ELSEVIER

Applied Numerical Mathematics 41 (2002) 325–344

APPLIED NUMERICAL MATHEMATICS

www.elsevier.com/locate/apnum

On the influence of numerical boundary conditions

Ercília Sousa [*], Ian Sobey

Oxford University Computing Laboratory, Wolfson Building, Parks Road, Oxford OX1 3QD, UK

Abstract

Our understanding about the behaviour of numerical solutions for evolutionary convection–diffusion equations is mainly based on analysis of infinite domains situations with stability given by von Neumann analysis. Almost all practical problems involve physical domains with boundaries. For evolution problems with Dirichlet boundary conditions, some algorithms can be used without alteration near a boundary. However, the application of higher order methods such as Quickest or second order upwinding introduces difficulty near an inflow boundary, since for interior points adjacent to the boundary there are insufficient upstream points for the high order scheme to be applied without alteration. For that reason such methods require a careful treatment on the inflow boundary, where additional numerical boundary conditions have to be introduced. The choice of numerical boundary conditions turns out to be crucial for stability. A test problem is described, showing the practical advantages of some numerical boundary conditions versus the others by comparison with an exact solution. © 2001 IMACS. Published by Elsevier Science B.V. All rights reserved.

Keywords: Finite differences; Quickest; Stability; Numerical boundary conditions

1. Introduction

Consider a one dimensional problem of convection with velocity V in the x-direction and diffusion with positive coefficient D:

$$\frac{\partial u}{\partial t} + V\frac{\partial u}{\partial x} = D\frac{\partial^2 u}{\partial x^2}. \tag{1}$$

Our interest is in the solution of (1) for $t > 0, x \geqslant 0$ with an initial condition

$$u(x,0) = f(x), \tag{2}$$

[*] Corresponding author. Supported by Sub-Programa Ciência e Tecnologia do 2 Quadro Comunitário de Apoio and by Coimbra University, Portugal. Current address: Universidade de Coimbra, Departamento de Matemática, Apartado 3008, 3001-454 Coimbra, Portugal.
E-mail address: ecs@mat.uc.pt (E. Sousa).

PII: S0168-9274(01)00122-2

given, and subject to the boundary conditions

$$u(x,t) \to 0,\ x \to \infty, \quad \text{and} \quad u(0,t) = g(t),\ t \geqslant 0. \tag{3}$$

If we choose a uniform space step Δx and time step Δt, there are two dimensionless quantities of importance to the properties of most numerical schemes:

$$\mu = \frac{D\Delta t}{(\Delta x)^2}, \qquad \nu = \frac{V\Delta t}{\Delta x},$$

where ν is the Courant (or CFL) number.

Our main results will concern the numerical scheme Quickest (Quadratic Upstream Interpolation for Convective Kinematics with Estimated Streaming Terms) and associated numerical boundary conditions near the inflow boundary at $x = 0$. Quickest is due to Leonard [10] who derived this scheme using control volume arguments.

In its original form Quickest used an explicit, Leith-type differencing [16] and third-order upwinding on the convective derivatives to yield a four-point upwinded scheme. In the limit $D \to 0$, Quickest is third order accurate in time. The use of third-order upwind differencing for convection greatly reduces the numerical diffusion associated with first-order upwinding (this was illustrated in Baum et al. [1]).

The motivation for our study of the convection diffusion equation is mainly related to the unsteady two dimensional Navier–Stokes equations although this aspect will not be focused on in this paper. There is now an extensive literature about Quickest and its use in flow simulation, see for instance [1,2,10]. Other references concerning the equivalent method for steady flow, called Quick, can be found in [6,11].

There are a number of other schemes which are of great practical importance, particularly for a convection equation but also to some extent for a convection diffusion equation. These include schemes using non-linear flux limiters where a major objective can be to inhibit or prevent oscillations. Such schemes, associated with the mnemonics TVD (total variation diminishing), ENO (essentially non oscillatory) and MUSCL (monotone upwind-centred scheme for conservation laws) are of great importance in compressible flow calculations and increasingly in schemes for incompressible Navier–Stokes equations, see for instance [13] or [20]. The work described in this paper is focussed on the Lax–Wendroff and Quickest schemes, which do allow non-physical oscillatory behaviour in the solution, and in particular on the consequences for stability of their implementation near a boundary. The extension of this work to non-linear schemes using flux limiters is an important objective for further study.

We analyse the one dimensional linear convection diffusion equation as a preliminary step to the study of the multidimensional case. In doing this we introduce an efficient way to deal with the numerical boundary condition and examine the stability and the accuracy of different numerical boundary conditions.

2. Finite difference schemes

Dominant convection often leads to algorithms derived by a method introduced by Lax and Wendroff [8], who considered a Taylor expansion

$$u(x,t+\Delta t) \approx u(x,t) + \Delta t \frac{\partial u}{\partial t} + \frac{\Delta t^2}{2}\frac{\partial^2 u}{\partial t^2} + \cdots \tag{4}$$

and then used Eq. (1) to replace the temporal derivatives with spatial derivatives. The advantage of the Lax–Wendroff scheme is its second order accuracy compared to the first order accuracy of a simple upwind scheme. Davis and Moore [2] have shown that Quickest can also be derived by considering the Δt^3 term in expansion (4) and making some subsequent approximations rather than by the method of Leonard.

Morton and Sobey [12] derived a generalised solution technique which gave both Lax–Wendroff and Quickest algorithms as special cases by using an exact solution of Eq. (1) applied to an approximation on a discrete mesh. They considered a problem on the whole real line which solved exactly using Fourier transforms in x to obtain the solution,

$$u(x,t) = \frac{1}{\sqrt{\pi}} \int_{-\infty}^{+\infty} f\bigl(x - Vt + 2\sqrt{Dt}\xi\bigr) \mathrm{e}^{-\xi^2}\, \mathrm{d}\xi.$$

They wrote this as an evolution over one time step

$$u(x, t_n + \Delta t) = \int_{-\infty}^{+\infty} u(\eta, t_n) G(x - \eta; \Delta t)\, \mathrm{d}\eta, \tag{5}$$

where $t_n = n\Delta t$ and the Green's function was given by

$$G(z;\tau) = \frac{1}{\sqrt{D\pi\tau}} \mathrm{e}^{-(z-V\tau)^2/4D\tau}.$$

To derive finite differences as in [12] they substituted a local polynomial approximation to $u(\eta, t_n)$ into the integral (5), and exploited the fact that the integration for a global polynomial could be carried out exactly. They supposed there were approximations $\boldsymbol{U}^n := \{U_j^n\}$ to the values $u(x_j, t_n)$ at the mesh points

$$x_j = j\Delta x, \quad j = 0, \pm 1, \pm 2, \ldots.$$

Now they associated with each point x_j a local interpolating polynomial through U_j^n and values at a number of neighbouring points, denoting each such polynomial by $p_j(x; \boldsymbol{U}^n)$, of degree R,

$$p_j\bigl(x; \boldsymbol{U}^n\bigr) = \sum_{r=0}^{R} b_{jr}(x - x_j)^r. \tag{6}$$

Then they generated finite difference schemes from

$$U_j^{n+1} = \int_{-\infty}^{+\infty} p_j\bigl(\eta; \boldsymbol{U}^n\bigr) G(x_j - \eta; \Delta t)\, \mathrm{d}\eta. \tag{7}$$

This method provided a family of algorithms, in principle of arbitrary order of accuracy and it followed that for the approximation (6),

$$\begin{aligned} U_j^{n+1} &= b_{j0} - b_{j1} V\Delta t + b_{j2}\bigl[V^2(\Delta t)^2 + 2D\Delta t\bigr] - b_{j3}\bigl[V^3(\Delta t)^3 + 6VD(\Delta t)^2\bigr] \\ &\quad + b_{j4}\bigl[V^4(\Delta t)^4 + 12V^2D(\Delta t)^3 + 12D^2(\Delta t)^2\bigr] + \cdots. \end{aligned}$$

Within this general framework Morton and Sobey [12] could obtain both Lax–Wendroff and Quickest schemes by interpolation on a uniform mesh. If the usual central, backward and second difference operators, are written

$$\Delta_0 U_j := \frac{1}{2}(U_{j+1} - U_{j-1}), \qquad \Delta_- U_j := U_j - U_{j-1}, \quad \text{and} \quad \delta^2 U_j := U_{j+1} - 2U_j + U_{j-1}$$

and used to evaluate the coefficients b_{jr} in terms of the nodal values $\boldsymbol{U}^n$ then the following results were obtained.

Quadratic interpolation—Lax–Wendroff

If a quadratic interpolant of U_{j-1}, U_j and U_{j+1} was used then

$$b_{j0} = U_j^n, \qquad b_{j1} = \frac{\Delta_0 U_j^n}{\Delta x}, \qquad b_{j2} = \frac{\delta^2 U_j^n}{2\Delta x^2},$$

and the approximation formula for U_j^{n+1} was the Lax–Wendroff scheme

$$U_j^{n+1} = \left[1 - \nu\Delta_0 + \left(\frac{1}{2}\nu^2 + \mu\right)\delta^2\right]U_j^n. \tag{8}$$

Cubic approximation—Quickest

If $p_j(x, \boldsymbol{U}^n)$ was extended to include a cubic term, then there would be a choice of points which can be interpolated. If the cubic expansion was obtained by interpolating U_{j-2}^n as well as U_{j-1}^n, U_j^n and U_{j+1}^n, that is by using two upstream points, then

$$b_{j0} = U_j^n, \qquad b_{j1} = \frac{\Delta_0 U_j^n}{\Delta x} - \frac{\delta^2 \Delta_- U_j^n}{6\Delta x^3}, \qquad b_{j2} = \frac{\delta^2 U_j^n}{2\Delta x^2}, \qquad b_{j3} = \frac{\delta^2 \Delta_- U_j^n}{6\Delta x^3},$$

and the approximation formula became the Quickest scheme:

$$U_j^{n+1} = \left[1 - \nu\Delta_0 + \left(\frac{1}{2}\nu^2 + \mu\right)\delta^2 + \nu\left(\frac{1}{6} - \frac{\nu^2}{6} - \mu\right)\delta^2\Delta_-\right]U_j^n. \tag{9}$$

The difficulty in applying Quickest near a boundary is immediately evident because adjacent to a boundary there will only be one upstream point whereas the scheme (9) requires two upstream points.

3. The numerical boundary condition

The model problem we consider here is a simplified form of (3) where, for the solution defined on the half-line, the inflow boundary condition is given by

$$u(0, t) = 0. \tag{10}$$

As Lax–Wendroff is a three point scheme it can be used at all interior points. On the other hand the Quickest scheme uses two points upstream and can not be applied on the first interior point of the mesh. At that point we need to apply a numerical boundary condition. In the next sections we discuss a number of different numerical boundary conditions which can be used at the first interior point of the scheme. The new results in this work concerns the consequences for stability and accuracy of the resulting schemes.

3.1. A numerical boundary condition suggested by Leonard

Leonard [10] suggested the following boundary condition based on control-volume arguments for a cell $[\Delta x/2, 3\Delta x/2]$: a hypothetical node is specified at $\Delta x/2$ downstream of the physical boundary at

$x = 0$. This node is denoted by B. It is assumed that the Dirichlet condition can be applied at this node, rather than at $x = 0$, so that in this case: $U_B = 0$. A linear interpolation, between the next boundary value and the first interior point, $U_B = (U_0^{n+1} + U_1^n)/2$ gives for this case

$$U_0^{n+1} = -U_1^n. \tag{11}$$

Then, using a control volume approach to determine fluxes across the mid-cell faces at $\Delta x/2$ and $3\Delta x/2$, it gives

$$U_1^{n+1} = U_1^n - \nu\left(U_r^n - U_l^n\right) + \mu\left(U_0^n + U_2^n - 2U_1^n\right), \tag{12}$$

where fictitious values U_r^n, U_l^n are evaluated at $3\Delta x/2$ and $\Delta x/2$, respectively. Applying the numerical boundary condition (10) at $\Delta x/2$ and an interpolation at $3\Delta x/2$ gives:

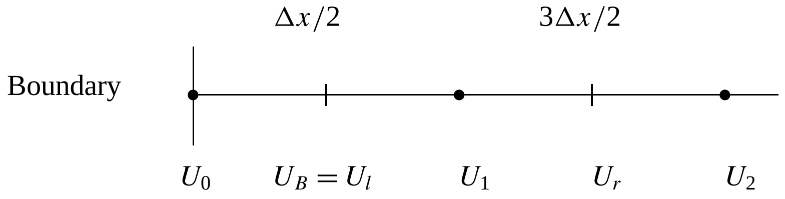
$$\begin{aligned} U_l^n &= U_B^n, \\ U_r^n &= \frac{1}{2}\left(U_1^n + U_2^n\right) - \frac{1}{8}\left(U_0^n + U_2^n - 2U_1^n\right). \end{aligned}$$

Since in this case $U_B^n = 0$, (12) can be rewritten

$$U_1^{n+1} = U_1^n - \frac{\nu}{8}\left(6U_1^n + 3U_2^n - U_0^n\right) + \mu\left(U_0^n + U_2^n - 2U_1^n\right). \tag{13}$$

This provides an algorithm for dealing with the first interior point which incorporates the numerical boundary condition.

A diagram with the relevant points is given below:



3.2. Downwind third difference

The derivation of the numerical scheme Quickest using (7) was based on a local cubic approximation. If at the first internal point of the scheme we choose the points used for interpolation as U_0^n, U_1^n, U_2^n and U_3^n we bring in a forward third difference instead of a backward third order difference, given a scheme

$$U_1^{n+1} = \left[1 - \nu\Delta_0 + \left(\frac{1}{2}\nu^2 + \mu\right)\delta^2 + \nu\left(\frac{1}{6} - \frac{\nu^2}{6} - \mu\right)\delta^2\Delta_+\right]U_1^n, \tag{14}$$

where Δ_+ is the forward operator defined by $\Delta_+ U_j := U_{j+1} - U_j$. Additionally we are considering

$$U_0^n = 0.$$

The use of this downwind third difference does not affect accuracy since it is still based on a local cubic approximation. However as we shall show, it does have penalties in terms of stability.

3.3. Lax–Wendroff

We assume the Dirichlet boundary condition

$$U_0^n = 0.$$

An alternative numerical boundary condition at the first interior point is obtained by applying a quadratic local approximation using the points U_0^n, U_1^n and U_2^n for interpolation. In that way we have the Lax–Wendroff method only at that point:

$$U_1^{n+1} = \left[1 - \nu\Delta_0 + \left(\frac{1}{2}\nu^2 + \mu\right)\delta^2\right]U_1^n. \tag{15}$$

3.4. A fictitious point U_{-1}

Let us suppose we apply a Quickest scheme to the first internal point without modification by assuming a fictitious point U_{-1}. In this section we describe one way to calculate this fictitious point using the boundary data.

We know that on the whole real line the exact solution of the convection–diffusion equation (1) subject to an initial condition is given by a version of (5),

$$u(x,t) = \int_{-\infty}^{+\infty} u(\eta,0)G(x-\eta;t)\,\mathrm{d}\eta. \tag{16}$$

In our case we only have initial data for $x \geqslant 0$ but the boundary data at $x = 0$ for $t > 0$ will correspond to (unknown) initial data for $x < 0$. In particular at $x = 0$, $g(t) = u(0,t)$ is given by

$$g(t) = \int_{-\infty}^{+\infty} u(\eta,0)G(-\eta;t)\,\mathrm{d}\eta. \tag{17}$$

If we define

$$\begin{aligned} u_+(\eta) &= u(\eta,0), \quad \eta \geqslant 0,\\ u_-(\eta) &= u(\eta,0), \quad \eta < 0, \end{aligned}$$

then we can write

$$\int_{-\infty}^{0} u_-(\eta)G(-\eta;t)\,\mathrm{d}\eta = g(t) - \int_{0}^{+\infty} u_+(\eta)G(-\eta;t)\,\mathrm{d}\eta.$$

Given u_+ and g, this defines an inverse problem for u_-. This gives one way to deal with a fictitious point on the left of $x = 0$. Rather than try to determine $u_-(\eta)$ analytically, we consider an application of (17) over one time step:

$$g(t_{n+1}) = \int_{-\infty}^{+\infty} u(\eta,t_n)G(-\eta;\Delta t)\,\mathrm{d}\eta. \tag{18}$$

Then approximating the solution $u(\eta, t_n)$ by a quadratic polynomial around $x = 0$ using U_{-1}^n, U_0^n and U_1^n gives

$$g^{n+1} = g^n - \frac{\nu}{2}\left(U_1^n - U_{-1}^n\right) + \left(\mu + \frac{\nu^2}{2}\right)\left(U_1^n - 2g^n + U_{-1}^n\right), \tag{19}$$

where $g^n := g(t_n)$. This is of course the same as (8) with $j = 0$ and $U_0^n = g^n$. Note that in particular we are assuming $g(t) = 0$ as imposed by (10).

Since g^n, g^{n+1} and U_1^n are known, from (19) we can calculate the fictitious value U_{-1}^n . After we have obtained the value, we can apply the correct third difference at the first interior point. As we shall show below this numerical boundary condition has a substantial advantage for stability.

4. Stability considerations

Stability analysis is usually only possible for fairly idealised situations, linear constant coefficient equations on infinite domains (although energy methods can be used for some situations). It is observed that most more complex situations still follow results which come from analysis of idealised model equations. On an infinite domain it is conventional to use von Neumann analysis to determine whether a discretisation scheme will be stable or unstable. In the case of a bounded domain it is no longer possible to use simple von Neumann analysis. We need to assure that the discretisation of the boundary conditions is also stable and then the overall discretisation will be stable, in the sense of Lax [15].

However, to have stability of a scheme subject to numerical boundary conditions, first of all we need to assure that the Cauchy problem is stable, that is that the scheme is von Neumann stable in the infinite domain.

Denote $\kappa(\xi)$ the Fourier amplification factor of a numerical scheme. A numerical scheme is said to be von Neumann stable if there is a constant K such that

$$\left|\kappa(\xi)\right| \leqslant 1 + K\Delta t, \quad \forall \xi \in \mathbb{R}. \tag{20}$$

However, for some problems the presence of the arbitrary constant in (20) is too generous for practical purposes, although being adequate for eventual convergence in the limit $\Delta t \to 0$. In practice, the inequality (20) is substituted by the following stronger condition.

Definition 1. A numerical scheme is said to be practically von Neumann stable if

$$\left|\kappa(\xi)\right| \leqslant 1, \quad \forall \xi \in \mathbb{R}. \tag{21}$$

In some cases condition (20) allows numerical modes to grow exponentially in time for finite values of Δt. Therefore, the practical, or strict, stability condition (21) is used in order to prevent numerical modes growing faster than physical modes solution of the differential equation.

In the next sections, when referring to a scheme as von Neumann stable, it means that the practical von Neumann condition (21) is satisfied.

All the explicit methods we discuss can be written in the form of a matrix iteration. Assume that the nodal points are U_j^n, $j = 0, \ldots, N$, and that the outflow boundary is such that

$$U_N^n = 0, \quad \forall n. \tag{22}$$

Introducing the vector $U^n = \{U_0^n, U_1^n, \ldots, U_{N-1}^n\}^{\mathrm{T}}$, all the schemes may be written as matrix equations

$$U^{n+1} = AU^n, \quad n = 0, 1, 2, \ldots, \tag{23}$$

where A is an $N \times N$ matrix and depends on the scheme used.

Leaving aside errors in the truncation of the original continuous equation, any errors E^n in a calculation based on (23) will grow according to

$$E^{n+1} = AE^n, \quad n = 0, 1, 2, \ldots, \tag{24}$$

where $E^n = u^n - U^n$ with u^n, U^n the exact and numerical solutions of (23), respectively, at $t = n\Delta t$.

Given $A \in \mathbb{R}^{N\times N}$ denote the spectral radius of A by $\rho(A)$ and the L_2-norm of the matrix A by $\|A\|$. We recall that

$$\|A\| = \rho(A) \quad \text{if } A \in \mathbb{R}^{N\times N} \text{ is normal.}$$

It is well known that for any $A \in \mathbb{R}^{N\times N}$

$$A^m \to 0 \text{ as } m \to \infty \quad \text{if and only if} \quad \rho(A) < 1.$$

A simple criterion for regulating the error growth governed by (24) is given by

$$\rho(A) \leqslant 1. \tag{25}$$

When the matrix A is not normal, the spectral radius gives no indication of the magnitude of E^n for finite n. In this case a condition of the form $\rho(A) < 1$ guarantees eventual decay of the solution, but does not control the intermediate growth of the solution.

A more severe condition for regulating error growth follows from (24). If the matrix norm, $\|A\|$, is consistent with the vector norm, $\|E\|$, then

$$\|E^{n+1}\| \leqslant \|A\| \|E^n\|, \quad n = 0, 1, 2, \ldots,$$

and the condition

$$\|A\| \leqslant 1, \tag{26}$$

is sufficient to ensure that the error cannot grow with n. This condition is very severe and replacing (26) by

$$\|A^m\| \leqslant K, \quad \forall m, \tag{27}$$

with a suitable choice of m and K, gives a more relaxed condition which allows a limited growth of the error vector after m time steps. The error is controlled by a reasonable constant for all $m \geqslant 0$, although in practice the concept of reasonable constant is not straightforward. Recently several authors [3,9,14] have carried out work related to non-normality effects and have found some sufficient conditions to bound $\|A^m\|$ for all $m \geqslant 0$.

By examining both the spectral radius and the matrix norm, we are able to find very accurate regions of stability for our methods.

It is also worth noting that the Godunov–Ryabenkii theory can be applied to these problems. Godunov and Ryabenkii [4] deduced necessary conditions occasioned by the boundary conditions. This work was further developed by Kreiss [7] and Gustafsson et al. [5]. This method is quite powerful, but often leads to very complex and intractable calculations, see also [17].

5. Practical stability regions

To have stability of a scheme subject to numerical boundary conditions, a necessary condition is that the scheme is von Neumann stable in the infinite domain. The Lax–Wendroff and Quickest schemes are von Neumann stable, provided μ, ν are such as to lie within the respective curves in Fig. 1 [10,12]. The regions plotted in Fig. 1 are sufficient and necessary for a von Neumann stability of these schemes.

This means that when the Quickest scheme is subject to numerical boundary conditions, any stability region should lie inside the stability region displayed in Fig. 1.

Our plan is to show curves which define $\rho(A) = 1$, curves which define $\|A\| = 1$ and curves which define $\|A^n\| = 1$ for some fixed n. These curves have been computed using Matlab for finite size matrices. The shaded area between two curves is where eigenvalue analysis would indicate stability but where matrix analysis tell us the error might grow by many orders of magnitude before eventually decaying. A simple guide for practical stability is to stay within the region where $\|A\| \leqslant 1$ but that can be very restrictive in some cases. A less restrictive condition is to consider the region where $\|A^n\| \leqslant 1$ for some $n \geqslant 1$ not very large. In general the size of the matrix A considered is $N = 30$ unless another size is mentioned. The outflow boundary condition considered is always the Dirichlet boundary condition (22).

Lax–Wendroff

For the Lax–Wendroff scheme and considering Dirichlet boundary conditions on the inflow and outflow, we know the stability region is given by the von Neumann condition, since we can consider periodic boundary conditions. In Fig. 2 we see that the region $\|A\| \leqslant 1$ coincides with the well known von Neumann condition: $\nu^2 + 2\mu \leqslant 1$. We can also observe that for finite matrices the spectral radius is greater than that indicated by von Neumann analysis.

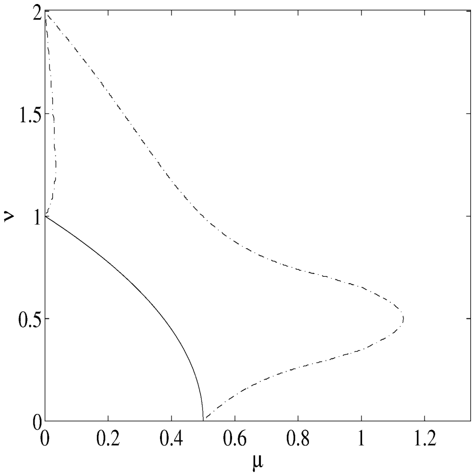


Fig. 1. Von Neumann stability regions for Lax–Wendroff (−) and Quickest (−·).

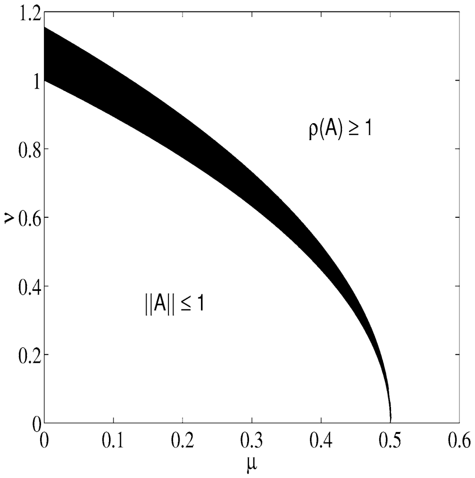
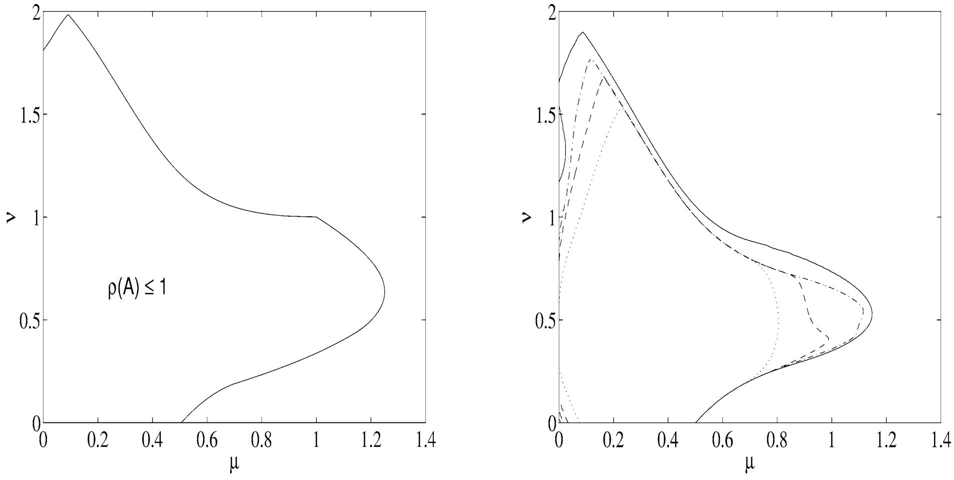


Fig. 2. Stability region for Lax–Wendroff.



(a)

(b)

Fig. 3. Stability region for Quickest combined with the numerical boundary condition suggested by Leonard (Section 3.1): (a) Region where the eigenvalues of A are less than one but the norm of A is bigger than one (in this figure we do not shade this region as in the previous and subsequent figures); (b) $\|A^3\| = 1$ $(\cdots)$, $\|A^6\| = 1$ (- -), $\|A^{12}\| = 1$ $(-\cdot)$, $\|A^{48}\| = 1$ $(-)$.

Quickest

As we have indicated, difficulty in applying Quickest is related to the right choice of the numerical boundary condition. Since the iterative matrix A is slightly different for different boundary conditions, stability results also differ. We apply inlet (10), and outlet (22), Dirichlet boundary conditions.

On applying the boundary method suggested by Leonard (Section 3.1), the region where the eigenvalues are less than one (Fig. 3(a)) almost contains all the von Neumann stability region (Fig. 1),

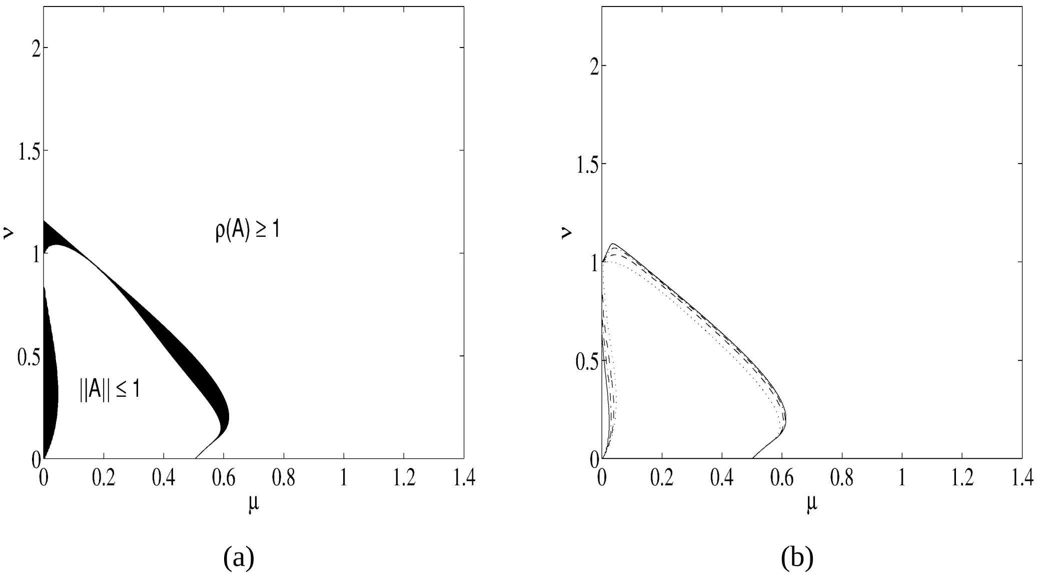


Fig. 4. Stability region for Quickest with a downwind numerical boundary condition (Section 3.2): (a) Norm and spectral radius for the iterative matrix A; (b) $\|A^3\| = 1$ (···), $\|A^6\| = 1$ (- -), $\|A^{12}\| = 1$ (—·), $\|A^{24}\| = 1$ (—).

except a small portion on the top left corner of Fig. 3(a), although the norm of the matrix A is never less than one. The fact that the norm is never less than one does not imply that the method is not stable, since $\|A\| \leqslant 1$ is only a sufficient condition for stability but not a necessary condition. It is interesting to see what happens when the norm of powers of A, $\|A^n\|$ is computed. We plot in Fig. 3(b) the regions $\|A^n\| \leqslant 1$ for $n = 3, 6, 12, 48$. The region defined by $\|A^{48}\| = 1$ is approximately the same as the von Neumann region. Of course the condition $\rho(A) \leqslant 1$ implies that $\|A^n\|$ tends to zero when $n \to \infty$, but the main point for a practical stability is that $\|A^n\|$ does not grow very strongly and starts to decay after few steps in time. The practical stability region for this case is approximately the von Neumann region given by the intersection of the condition $\rho(A) \leqslant 1$ (Fig. 3(a)) with the von Neumann condition (Fig. 1).

Using the Quickest scheme with a downwind third order difference applied to the first mesh point we lose a substantial part of the stability region (Fig. 4(a)). When plotting the regions $\|A^n\| \leqslant 1$, $n = 3, 6, 12, 24$ (Fig. 4(b)) we can observe that as we increase n the region $\|A^n\| \leqslant 1$ approximates the region defined by $\rho(A) \leqslant 1$. Note also that since we are considering the Quickest scheme with a numerical boundary condition, when $N \to \infty$ the curve $\rho(A) = 1$ does not necessarily approach the boundary of the von Neumann region as would happen, for instance, in the case of the Lax–Wendroff scheme (see Fig. 1), where numerical boundary conditions are not present.

There is a small portion for μ small (Fig. 4(b)) where $\|A^n\|$ does not become less than one for a relatively small n, although it does not grow significantly either as shown in Fig. 5. If we consider the area where $\|A\| > 1$ and $\rho(A) \leqslant 1$ for small μ (the shaded region on Fig. 4(a)) then for $\mu = 0.001$ and $\nu = 0.5$, if N (the size of the matrix A) is increased, the maximum value of $\|A^n\|$ does not increase, i.e., $\|A^n\| \leqslant 1.2$ for all n and N considered (Fig. 5(a)). In Fig. 5(b) for $\mu = 0.001$ and $\nu = 0.1$ we can observe that for the matrix size $N = 30$ the norm starts to be less than one around $n = 300$. We also have that $\|A^n\| \leqslant 1.6$ for all n and N considered. In this region more steps in time are needed before $\|A^n\|$ becomes less than one.

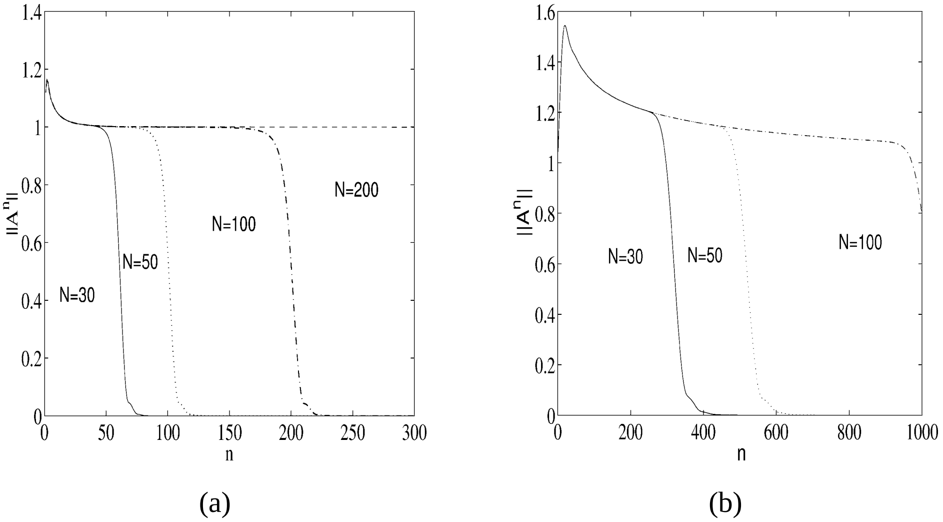


Fig. 5. Evolution of matrix norm for Quickest with a downwind numerical boundary condition (Section 3.2): (a) Behaviour of the function $\|A^n\|$ at $\mu = 0.001,\ \nu = 0.5$ for different matrices sizes (N); (b) behaviour of the function $\|A^n\|$ at $\mu = 0.001,\ \nu = 0.1$ for different matrices sizes (N).

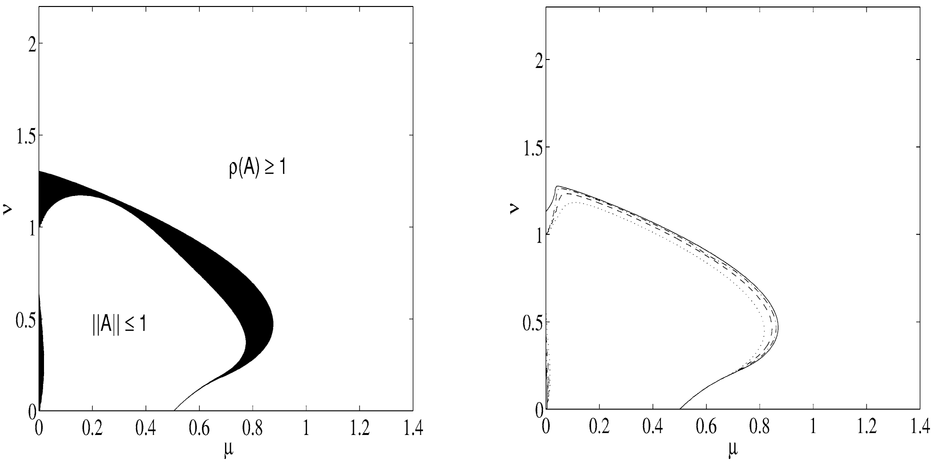
The curves in Fig. 5 show that for these values and parameters the norm of $\|A^n\|$ as n increases has a characteristic early growth, plateau values and later a rapid decrease to zero. We do not have a satisfactory explanation for this behaviour.

In cases where μ is small, Fig. 4 indicates a region of potential instability but it is in fact a stable region where the condition (27) is satisfied with values of K not much larger than one, see Fig. 5. Consequently the practical stability region is given by the condition $\rho(A) \leqslant 1$ that lies inside the von Neumann region. The fact that the stability region for the Quickest scheme with the downwind third difference numerical boundary condition is given by this region was also recently proved in [17] using Godunov–Ryabenkii theory.

The effect on stability of applying the Lax–Wendroff scheme to the first interior point of the scheme is shown in Fig. 6. The region of stability is larger than with a downwinded third difference. The shaded area in Fig. 6 that lies inside the von Neumann stability region (Fig. 1) is still a region where we have practical stability although the norm exceeds one, as we can conclude by the behaviour of $\|A^n\|$ as n increases in Fig. 6(b), where the regions $\|A^n\| \leqslant 1,\ n = 6, 12, 24, 48$ are plotted.

The numerical boundary condition which used a fictitious point value to apply an upwinded third difference, associated with inlet and outlet Dirichlet boundary conditions gives essentially the same stability region as the von Neumann condition. The region where $\|A\| \leqslant 1$ (see Fig. 7) is coincident with the region where the interior scheme Quickest is von Neumann stable. We can conclude that we have practical stability for that scheme in the region given by the condition $\|A\| \leqslant 1$.

An important point is whether the results presented are sensitive to changes in the size of the iterative matrix. Our numerical experience is that changing the matrix size does not change our general conclusions. This is illustrated by showing the effect of changing the size of the matrices on the eigenvalues and norm in Fig. 8. Although these results are for a Quickest scheme with Lax–Wendroff as a numerical boundary condition, these are also a general profile for the other methods. Fig. 8 shows



(a)

(b)

Fig. 6. Stability region for Quickest with Lax–Wendroff as the numerical boundary condition (Section 3.3): (a) Norm and spectral radius for the iterative matrix A; (b) $\|A^6\| = 1$ ($\cdots$), $\|A^{12}\| = 1$ (- -), $\|A^{24}\| = 1$ ($-\cdot$), $\|A^{48}\| = 1$ ($-$).

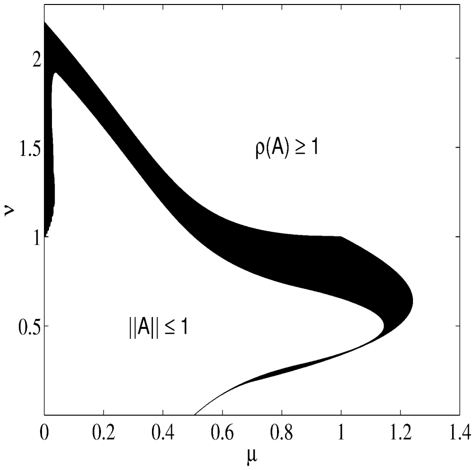


Fig. 7. Stability region for Quickest using a fictitious point (Section 3.4).

what happens to the spectral radius and the norm for two cases of interest, one where the norm and eigenvalues are simultaneously less than one (Fig. 8(a)) and the other region where the spectral radius is still less than one but the norm is not (Fig. 8(b)). We observe slight changes in the spectral radius with the dimension of the matrix but it does not become larger than one. The matrix norm seems more stable to changes of the matrix size; neither indicator is much affected by increasing the matrix size.

Although when we are dealing with non-normal matrices the eigenvalues are not reliable indicators of stability, in our examples the intersection of the region where the eigenvalues are less than one with the von Neumann stability region gives us a quite accurate practical stability region.

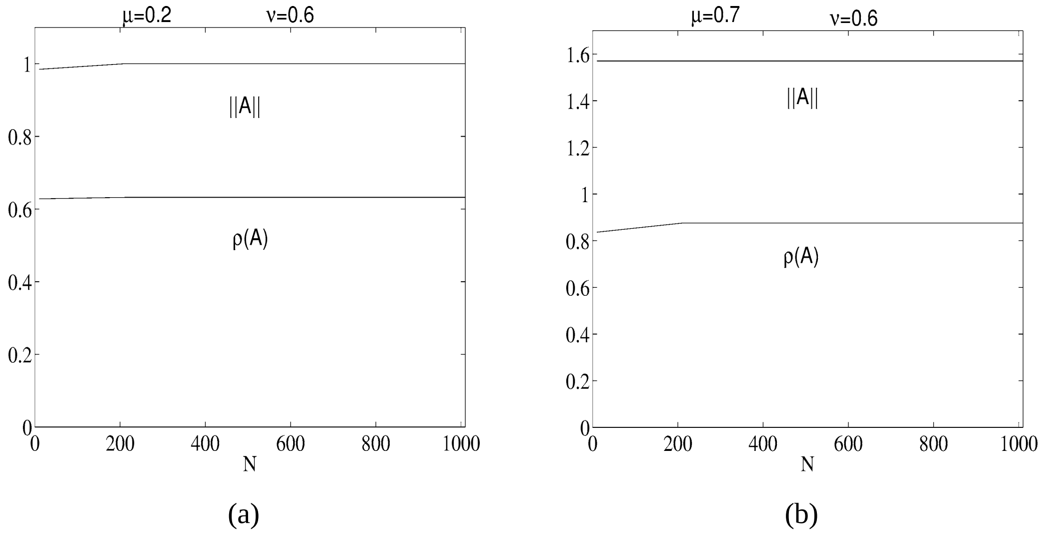


Fig. 8. Effect of the matrix size on the spectral values and matrix norm for Quickest with Lax–Wendroff as a numerical boundary condition: (a) $\mu = 0.2$, $\nu = 0.6$; (b) $\mu = 0.7$, $\nu = 0.6$.

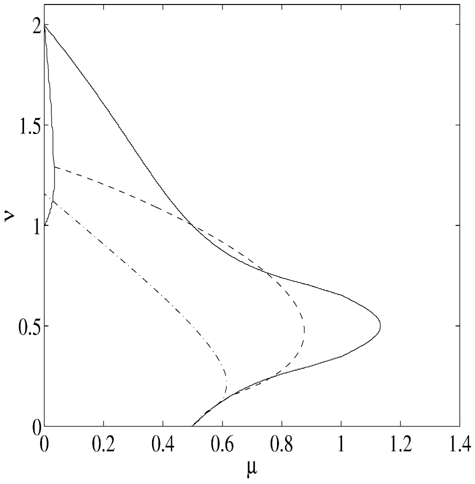


Fig. 9. Von Neumann stability region (—); region where the spectral radius is less than one for the iterative matrix A when the Lax–Wendroff numerical boundary condition is considered (— — —); region where the spectral radius is less than one for the iterative matrix A when the downwind numerical boundary condition is considered (— · —).

To give a better comparative idea of the stability regions for the different numerical boundary conditions, we show in Fig. 9 the von Neumann stability region alongside the regions where the spectral radius of the iterative matrix A is less than one when the Lax–Wendroff numerical boundary condition and the downwind numerical boundary condition are considered. For the numerical boundary condition with the fictitious point the stability region is given by the von Neumann stability region (see Fig. 7). For the Leonard numerical boundary condition the stability region is approximately the von Neumann stability region, since the region where the spectral radius of the iterative matrix A is less than one almost contains the von Neumann stability region, except for a very small portion, pointed out previously, that is located in the top left of the Fig. 3(a).

6. Accuracy and test problem

To analyse the accuracy of the methods it is usual to consider a local truncation error. The local truncation error of the Lax–Wendroff scheme and Quickest scheme can be derived using the modified equation [19] or the Peano kernel theorem [12].

On theoretical grounds, over a finite interval of time, we expect the Lax–Wendroff method to be close to $\mathrm{O}(\Delta x^2)$ accurate while Quickest methods should be $\mathrm{O}(\Delta x^3)$ accurate. These estimates are not rigorous since there will be variation of the error with μ and ν depending on how Δx and Δt are related when the mesh is refined and also depending on the different choices of numerical boundary conditions.

We compare the effect of different numerical boundary conditions using the following test problem. If we consider the convection–diffusion problem (1)–(3), then an exact solution of this system on the half line $x \geqslant 0$ can be found using Laplace Transforms:

$$
\begin{aligned}
u(x,t) = {} & \frac{1}{\sqrt{\pi}} \int_0^t g(t-\hat{\tau}) G^*(x,\hat{\tau})\,\mathrm{d}\hat{\tau} + \frac{1}{\sqrt{\pi}} \int_{\frac{Vt-x}{2\sqrt{Dt}}}^{+\infty} f\left(x - Vt + 2\sqrt{Dt}\xi\right) \mathrm{e}^{-\xi^2}\,\mathrm{d}\xi \\
& - \frac{1}{\sqrt{\pi}} \int_{\frac{Vt+x}{2\sqrt{Dt}}}^{+\infty} f\left(-x - Vt + 2\sqrt{Dt}\xi\right) \mathrm{e}^{Vx/D} \mathrm{e}^{-\xi^2}\,\mathrm{d}\xi,
\end{aligned}
$$

where the function $G^*(x,\hat{\tau})$ is given by

$$
G^*(x,\hat{\tau}) = \frac{x}{2\sqrt{D}\hat{\tau}^{3/2}} \mathrm{e}^{-(x-V\hat{\tau})^2/4D\hat{\tau}}.
$$

To measure accuracy of the different Quickest schemes we have considered a test problem with initial data

$$
u(x,0) = \mathrm{e}^{-x^2/L^2}, \quad x \geqslant 0, \qquad u(0,t) = 0,
$$

where L is an arbitrary length scale. We will eventually take $L = 1$ but we retain it for the present to keep track of dimensions in the solution. Our reason for considering this test case is that it is straightforward to calculate an exact solution for this initial profile:

$$
\begin{aligned}
u(x,t) = \frac{L}{2\sqrt{4Dt + L^2}} \Bigg[& \mathrm{e}^{\frac{(x-Vt)^2}{4Dt+L^2}} \operatorname{Erfc}\left(-\frac{(x-Vt)L}{2\sqrt{Dt(4Dt+L^2)}} \right) \\
& - \mathrm{e}^{-\frac{(x+Vt)^2}{4Dt+L^2} + \frac{Vx}{D}} \operatorname{Erfc}\left(\frac{(x+Vt)L}{2\sqrt{Dt(4Dt+L^2)}} \right) \Bigg],
\end{aligned}
$$

where $\operatorname{Erfc}(x) = \frac{2}{\sqrt{\pi}} \int_x^\infty \mathrm{e}^{-s^2}\,\mathrm{d}s$. The time evolution of the solution is shown in Fig. 10 for $L = 1$.

The following test problem results are for the section $0 \leqslant x \leqslant 20$ and for $0 \leqslant t \leqslant 20$. There seems nothing particular about these ranges which change the general nature of our conclusions.

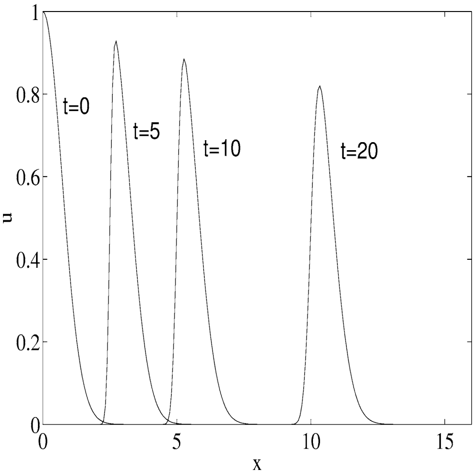


Fig. 10. Exact solution defined by (28) at the times $t = 0, 5, 10, 20$.

For the initial solution $u(x, 0) = \mathrm{e}^{-x^2}$, $V = 0.5$, $D = 0.001$ we compute the approximated solutions given respectively by the Lax–Wendroff scheme and by the Quickest scheme associated with the different numerical boundary conditions for a finite domain $0 \leqslant x \leqslant 20$. We plot the results in Fig. 11 at $t = 20$, for a discrete mesh $x_j = j\Delta x$, $j = 1, \ldots, 300$, $\Delta x = 20/300$ and $\Delta t = \Delta x^2$.

Consider the vector $u_{\mathrm{ex}} = (u(x_0, t), u(x_1, t), \ldots, u(x_N, t))$, where u is the exact solution (28) and the vector $U_{\mathrm{app}} = (U(x_0, t), U(x_1, t), \ldots, U(x_N, t))$, where U is the approximated solution given by the respective numerical scheme. The error is then given by

$$\mathrm{Error}(\Delta x) = \big\|u_{\mathrm{ex}}(\Delta x) - U_{\mathrm{app}}(\Delta x)\big\|,$$

where $\|\cdot\|$ is the L_2 norm.

In Fig. 12 we plot the error versus the mesh size for the Lax–Wendroff scheme and for the Quickest scheme associated with different numerical boundary conditions. In Table 1 we give estimates for the convergence rate, p, assuming that the error behaves like $(\Delta x)^p$. In theory, Lax–Wendroff schemes should be second order ($p = 2$) and Quickest schemes third order ($p = 3$). However the practical order of convergence is highly dependent on the refinement path.

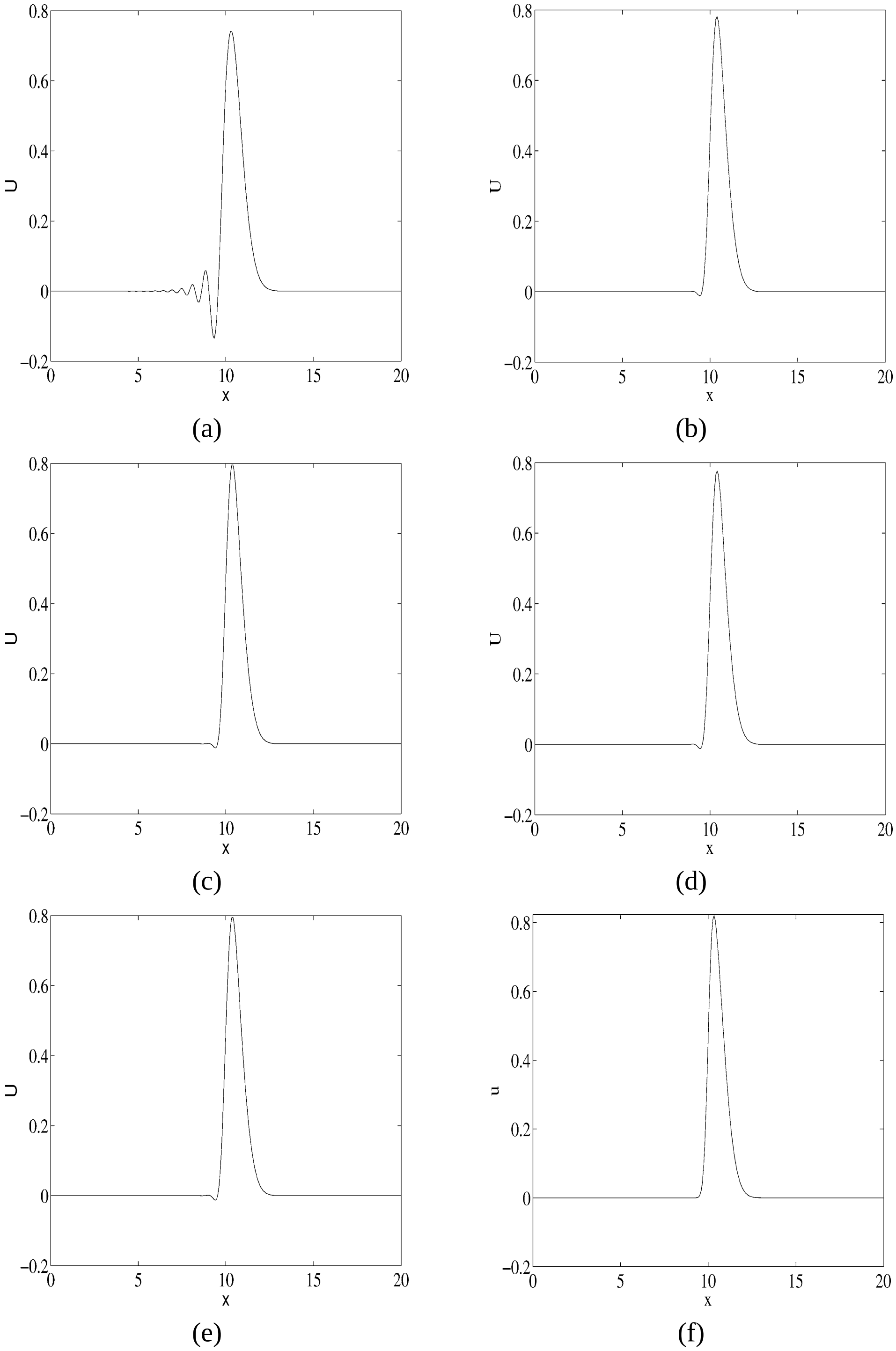
Denoting by T^n the truncation error, we have the following for the Lax–Wendroff and Quickest schemes (see [12,18]):

Lax–Wendroff

$$\begin{aligned}\Delta t T_j^n = {} & \frac{1}{6}\Delta x^3 \nu\big(1 - \nu^2 - 6\mu\big)U_{x^3}^n(x_j) \\ & + \frac{1}{24}\Delta x^4\big(12\mu^2 - 2\mu + 3\nu^2\big(1 - \nu^2 - 4\mu\big)\big)U_{x^4}^n(x_j) + \cdots.\end{aligned} \tag{28}$$

Quickest

$$\Delta t T_j^n = \frac{1}{24}\Delta x^4\big(12\mu^2 - 2\mu - 12\mu\nu(1 - \nu) + \nu\big(1 - \nu^2\big)(2 - \nu)\big)U_{x^4}^n(x_j) + \cdots. \tag{29}$$



(a)

(b)

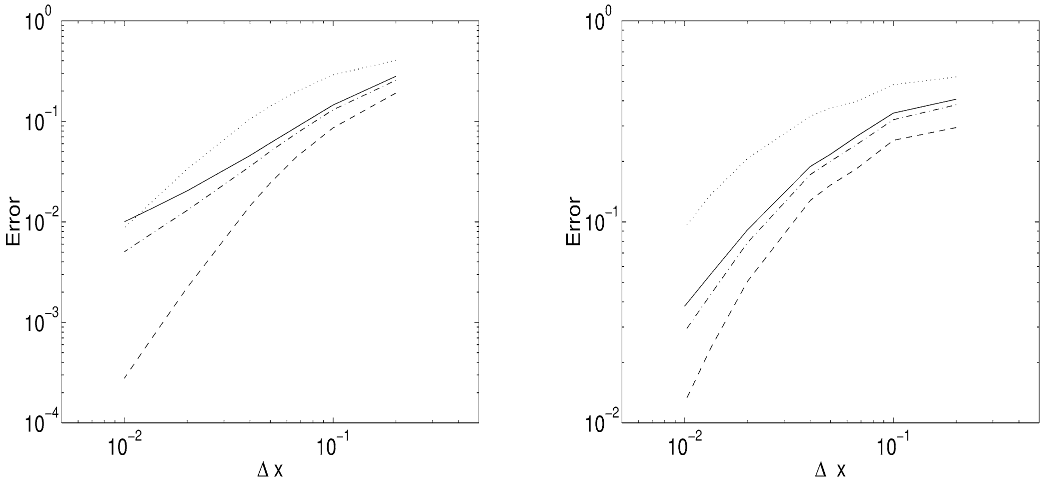
(c)

(d)

(e)

(f)

Fig. 11. Approximated solutions and exact solution at $t = 20$. (a) Lax–Wendroff; (b) Quickest with fictitious point; (c) Quickest with downwind; (d) Quickest with Leonard; (e) Quickest with Lax–Wendroff; (f) exact solution.



(a) (b)

Fig. 12. Error function as mesh is refined for Lax–Wendroff scheme (···), Quickest scheme with the respective numerical boundary conditions: Boundary suggested by Leonard (Section 3.1) (—); fictitious point boundary (Section 3.4) (— · —); downwind third difference boundary (Section 3.2) and Lax–Wendroff boundary (Section 3.3) (— — —). (a) $\mu = 0.001$; (b) $\nu = 0.1$.

Table 1
Estimated convergence rate p for error, assuming Error $\sim (\Delta x)^p$. The case μ fixed is for $V = 0.5$, $D = 0.001$, $\mu = 0.001$ and the case ν fixed is for $V = 0.25$, $D = 0.0001$, $\nu = 0.1$

	μ fixed	ν fixed
Quickest with downwind	3.03	2.18
Quickest with fictitious point	1.39	1.49
Quickest with Leonard	1.01	1.28
Quickest with Lax–Wendroff	3.04	2.18
Lax–Wendroff	1.98	1.32

For instance, for the Lax–Wendroff scheme on the refinement path for μ fixed, $\Delta t = \mathrm{O}(\Delta x^2)$, we have

$$T_j^n = \frac{1}{6} V \Delta x^2 \left(1 - V^2 \mu^2 \frac{\Delta x^2}{D^2} - 6\mu\right) U_{x^3}^n(x_j) + \cdots, \tag{30}$$

so that the truncation error is second order. Of course the truncation error still has to be related to the error and that can introduce changes in the convergence rate.

On the refinement path for ν fixed, $\Delta t = \mathrm{O}(\Delta x)$, the truncation error for the Lax–Wendroff scheme is

$$T_j^n = \frac{1}{6} V \Delta x^2 \left(1 - \nu^2 - \frac{D\nu}{V \Delta x}\right) U_{x^3}^n(x_j) + \cdots \tag{31}$$

and it is only first order. This refinement path cannot be continued to $\Delta x \to 0$ because it corresponds to $\mu \to \infty$ and at some point the stability boundary will be passed and there will be no stable solutions for smaller values of Δx. The refinement path, μ fixed, can be continued to $\Delta x \to 0$ since it corresponds to $\nu \to 0$ which does not pass a stability boundary.

Doing a similar analysis for the Quickest scheme, from (29) we can infer that the truncation error for the Quickest scheme (boundary conditions are not taken into account), should be second order for the refinement path with μ fixed, but only first order for the refinement path with ν fixed. Of course in our examples, since we are in the presence of numerical boundary conditions, relevant differences between the order of accuracy of the truncation error and the global error occur as we observe in Fig. 12 and Table 1.

The behaviour of the error is illustrated for two refinement paths. In Fig. 12(a) the refinement path for $\mu = 0.001$, with $V = 0.5$ and $D = 0.001$ shows that the practical convergence rate for the Quickest scheme with the different numerical boundary conditions can vary between $p = 1$ and $p = 3$. The Lax–Wendroff scheme is very close to its theoretical second order convergence. Of course the practical convergence rate has to be offset against the practical stability region. In Fig. 12(b) the refinement path for $\nu = 0.1$, with $V = 0.25$ and $D = 0.0001$ is illustrated for values of Δx where the solution is stable. As expected from the discussion of truncation errors, the best convergence rate for ν fixed, is nearly one power less than for the refinement path with μ fixed, for both Lax–Wendroff and Quickest schemes.

It is evident that there are some gains in accuracy by using the numerical boundary conditions described in Sections 3.2 and 3.3. We notice too in Fig. 12(a) that there is an advantage to Quickest schemes compared with Lax–Wendroff when convection is dominant, that is, μ is small.

7. Conclusion

We have studied constant velocity convection diffusion on half line in order to examine how a higher-order finite difference scheme can be implemented and proper account taken of numerical boundary conditions. Lax–Wendroff is a very good scheme but if accuracy is a concern then a higher-order scheme like Quickest is very important but so is the treatment of points adjacent to a boundary. The stability regions are substantially affected by the numerical boundary conditions and in the cases we have examined they can be determined quite accurately by using a von Neumann analysis associated with the spectral radius and matrix analysis. When we choose the downwind third difference numerical boundary condition (Section 3.2) or a Lax–Wendroff boundary condition (Section 3.3) we maintain a good accuracy but we loose some stability. When we require a large region of stability, the numerical boundary condition involving a fictitious point (Section 3.4) seems to be a very good choice. Further work to generalise these results to multidimensional problems is in progress.

Acknowledgements

We acknowledge the referees for their useful comments.

References

[1] H.R. Baum, M. Ciment, R.W. Davis, E.F. Moore, Numerical solutions for a moving shear layer in a swirling axisymmetric flow, in: W.C. Reynolds, R.W. MacCormack (Eds.), Proc. 7th Internat. Conf. on Numerical Methods in Fluid Dynamics, Lect. Notes in Physics, Vol. 141, Springer, Berlin, 1981, pp. 74–79.

[2] R.W. Davis, E.F. Moore, A numerical study of vortex shedding from rectangles, J. Fluid Mech. 116 (1982) 475–506.

[3] J.L.M. van Dorsselaer, J.F.B.M. Kraaijevanger, M.N. Spijker, Linear stability analysis in the numerical solution of initial value problems, Acta Numer. 2 (1993) 199–237.

[4] S.K. Godunov, V.S. Ryabenkii, Spectral stability criteria of boundary value problems for non-self-adjoint difference equations, Russian Math. Survey 18 (1963) 1–12.

[5] B. Gustafsson, H.-O. Kreiss, A. Sundstrom, Stability theory of difference approximations for mixed initial boundary value problems, II, Math. Comp. 26 (1972) 649–686.

[6] R.W. Johnson, R.J. MacKinnon, Equivalent versions of the Quick scheme for finite-difference and finite-volume numerical methods, Comm. Appl. Numer. Methods 8 (1992) 841–847.

[7] H.-O. Kreiss, Stability theory for difference approximations of mixed initial boundary value problems. I, Math. Comp. 22 (1968) 703–714.

[8] P.D. Lax, B. Wendroff, Difference schemes for hyperbolic equations with high order of accuracy, Comm. Pure Appl. Math. 17 (1964) 381–398.

[9] H.W.J. Lenferink, M.N. Spijker, On the use of stability regions in the numerical analysis of initial value problems, Math. Comp. 57 (1991) 221–237.

[10] B.P. Leonard, A stable and accurate convective modelling procedure, Comput. Methods Appl. Mechanics Engrg. 19 (1979) 59–98.

[11] B.P. Leonard, S. Mokhtari, Beyond first-order upwinding the ultra-sharp alternative for non-oscillatory steady-state simulation of convection, Internat. J. Numer. Methods Engrg. 30 (1990) 729–766.

[12] K.W. Morton, I.J. Sobey, Discretisation of a convection–diffusion equation, IMA J. Numer. Anal. 13 (1993) 141–160.

[13] K.W. Morton, Numerical Solution of Convection–Diffusion Problems, Chapman and Hall, London, 1996.

[14] S.C. Reddy, L.N. Trefethen, Pseudospectra of the convection–diffusion operator, SIAM J. Appl. Math. 54 (1994) 1634–1649.

[15] R.D. Richtmyer, K.W. Morton, Difference Methods for Initial-Value Problems, 2nd edn., Wiley-Interscience, New York, 1967.

[16] P.J. Roache, Computational Fluid Dynamics, Hermosa, Albuquerque, NM, 1972.

[17] E. Sousa, A Godunov–Ryabenkii instability for a Quickest scheme, in: J. Wasniewski, L. Vulkov, P. Yalamov (Eds.), Numerical Analysis and Applications, Lecture Notes in Computer Science, Vol. 1988, Springer, Berlin, 2001, pp. 732–740.

[18] E. Sousa, Finite Differences for the convection–diffusion equation: On stability and boundary conditions, Ph.D. Thesis, Oxford University, Oxford, 2001.

[19] R.F. Warming, B.J. Hyett, The modified equation approach to the stability and accuracy analysis of finite-difference methods, J. Comput. Phys. 14 (1974) 159–179.

[20] N.P. Waterson, H. Deconinck, A unified approach to the design and application of bounded higher-order convection schemes, VKI Preprint 1995-21, 1995.