac{1}{12}(6s_y^2 - s_y)\tau^*_{j,k+2}\end{aligned} \tag{8.7}$$

by sweeping in the y direction for each x value. The constituent formulae are locally stable if $s_x \leq 2/3$ and $s_y \leq 2/3$, so the LOD method is locally stable when both conditions are satisfied.

The modified fourth-order NH4 method can be implemented in the same way, so that

$$\tau^*_{j,k} = \frac{1}{12}(6s_x^2 - s_x - 12Q)\tau^n_{j-2,k} + \frac{1}{3}(-6s_x^2 + 4s_x + 6Q + 3P)\tau^n_{j-1,k} + \frac{1}{2}(2 + 6s_x^2 - 5s_x - 4P)\tau^n_{j,k}$$
$$+\frac{1}{3}(-6s_x^2 + 4s_x - 6Q + 3P)\tau^n_{j+1,k} + \frac{1}{12}(6s_x^2 - s_x + 12Q)\tau^n_{j+2,k} \tag{8.8}$$

is solved by sweeping in the x direction for each y value, and

$$\tau^{n+1}_{j,k} = \frac{1}{12}(6s_y^2 - s_y - 12Q)\tau^*_{j,k-2} + \frac{1}{3}(-6s_y^2 + 4s_y + 6Q + 3P)\tau^*_{j,k-1} + \frac{1}{2}(2 + 6s_y^2 - 5s_y - 4P)\tau^*_{j,k}$$
$$+\frac{1}{3}(-6s_y^2 + 4s_y - 6Q + 3P)\tau^*_{j,k+1} + \frac{1}{12}(6s_y^2 - s_y + 12Q)\tau^*_{j,k+2} \tag{8.9}$$

is solved by sweeping in the y direction for each value of x. The factors P and Q are defined by (7.21), with y replacing x in P and Q for use in (8.9). A necessary condition for the stability of the LOD method is that $s_x \leq 2/3$ and $s_y \leq 2/3$.

8.2.1 Near Boundary Values

Because of their (1,5) stencils, the NH2 and NH4 methods can only be used for $j = 2(1)J - 2$ for each x sweep, and for $k = 2(1)K - 2$ for each y sweep (see Noye and Hayman 1994). The values near the boundaries can be found as follows. Consider Mitchell's method written in the form

$$L1\,\tau^*_{j-1,k} + L2\,\tau^*_{j,k} + L3\,\tau^*_{j+1,k} = R1\,\tau^n_{j-1,k} + R2\,\tau^n_{j,k} + R3\,\tau^n_{j+1,k} \tag{8.10}$$

where $L1, L2, L2, R1, R2, R3$ are the coefficients given by (7.33). Then, setting $j = 2$ in (8.10), and rearranging, yields

$$\tau^*_{1,k} = (R1\,\tau^n_{1,k} + R2\,\tau^n_{2,k} + R3\,\tau^n_{3,k} - L2\,\tau^*_{2,k} - L3\,\tau^*_{3,k})/L1, \tag{8.11}$$

where the NH2 or NH4 method has already been used to find the intermediate values for $j = 2(1)J - 2$. Similarly, setting $j = J-2$ in (8.10) provides

$$\tau^*_{J-1,k} = (R1\,\tau^n_{J-3,k} + R2\,\tau^n_{J-2,k} + R3\,\tau^n_{J-1,k} - L1\,\tau^*_{J-3,k} - L2\,\tau^*_{J-2,k})/L3. \tag{8.12}$$

If Mitchell's method is written in the form

$$L1\,\tau^{n+1}_{j,k-1} + L2\,\tau^{n+1}_{j,k} + L3\,\tau^{n+1}_{j,k+1} = R1\,\tau^*_{j,k-1} + R2\,\tau^*_{j,k} + R3\,\tau^*_{j,k+1}, \tag{8.13}$$

in which y replaces x in the coefficients given by (7.33), then by setting $k = 2$ and $k = K-2$ in (8.13), and rearranging, the values of $\tau^{n+1}_{j,1}$ and $\tau^{n+1}_{j,K-1}$ may be found.

8.3 Mitchell's Method

Mitchell's method can be used to solve the two-dimensional diffusion equation as follows:

$$L1\,\tau^*_{j-1,k} + L2\,\tau^*_{j,k} + L3\,\tau^*_{j+1,k} = R1\,\tau^n_{j-1,k} + R2\,\tau^n_{j,k} + R3\,\tau^n_{j+1,k} \tag{8.14}$$

is applied by sweeping in the x direction for each y value, where $L1, L2, L2, R1, R2, R3$ are the coefficients given by (7.33), and

$$L1\,\tau^{n+1}_{j,k-1} + L2\,\tau^{n+1}_{j,k} + L3\,\tau^{n+1}_{j,k+1} = R1\,\tau^*_{j,k-1} + R2\,\tau^*_{j,k} + R3\,\tau^*_{j,k+1}, \tag{8.15}$$

is solved by sweeping in the y direction for each x value, where y replaces x in the coefficients of (7.33). Mitchell's method is unconditionally stable and solvable for all $s_x > 0$ and $s_y > 0$, so the LOD method can be implemented unconditionally.

8.4 Numerical Tests

The two-dimensional diffusion equation, in which the diffusion coefficient takes one of the three functional forms described in Section 7.8, is solved on the domain [0,1] in each space dimension up to the final time T, for various values of the grid number ranging from 40 to 200. Rather than setting $N = J^2$ we set $N = J^2/2$, halving the time required to solve the problem without altering the expected orders of convergence, as Δt is still proportional to $(\Delta x)^2$. Choosing $T = 2$, the maximum value of the diffusion number for each profile is the same as in the one-dimensional case.

The analytical solution of the two-dimensional equation is the product of the exact solution of (8.2) and the exact solution of (8.3). For Equation (8.2) the exact solution is given in Section 7.8, with y replacing x to give the exact solution of (8.3). The initial and boundary values are given by the exact solution, and the intermediate boundary conditions are given by (8.16), see below. A summary of the data required to implement the numerical test for the asymptotic profile is given in Table 8.1.

8.4.1 Implementation

For each time-step the LOD technique is implemented as indicated in Table 8.2. Noye and Hayman (1994) also give an in depth description of the implementation of the LOD technique for the FTCS method and the NH2 method applied to the two-dimensional constant coefficient diffusion equation. The implementation for variable coefficients is identical.

1. Exact solution: $\hat{\tau}(x,y,t) = (1+t/10)\exp\{-(x-0.5)^2-(y-0.5)^2-0.2t\}$

2. Initial condition: $\hat{\tau}(x,y,0)$ given by the exact solution

3. Boundary conditions: $\hat{\tau}(0,y,t)$, $\hat{\tau}(1,y,t)$, $\hat{\tau}(x,0,t)$, $\hat{\tau}(x,1,t)$ given by exact solution

4. Intermediate BCs: $\hat{\tau}(0,y,t_*)$, $\hat{\tau}(1,y,t_*)$, $\hat{\tau}(x,0,t_*)$, $\hat{\tau}(x,1,t_*)$ given by (8.16)

5. Diffusion numbers: $s_x = \alpha_x \Delta t/(\Delta x)^2 = \alpha_x T J^2/N$, $s_y = \alpha_y \Delta t/(\Delta y)^2 = \alpha_y T K^2/N$

6. Asymptotic profile: $\alpha_x(x,t) = 0.05(t+5)(t+10)^{-1}[1-2(x-0.5)^2]^{-1}$, $\alpha_y(y,t) = \alpha_x(y,t)$

7. maximum value of α_x and α_y is 0.05, $T = 2$

8. $N = J^2/2 = K^2/2$ with $J = 40, 50, 80, 100, 200$ gives $s\max \approx 0.26$ for s_x and s_y

Table 8.1: Data for the LOD problem for the asymptotic diffusion profile.

The boundary conditions given for the full two-dimensional problem cannot be applied at the intermediate level, because they incorporate diffusion in both spatial directions, whereas after the first stage only the diffusion in the x direction has been approximated (Noye and Hayman 1994). The exact solution at the intermediate level is given by

$$\hat{\tau}(x,y,t_*) = \hat{\tau}_1(x,t_{n+1})\hat{\tau}_2(y,t_n), \tag{8.16}$$

where $\hat{\tau}_1$ and $\hat{\tau}_2$ are the exact solutions of (8.2) and (8.3), respectively. This can be used to supply the correct intermediate boundary conditions, because at the new time-level only the diffusion in the x direction is incorporated, and none of the diffusion in the y direction (Noye and Hayman 1994).

For the explicit schemes, the single unknown at the new time-level is given directly from the known values at the old time-level. The NH2 and NH4 methods are supplemented using Mitchell's method, to give the interior values near to the boundaries (cf. Section 8.2.1). The Thomas algorithm is used to solve the system of linear algebraic equations arising when Mitchell's method is used. Recall that the coefficients of the explicit schemes are evaluated at time t_n, but at $t_{n+1/2}$ for Mitchell's method.

1. The FDE is first used in the x direction as it would be for the one-dimensional equation (8.2).

2. The application is repeated for each y value, yielding the intermediate values $\tau^*_{j,k}$ over the spatial domain.

3. Then using the $\tau^*_{j,k}$ as the initial condition, the same FDE is used in the y direction as it would normally be used to solve the one-dimensional equation (8.3).

4. This is repeated for each x value, yielding the values $\tau^{n+1}_{j,k}$ at the end of the time-level $(n+1)$.

Table 8.2: Implementation of the LOD technique.

8.4.2 Results

The convergence and efficiency of the schemes for the asymptotic diffusion profile are shown in Figures 8.1 and 8.2. Table 8.3 shows that the results for the linear and exponential profiles are qualitatively similar to those of the asymptotic profile, so the convergence and efficiency plots are not presented. The cpu times given in Table 8.4 are the average of 10 consecutive runs of the algorithm. These show that the cpu times are longest for the profile incorporating exponential time variation. For all methods and profiles, doubling the grid number leads approximately to a 16 fold increase in the computational times.

The second-order NH2 method is more accurate than the second-order FTCS method unless $J = 40$. The FTCS method is however faster to execute: a reflection of its simpler (1,3) computational stencil. Because of the trade-off between speed and accuracy, neither method has any significant advantage over the other in terms of efficiency (see Figure 8.2). Although Mitchell's implicit method requires more cpu time than the explicit NH4 method, it is always more accurate and more efficient to use. For example, for the asymptotic diffusion profile, using Mitchell's method with $J = 40$ is more accurate than using the NH4 method with $J = 50$, and takes only about 54% the cpu time.

The advantage of using the fourth-order schemes over the second-order ones is obvious in terms of overall accuracy and efficiency. In terms of raw computational time, the fourth-order schemes are however more time consuming. For example, for the linear diffusion profile, the NH4 method requires about 2.7 times longer to run for each grid than the FTCS method, and about 1.7 times longer than the NH2 method.

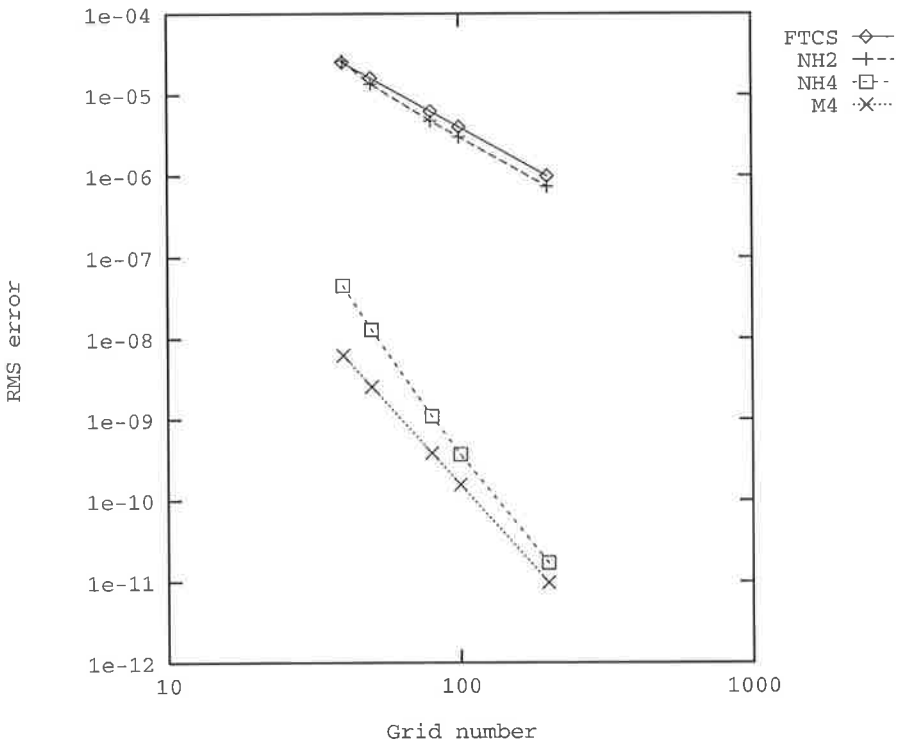

Figure 8.1: Convergence for asymptotic diffusion profile.

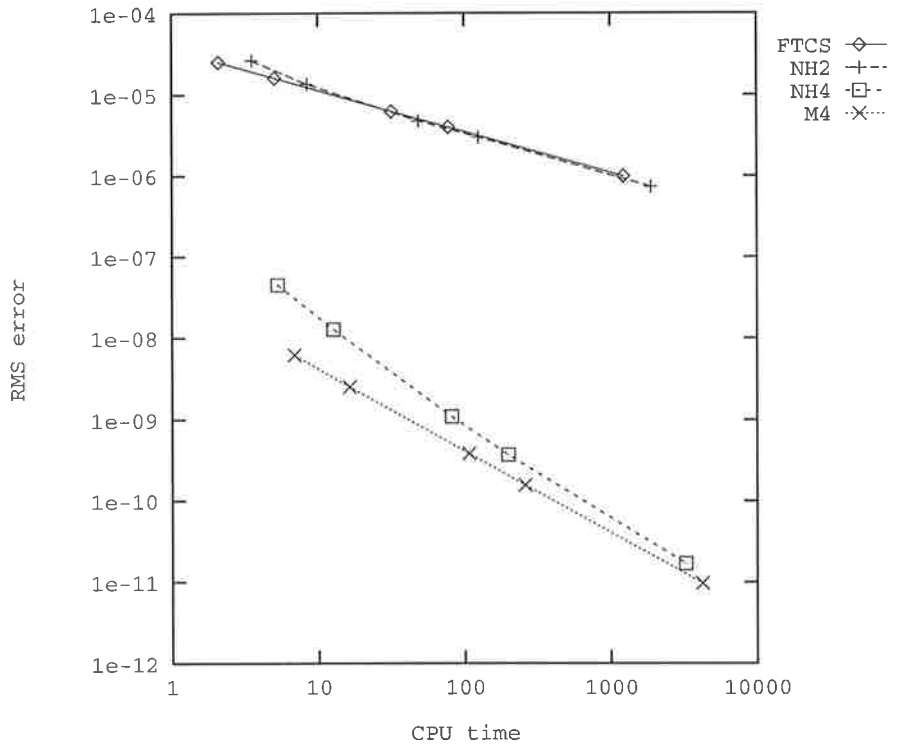

Figure 8.2: Efficiency for asymptotic diffusion profile.

	Linear Time Variation			
J	FTCS	NH2	NH4	M4
40	2.70e−05	2.83e−05	4.17e−08	6.49e−09
50	1.72e−05	1.49e−05	1.18e−08	2.65e−09
80	6.66e−06	5.23e−06	1.01e−09	4.01e−10
100	4.25e−06	3.29e−06	3.42e−10	1.64e−10
200	1.06e−06	8.09e−07	1.56e−11	1.02e−11
	Asymptotic Time Variation			
J	FTCS	NH2	NH4	M4
40	2.53e−05	2.64e−05	4.50e−08	6.18e−09
50	1.61e−05	1.36e−05	1.27e−08	2.52e−09
80	6.24e−06	4.75e−06	1.09e−09	3.81e−10
100	3.98e−06	2.99e−06	3.69e−10	1.56e−10
200	9.91e−07	7.34e−07	1.69e−11	9.69e−12
	Exponential Time Variation			
J	FTCS	NH2	NH4	M4
40	2.60e−05	2.72e−05	4.35e−08	6.31e−09
50	1.66e−05	1.42e−05	1.23e−08	2.57e−09
80	6.42e−06	4.96e−06	1.06e−09	3.89e−10
100	4.10e−06	3.12e−06	3.58e−10	1.59e−10
200	1.02e−06	7.67e−07	1.64e−11	9.88e−12

Table 8.3: RMS errors for two-dimensional diffusion problem.

	Linear Time Variation			
J	FTCS	NH2	NH4	M4
40	1.82e+00	2.80e+00	4.89e+00	6.43e+00
50	4.47e+00	6.92e+00	1.18e+01	1.56e+01
80	2.82e+01	4.39e+01	7.64e+01	1.03e+02
100	7.02e+01	1.08e+02	1.88e+02	2.49e+02
200	1.14e+03	1.78e+03	3.08e+03	4.05e+03
	Asymptotic Time Variation			
J	FTCS	NH2	NH4	M4
40	2.09e+00	3.52e+00	5.27e+00	6.86e+00
50	5.05e+00	8.40e+00	1.27e+01	1.64e+01
80	3.17e+01	4.86e+01	8.14e+01	1.07e+02
100	7.75e+01	1.25e+02	1.98e+02	2.59e+02
200	1.23e+03	1.89e+03	3.26e+03	4.24e+03
	Exponential Time Variation			
J	FTCS	NH2	NH4	M4
40	3.74e+00	4.83e+00	1.02e+01	1.05e+01
50	8.62e+00	1.14e+01	2.53e+01	2.54e+01
80	5.45e+01	7.55e+01	1.63e+02	1.66e+02
100	1.32e+02	1.80e+02	3.97e+02	4.05e+02
200	2.15e+03	2.67e+03	6.34e+03	6.46e+03

Table 8.4: CPU times for two-dimensional diffusion problem.

8.4.3 Comment

Because of the lack of availability of exact solutions for more general problems, the methods in this chapter were tested for the special case α_x independent of y and α_y independent of x. However, as already discussed in Section 5.8.2, because for each sweep in the x direction the value of y is held constant, and for each sweep in the y direction the value of x is held constant, the LOD technique should be just as successful if it is implemented for the more general case.

Chapter 9

One-Dimensional Diffusion: General Case

The most general version of the diffusion equation is given by

$$\frac{\partial \hat{\tau}}{\partial t} - \frac{\partial}{\partial x}\left(\alpha \frac{\partial \hat{\tau}}{\partial x}\right) = 0, \tag{9.1}$$

in which the diffusion coefficient $\alpha(x,t) > 0$. This may be written as

$$\frac{\partial \hat{\tau}}{\partial t} - \frac{\partial \alpha(x,t)}{\partial x}\frac{\partial \hat{\tau}}{\partial x} - \alpha(x,t)\frac{\partial^2 \hat{\tau}}{\partial x^2} = 0, \tag{9.2}$$

which is equivalent to the variable-coefficient transport equation, namely

$$\frac{\partial \hat{\tau}}{\partial t} + u(x,t)\frac{\partial \hat{\tau}}{\partial x} - \alpha(x,t)\frac{\partial^2 \hat{\tau}}{\partial x^2} = 0, \tag{9.3}$$

where $u = -\partial\alpha/\partial x$. Equations (9.2) and (9.3) reveal the presence of an intrinsic advective component in the diffusion process, which was neglected in Chapters 7 and 8. In this chapter, we will investigate how this advection component can be incorporated, so that (9.1) can be accurately approximated. The first approach is to approximate (9.3) using several well-known methods for the constant-coefficient transport equation (see Hogarth et al. 1990, Noye 1987b, Noye 1987c). The accuracy of these methods will be analyzed, and it will be shown how they can be modified to give high-order convergence in the variable-diffusion situation. The analysis of the methods will be kept as general as possible so that they may readily be applied to either (9.2), or to (9.3), where u and α are arbitrary functions. The second approach is to use process splitting, whereby (9.3) is divided into the two separate processes of advection and diffusion, each of which is solved separately every time-step. Any of the methods developed in Chapters 3 or 7 may be used for the advection and diffusion components, respectively.

9.1 The FTCS Method

If (9.3) is discretized at (x_j, t_n) using forward-time and centered-space differencing (Noye 1987b), then

$$\frac{\hat{\tau}_j^{n+1} - \hat{\tau}_j^n}{\Delta t} + u\frac{\hat{\tau}_{j+1}^n - \hat{\tau}_{j-1}^n}{2\Delta x} - \alpha\frac{\hat{\tau}_{j+1}^n - 2\hat{\tau}_j^n + \hat{\tau}_{j-1}^n}{(\Delta x)^2} + O\{\Delta t, (\Delta x)^2\} = 0. \tag{9.4}$$

Omitting the second-order error terms, where Δt is proportional to $(\Delta x)^2$, gives the second-order FTCS method, namely

$$\tau_j^{n+1} = \frac{1}{2}(2s + c)\tau_{j-1}^n + (1 - 2s)\tau_j^n + \frac{1}{2}(2s - c)\tau_{j+1}^n, \tag{9.5}$$

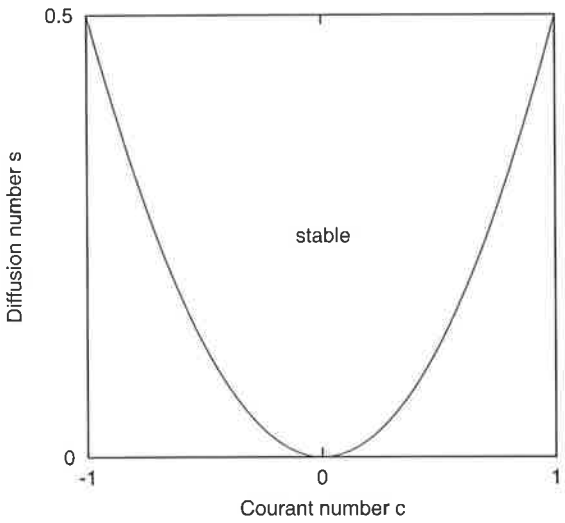

Figure 9.1: Stability region for the FTCS method.

which is stable for $c^2 \leq 2s \leq 1$ (Noye 1987b). This region, shown in Figure 9.1, indicates that (9.5) is particularly suited for application to diffusion dominated transport problems. However, if advection dominates, (9.5) may be unstable. The FTCS method may readily be applied to (9.2) by noting that

$$c = -\frac{\partial \alpha}{\partial x}\frac{\Delta t}{\Delta x}. \tag{9.6}$$

All variable quantities are evaluated at (x_j, t_n), but the superscripts and subscripts have been omitted for convenience.

9.2 The Lax-Wendroff Method

A second-order explicit method for solving the constant-coefficient transport equation is the Lax-Wendroff method (Lax and Wendroff 1964), namely

$$\tau_j^{n+1} = \frac{1}{2}(2s + c + c^2)\tau_{j-1}^n + (1 - 2s - c^2)\tau_j^n + \frac{1}{2}(2s - c + c^2)\tau_{j+1}^n. \tag{9.7}$$

Setting $s = 0$ in (9.7) gives Leith's method for pure advection, while setting $c = 0$ yields the FTCS method for pure diffusion. The Lax-Wendroff method, denoted LW2, is stable if $0 < s \leq (1 - c^2)/2$ (Lax and Wendroff 1964).

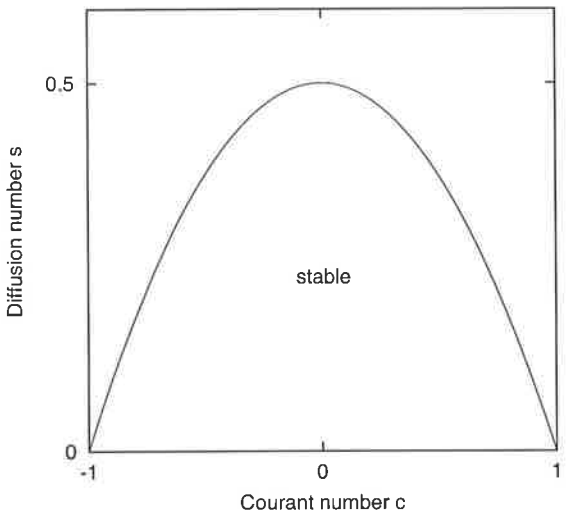

Figure 9.2: Stability region for the Lax-Wendroff method.

The size of the stability region is the same as for the FTCS method, but this method is suitable for advection dominated transport problems as well as for diffusion dominated problems. For variable-coefficients, the convergence of the LW2 method may be determined from a consistency analysis, resulting in

$$\mathcal{F}\{\hat{\tau}\} = -\Delta t E_{\mathrm{T}}, \tag{9.8}$$

where

$$E_{\mathrm{T}} = -\frac{1}{2}\Delta t\left(\frac{\partial^2\hat{\tau}}{\partial t^2} - u^2\frac{\partial^2\hat{\tau}}{\partial x^2} + \frac{1}{3}\frac{u(\Delta x)^2}{\Delta t}\frac{\partial^3\hat{\tau}}{\partial x^3} - \frac{1}{6}\frac{\alpha(\Delta x)^2}{\Delta t}\frac{\partial^4\hat{\tau}}{\partial x^4}\right) + O\{4\}. \tag{9.9}$$

The time derivative can be converted to space derivatives as follows. First (9.3) is written in the form

$$\frac{\partial\hat{\tau}}{\partial t} = -u\frac{\partial\hat{\tau}}{\partial x} + \alpha\frac{\partial^2\hat{\tau}}{\partial x^2}, \tag{9.10}$$

then, differentiating with respect to x, gives

$$\frac{\partial^2\hat{\tau}}{\partial x\partial t} = -\frac{\partial u}{\partial x}\frac{\partial\hat{\tau}}{\partial x} + \left(\frac{\partial\alpha}{\partial x} - u\right)\frac{\partial^2\hat{\tau}}{\partial x^2} + \alpha\frac{\partial^3\hat{\tau}}{\partial x^3}. \tag{9.11}$$

Differentiating (9.11) with respect to x, yields

$$\frac{\partial^3\hat{\tau}}{\partial x^2\partial t} = -\frac{\partial^2 u}{\partial x^2}\frac{\partial\hat{\tau}}{\partial x} + \left(\frac{\partial^2\alpha}{\partial x^2} - 2\frac{\partial u}{\partial x}\right)\frac{\partial^2\hat{\tau}}{\partial x^2} + \left(2\frac{\partial\alpha}{\partial x} - u\right)\frac{\partial^3\hat{\tau}}{\partial x^3} + \alpha\frac{\partial^4\hat{\tau}}{\partial x^4}. \tag{9.12}$$

Now, if (9.10) is differentiated with respect to t, and (9.11) and (9.12) substituted, this produces

$$\frac{\partial^2\hat{\tau}}{\partial t^2} = \mathrm{P}\frac{\partial\hat{\tau}}{\partial x} + \mathrm{Q}\frac{\partial^2\hat{\tau}}{\partial x^2} + \mathrm{R}\frac{\partial^3\hat{\tau}}{\partial x^3} + \alpha^2\frac{\partial^4\hat{\tau}}{\partial x^4}, \tag{9.13}$$

where

$$\begin{aligned}
\mathrm{P} &= -\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} - \alpha\frac{\partial^2 u}{\partial x^2},\\
\mathrm{Q} &= \frac{\partial\alpha}{\partial t} - u\frac{\partial\alpha}{\partial x} + \alpha\frac{\partial^2\alpha}{\partial x^2} - 2\alpha\frac{\partial u}{\partial x} + u^2,\\
\mathrm{R} &= 2\alpha\frac{\partial\alpha}{\partial x} - 2\alpha u.
\end{aligned} \tag{9.14}$$

In the special case $u = -\partial\alpha/\partial x$, this may also be written

$$\begin{aligned}
\mathrm{P}' &= \frac{\partial^2\alpha}{\partial x\partial t} + \frac{\partial\alpha}{\partial x}\frac{\partial^2\alpha}{\partial x^2} + \alpha\frac{\partial^3\alpha}{\partial x^3},\\
\mathrm{Q}' &= \frac{\partial\alpha}{\partial t} + 2\left(\frac{\partial\alpha}{\partial x}\right)^2 + 3\alpha\frac{\partial^2\alpha}{\partial x^2},\\
\mathrm{R}' &= 4\alpha\frac{\partial\alpha}{\partial x}.
\end{aligned} \tag{9.15}$$

To keep the analysis as general as possible, (9.14) rather than (9.15) will usually be used.
Substituting (9.13) into (9.9) then gives

$$E_{\mathrm{T}} = -\frac{1}{2}\Delta t\left[\mathrm{A}\frac{\partial\hat{\tau}}{\partial x} + \mathrm{B}\frac{\partial^2\hat{\tau}}{\partial x^2} + \mathrm{C}\frac{\partial^3\hat{\tau}}{\partial x^3} + \mathrm{D}\frac{\partial^4\hat{\tau}}{\partial x^4}\right] + O\{4\}, \tag{9.16}$$

where

$$\begin{aligned} \mathrm{A} &= \mathrm{P}, \\ \mathrm{B} &= \mathrm{Q} - u^2, \\ \mathrm{C} &= \mathrm{R} + \frac{1}{3}\frac{u(\Delta x)^2}{\Delta t}, \\ \mathrm{D} &= \alpha^2 - \frac{1}{6}\frac{\alpha(\Delta x)^2}{\Delta t}. \end{aligned} \tag{9.17}$$

If Δt is proportional to $(\Delta x)^2$, then the truncation error corresponds to second-order convergence. The truncation error can be modified to give fourth-order convergence as follows.

9.2.1 Modification

Substituting (9.16) into (9.8) gives

$$\mathcal{F}\{\hat{\tau}\} = \frac{1}{2}(\Delta t)^2 \left[\mathrm{A}\frac{\partial \hat{\tau}}{\partial x} + \mathrm{B}\frac{\partial^2 \hat{\tau}}{\partial x^2} + \mathrm{C}\frac{\partial^3 \hat{\tau}}{\partial x^3} + \mathrm{D}\frac{\partial^4 \hat{\tau}}{\partial x^4}\right] + O\{6\}. \tag{9.18}$$

Then, if all the space derivatives in (9.18) are replaced by second-order centered-space difference forms, and the sixth-order error terms are omitted, the fourth-order LW4 method is obtained, namely

$$\begin{aligned} \tau_j^{n+1} = (f-h)\tau_{j-2}^n + \frac{1}{2}(2s + c + c^2 - 8f + 4h + 2b - 2g)\tau_{j-1}^n + (1 - 2s - c^2 + 6f - 2b)\tau_j^n \\ + \frac{1}{2}(2s - c + c^2 - 8f - 4h + 2b + 2g)\tau_{j+1}^n + (f+h)\tau_{j+2}^n, \end{aligned} \tag{9.19}$$

where the correction factors are defined as

$$g = \frac{(\Delta t)^2}{4\Delta x}\mathrm{A}, \qquad b = \frac{(\Delta t)^2}{2(\Delta x)^2}\mathrm{B}, \qquad h = \frac{(\Delta t)^2}{4(\Delta x)^3}\mathrm{C}, \qquad f = \frac{(\Delta t)^2}{2(\Delta x)^4}\mathrm{D}. \tag{9.20}$$

9.2.2 The Inverted (5,1) Method

Since the LW4 method can only be applied for $j = 2(1)J-2$, it is inverted to give a fourth-order method for application near the boundaries. This is achieved as follows. Consider (9.19) written in the form

$$\tau_j^{n+1} = a_1\tau_{j-2}^n + a_2\tau_{j-1}^n + a_3\tau_j^n + a_4\tau_{j+1}^n + a_5\tau_{j+2}^n, \tag{9.21}$$

where the coefficients $a_1, \ldots, a_5$ are evaluated at (x_j, t_n). If Δt is replaced by $-\Delta t$, then $(n-1)$ replaces $(n+1)$ in (9.21), and shifting the index so that $(n+1)$ replaces n in the resultant expression, yields the

implicit (5,1) method given by

$$\tau_j^n = a_1\tau_{j-2}^{n+1} + a_2\tau_{j-1}^{n+1} + a_3\tau_j^{n+1} + a_4\tau_{j+1}^{n+1} + a_5\tau_{j+2}^{n+1}, \tag{9.22}$$

where the coefficients $a_1, \ldots, a_5$ are evaluated at (x_j, t_{n+1}). So, in order to solve for τ_1^{n+1}, c is replaced by $-c$, s is replaced by $-s$, and

$$\begin{aligned} \mathrm{C}^* &= \mathrm{R} - \frac{1}{3}\frac{u(\Delta x)^2}{\Delta t}, \\ \mathrm{D}^* &= \alpha^2 + \frac{1}{6}\frac{\alpha(\Delta x)^2}{\Delta t}, \end{aligned} \tag{9.23}$$

replace C and D in (9.22). By setting $j = 2$ we obtain

$$\tau_2^n = a_1\tau_0^{n+1} + a_2\tau_1^{n+1} + a_3\tau_2^{n+1} + a_4\tau_3^{n+1} + a_5\tau_4^{n+1}. \tag{9.24}$$

Rearranging (9.24) gives an explicit expression for τ_1^{n+1}, where τ_0^{n+1} is given by the left hand boundary condition, and the other values of τ at the new time level have already been calculated using (9.19). The coefficients $a_1, \ldots, a_5$ are evaluated at (x_2, t_{n+1}). The value of τ_{J-1}^{n+1} is obtained in the same fashion. Setting $j = J-2$ in (9.22), gives

$$\tau_{J-2}^n = a_1\tau_{J-4}^{n+1} + a_2\tau_{J-3}^{n+1} + a_3\tau_{J-2}^{n+1} + a_4\tau_{J-1}^{n+1} + a_5\tau_J^{n+1}, \tag{9.25}$$

in which the coefficients $a_1, \ldots, a_5$ are evaluated at (x_{J-2}, t_{n+1}). The value of τ_J^{n+1} is given by the right hand boundary condition.

9.3 The Noye and Tan Method

The greatest order of convergence for solving the constant-coefficient transport equation using a (2,3) stencil is 3 (see Noye 1987c). Such a method has been developed by Noye and Tan (1988), namely

$$\begin{aligned} &-2c(6s - 1 + c^2)\tau_{j-1}^{n+1} + 2(1+c)(6s + 2c + c^2)\tau_j^{n+1} \\ &\quad = \left(12s^2 + c(1+c)^2(2+c)\right)\tau_{j-1}^n - 2\left(12s^2 - 6s + c(c^2-1)(c+2)\right)\tau_j^n + (12s^2 - c^2 + c^4)\tau_{j+1}^n, \end{aligned} \tag{9.26}$$

which can be marched across the spatial grid from left to right when $u > 0$, making the computational procedure explicit in nature, thereby reducing the cpu times considerably. The Noye and Tan method, denoted NAT, is stable when (see Noye and Tan 1988)

$$\frac{1}{12}(c+1)\left(3 - \sqrt{3(3-2c)(2c+1)}\right) \le s \le \frac{1}{12}(c+1)\left(3 + \sqrt{3(3-2c)(2c+1)}\right), \tag{9.27}$$

which requires $0 \leq c \leq 3/2$. The Noye and Tan method is marching stable when $s > 0$, $c \geq 0$. Therefore, it may be used in the region displayed in Figure 9.3. A different formula, which may be marched in the negative x direction, that is, from right to left across the spatial grid, can be developed by replacing c by $-c$ in (9.26). The resultant formula is marching stable if $c \leq 0$, and may be used in the region obtained by replacing c by $-c$ in (9.27), provided $c \leq 0$.

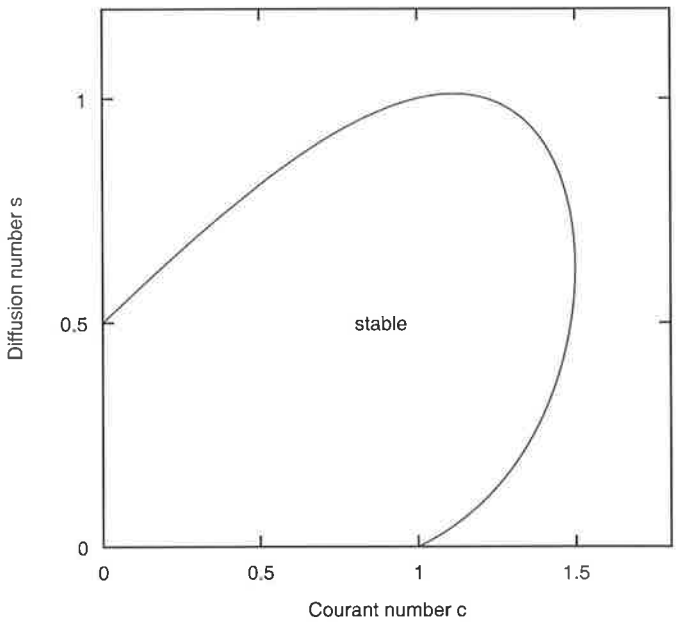

Figure 9.3: Usable region for the Noye and Tan method.

The region shown in Figure 9.3 is larger than for the explicit methods considered. The Noye and Tan method, which is third-order for constant coefficients, becomes second-order for variable c and s.

9.4 Optimal Two-Weight Method

If (9.3) is discretized at the point $(j\Delta x, (n+\theta)\Delta t)$, in the way described by Noye (1987b), then the following implicit (3,3) equation is obtained

$$\begin{aligned} &-\theta(c+\phi)\tau_{j-1}^{n+1} + 2(1+\theta\phi)\tau_j^{n+1} + \theta(c-\phi)\tau_{j+1}^{n+1} \\ &\quad = (1-\theta)(c+\phi)\tau_{j-1}^{n} + 2(1-\phi(1-\theta))\tau_j^{n} - (1-\theta)(c-\phi)\tau_{j+1}^{n}. \end{aligned} \tag{9.28}$$

For constant coefficients, the optimal third-order FDE is obtained by setting the weights

$$\begin{aligned}\theta &= [\,3c^2 + 6s - \sqrt{6c^2 + 3c^4 + 36s^2}\,]/(6c^2),\\ \phi &= \psi c + 2s,\\ \psi &= c(1 - 2\theta).\end{aligned} \tag{9.29}$$

It can be shown (Noye 1987b) that this method, denoted OPT2 in the following, is stable when

$$\begin{aligned}&s > 0, && \text{if} \quad |c| \le 1, \;\; c \neq 0\\ &s \ge c(c^2 - 1)/\sqrt{12(4 - c^2)}, && \text{if} \quad 1 \le c < 2,\\ &s \ge -c(c^2 - 1)/\sqrt{12(4 - c^2)}, && \text{if} \quad -2 < c \le -1,\end{aligned} \tag{9.30}$$

and diagonally dominant provided $c \neq 0$ (Noye 1987b). This region is shown in Figure 9.4 for $c > 0$. Although the OPT2 method has the greatest stability region, it cannot be applied when $c = 0$.

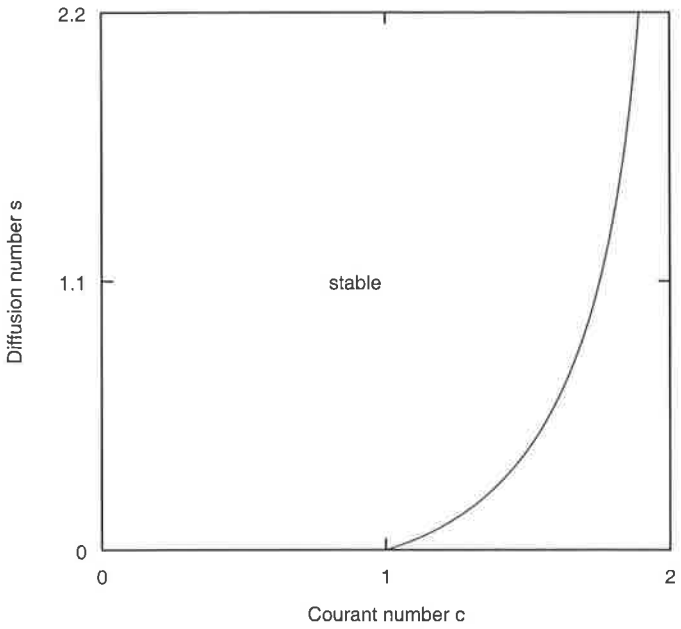

Figure 9.4: Stability region for the OPT2 method.

The order of convergence of the OPT2 method for variable coefficients can be determined by taking the Taylor series expansion of each term in (9.28) about (x_j, t_n), yielding

$$\mathcal{F}\{\hat{\tau}\} = -2\Delta t E_{\mathrm{T}}, \tag{9.31}$$

where

$$E_{\mathrm{T}} = -\frac{\Delta t}{2}\left(\frac{\partial^2\hat{\tau}}{\partial t^2} + 2u\theta\frac{\partial^2\hat{\tau}}{\partial x\partial t} - u^2(1-2\theta)\frac{\partial^2\hat{\tau}}{\partial x^2} - 2\alpha\theta\frac{\partial^3\hat{\tau}}{\partial x^2\partial t} + \frac{1}{3}\frac{u(\Delta x)^2}{\Delta t}\frac{\partial^3\hat{\tau}}{\partial x^3} - \frac{1}{6}\frac{\alpha(\Delta x)^2}{\Delta t}\frac{\partial^4\hat{\tau}}{\partial x^4}\right) + O\{4\}.$$

9.4.1 Modification

If the time derivatives are converted to space derivatives, (9.31) becomes

$$\mathcal{F}\{\hat{\tau}\} = (\Delta t)^2\left[\mathrm{A}\frac{\partial\hat{\tau}}{\partial x} + \mathrm{B}\frac{\partial^2\hat{\tau}}{\partial x^2} + \mathrm{C}\frac{\partial^3\hat{\tau}}{\partial x^3} + \mathrm{D}\frac{\partial^4\hat{\tau}}{\partial x^4}\right] + O\{6\}, \tag{9.32}$$

where

$$\begin{aligned}
\mathrm{A} &= -\frac{\partial u}{\partial t} + u\beta\frac{\partial u}{\partial x} - \alpha\beta\frac{\partial^2 u}{\partial x^2},\\
\mathrm{B} &= \frac{\partial\alpha}{\partial t} - u\beta\frac{\partial\alpha}{\partial x} + \alpha\beta\frac{\partial^2\alpha}{\partial x^2} - 2\alpha\beta\frac{\partial u}{\partial x},\\
\mathrm{C} &= 2\alpha\beta\frac{\partial\alpha}{\partial x} - 2u\alpha\beta + \frac{1}{3}u\alpha s^{-1},\\
\mathrm{D} &= \alpha^2\beta - \frac{1}{6}\alpha^2 s^{-1},
\end{aligned} \tag{9.33}$$

in which $\beta = 1 - 2\theta$. The spatial derivatives are replaced by second-order centered-space difference forms, yielding the fourth-order OPT4 method, namely

$$\begin{aligned}
&-\theta(c+\phi)\tau_{j-1}^{n+1} + 2(1+\theta\phi)\tau_j^{n+1} + \theta(c-\phi)\tau_{j+1}^{n+1} =\\
&\quad (f-h)\tau_{j-2}^n + \big((1-\theta)(c+\phi) - 4f + 2h + b - g\big)\tau_{j-1}^n + 2(1-\phi(1-\theta) + 3f - b)\tau_j^n\\
&\qquad +(-(1-\theta)(c-\phi) - 4f - 2h + b + g)\tau_{j+1}^n + (f+h)\tau_{j+2}^n,
\end{aligned} \tag{9.34}$$

where the weights are given by (9.29), and the correction factors are given by

$$g = \frac{(\Delta t)^2}{2\Delta x}\mathrm{A}, \qquad b = \frac{(\Delta t)^2}{(\Delta x)^2}\mathrm{B}, \qquad h = \frac{(\Delta t)^2}{2(\Delta x)^3}\mathrm{C}, \qquad f = \frac{(\Delta t)^2}{(\Delta x)^4}\mathrm{D}. \tag{9.35}$$

Because of its computational stencil, the OPT4 method can only be applied for $j = 2(1)J-2$, and so off-centered versions were developed to supplement it near the boundaries. If (A.8) and (A.10) are used for the third and fourth spatial derivatives in (9.32), then a method which can be used with $j = 1$ is obtained. If Δx is replaced by $-\Delta x$ in (A.8) and (A.10), and the resultant difference forms are used for the third and fourth spatial derivatives, then a method which can be used with $j = J - 1$ is obtained.

9.5 Process Splitting

It is possible to approximate solutions to (9.3) by solving the advection and diffusion components separately. A Strang type splitting algorithm (Strang 1968) must be used, whereby the order of the subprocesses is reversed each time-step, so that the composite solution retains the accuracy of the individual schemes.

A review of the literature suggests that all Strang type splitting algorithms are second-order, even if the subprocesses are treated exactly (Khan and Liu 1995, Vreugdenhil and Koren 1993, Leveque and Oliger 1983, Gourlay and Mitchell 1972). Consequently, little attention has been paid in the past to achieve higher than second-order splitting algorithms for either constant or variable-coefficient problems.

It has already been seen in Chapter 6, where the advection-decay equation was solved using a Strang type splitting algorithm, that higher than second-order convergence can be attained by using high-order algorithms for the component equations.

For the problem considered here, any of the two-level methods described in Chapters 3 and 7 can be used for the advection and diffusion components, respectively. However, because the boundary conditions are defined for the governing equation, it is necessary to determine appropriate boundary conditions for the intermediate step.

Very little work has been done to derive appropriate intermediate boundary conditions for the splitting of the transport equation. However, a good review is given by Khan and Liu (1995), who derive intermediate boundary conditions for the advection dominated transport equation, but assume u and α to be locally constant. Much additional work is required if these are allowed to vary.

A way to overcome the difficulty of deriving intermediate boundary conditions is to use, for example, an explicit (1,3) method for the advection process. The intermediate boundary values τ_0^* and τ_J^* are then computed separately using a (3,3) method. The values τ_j^*, $j = 0(1)J$, are then used as the initial conditions for the diffusion stage. The boundary conditions for this stage are the given conditions.

Because the order of the subprocesses is reversed, intermediate boundary conditions are then required for the diffusion process. Again, if a (1,3) stencil is used, then the intermediate boundary conditions are found separately using a (3,3) method. This process is repeated until the final time is reached. Since the optimal order of convergence for a (1,3) stencil is second-order, this is the maximum order of convergence that can be obtained from such a splitting procedure.

In the numerical tests to follow, PS1 will be used to denote the process splitting method when the mod_L method is used for the advection step and the FTCS method, Equation (7.8), is used for the diffusion step. The boundary values at the intermediate level are evaluated using the second-order OPT method for advection, and the second-order Crank-Nicolson method (Crank and Nicolson 1947) for diffusion.

A splitting algorithm in which the component equations have (1,5) stencils can be defined as follows: the initial condition is extrapolated to give fictitious values near the boundaries outside the computational domain, so that the first method can be used for $j = 0(1)J$. The intermediate values are the initial conditions for the second stage, and the boundary conditions are those defined for the given problem.

The intermediate values are then extrapolated, so that the second method can be used for $j = 0(1)J$. The order of the processes is reversed, and the procedure is repeated until the final time is reached. The optimal convergence for a (1,5) stencil is fourth-order, so this is the maximum order that can be expected from this splitting algorithm.

In the numerical tests to follow, PS2 will be used to denote the process splitting method when the mod2_R method is used for advection and the NH4 method is used for diffusion. The analytical solution developed in the next section will be used to extend the computational domain as required.

Although the mod2_R method is third-order convergent, it gave fourth-order results for grid numbers not exceeding 500 in the numerical test described in Section 4.1 (see Figure 4.4). Since the largest grid number used in the following tests is 500, fourth-order results can be obtained from the PS2 algorithm.

Once intermediate boundary conditions are known, methods with implicit (3,3) stencils can be used as the component equations for the splitting algorithm. PS3 will be used to denote the application of Mitchell's method for diffusion and the mod2_O method for advection, after the intermediate boundary values have been determined using the NH4 and mod2_R methods.

The splitting algorithms are stable provided the component equations are stable over the step for which they are used (a proof of this is given in Noye 1987c). This generally ensures a greater stability range than for methods based on the discretization of the complete transport equation.

A problem may arise if the stability of the schemes used to determine the intermediate boundary conditions is more restricted than that of the component equations. For the algorithms described above, this problem only affects the stability of the PS3 algorithm. Since Mitchell's method is supplemented by the NH4 method, the overall stability of the diffusion stage is reduced to $s \leq 2/3$. Clearly, it would be better to be able to derive scheme independent boundary conditions, so that this problem is eliminated.

9.6 An Analytical Solution

Consider (9.3) with $u = -\partial\alpha/\partial x$. A solution will be sought when $\hat{\tau} = \hat{\tau}(x,t)$ is in the separable form

$$\hat{\tau}(x,t) = X(x)T(t), \tag{9.36}$$

and $\alpha(x,t) = f(x)g(t)$ is also separable. It follows that $u(x,t) = -f'(x)g(t)$. Substituting in (9.3) gives

$$T'X - f'gTX' - fgTX'' = 0, \tag{9.37}$$

which yields the two ordinary differential equations

$$\frac{T'}{gT} = \frac{f'X'}{X} + \frac{fX''}{X} = K, \ \ K \text{ a constant.} \tag{9.38}$$

The solution to the first-order differential equation involving time t, is

$$T(t) = A\exp\{K\int g(t)dt\}, \ \ A \text{ a constant.} \tag{9.39}$$

A solution to the second-order differential equation involving space x, namely

$$fX'' + f'X' - KX = 0, \tag{9.40}$$

is given by the solution of Legendre's equation

$$(1-x^2)X'' - 2xX' + n(n+1)X = 0, \tag{9.41}$$

if the space-varying component of the diffusion coefficient is $f(x) = 1 - x^2$ and $K = -2$. In this case,

$$\begin{aligned} X(x) &= a_1X_1(x) + a_0X_2(x) \\ &= a_1x + a_0(1 - 0.5x\ln(1+x) + 0.5x\ln(1-x)), \end{aligned} \tag{9.42}$$

where $X_1(x)$ and $X_2(x)$ are linearly independent solutions to (9.41). In the following, $a_0 = 1$, $a_1 = 0$; and so an exact solution to (9.3) is given by

$$\hat{\tau}(x,t) = A(1 - 0.5x\ln(1+x) + 0.5x\ln(1-x))\exp\{-2\int g(t)dt\}. \tag{9.43}$$

If an asymptotic function is chosen for the time-varying component of the diffusion coefficient, namely

$$g(t) = D_0 \frac{t}{t+k} + D_m \tag{9.44}$$

with $D_0 = D_m = 1/30$ and $k = 10$, then

$$\alpha(x,t) = \frac{1}{15}(1-x^2)\left(\frac{t+5}{t+10}\right), \tag{9.45}$$

and

$$u(x,t) = \frac{2}{15}x\left(\frac{t+5}{t+10}\right). \tag{9.46}$$

An exact solution to (9.2) is then given by

$$\hat{\tau}(x,t) = A(1-0.5x\ln(1+x)+0.5x\ln(1-x))\exp\{-2t/15+2\ln(t+10)/3\}, \quad A = 10^{-2/3} \tag{9.47}$$

Note that the exact solution is unbounded when $x = 1$, so a restricted spatial domain must be used. The exact solution is displayed in Figure 9.5 for $t = 0, 5$ and 10. Since solutions for which $\hat{\tau}(x,t) < 0$, which correspond to non-physical quantities, are not considered, the diffusion equation is solved on the spatial domain $0 \leq x \leq 0.8$. Furthermore, if $D_0 = D_m = 1/20$, as in Section 7.8, then $s\max > 0.5$, so the FTCS method is unstable.

9.7 Numerical Test

A numerical solution is sought at $T = 5$ with $N = J^2$, so that Δt is proportional to $(\Delta x)^2$. The initial and boundary conditions are given by the analytical solution. The maximum value of the diffusion number is $s\max \approx 0.35$. The problem is strongly diffusion dominated, with $c\max \approx 0.02$. A summary of the data required to implement the numerical test is given in Table 9.3 in Section 9.8.

The explicit methods are solved by sweeping in the x direction, with the single unknown at the new time-level being given directly from the known values at the previous time-level. The LW4 method is supplemented near the boundaries using the inverted scheme as indicated in Section 9.2.2. The NAT method is marched across the computational domain in an explicit fashion.

The implicit methods are solved using the Thomas algorithm, with the OPT4 method being supplemented near the boundaries as indicated in Section 9.4. The methods based on splitting the governing equation are implemented in the way described in Section 9.5.

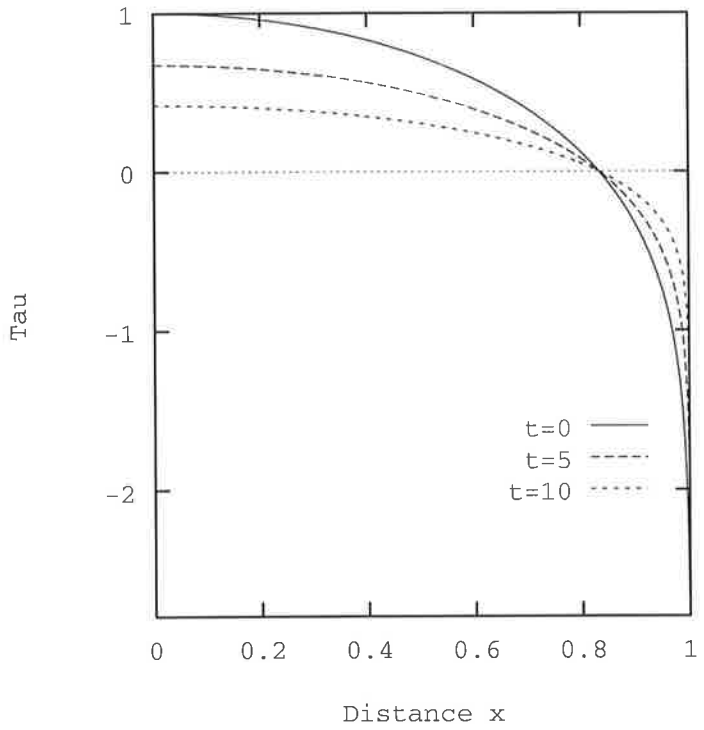

Figure 9.5: Initial condition and analytical solution at $t = 5$ and $t = 10$.

The rms error in the numerical solution relative to the analytical solution at the final time is plotted against the grid number on logarithmic scales to illustrate convergence. Graphs showing the cpu times versus the errors on logarithmic scales give an indication of the efficiency of the schemes. Diagrams showing the convergence, efficiency and computational times of the schemes are depicted in Figure 9.6. Additionally, the errors and times for the numerical test are presented in Tables 9.1 and 9.2.

The slope of the line of best fit to the data gives the orders of convergence of the methods. For the second-order schemes the orders are found to be 2.00 for all methods except for the PS1 scheme, whose order of convergence is 2.30. The orders of convergence for the fourth-order LW4, OPT4, PS2 and PS3 methods are 4.03, 3.89, 3.99 and 4.04, respectively.

Of the methods based on the discretization of the transport equation, the FTCS method is the most accurate second-order scheme (see Table 9.1). Since it also has the simplest coefficients and a (1,3) computational stencil, it also gives the smallest cpu times (see Table 9.2), making it the most efficient second-order scheme, based on the discretization of the transport equation.

The second-order PS1 process splitting algorithm, however, gives the most accurate second-order results overall (see top diagrams in Figure 9.6), giving errors one quarter the size of the FTCS method. Because the PS1 technique involves the solution of two (1,3) methods each time-step, it requires about twice the time to run for each grid than the FTCS method (see Table 9.2). Although the PS1 method takes longer to run than the FTCS method, it is clearly more efficient to use, and is the most efficient second-order scheme overall (compare the middle diagrams in Figure 9.6).

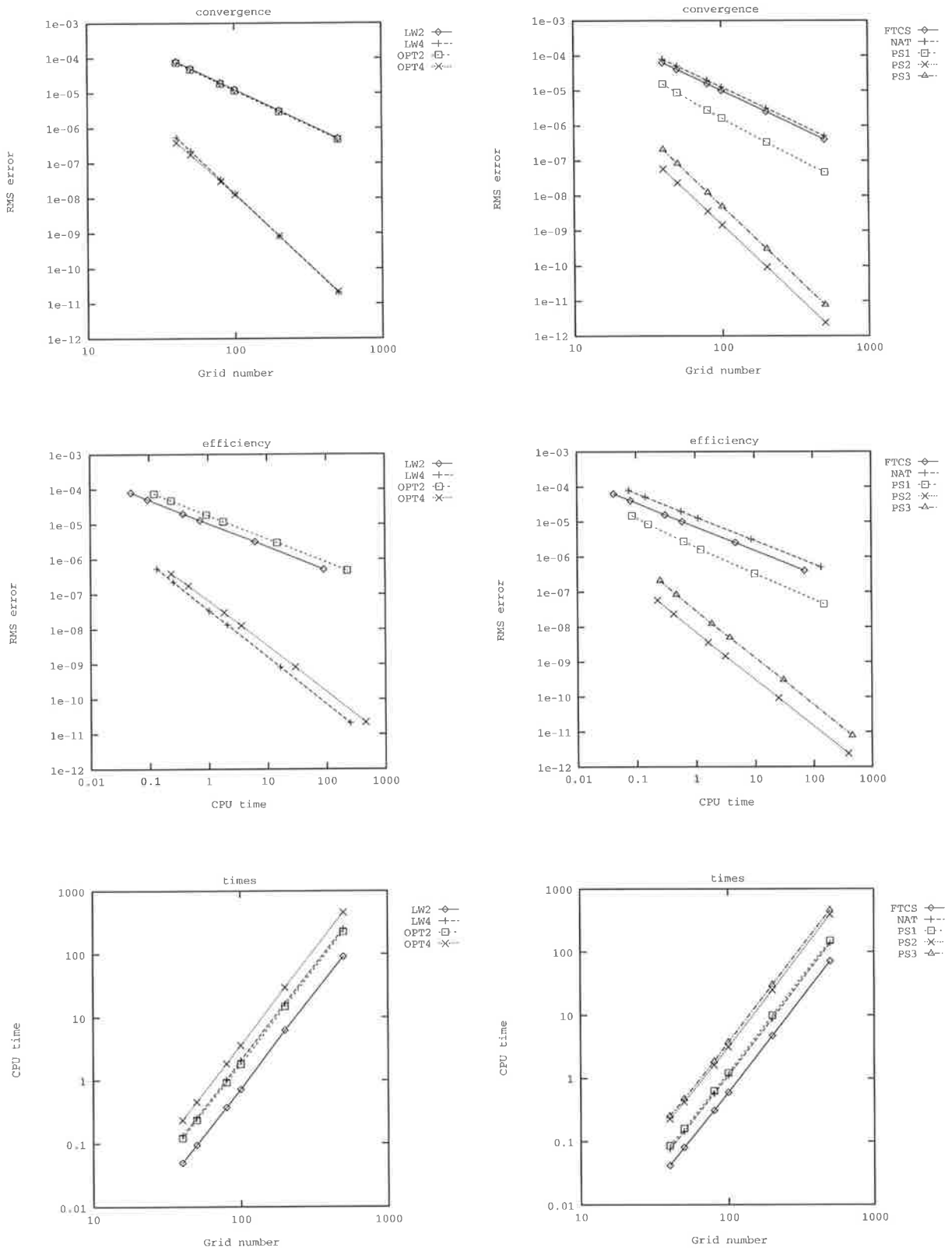

Fig 9.6: Convergence, efficiency and times shown for LW and OPT methods (left) and FTCS, NAT and PS methods (right).

J	FTCS	NAT	PS1
40	6.35e−05	7.95e−05	1.51e−05
50	4.06e−05	5.08e−05	8.63e−06
80	1.58e−05	1.98e−05	2.74e−06
100	1.01e−05	1.27e−05	1.61e−06
200	2.52e−06	3.16e−06	3.30e−07
500	4.02e−07	5.05e−07	4.57e−08
J	LW2	OPT2	PS2
40	8.04e−05	7.49e−05	5.73e−08
50	5.13e−05	4.78e−05	2.34e−08
80	2.00e−05	1.86e−05	3.59e−09
100	1.28e−05	1.19e−05	1.47e−09
200	3.19e−06	2.97e−06	9.27e−11
500	5.09e−07	4.75e−07	2.39e−12
J	LW4	OPT4	PS3
40	5.43e−07	3.87e−07	2.07e−07
50	2.22e−07	1.74e−07	8.25e−08
80	3.32e−08	3.00e−08	1.22e−08
100	1.34e−08	1.27e−08	4.93e−09
200	8.21e−10	8.32e−10	3.03e−10
500	2.08e−11	2.16e−11	7.67e−12

Table 9.1: RMS errors for the general case diffusion equation.

J	FTCS	NAT	PS1
40	4.18e−02	7.54e−02	8.52e−02
50	8.07e−02	1.44e−01	1.59e−01
80	3.08e−01	5.74e−01	6.32e−01
100	5.99e−01	1.11e+00	1.22e+00
200	4.73e+00	8.82e+00	9.94e+00
500	7.13e+01	1.37e+02	1.50e+02
J	LW2	OPT2	PS2
40	4.95e−02	1.21e−01	2.26e−01
50	9.44e−02	2.34e−01	4.21e−01
80	3.73e−01	9.29e−01	1.62e+00
100	7.22e−01	1.81e+00	3.13e+00
200	6.23e+00	1.48e+01	2.49e+01
500	9.05e+01	2.25e+02	3.90e+02
J	LW4	OPT4	PS3
40	1.33e−01	2.32e−01	2.52e−01
50	2.54e−01	4.53e−01	4.70e−01
80	1.02e+00	1.84e+00	1.85e+00
100	2.09e+00	3.59e+00	3.73e+00
200	1.65e+01	2.95e+01	3.02e+01
500	2.48e+02	4.55e+02	4.57e+02

Table 9.2: CPU times for the general case diffusion equation.

Although the NAT method is implemented in a marching fashion, it is still quite expensive to use, because of the complicated coefficients in its (2,3) computational stencil. The most expensive second-order scheme is, however, the implicit OPT2 method, which requires three times as long as the FTCS method to run for each grid. The OPT2 method, although being more accurate than the LW2 method, is somewhat less efficient to use (see middle left diagram in Figure 9.6).

As shown in the top left diagram of Figure 9.6, the two modified fourth-order schemes based on the discretization of the transport equation are of quite similar accuracy, but the explicit LW4 method is clearly more efficient to use. Both modified methods incorporate four correction factors and require special procedures near the boundaries, making them quite complicated to implement compared to the base methods. These methods do, however, give far superior results in terms of overall accuracy and efficiency to the base methods.

As shown in Table 9.1, the fourth-order process splitting algorithms, PS2 and PS3, are more accurate than the fourth-order methods based on the discretization of the transport equation. The PS2 algorithm, which involves the solution of two (1,5) methods every time-step, requires nearly twice as long to run as the explicit LW4 method, which also has a (1,5) stencil. The PS3 algorithm, which requires the solution of two systems of tridiagonal linear algebraic equations each time-step using the Thomas algorithm, is the most time consuming method of all.

9.7.1 Summary

Several techniques for solving the transport equation for the case $u = -\partial\alpha/\partial x$ were presented. Of the second-order methods based on the discretization of the transport equation, the FTCS method was the most accurate and efficient scheme, while the LW2 method was slightly less accurate than the OPT2 and NAT methods. The OPT2 and LW2 methods were modified from second to fourth-order, resulting in schemes with (3,5) and (1,5) stencils respectively. Off-centered (in the case of the OPT4 method) and inverted (in the case of the LW4 method) schemes were developed to supplement these methods near the boundaries. The results of the numerical tests showed that the modified methods gave superior results in terms of accuracy and efficiency to the base methods, but also required longer to run.

The process splitting algorithms yielded the most accurate and efficient results. The PS1 algorithm gave the most accurate and efficient second-order results, while the PS2 algorithm gave the most accurate and efficient fourth-order results. Scheme dependent intermediate boundary conditions were used in the numerical tests. As indicated by Khan and Liu (1995), it would probably be better to derive intermediate boundary conditions which are independent of the schemes used for the separate components. This may be important in situations where the conditions outside the computational domain do not match those in the interior, and would eliminate the problem of reduced stability if the supplementary schemes have more limited stability than the component equations.

9.8 Parameter Analysis

For constant coefficients, the leading term in the truncation error of the MEPDE of a FDE consistent with the transport equation (see Noye 1987b), gives an indication of how the accuracy of the scheme depends on the size of the diffusion number s and Courant number c. The MEPDE also provides information about whether the dominant error affects the scheme's ability to model the amplitude or the wave-speed of the numerical solution (see Hogarth et al. 1990, Noye 1987b).

A comparative study of finite-difference methods for the constant-coefficient transport equation is given in Hogarth et al. (1990). The methods based on the discretization of the constant-coefficient transport equation discussed in the present chapter are given in that paper, as well as the leading term in the truncation error of the MEPDE, and other relevant comments about the accuracy and efficiency of the schemes. For variable-coefficient problems, the methods must be compared in a numerical test for different values of the maximum diffusion number.

9.8.1 Numerical Test

The methods are tested for several values of the maximum diffusion number, as indicated in Table 9.3. Only the OPT2 and OPT4 methods are stable when $s\max = 1.40$. The PS3 algorithm could be used with $s\max = 0.70$, even though the NH4 method used to calculate the intermediate boundary values for the diffusion step, is only stable if $s \leq 2/3$. This is because the NH4 method is only used for a few points in the domain for which $s > 2/3$, and so round-off errors do not accumulate. The rms errors are given in Table 9.4. The cpu times are omitted because they follow from those given in Table 9.2, in the sense that doubling the number of time steps doubles the computational times, halving the number of time steps halves the times, and so on.

The convergence and efficiency of the process splitting algorithms are shown in Figure 9.8. For the methods based on the discretization of the transport equation, only the efficiency plots are presented in Figure 9.7. This is because Table 9.4 shows that for almost all of these schemes, the rms errors do not change significantly when the size of the maximum diffusion number is altered, and so the lines of convergence overlap to a large extent. From Table 9.4, the FTCS, LW2 and NAT methods are most accurate if $s\max = 0.09$, the LW4 method is most accurate for $s\max = 0.35$, and the OPT2 and OPT4 methods perform best when $s\max = 1.40$.

The OPT2 method used with $s\max = 1.40$ is more accurate than the second-order FTCS, LW2 and NAT methods, and since the cpu times for this value of $s\max$ are shortest, it is also more efficient than these schemes. Likewise, the fourth-order OPT4 method used with $s\max = 1.40$ is more accurate and efficient than the fourth-order LW4 method (compare right middle and bottom diagrams in Figure 9.7). Figure 9.8 shows that the second-order PS1 and fourth-order PS2 splitting algorithms are most accurate and efficient when $s\max = 0.35$, while the fourth-order PS3 algorithm performs best for $s\max = 0.70$.

1. Exact solution: $\hat{\tau}(x,t) = (0.1t+1)^{3/2}(1 - 0.5x\ln(1+x) + 0.5x\ln(1-x))\exp\{-2t/15\}$

2. Initial condition: $\hat{\tau}(x,0)$ is given by the exact solution

3. Boundary conditions: $\hat{\tau}(0,t)$ and $\hat{\tau}(0.8,t)$ are given by the exact solution

4. Diffusion number: $s = \alpha\Delta t/(\Delta x)^2$, $\Delta t = T/N$, $\Delta x = 0.8/J$

5. Asymptotic profile: $\alpha(x,t) = (1-x^2)(t+5)(t+10)^{-1}/15$

6. $\alpha\max = 2/45$, $T = 5$, $J = 40, 50, 80, 100, 200$

7. setting $N = J^2/4,\ J^2/2,\ J^2, 2J^2, 4J^2$ gives $s\max \approx 1.40,\ 0.70,\ 0.35,\ 0.18,\ 0.09$

Table 9.3: Data for Parameter Analysis.

9.8.2 Summary

The most accurate and efficient second-order results overall are achieved by using the PS1 algorithm with $s\max = 0.35$. For this $s\max$, the errors given by the PS1 algorithm are less than 27% of those given by the second-order OPT2 method used with $s\max = 1.40$. Additionally, from their efficiency plots it can be deduced that it takes more than 10 seconds to achieve an accuracy of 1.0e-06 using the OPT2 method, but less than 10 seconds to gain this accuracy using the PS1 algorithm.

The most accurate and efficient fourth-order results overall are attained by using the PS2 algorithm with $s\max = 0.35$. For this $s\max$, the errors given by the PS2 algorithm are less than 16% of those given by the fourth-order OPT4 method used with $s\max = 1.40$. The PS2 algorithm used with $s\max = 0.35$ requires about four times longer to run than the OPT4 method used with $s\max = 1.40$. The additional cpu time is compensated for by the superior accuracy of the PS2 algorithm, which is slightly more efficient than the OPT4 method.

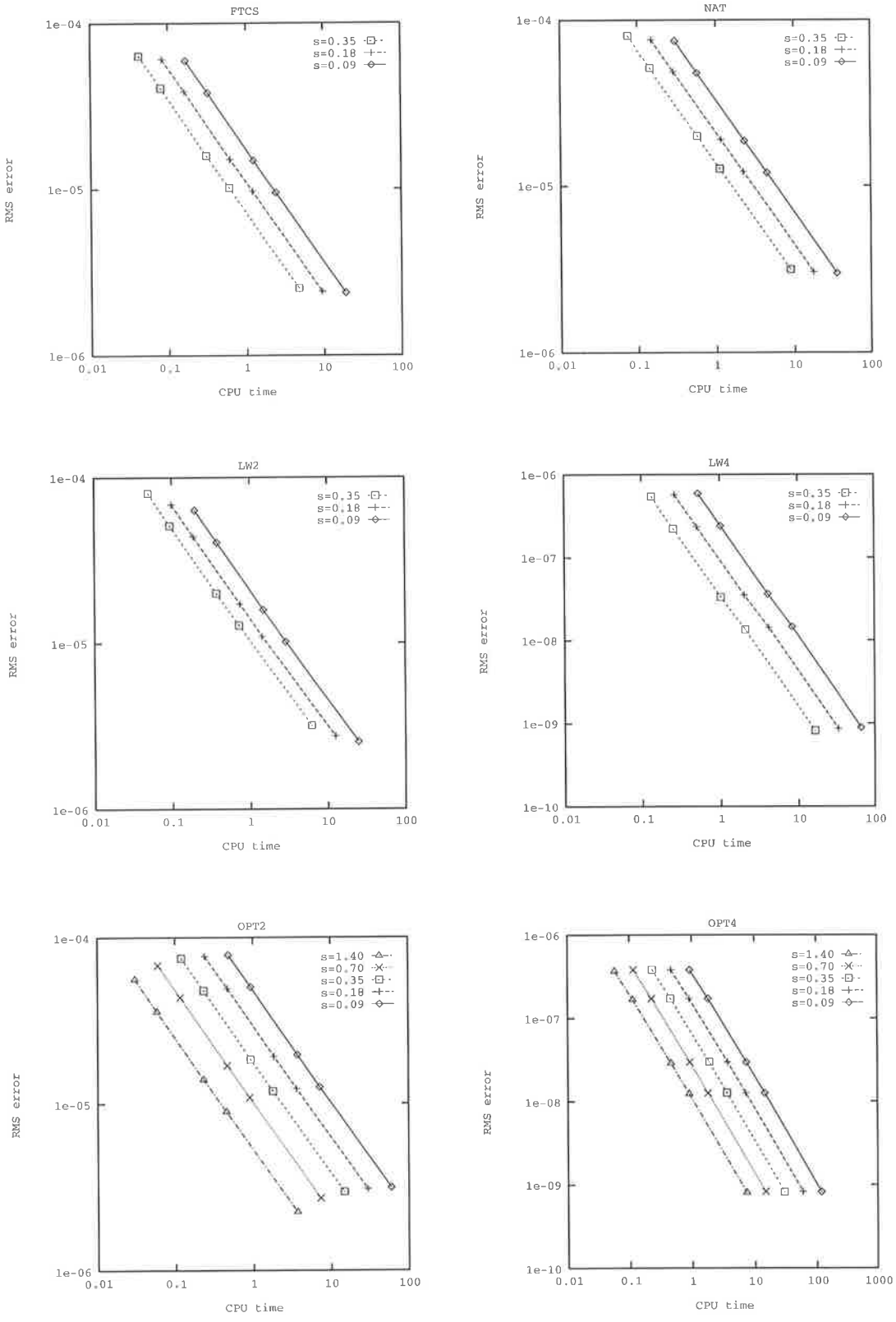

Fig 9.7: Efficiency of the methods based on the discretization of the transport equation shown for various values of $s$max.

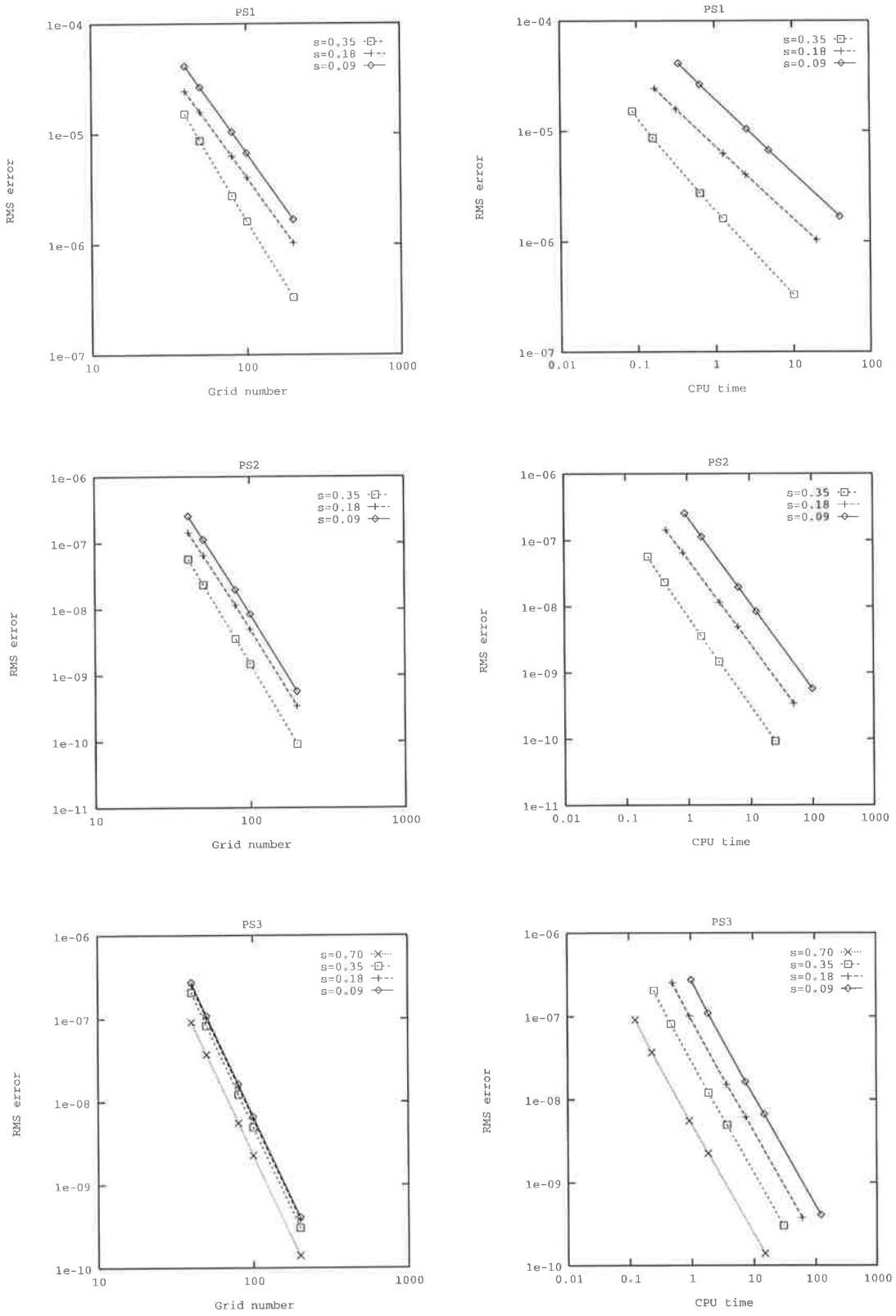

Fig 9.8: Convergence shown on left and efficiency shown on right for the process splitting algorithms for various values of $s$max.

	smax=0.09								
J	FTCS	LW2	LW4	NAT	OPT2	OPT4	PS1	PS2	PS3
40	5.94e-05	6.36e-05	5.93e-07	7.47e-05	7.85e-05	3.80e-07	4.11e-05	2.54e-07	2.73e-07
50	3.80e-05	4.07e-05	2.42e-07	4.79e-05	5.03e-05	1.72e-07	2.64e-05	1.13e-07	1.09e-07
80	1.49e-05	1.59e-05	3.61e-08	1.87e-05	1.97e-05	2.97e-08	1.04e-05	1.97e-08	1.63e-08
100	9.52e-06	1.02e-05	1.46e-08	1.20e-05	1.26e-05	1.26e-08	6.69e-06	8.42e-09	6.63e-09
200	2.38e-06	2.55e-06	8.94e-10	3.00e-06	3.15e-06	8.27e-10	1.68e-06	5.75e-10	4.10e-10
	smax=0.18								
J	FTCS	LW2	LW4	NAT	OPT2	OPT4	PS1	PS2	PS3
40	6.05e-05	6.88e-05	5.74e-07	7.60e-05	7.70e-05	3.81e-07	2.42e-05	1.43e-07	2.53e-07
50	3.87e-05	4.41e-05	2.35e-07	4.87e-05	4.93e-05	1.72e-07	1.57e-05	6.48e-08	1.01e-07
80	1.51e-05	1.72e-05	3.51e-08	1.90e-05	1.93e-05	2.98e-08	6.26e-06	1.15e-08	1.51e-08
100	9.69e-06	1.10e-05	1.42e-08	1.22e-05	1.23e-05	1.26e-08	4.04e-06	4.95e-08	6.12e-09
200	2.42e-06	2.76e-06	8.69e-10	3.05e-06	3.09e-06	8.28e-10	1.03e-06	3.43e-10	3.78e-10
	smax=0.35								
J	FTCS	LW2	LW4	NAT	OPT2	OPT4	PS1	PS2	PS3
40	6.35e-05	8.04e-05	5.43e-07	7.95e-05	7.49e-05	3.87e-07	1.51e-05	5.73e-08	2.07e-07
50	4.06e-05	5.13e-05	2.22e-07	5.08e-05	4.78e-05	1.74e-07	8.63e-06	2.34e-08	8.25e-08
80	1.58e-05	2.00e-05	3.32e-08	1.98e-05	1.86e-05	3.00e-08	2.74e-06	3.59e-09	1.22e-08
100	1.01e-05	1.28e-05	1.34e-08	1.27e-05	1.19e-05	1.27e-08	1.61e-06	1.47e-09	4.93e-09
200	2.52e-06	3.19e-06	8.21e-10	3.16e-06	2.97e-06	8.32e-10	3.30e-07	9.27e-11	3.03e-10
	smax=0.70								
J	FTCS	LW2	LW4	NAT	OPT2	OPT4	PS1	PS2	PS3
40					6.78e-05	3.81e-07			9.10e-08
50					4.35e-05	1.72e-07			3.70e-08
80	unstable	unstable	unstable	unstable	1.70e-05	2.98e-08	unstable	unstable	5.55e-09
100					1.09e-05	1.26e-08			2.26e-09
200					2.72e-06	8.29e-10			1.40e-10
	smax=1.40								
J	FTCS	LW2	LW4	NAT	OPT2	OPT4	PS1	PS2	PS3
40					5.59e-05	3.67e-07			
50					3.58e-05	1.67e-07			
80	unstable	unstable	unstable	unstable	1.40e-05	2.89e-08	unstable	unstable	unstable
100					8.96e-06	1.23e-08			
200					2.24e-06	8.07e-10			

Table 9.4: RMS errors for various values of the maximum diffusion number.

Chapter 10

Conclusion

This thesis was motivated by the lack of high-order finite-difference methods available in the literature for realistic variable-coefficient flow problems. In the past, it was necessary to use finite-difference methods based on the discretization of a constant-coefficient partial differential equation for variable-coefficient problems. These schemes ignore the variability of the advective velocity and the diffusion coefficient — a serious shortcoming considering that, in most applications, these vary in both space and time. Not surprisingly, such finite-difference methods, which may be of high-order for constant coefficients, become low-order when applied to variable-coefficient problems.

The main objective of this thesis was to modify these schemes, so that they retain their constant-coefficient order of convergence in variable-coefficient problems where the advective velocity and the diffusion field are allowed to vary in both space and time. The only constraint made on the variable-coefficients is that they are assumed to be smooth. In other words, they must be continuous and sufficiently differentiable.

A secondary goal was to compare the new modified methods with the original base methods in terms of their computational efficiency, and the introduction of certain types of errors (such as amplitude and wave-speed errors) into the numerical solution. Because of the lack of available analytical solutions in the literature for variable-coefficient problems, it was necessary to develop exact solutions for the variable-coefficient problems considered, so that such comparisons could be made.

The scope of this thesis included both the non-conservative and conservative forms of the one-dimensional advection equation, and the diffusion equation. The techniques were also extended to solve two-dimensional problems using locally one-dimensional methods. The scope of this thesis is rather broad, especially considering that the new methods can readily be incorporated into larger flow problems by using the technique of process splitting. An outline of the general approach taken for each method follows.

The convergence of the base method was established by performing a consistency analysis, yielding the truncation error of the scheme. The temporal derivatives in the truncation error were then converted to spatial derivatives using the original partial differential equation, so that the convergence of the scheme is given by the order of the leading term in the truncation error in terms of the grid spacing Δx. For each scheme the types of error that are introduced into the numerical solution were identified. The main sources of error were recognized as amplitude errors, which are quantified by the coefficients of the even spatial derivatives in the truncation error, and wave speed errors, governed by the odd spatial derivatives in the truncation error.

The modification procedure developed to improve the convergence for variable-coefficients involved incorporating the leading term of the truncation error into the finite-difference equation. This was achieved by replacing the spatial derivatives by finite-difference approximations of appropriate order. Although the modification procedure itself introduces new errors into the numerical solution, these are of a higher order than the original errors. The approximations were usually chosen so that the computational stencil of the original scheme could be retained, but this is not always possible. Schemes with wide computational stencils are more difficult to implement than compact schemes, and additional procedures to determine the numerical solution adjacent to the boundaries are required.

The stability of finite-difference methods for variable-coefficient problems can usually only be determined by performing a local stability analysis. This involves freezing the coefficients, so that the velocity and/or diffusion may be considered to be constant within the computational stencil. The local stability condition then gives a necessary condition for the stability of the variable-coefficient problem. The method for determining the local stability of the methods considered in this work was the von Neumann stability analysis. The key concepts and main results of each chapter are summarized below.

In Chapter 3, the accuracy of several well-known methods was analyzed when they are applied to the non-conservative variable-coefficient advection equation. All of the methods were shown to suffer from a reduction in convergence rate. Except for Rusanov's method, which was noted to contain a fourth-order temporal derivative in its truncation error, making it more complicated to modify than the other schemes, the methods were modified to retain their constant-coefficient order. The maximum order of convergence was achieved by modifying the implicit optimal method, which reverted to second-order for variable-coefficients, back to its constant-coefficient rate, resulting in the fourth-order mod2_O method.

All of the methods described in Chapter 3 were tested in Chapter 4. The data used in the numerical tests were chosen to be smooth because the existence of a smooth solution is the underlying assumption of high-order methods. The numerical tests illustrated the improved orders of convergence of the modified methods relative to the base methods. Moreover, the tests showed that the modified methods were almost always more accurate and efficient than the base methods.

The schemes were also compared in terms of their ability to simulate the advection of a pulse. It was seen that although the modified methods could not overcome all the difficulties associated with the introduction of spurious oscillation, they often captured the height of the pulse and the peak position

better than the base methods. It was demonstrated that unwanted negative values can be removed by using the first-order upwind formula whenever the numerical solution becomes negative. The effect of altering the size of the maximum Courant number on the accuracy and the efficiency of the schemes was investigated. It was concluded that the third-order mod2_R method and the fourth-order mod2_O method are the most accurate and efficient explicit and implicit schemes, respectively.

For the two-dimensional advection equation, a locally one-dimensional approach was taken so that the methods described in Chapter 3 could be used directly for either spatial direction. The complexities associated with applying a three-dimensional computational stencil are avoided by using locally one-dimensional techniques, which are relatively straightforward to implement, except that intermediate boundary conditions are required if an implicit scheme is used for the first spatial direction. Numerical tests established that the convergence of the locally one-dimensional method matches that of the component equations, and verified the superior accuracy and efficiency of the modified methods, compared to the base methods.

It was shown in Chapter 6 that the exact solution of the non-conservative and conservative forms of the advection equation to the same test problem are quite different. In the non-conservative case, the concentration front travelled slower, the peak was unattenuated, and the area under the solution profile increased. Several approaches to incorporate the decay (or growth) term into the solution were examined. It was demonstrated that a discretization procedure could be used to improve the convergence of methods, when the decay term is treated like a sink term. This proved to be too complicated for most schemes, so more attention was focussed on how the processes of advection and decay could be approximated separately.

A Strang-type splitting algorithm was utilized because errors are introduced if the splitting process is not symmetric. Such errors are avoided by reversing the order of the processes each time step. The advection component was approximated using the methods described in Chapter 3, while the decay was handled using an ODE solver such as the fourth-order Runge-Kutta method or Heun's second-order method, depending on the required accuracy. Although the splitting algorithm was generally successful, a decline in the convergence rate was observed for some of the high-order schemes when very fine grids were used. Nevertheless, higher convergence rates could be attained than by employing conventional methods based on the integral formulation of the conservative equation, or by applying the discretization procedure to the complete unsplit conservative equation. Moreover, the modified methods gave significant gains in accuracy and efficiency compared to the base schemes.

It has been recognized that constant-coefficient models are inadequate to describe, for example, diffusion through porous media. Field experiments have demonstrated that the diffusion is not constant but may depend on the travel time or solute displacement distance. However, in the past, most research has gone into improving discretization methods for the constant-coefficient diffusion equation. In Chapter 7, the accuracy of several methods was analyzed when they are used for variable-coefficient problems. It was shown how the Noye-Hayman and N131 methods, which are fourth-order for constant diffusion, can be modified to retain this convergence rate.

Analytical solutions were developed for three functional forms of the diffusion coefficient, and numerical tests established the improved convergence, accuracy and efficiency of the modified methods. The results of the methods for the three diffusion profiles considered were rather similar; an outcome also observed by other researchers. The schemes were also compared for various values of the maximum diffusion number. It was concluded that Mitchell's fourth-order implicit method could produce the most accurate and efficient results. Because of its unconditional stability and solvability, Mitchell's method has a considerable advantage over the other schemes. Except for the three-level methods, which cannot be applied in a locally one-dimensional fashion, the methods were then successfully applied in Chapter 8, to the two-dimensional diffusion equation.

If methods for the diffusion equation are used directly for variable-coefficient diffusion problems, the intrinsic advective component is not incorporated. This problem was addressed in Chapter 9, where the most general form of the one-dimensional diffusion equation was studied. Two approaches were taken to incorporate the advective component. The first approach was to recognize that by expanding the diffusion equation, a special case of the variable-coefficient transport equation is obtained. Consequently, by modifying methods for the constant-coefficient transport equation, two new fourth-order methods for the one-dimensional diffusion equation were obtained. Because the analysis was kept as general as possible, these schemes can also be applied to the general variable-coefficient transport equation. The second approach was to use some of the methods developed in Chapters 3 and 7 to approximate the advection and diffusion components separately. A Strang-type splitting algorithm was used, so that the convergence matches that of the component equations. Both approaches were successful, yielding far superior results to those given by the conventional schemes.

To summarize, in this thesis, a technique was developed to give high-order finite-difference methods for variable-coefficient problems. This technique has been successfully applied to variable-coefficient advection and diffusion problems. It has been demonstrated that the new schemes may readily be incorporated into multi-dimensional problems by using locally one-dimensional techniques, or that they may be used in process splitting algorithms to solve complicated time-dependent partial differential equations.

Bibliography

[1] D.A. Barry and G. Sposito, Analytical Solution of a Convection-Dispersion Model with Time-Dependent Transport Coefficients, *Water Resources Research* 25(12), 2407-2416 (1989).

[2] H.A. Basha and F.S. El-Habel, Analytical Solution of the One-Dimensional Time-Dependent Transport Equation, *Water Resources Research* 29(9), 3209-3214 (1993).

[3] J.P. Boris and D.L. Book, Flux-Corrected Transport. 1. SHASTA, A Fluid Transport Algorithm that Works, *Journal of Computational Physics* 11, 38-69 (1973).

[4] J.P. Boris and D.L. Book, Flux-Corrected Transport. 11. Generalizations of the Method, *Journal of Computational Physics* 18, 248-283 (1975).

[5] J.P. Boris and D.L. Book, Flux-Corrected Transport. 111. Minimal-Error FCT Algorithms, *Journal of Computational Physics* 20, 397-431 (1976).

[6] J.R. Cash, Two New Finite Difference Methods for Parabolic Equations, *SIAM Journal on Numerical Analysis* 21(3) 433-446 (1984).

[7] J.C. Corey, R.H. Hawkins, R.F. Overman and R.E. Green, Miscible Displacement Measurements within Laboratory Columns using the Gamma Photoneutron Method, *Soil Science Society of America* 34, 854-858 (1970).

[8] S.H. Crandall, An Optimum Implicit Recurrence Formula for the Heat Conduction Equation, *Quarterly of Applied Mathematics* 13(3), 318-320 (1955).

[9] J. Crank, The Mathematics of Diffusion, Oxford University Press (1956).

[10] J. Crank and P. Nicolson, A Practical Method for Numerical Evaluation of Solutions of Partial Differential Equations of the Heat Conduction Type, *Proceedings of the Cambridge Philosophical Society* 43(50), 50-67 (1947).

[11] A.E. Croucher and M.H. O'Sullivan, Numerical Methods for Contaminant Transport in Rivers and Estuaries, *Computers & Fluids* 27(8) 861-878 (1998).

[12] W.P. Crowley, Numerical Advection Experiments, *Monthly Weather Review* 96, 1-11 (1968).

[13] Y.G. D'Yakonov, Difference Schemes with Split Operators for Multidimensional Unsteady Problems, *U.S.S.R. Computational Mathematics* 4(2), 92-110 (1963).

[14] J.H. Ferziger and M. Perić, Computational Methods for Fluid Dynamics, Springer Verlag, New York (1996).

[15] S.K. Godunov, Finite Difference Methods for Numerical Computation of Discontinuous Solutions of the Equations of Fluid Dynamics, *Mathematicheskii Sbornik* 47, 271-306 (1959).

[16] A.R. Gourlay and A.R. Mitchell, A Classification of Split Difference Methods for Hyperbolic Equations in Several Space Dimensions, *SIAM Journal on Numerical Analysis* 6(1), 62-71 (1972).

[17] R.W. Healy and T.F. Russell, A Finite-Volume Eulerian-Lagrangian Localized Adjoint Method for Solution of the Advection-Dispersion Equation, *Water Resources Research* 29(7), 2399-2413 (1993).

[18] C. Hirsch, Numerical Computation of Internal and External Flows, Volume 1 Fundamentals of Numerical Discretization, Wiley series in numerical methods in engineering (1990).

[19] W.L. Hogarth, B.J. Noye, J. Stagnitti, J. Parlange and G. Bolt, A comparative study of Finite Difference Methods for Solving the One-Dimensional Transport Equation with an Initial-Boundary Value Discontinuity, *Computers and Mathematics with Applications* 20(11), 67-82 (1990).

[20] P. Holmgren, An Advection Algorithm and an Atmospheric Airflow Application, *Journal of Computational Physics* 115, 27-42 (1994).

[21] G.M. Hornberger, A.L. Mills and J.S. Herman, Bacterial Transport in Porous Media: Evaluation of a Model Using Laboratory Observations, *Water Resources Research* 28(3), 915-938 (1992).

[22] M.E. Hubbard and M.J. Baines, Conservative Multidimensional Upwinding for the Steady Two-Dimensional Shallow Water Equations, *Journal of Computational Physics* 138, 419-448 (1997).

[23] A. Kay, Advection-Diffusion in Reversing and Oscillating Flows: 2. Flows with Multiple Reversals, *IMA Journal of Applied Mathematics* 58, 185-210 (1997).

[24] L.A. Khan and P.L. Liu, Intermediate Dirichlet Boundary Conditions for Operator Splitting Algorithms for the Advection-Diffusion Equation, *Computers & Fluids* 24(4), 447-458 (1995).

[25] P.D. Lax and R.D. Richtmyer, Survey of the Stability of Linear Finite Difference Equations, *Communications on Pure and Applied Mathematics* 9, 267-293 (1956).

[26] P.D. Lax and B. Wendroff, Difference Schemes for Hyperbolic Equations with High Order of Accuracy, *Communications on Pure and Applied Mathematics* 17, 381-398 (1964).

[27] C.E. Leith, Numerical Simulation of the Earth's Atmosphere, Report UCRL 7986-T (1964).

[28] B.P. Leonard, A Stable and Accurate Convective Modelling Procedure Based on Quadratic Upstream Interpolation, *Computer Methods in Applied and Mechanical Engineering* 19, 59-98 (1979).

[29] B.P. Leonard, A Survey of Finite Differerences with Upwinding for Numerical Modelling of the Incompressible Convective Diffusion Equation, *Computational Techniques in Transient and Turbulent Flows Volume 2 in Series Recent Advances in Numerical Methods in Fluids* 1-35 (1981).

[30] B.P. Leonard, The ULTIMATE Conservative Difference Scheme Applied to Unsteady One-Dimensional Advection, *Computer Methods in Applied Mechanics and Engineering* 88, 17-74 (1991).

[31] B.P. Leonard and H.S. Niknafs, The Ultimate CFD Scheme with Adaptive Discriminator for High Resolution and Narrow Extrema, *Computational Techniques and Applications: CTAC-89*, editors W.L. Hogarth and B.J. Noye, 303-310 (1990).

[32] R.J. Leveque, Wave Propagation Algorithms for Multidimensional Hyperbolic Systems, *Journal of Computational Physics* 131, 327-353 (1997).

[33] R.J. Leveque and J. Oliger, Numerical Methods Based on Additive Splittings for Hyperbolic Partial Differential Equations, *Mathematics of Computation* 40(162), 469-497 (1983).

[34] E. Livne and A. Glasner, A Finite Difference Scheme for the Heat Conduction Equation, *Journal of Computational Physics* 58, 59-66 (1986).

[35] G.I. Marchuk, Methods of Numerical Mathematics, Springer-Verlag (1975).

[36] R. May and J. Noye, The Numerical Solution of Ordinary Differential Equations: Initial Value Problems, *Computational Techniques for Differential Equations* 83, edited by J. Noye, 1-94 (1984).

[37] A.R. Mitchell, Computational Methods in Partial Differential Equations, John Wiley & Sons (1969).

[38] A.R. Mitchell and D.F. Griffiths, The Finite Difference Method in Partial Differential Equations, John Wiley & Sons (1980).

[39] J.L. Morris, On the Numerical Solution of a Heat Equation Associated with a Thermal Print-Head, *Journal of Computational Physics* 5, 208-228 (1970).

[40] J.L. Morris and A.R. Gourlay, Modified Locally One Dimensional Methods for Parabolic Partial Differential Equations in Two Space Dimensions, *Journal of the Institute of Mathematics and Its Applications* 12, 349-353 (1973).

[41] R. Morrow, Numerical Solution of Hyperbolic Equations for Electron Drift in Strongly Non-Uniform Electric Fields, *Journal of Computational Physics* 43, 1-15 (1981).

[42] J. Noye, Finite Difference Techniques for Partial Differential Equations, *Computational Techniques for Differential Equations* 83, edited by J. Noye, 95-354 (1984a).

[43] J. Noye, Analysis of Explicit Finite Difference Methods used in Computational Fluid Mechanics, *Contributions of Mathematical Analysis to the Numerical Solution of Partial Differential Equations*, edited by A. Miller, Proceedings of the Centre for Mathematical Analysis, Australian National University 7, 106-118 (1984b).

[44] B.J. Noye, Three-Point Two-Level Finite-Difference Methods for the One-Dimensional Advection Equation, *Computational Techniques and Applications: CTAC-85*, editors B.J. Noye and R.L. May, 159-192 (1986).

[45] B.J. Noye, Numerical Methods for Solving the Transport Equation, *Numerical Modelling: Applications to Marine Systems*, edited by B.J. Noye, 145, 195-229 (1987a).

[46] B.J. Noye, Finite Difference Methods for Solving the One-Dimensional Transport Equation, *Numerical Modelling: Applications to Marine Systems*, edited by B.J. Noye, 145, 231-256 (1987b).

[47] B.J. Noye, Time-Splitting the One-Dimensional Transport Equation, *Numerical Modelling: Applications to Marine Systems*, edited by B.J. Noye, 145, 271-295 (1987c).

[48] B.J. Noye, Some Three-Level Finite Difference Methods for Simulating Advection in Fluids, *Computers & Fluids* 19(1), 119-140 (1991).

[49] J. Noye, Honours Lecture Notes (unpublished), The University of Adelaide (1998).

[50] B.J. Noye, P.J. Bills, K. I'Anson and C.C. Wong, Prediction of Prawn Larvae Movement in a Coastal Sea, *Computational Techniques and Applications: CTAC-91*, editors B.J. Noye, B. Benjamin and L. Coylan, 385-382 (1992).

[51] J. Noye and K. Hayman, Accurate Finite Difference Methods for Solving the Advection-Diffusion Equation, *Computational Techniques and Applications: CTAC-85*, editors B.J. Noye and R.L. May, 137-146 (1986a).

[52] J. Noye and K. Hayman, An Accurate Five-Point Explicit Finite Difference Method for Solving the One-Dimensional Linear Diffusion Equation, *Computational Techniques and Applications: CTAC-85*, editors B.J. Noye and R.L. May, 205-216 (1986b).

[53] J. Noye and K. Hayman, New LOD and ADI Methods for the Two-Dimensional Diffusion Equation, *International Journal of Computer Mathematics* 51, 215-228 (1994).

[54] B.J. Noye and H.H. Tan, A Third-Order Semi-Implicit Finite Difference Method for Solving the One-Dimensional Convection-Diffusion Equation, *International Journal for Numerical Methods in Engineering* 26, 1616-1629 (1988).

[55] B.J. Noye and A. von Trojan, An Explicit Three-Level Fifth-Order Finite Difference Method for Advection, *Computational Techniques and Applications: CTAC-95*, editors R.L. May and A.K. Easton, 603-610 (1996).

[56] J. Noye and A. von Trojan, An Explicit Finite-Difference Method for Variable Velocity Advection, *Computational Techniques and Applications: CTAC-97*, editors B.J. Noye, M. Teubner and A. Gill, 513-520 (1998).

[57] J.F. Pickens and G.E. Grisak, Scale-Dependent Dispersion in a Stratified Granular Aquifer, *Water Resources Research* 17(4), 1191-1211 (1981a).

[58] J.F. Pickens and G.E. Grisak, Modeling of Scale-Dependent Dispersion in Hydrogeologic Systems, *Water Resources Research* 17(6), 1701-1711 (1981b).

[59] I. Porro, P.J. Wierenga and R.G. Hills, Solute Transport Through Large Uniform and Layered Soil Columns, *Water Resources Research* 29(4), 1321-1330 (1993).

[60] L. Portela, L. Cancino and R. Neyes, Modelling of Tidal Flow and Transport Processes: A Case Study in the Tejo Estuary, *Computer Modelling of Seas and Coastal Regions*, edited by B.J. Noye, 449-461 (1992).

[61] R.D. Richtmyer and K.W. Morton, Difference Methods for Initial-Value Problems Second Edition, Interscience Publishers (1967).

[62] K.V. Roberts and N.O. Weiss, Convective Difference Schemes, *Mathematics of Computation* 20(94), 272-299 (1966).

[63] V.V. Rusanov, On Difference Schemes of Third Order Accuracy for Nonlinear Hyperbolic Systems, *Journal of Computational Physics* 5, 507-516 (1970).

[64] J.M. Sanz-Serna, J.G. Verwer and W.H. Hundsdorfer, Corvergence and Order Reduction of Runge-Kutta Schemes Applied to Evolutionary Problems in Partial Differential Equations, *Numerische Mathematik* 150, 405-418 (1987).

[65] J.P. Sauty, An Analysis of Hydrodispersive Transfer in Aquifers, *Water Resources Research* 16, 145-158 (1980).

[66] R. Shapiro, Smoothing, Filtering and Boundary Effects, *Review of Geophysics and Space Physics* 8(2), 359-387 (1970).

[67] M.A.R. Sharif and A.A. Busnaina, Assessment of Finite Difference Approximations for the Advection Terms in the Simulation of Practical Flow Problems, *Journal of Computational Physics* 74, 143-176 (1988).

[68] P.K. Smolarkiewicz, A Simple Positive Definite Advection Scheme with Small Implicit Diffusion, *Monthly Weather Review* 111, 479-486 (1983).

[69] P.K. Smolarkiewicz, On the Accuracy of the Crowley Advection Scheme, *Monthly Weather Review* 113, 1425-1429 (1985).

[70] A. Staniforth and J. Côté, Semi-Lagrangian Integration Schemes for Atmospheric Models – A Review, *Monthly Weather Review* 119, 2206-2223 (1991).

[71] P. Steinle, Finite Difference Methods for the Advection Equation, PHD Thesis, The University of Adelaide (1994).

[72] P. Steinle and R. Morrow, An Implicit Flux Corrected Transport Algorithm, *Journal of Computational Physics* 80, 61-71 (1989).

[73] P. Steinle, R. Morrow and A. Roberts, Use of Implicit and Explicit Flux-Corrected Transport Algorithms in Gas Discharge Problems Involving Non-Uniform Velocity Fields, *Journal of Computational Physics* 85, 493-499 (1989).

[74] J.M. Stone and F. Mihalas, Upwind Monotonic Interpolation Methods for the Solution of the Time-Dependent Radiative Transfer Equation, *Journal of Computational Physics* 100, 402-408 (1992).

[75] G. Strang, On the Construction and Comparison of Difference Schemes, *SIAM Journal on Numerical Analysis* 5, 506-517 (1968).

[76] J.C. Strikwerda, Finite Difference Schemes and Partial Differential Equations, Wadsworth & Brooks/Cole Mathematics Series (1989).

[77] H. Takewaki and T. Yabe, The Cubic-Interpolated Pseudo-Particle (CIP) Method: Application to NonLinear and Multi-Dimensional Hyperbolic Equations, *Journal of Computational Physics* 70, 355-372 (1987).

[78] G. I. Taylor, Dispersion of Soluble Matter in Solvent Flowing Through a Tube, *Proceedings of the Royal Society of London Series A* 219, 186-204 (1953).

[79] S.R. Yates, An Analytical Solution for One-Dimensional Transport in Porous Media With an Exponential Dispersion Function, *Water Resources Research* 28(8), 2149-2154 (1992).

[80] G.T. Yeh, A Lagrangian-Eulerian Method with Zoomable Hidden Fine-Mesh Approach to Solving Advection-Dispersion Equations, *Water Resources Research* 26(6), 1133-1144 (1990).

[81] G.T. Yeh, J.R. Chang and T.E. Short, An Exact Peak Capturing and Oscillation-Free Scheme to Solve Advection-Dispersion Transport Equations, *Water Resources Research* 28(11), 2937-2951 (1992).

[82] B. van Leer, Towards the Ultimate Conservative Difference Scheme: II. Monotonicity Combined in a Second-order Scheme, *Journal of Computational Physics* 14, 361-370 (1974).

[83] B. van Leer, Towards the Ultimate Conservative Difference Scheme: IV. A New Approach to Numerical Convection, *Journal of Computational Physics* 23, 276-299 (1977).

[84] V.B. Vreugdenhil and B. Koren (Editors), Numerical Methods for Advection-Diffusion Problems, Notes on Numerical Fluid Mechanics 45 (1993).

[85] R. Zhang, K. Huang and M.T. van Genuchten, An Efficient Eulerian-Lagrangian Method for Solving Solute Transport Problems in Steady and Transient Flow Fields, *Water Resources Research* 29(12), 4131-4138 (1993).

[86] C. Zoppou and J.H. Knight, Analytical Solutions for Advection and Advection-Diffusion Equations with Spatially Variable Coefficients, *Journal of Hydraulic Engineering* 123(2), 144-148 (1997a).

[87] C. Zoppou and J.H. Knight, Analytical Solution of the Spatially Variable Coefficient Advective-Diffusion Equation in One-, Two- and Three-Dimensions, Mathematics Research Report No. MRR 056-97 (1997b).

Appendix A: Difference Forms

$$\frac{\partial \hat{\tau}}{\partial x} = \frac{\hat{\tau}^n_{j+1} - \hat{\tau}^n_j}{\Delta x} - \frac{\Delta x}{2}\frac{\partial^2 \hat{\tau}}{\partial x^2} + O\{2\}, \tag{A.1}$$

$$\frac{\partial \hat{\tau}}{\partial x} = \frac{\hat{\tau}^n_j - \hat{\tau}^n_{j-1}}{\Delta x} + \frac{\Delta x}{2}\frac{\partial^2 \hat{\tau}}{\partial x^2} + O\{2\}, \tag{A.2}$$

$$\frac{\partial \hat{\tau}}{\partial x} = \frac{\hat{\tau}^n_{j+1} - \hat{\tau}^n_{j-1}}{2\Delta x} - \frac{(\Delta x)^2}{6}\frac{\partial^3 \hat{\tau}}{\partial x^3} + O\{4\}, \tag{A.3}$$

$$\frac{\partial \hat{\tau}}{\partial x} = \frac{3\hat{\tau}^n_j - 4\hat{\tau}^n_{j-1} + \hat{\tau}^n_{j-2}}{2\Delta x} + \frac{(\Delta x)^2}{3}\frac{\partial^3 \hat{\tau}}{\partial x^3} + O\{3\}, \tag{A.4}$$

$$\frac{\partial \hat{\tau}}{\partial x} = \frac{-\hat{\tau}^n_{j+2} + 4\hat{\tau}^n_{j+1} - 3\hat{\tau}^n_j}{2\Delta x} + \frac{(\Delta x)^2}{3}\frac{\partial^3 \hat{\tau}}{\partial x^3} + O\{3\}, \tag{A.5}$$

$$\frac{\partial^2 \hat{\tau}}{\partial x^2} = \frac{\hat{\tau}^n_{j+1} - 2\hat{\tau}^n_j + \hat{\tau}^n_{j-1}}{(\Delta x)^2} - \frac{(\Delta x)^2}{12}\frac{\partial^4 \hat{\tau}}{\partial x^4} + O\{4\}, \tag{A.6}$$

$$\frac{\partial^3 \hat{\tau}}{\partial x^3} = \frac{\hat{\tau}^n_{j+2} - 2\hat{\tau}^n_{j+1} + 2\hat{\tau}^n_{j-1} - \hat{\tau}^n_{j-2}}{2(\Delta x)^3} + O\{2\}, \tag{A.7}$$

$$\frac{\partial^3 \hat{\tau}}{\partial x^3} = \frac{-\hat{\tau}^n_{j+3} + 6\hat{\tau}^n_{j+2} - 12\hat{\tau}^n_{j+1} + 10\hat{\tau}^n_j - 3\hat{\tau}^n_{j-1}}{2(\Delta x)^3} + O\{2\}, \tag{A.8}$$

$$\frac{\partial^4 \hat{\tau}}{\partial x^4} = \frac{\hat{\tau}^n_{j+2} - 4\hat{\tau}^n_{j+1} + 6\hat{\tau}^n_j - 4\hat{\tau}^n_{j-1} + \hat{\tau}^n_{j-2}}{(\Delta x)^4} + O\{2\}. \tag{A.9}$$

$$\frac{\partial^4 \hat{\tau}}{\partial x^4} = \frac{-\hat{\tau}^n_{j+4} + 6\hat{\tau}^n_{j+3} - 14\hat{\tau}^n_{j+2} + 16\hat{\tau}^n_{j+1} - 9\hat{\tau}^n_j + 2\hat{\tau}^n_{j-1}}{(\Delta x)^4} + O\{2\}. \tag{A.10}$$

Table A.1: A selection of first and second-order difference forms.

Note that (A.1) and (A.5) are first and second-order forward space forms respectively, (A.2) and (A.4) are first and second-order backward space forms respectively, while (A.3), (A.6), (A.7) and (A.9) are second-order centered space forms.

Appendix B: Gauss Results

J	LTH	UW15	RUS	OPT
50	2.43e−02	4.16e−02	1.63e−02	4.82e−16
100	8.31e−03	1.22e−02	7.12e−03	6.00e−16
200	3.62e−03	4.01e−03	3.50e−03	7.59e−16
500	1.40e−03	1.42e−03	1.39e−03	1.20e−15
1000	6.96e−04	6.98e−04	6.96e−04	1.86e−15
2000	3.48e−04	3.48e−04	3.47e−04	2.98e−15
5000	1.89e−04	1.39e−04	1.39e−04	4.62e−15
J	mod_L	mod_U	mod_R	mod_O
50	3.50e−02	5.05e−02	1.40e−02	5.17e−04
100	9.33e−03	1.40e−02	1.20e−03	6.18e−05
200	2.14e−03	2.91e−03	8.74e−05	7.63e−06
500	3.11e−04	3.62e−04	3.47e−06	4.87e−07
1000	7.50e−05	8.12e−05	3.63e−07	6.08e−08
2000	1.84e−05	1.91e−05	4.16e−08	7.60e−09
5000	2.90e−06	2.95e−06	2.53e−09	4.87e−10
J	mod2_L	mod2_U	mod2_R	mod2_O
50	1.79e−02	3.78e−02	5.71e−03	2.47e−16
100	3.23e−03	8.61e−03	2.91e−04	4.75e−16
200	4.42e−04	1.29e−03	2.81e−05	6.16e−16
500	2.88e−05	8.57e−05	1.96e−06	1.16e−15
1000	3.61e−06	1.08e−05	2.50e−07	1.41e−15
2000	4.52e−07	1.34e−06	3.14e−08	2.08e−15
5000	2.89e−08	8.61e−08	2.01e−09	3.34e−15

Table B.1: RMS errors after one time cycle when the Gauss test is applied to the one-dimensional non-conservative advection problem.

J	LTH	UW15	RUS	OPT
50	2.44e−02	4.02e−02	1.77e−02	8.46e−09
100	8.81e−03	1.19e−02	8.02e−03	4.01e−11
200	4.02e−03	4.26e−03	3.96e−03	1.25e−12
500	1.58e−03	1.59e−03	1.58e−03	1.30e−14
1000	7.87e−04	7.88e−04	7.87e−04	2.06e−15
2000	3.93e−04	3.93e−04	3.93e−04	2.83e−15
5000	1.57e−04	1.57e−04	1.57e−04	4.80e−15
J	mod_L	mod_U	mod_R	mod_O
50	3.48e−02	4.97e−02	1.45e−02	5.17e−04
100	9.26e−03	1.38e−02	1.15e−03	6.19e−05
200	2.12e−03	2.87e−03	1.06e−04	7.64e−06
500	3.09e−04	3.58e−04	4.70e−06	4.88e−07
1000	7.45e−05	8.05e−05	5.13e−07	6.09e−08
2000	1.83e−05	1.90e−05	5.98e−08	7.62e−09
5000	2.89e−06	2.93e−06	3.67e−09	4.87e−10
J	mod2_L	mod2_U	mod2_R	mod2_O
50	1.78e−02	3.71e−02	5.51e−03	3.67e−09
100	3.20e−03	8.42e−03	2.78e−04	4.01e−11
200	4.37e−04	1.26e−03	3.08e−05	1.25e−12
500	2.85e−05	8.37e−05	2.21e−06	1.28e−14
1000	3.57e−06	1.05e−05	2.82e−07	1.72e−15
2000	4.47e−07	1.31e−06	3.54e−08	2.02e−15
5000	2.86e−08	8.41e−08	2.27e−09	3.40e−15

Table B.2: RMS errors after one time cycle when the Gauss test is applied to the one-dimensional conservative advection problem.

Appendix C: Non-Negativity

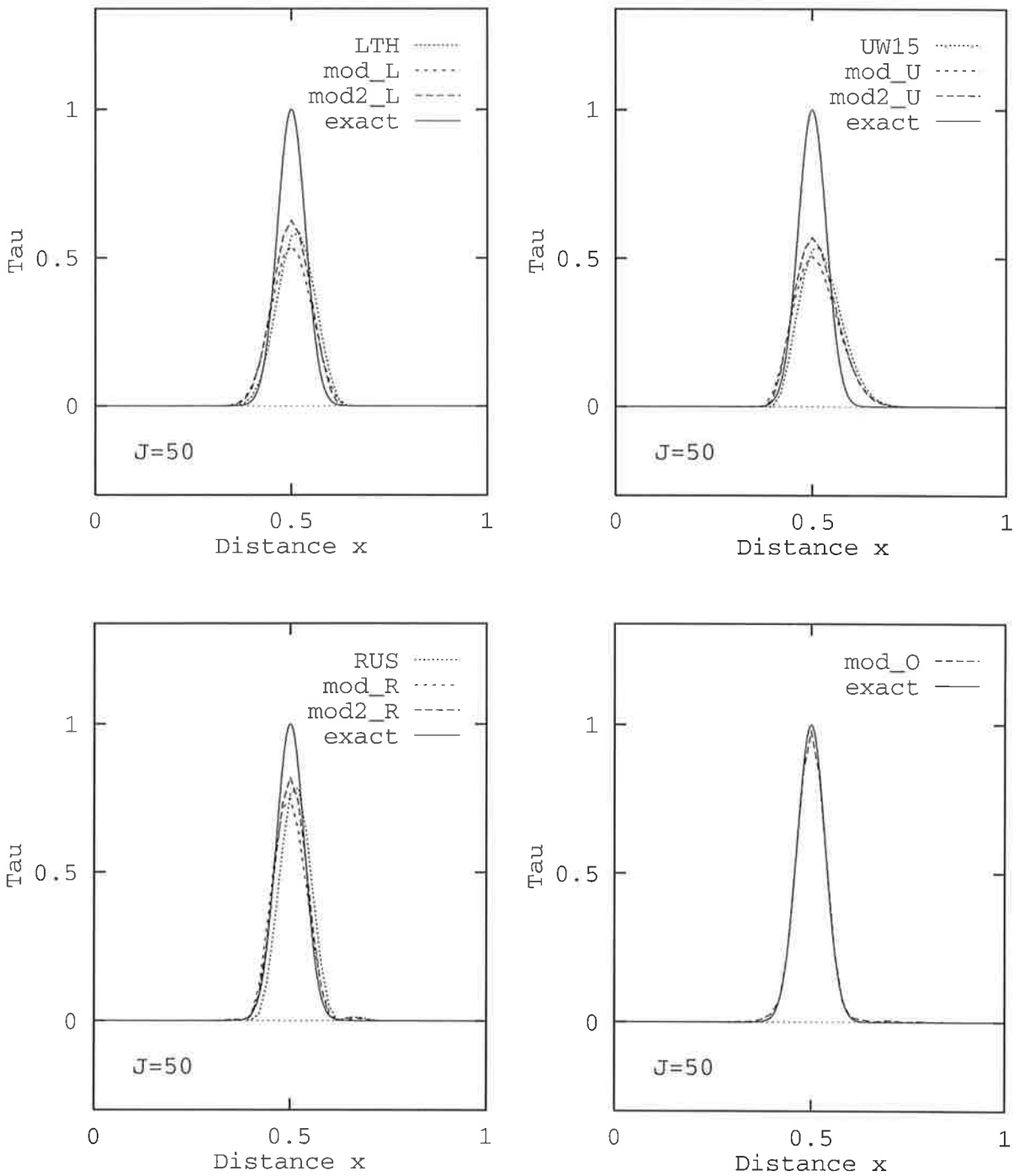

Figure C.1: The upwind method is used to eliminate negative values when the Gauss test is applied to the two-dimensional non-conservative advection equation. The results shown are for $J = 50$.