Development of finite difference schemes near an inflow boundary

E. Sousa[*, †]

Department of Mathematics, University of Coimbra, Portugal

SUMMARY

Numerical schemes for a convection–diffusion problem defined on the whole real line have been derived by Morton and Sobey (*IMA J. Numer. Anal.* 1993; **13**:141–160) using the exact evolution operator through one time step. In this paper we derive new numerical schemes by using the exact evolution operator for a convection–diffusion problem defined on the half-line. We obtain a third-order method that requires the use of a numerical boundary condition which is also derived using the same evolution operator. We determine whether there are advantages from the point of view of stability and accuracy in using these new schemes, when compared with similar methods obtained for the whole line. We conclude that the third-order scheme provides gains in terms of stability and although it does not improve the practical accuracy of existing methods faraway from the inflow boundary, it does improve the accuracy next to the inflow boundary. Copyright © 2006 John Wiley & Sons, Ltd.

KEY WORDS: finite differences; convection–diffusion; stability; accuracy

1. INTRODUCTION

The mechanism of convection diffusion appears in many physical applications and accurate modelling of the interaction between convective and diffusive processes can be a difficult task. Although the majority of physical experiments are performed in the presence of boundaries if we consider the approximation of the unsteady convection–diffusion problem, we can observe that much of the literature is concerned with choices for the whole real line.

It is very common that the approximate solutions we derive for the whole line present some difficulties when we need to deal with the presence of a physical boundary. This difficulty is more obvious if we are interested in simulations next to the boundary and at short times. Even if they perform efficiently far away from the physical boundary, next to it they can have

---

[*]Correspondence to: E. Sousa, Department of Mathematics, University of Coimbra, Portugal.
[†]E-mail: ecs@mat.uc.pt

Copyright © 2006 John Wiley & Sons, Ltd.

a poor performance. In this paper, we present new finite difference schemes derived taking into account the existence of an inflow physical boundary.

Finite difference schemes typically consist of replacement of the individual derivative terms in the partial differential equation by a set of discretized approximations (see, e.g. Reference [1]). However, recently different techniques were suggested for deriving finite differences for the unsteady convection–diffusion equation (see, e.g. References [2, 3]). In the next section, we use the framework described in Reference [2] to obtain finite difference schemes taking into account the presence of a physical boundary.

Related with the convergence of a finite difference scheme we encounter questions about stability and accuracy and the presence of a boundary most likely will affect stability and accuracy of the overall numerical scheme. In the third section we study the stability and accuracy of the numerical schemes and in the fourth section, to analyse the performance of the third-order scheme, we present three test problems.

2. THE FINITE DIFFERENCE SCHEMES

Consider the one-dimensional problem of convection with constant velocity V in the positive x direction and constant diffusion $D > 0$

$$\frac{\partial u}{\partial t} + V\frac{\partial u}{\partial x} = D\frac{\partial^2 u}{\partial x^2}, \quad t > 0, \ \ x > 0 \tag{1}$$

with the initial condition

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

and subject to the boundary conditions

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

The exact solution of system (1), (2) and (3) can be found using Laplace transforms in t

$$u(x, t) = \frac{1}{\sqrt{\pi}}\int_0^t g(t - \tau)G^*(x, \tau)\,\mathrm{d}\tau$$

$$+ \frac{1}{\sqrt{\pi}}\int_{(Vt-x)/2\sqrt{Dt}}^{+\infty} f(x - Vt + 2\sqrt{Dt}\xi)\mathrm{e}^{-\xi^2}\,\mathrm{d}\xi$$

$$- \frac{1}{\sqrt{\pi}}\int_{(Vt+x)/2\sqrt{Dt}}^{+\infty} f(-x - Vt + 2\sqrt{Dt}\xi)\mathrm{e}^{Vx/D}\mathrm{e}^{-\xi^2}\,\mathrm{d}\xi \tag{4}$$

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

$$G^*(x, \tau) = \frac{x}{2\sqrt{D}\tau^{3/2}}\mathrm{e}^{-(x-V\tau)^2/4D\tau} \tag{5}$$

Applying the result to evolution over one time step, we write

$$u(x, t_n + \Delta t) = \frac{1}{\sqrt{\pi}} \int_0^{\Delta t} g(t_n + \Delta t - \tau) G^*(x, \tau) \, d\tau$$

$$+ \frac{1}{\sqrt{\pi}} \int_{(V\Delta t - x)/2\sqrt{D\Delta t}}^{+\infty} u(x - V\Delta t + 2\sqrt{D\Delta t}\,\xi, t_n) e^{-\xi^2} \, d\xi$$

$$- \frac{1}{\sqrt{\pi}} \int_{(V\Delta t + x)/2\sqrt{D\Delta t}}^{+\infty} u(-x - V\Delta t + 2\sqrt{D\Delta t}\,\xi, t_n) e^{Vx/D} e^{-\xi^2} \, d\xi \qquad (6)$$

The exact solution for this model problem differs from the solution of a convection–diffusion problem on the whole real line. This is the fundamental solution we shall use, to derive approximation schemes by allowing a local solution to evolve and then restricting the evolved solution to an approximation space.

We can rewrite the evolution operator over one time step, given by (6), in terms of a Green's function

$$u(x, t_n + \Delta t) = \frac{1}{\sqrt{\pi}} \int_0^{\Delta t} g(t_n + \Delta t - \tau) G^*(x, \tau) \, d\tau$$

$$+ \frac{1}{\sqrt{\pi}} \int_0^{+\infty} u(\eta, t_n) G_1(x, \eta, \Delta t) \, d\eta \qquad (7)$$

where

$$G_1(x, \eta, \beta) = \frac{e^{-(\eta - x - V\beta)^2/4D\beta}}{2\sqrt{D\beta}} [1 - e^{\eta x/D\beta}]$$

To derive finite difference approximations we substitute a local polynomial approximation to $u(\eta, t_n)$ in (7), and then carry out the integration of a global polynomial. Suppose we have approximations $\mathbf{U}^n := \{U_j^n\}$ to the values $u(x_j, t_n)$ at the mesh points

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

We associate with each point x_j a local interpolating polynomial through U_j^n and the values at a certain number of neighbouring points. Denoting each such polynomial by $p_j(x; \mathbf{U}^n)$, we generate finite difference schemes from

$$U_j^{n+1} = \frac{1}{\sqrt{\pi}} \int_0^{\Delta t} g(t_n + \Delta t - \tau) G^*(x, \tau) \, d\tau$$

$$+ \frac{1}{\sqrt{\pi}} \int_0^{+\infty} p_j(\eta; \mathbf{U}^n) G_1(x_j, \eta; \Delta t) \, d\eta \qquad (8)$$

The approximation scheme which we obtain comes from approximating $\mathbf{U}^n$ near x_j by a polynomial $p_j(x; \mathbf{U}^n)$, of degree R

$$p_j(x; \mathbf{U}^n) = \sum_{r=0}^{R} b_{jr}(x - x_j)^r$$

Then

$$
\begin{aligned}
U_j^{n+1} = {} & \frac{1}{\sqrt{\pi}} \int_0^{\Delta t} g(t_n + \Delta t - \tau) G^*(x, \tau)\, \mathrm{d}\tau \\
& + \frac{1}{\sqrt{\pi}} \int_{(v-j)/2\sqrt{\mu}}^{+\infty} p_j(x_j - V\Delta t + 2\sqrt{D\Delta t}\,\xi; \mathbf{U}^n) \mathrm{e}^{-\xi^2}\, \mathrm{d}\xi \\
& - \frac{1}{\sqrt{\pi}} \int_{(v+j)/2\sqrt{\mu}}^{+\infty} p_j(-x_j - V\Delta t + 2\sqrt{D\Delta t}\,\xi; \mathbf{U}^n) \mathrm{e}^{jv/\mu} \mathrm{e}^{-\xi^2}\, \mathrm{d}\xi
\end{aligned}
\tag{9}
$$

where

$$v = \frac{V\Delta t}{\Delta x} \quad \text{and} \quad \mu = \frac{D\Delta t}{\Delta x^2}$$

First, for clarity, we assume that the left boundary condition is zero, that is, $g(t) = 0$. Then the first integral in (9) is zero and we can write after integration of the polynomial form

$$
\begin{aligned}
U_j^{n+1} = {} & b_{j0} \left[\frac{1}{2}\mathrm{Erfc}\left(\frac{v-j}{2\sqrt{\mu}}\right) - \frac{1}{2}\mathrm{e}^{vj/\mu}\mathrm{Erfc}\left(\frac{v+j}{2\sqrt{\mu}}\right) \right] \\
& + b_{j1}\left[-V\Delta t\,\frac{1}{2}\mathrm{Erfc}\left(\frac{v-j}{2\sqrt{\mu}}\right) + (2x_j + V\Delta t)\frac{1}{2}\mathrm{e}^{vj/\mu}\mathrm{Erfc}\left(\frac{v+j}{2\sqrt{\mu}}\right) \right] \\
& + b_{j2}\left[(V^2(\Delta t)^2 + 2D\Delta t)\frac{1}{2}\mathrm{Erfc}\left(\frac{v-j}{2\sqrt{\mu}}\right) \right. \\
& \left. \quad - ((2x_j + V\Delta t)^2 + 2D\Delta t)\frac{1}{2}\mathrm{e}^{vj/\mu}\mathrm{Erfc}\left(\frac{v+j}{2\sqrt{\mu}}\right) + 2\frac{\sqrt{D\Delta t}}{\sqrt{\pi}}x_j\mathrm{e}^{-(v-j)^2/4\mu} \right] \\
& + b_{j3}\left[-(V^3(\Delta t)^3 + 6VD(\Delta t)^2)\frac{1}{2}\mathrm{Erfc}\left(\frac{v-j}{2\sqrt{\mu}}\right) \right. \\
& \quad + (2x_j + V\Delta t)((2x_j + V\Delta t)^2 + 6D\Delta t)\frac{1}{2}\mathrm{e}^{vj/\mu}\mathrm{Erfc}\left(\frac{v+j}{2\sqrt{\mu}}\right) \\
& \left. \quad - 2(2V\Delta t + 3x_j)\frac{\sqrt{D\Delta t}}{\sqrt{\pi}}x_j\mathrm{e}^{-(v-j)^2/4\mu} \right] + \cdots
\end{aligned}
\tag{10}
$$

where $\mathrm{Erfc}(x)$ is the complementary error function $\mathrm{Erfc}(x) = (2/\sqrt{\pi})\int_x^{\infty} \mathrm{e}^{-t^2}\, \mathrm{d}t$.

Within this general framework we can now obtain finite difference schemes by interpolation on a uniform mesh. We use the usual central, backward and second difference operators to evaluate the coefficients b_{jr}, $r = 0, 1, 2, \ldots$ in terms of the nodal values $\mathbf{U}^n$

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

We present two new numerical schemes using a quadratic interpolant and a cubic interpolant, obtaining in that way the schemes that we call the Modified Lax–Wendroff scheme and the Modified Quickest scheme. Other methods of higher order could be obtained by using higher order interpolants.

Using the quadratic interpolant of U_{j-1}^n, U_j^n and U_{j+1}^n we have the approximation formula for U_j^{n+1}, $j \geqslant 1$

$$U_j^{n+1} = a(j) \left[1 - v\Delta_0 + \left(\frac{v^2}{2} + \mu \right) \delta^2 \right] U_j^n + b(j)\Delta_0 U_j^n + c(j)\delta^2 U_j^n \tag{11}$$

where

$$a(j) = \frac{1}{2} \mathrm{Erfc} \left(\frac{v - j}{2\sqrt{\mu}} \right) - \frac{1}{2} e^{vj/\mu} \mathrm{Erfc} \left(\frac{v + j}{2\sqrt{\mu}} \right)$$

$$b(j) = j e^{vj/\mu} \mathrm{Erfc} \left(\frac{v + j}{2\sqrt{\mu}} \right) \tag{12}$$

$$c(j) = -j(j + v) e^{vj/\mu} \mathrm{Erfc} \left(\frac{v + j}{2\sqrt{\mu}} \right) + Z(j)$$

where $Z(j) = (\sqrt{\mu}/\sqrt{\pi})j e^{-(v-j)^2/4\mu}$. We call this scheme the Modified Lax–Wendroff scheme since for $a(j) = 1$, $b(j) = 0$ and $c(j) = 0$, we obtain the Lax–Wendroff scheme [4, 5].

Although we will concentrate our attention on the scheme that follows, which is obtained using a cubic term, we have shown above, for completeness, the derivation of the Modified Lax–Wendroff scheme.

If $p_j(x, \mathbf{U}^n)$ is extended to include a cubic term, using the interpolation points U_{j-2}^n, U_{j-1}^n, U_j^n and U_{j+1}^n then the approximation formula for $j \geqslant 2$ becomes

$$U_j^{n+1} = a(j) \left[1 - v\Delta_0 + \left(\frac{v^2}{2} + \mu \right) \delta^2 + \frac{v}{6}(1 - v^2 - 6\mu)\delta^2\Delta_- \right] U_j^n$$

$$+ b(j)\Delta_0 U_j^n + c(j)\delta^2 U_j^n + d(j)\delta^2\Delta_- U_j^n \tag{13}$$

where

$$d(j) = -\frac{1}{6}b(j) + \frac{1}{6}e(j)$$

$$e(j) = (4j^3 + 2j^2 v + jv^2 + 6j\mu)e^{vj/\mu} \mathrm{Erfc} \left(\frac{v + j}{2\sqrt{\mu}} \right) - 2(2v + 3j)Z(j)$$

We call this scheme the Modified Quickest scheme since in (13), for $a(j) = 1$, $b(j) = 0$, $c(j) = 0$ and $d(j) = 0$ we obtain the Quickest scheme [6].

To obtain the scheme (13) we interpolate at two points upwind but we do not have these points for interpolation around the first point of the mesh. Here, therefore, we need to consider a numerical boundary condition at the first mesh point. At this point we perform a cubic interpolation of the points U_0^n, U_1^n, U_2^n, U_3^n, namely

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

$$+ b(1)\Delta_0 U_1^n + c(1)\delta^2 U_1^n + d(1)\delta^2\Delta_+ U_1^n \qquad (14)$$

where Δ_+ is the forward difference operator defined by $\Delta_+ U_j^n = U_{j+1}^n - U_j^n$.

When $j \to \infty$ we have

$$\text{Erfc}\left(\frac{v - j}{2\sqrt{\mu}} \right) \to 1, \quad \text{Erfc}\left(\frac{v + j}{2\sqrt{\mu}} \right) \to 0, \quad Z(j) \to 0$$

and

$$a(j) \to 1 \quad b(j) \to 0 \quad c(j) \to 0 \quad d(j) \to 0$$

We plot the coefficients $a(j), b(j), c(j), d(j)$ in Figures 1 and 2 for some values of v and μ. For different values of v, μ the profiles of the coefficients are similar to the ones presented in those figures. The amplitude can change and the wave move slightly to the left or right.

Although, these new schemes are considerably different from the Lax–Wendroff scheme and Quickest scheme at the first points of the mesh, they are only slightly different at the other mesh points. In the next section we discuss the stability and accuracy of the new schemes.
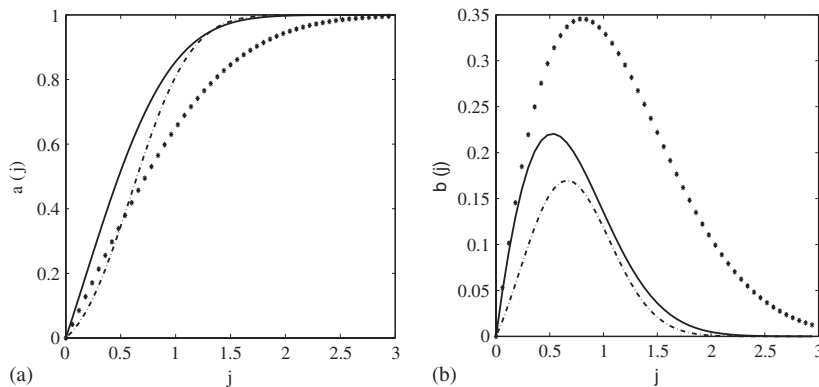

Figure 1. Coefficients $a(j), b(j)$ of (11) and (13) for: $v = 0.1$ and $\mu = 0.2$ (—), $v = 0.5$ and $\mu = 0.1$ (— · —·), $v = 0.1$ and $\mu = 0.5$ (∗∗).
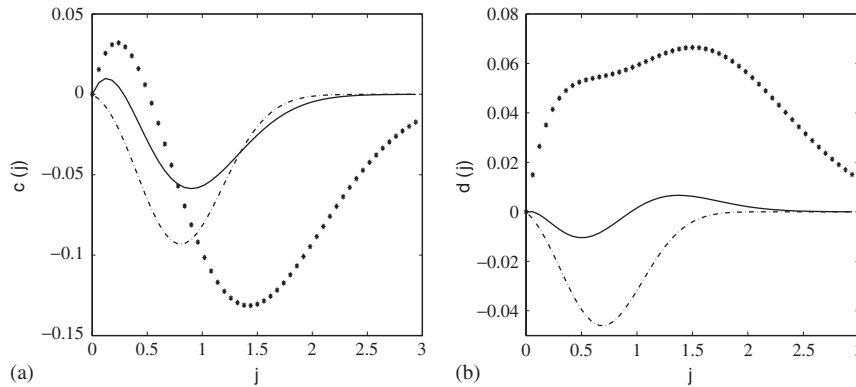
Figure 2. Coefficients $c(j), d(j)$ of (11) and (13) for: $v = 0.1$ and $\mu = 0.2$ (—), $v = 0.5$ and $\mu = 0.1$ ($- \cdot -\cdot$), $v = 0.1$ and $\mu = 0.5$ ($**$).

3. ACCURACY AND STABILITY

To analyse the stability of the new schemes we cannot use the von Neumann stability analysis since the coefficients are not constant, although under general conditions (see Reference [7]) it can be proved that for linear, non-constant coefficient problems a local von Neumann analysis will provide a necessary condition for stability. The more natural option in this case is to use the spectrum and matrix analysis based on the observation of the norm and spectrum behaviour of the iterative matrix. Also the matrix method provides information on the influence of boundary conditions.

Concerning accuracy, to calculate the local truncation error we cannot apply the modified equation as described in Reference [8], since we have non-constant terms. On the other hand, we can derive formal truncation error estimates in the same way as suggested in Morton and Sobey [2] by applying the Peano kernel theorem (see Reference [9]).

3.1. Accuracy of the new schemes

In this section we derive truncation error estimates. The exact solution (7) can be written

$$u(\cdot, t + \Delta t) = E(\Delta t)u(\cdot, t)$$

where $E(\Delta t)$ is an evolution operator. Consider I_p the local polynomial interpolation operator of degree p based on the nodal values U_j^n. Suppose R is a restriction operator onto the nodes, then the truncation error is given by

$$\Delta t T^n = RE(\Delta t)(u^n - I_p R u^n)$$

We have the following results for the schemes considered.

Theorem
For the scheme derived using the quadratic interpolant, the Modified Lax–Wendroff, the truncation error is given by

$$\Delta t T^n_{|x_j} = \tfrac{1}{6}\Delta x^3 u^n_{xxx} g_j^2(v, \mu) + O(\Delta x^4 u^n_{x^4}) \tag{15}$$

where

$$g_j^2(v, \mu) = v(1 - v^2 - 6\mu)a(j) - b(j) + e(j)$$

For the Modified Quickest scheme, obtained using a cubic interpolant, the truncation error for x_j, $j \geqslant 2$ is given by the cumbersome expression

$$\Delta t T_{|x_j}^n = \tfrac{1}{24}\Delta x^4 u_{x^4}^n g_j^3(v, \mu) + O(\Delta x^5 u_{x^5}^n) \tag{16}$$

with

$$g_j^3(v, \mu) = (12\mu^2 - 2\mu - 12\mu v(1 - v) + v(1 - v^2)(2 - v))a(j)$$

$$- 2b(j)(12\mu v + 2v^3 + 2jv + 1 + 2j^3)$$

$$+ 2c(j)(1 + 6j^2) + 2(1 - 2j)e(j) + 2Z(j)(3v^2 + j^2 + 10\mu)$$

Proof
The truncation error estimates are derived by applying the Peano kernel theorem. The details of the proof can be seen in Appendix A.   □

When $j \to \infty$ then $a(j) = 1$, $b(j) = c(j) = e(j) = 0$ and expressions (15) and (16) are the truncation errors obtained for the Lax–Wendroff scheme and Quickest scheme, respectively. Consequently, for each scheme, near the boundary we shall have a different truncation error which nevertheless does not have an inferior order.

These results indicate that the new schemes have similarities in terms of accuracy with the Lax–Wendroff and Quickest schemes, respectively.

3.2. Stability analysis of the new schemes

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

$$U_N^n = 0 \quad \forall n \tag{17}$$

The choice of this outflow boundary is motivated by the fact that we assume that the exact solution goes to zero when x goes to infinity.

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

$$U^{n+1} = AU^n + v^n, \quad n = 0, 1, 2, \ldots \tag{18}$$

where A is an $N \times N$ matrix and depends on the scheme used and v^n appears when the inflow boundary condition is not zero.

Any errors E^n in a calculation based on (18) will grow according to

$$E^{n+1} = AE^n, \quad n = 0, 1, 2, \ldots \tag{19}$$

where $E^n = u^n - U^n$ with u^n, U^n the exact and numerical solutions of (18), 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

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

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

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

and that

$$\rho(A) \leqslant \|A\|$$

A simple criterion for regulating the error growth governed by (19) is given by

$$\rho(A) \leqslant 1 \tag{20}$$

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 (19). 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

$$\|A\| \leqslant 1 \tag{21}$$

is sufficient to ensure that the error cannot grow with n.

From (19) we have

$$E^n = A^n E^0, \quad n = 1, 2, \ldots \tag{22}$$

Expression (22) shows that in order for all E^n to remain bounded and scheme (18) to remain stable the infinite set of operators A^n has to be uniformly bounded for all n, Δt and Δx.

Our stability analysis consists essentially in applying the sufficient condition for stability $\|A\| \leqslant 1$ with the necessary stability condition $\rho(A) \leqslant 1$. We will plot the regions using MATLAB and for the matrix size $N = 30$.

The stability region for the scheme derived using a quadratic polynomial approximation, which we called the Modified Lax–Wendroff scheme, is given by Figure 3.

The von Neumann stability condition for the Lax–Wendroff scheme is the well known condition $v^2 + 2\mu \leqslant 1$. Assuming that the stability region for the Modified Lax–Wendroff is defined by $\|A\| \leqslant 1$, displayed in Figure 3, we can infer that we do not have an advantage in terms of stability by choosing the Modified Lax–Wendroff scheme instead of the Lax–Wendroff scheme.

Consider the Modified Quickest scheme as being the scheme (13) associated with the numerical boundary condition (14). We display in Figure 4 the stability region for that scheme and additionally we plot in Figure 4(b) the practical von Neumann stability region for the Quickest scheme. This allows us to see in what way it relates with the region $\|A\| \leqslant 1$, where A is the iterative matrix of the Modified Quickest scheme.
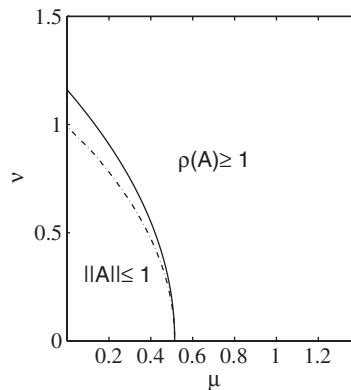
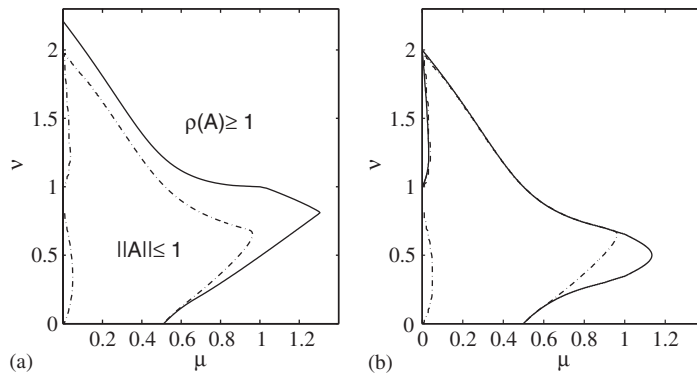Figure 3. Stability region for the Modified Lax–Wendroff scheme: $\rho(A) = 1$ (—) and $\|A\| = 1$ ($- \cdot -$).

Figure 4. Stability region for the Modified Quickest scheme with the numerical boundary condition (14): (a) $\rho(A) = 1$ (—) and $\|A\| = 1$ ($- \cdot -$); and (b) $\|A\| = 1$ ($- \cdot -$) and practical von Neumann stability for the Quickest scheme (—).

It is important to remember that $\|A\| \leqslant 1$ is a sufficient condition for stability and $\rho(A) \leqslant 1$ a necessary one. This means we have instability when the eigenvalues of the matrix A are larger than one, stability when the norm of the matrix is less than one and a region of uncertainty where although the norm is bigger than one, the necessary condition $\rho(A) \leqslant 1$ is still satisfied.

In fact, the new scheme seems to be stable in the bottom left region, for small μ, where $\|A\| \geqslant 1$ and $\rho(A) \leqslant 1$, plotted in Figures 4(a) and (b). This was checked by running numerical experiments on this region. On the other hand, in the top left, for small μ, the scheme is unstable. This is consistent with the von Neumann stability region given in Figure 4(b) for the Quickest scheme, which is also unstable in the same small region. Recall that the Modified Quickest becomes the Quickest scheme as $j \to \infty$, so it is not surprising that the von Neumann unstable Fourier nodes of the Quickest also affect the stability of the Modified Quickest scheme.

We also plot the stability region of the Modified Quickest scheme with the Lax–Wendroff numerical boundary condition in Figure 5.
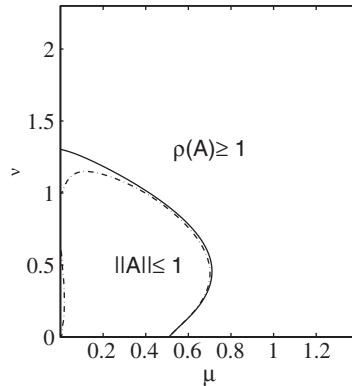
Figure 5. Stability region for the Modified Quickest scheme with the Lax–Wendroff numerical boundary condition: $\rho(A) = 1$ (—) and $\|A\| = 1$ (− · −).

To conclude, we observe there are advantages in terms of stability in using the Modified Quickest scheme associated with the suggested numerical boundary condition, since we do not have the usual penalties in stability associated with the presence of a numerical boundary condition (see also Reference [10]). In fact, the Modified Quickest scheme preserves all the von Neumann stability region of the Quickest scheme.

The numerical boundary conditions in general do interfere with the stability region of a scheme. This raises the question whether by using the Quickest scheme, instead of the Modified Quickest scheme, with the numerical boundary condition (14), we would still have gains in stability. We check this possibility in the next section.

3.3. Stability of a mixed scheme

It was seen in Reference [10] that the choice of numerical boundary conditions may strongly affect the stability of a Quickest scheme even if the accuracy is not affected.

When $j \rightarrow \infty$ the Modified Quickest scheme is identical to the Quickest scheme. It is in the first points of the scheme that a considerable difference may occur and this seems to affect strongly the stability. This fact motivates us to consider the use of the Quickest scheme together with the numerical boundary condition (14), specially in situations where we are interested in long time behaviour of solutions.

We plot, in Figure 6(a), the region $\|A\| \leqslant 1$, where A is the iteration matrix for the Quickest scheme associated with the numerical boundary conditions (14).

We also consider the Quickest scheme with the natural numerical boundary condition similar to (14), with $a(1) = 1$, $b(1) = c(1) = d(1) = 0$, that is

$$U_1^{n+1} = \left[1 - v\Delta_0 + \left(\frac{v^2}{2} + \mu \right) \delta^2 + \frac{v}{6}(1 - v^2 - 6\mu)\delta^2\Delta_+ \right] U_1^n \tag{23}$$

We show the stability region in Figure 6(b), that also appears in Reference [10]. We observe that in Figure 6(a) we recover the von Neumann stability region lost when using other numerical boundary conditions with the Quickest scheme.

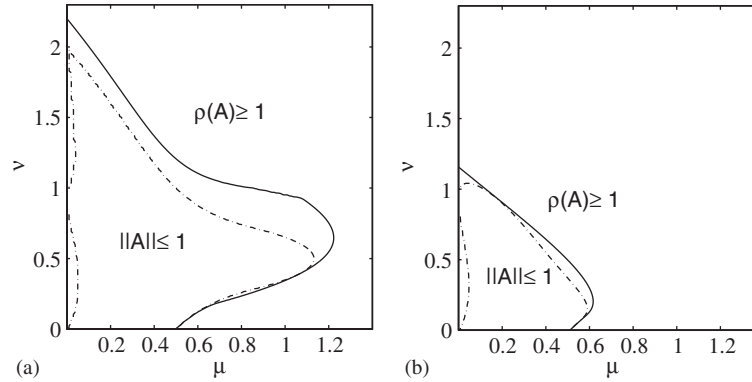*Int. J. Numer. Meth. Engng* 2006; **68**:210–230

Figure 6. Stability region for Quickest scheme with: (a) the numerical boundary condition (14); and (b) the numerical boundary condition (23).

In the next section, in our test problem, when referring only to the Quickest scheme we assume we are considering the Quickest scheme with the numerical boundary condition (23) and we call Mixed scheme when we consider the numerical boundary condition (14).

4. TEST PROBLEMS

To analyse the accuracy of the Modified Quickest and Quickest schemes with different numerical boundary conditions, we consider three test problems.

We star to consider a test problem where the initial condition and the inflow boundary condition are given as follows:

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

The exact solution is given by

$$u(x, t) = \frac{1}{2\sqrt{4Dt + 1}} \left[e^{-(x - Vt)^2/(4Dt + 1)} \, \text{Erfc} \left(-\frac{(x - Vt)}{2\sqrt{Dt(4Dt + 1)}} \right) \right.$$

$$\left. - e^{-((x + Vt)^2/(4Dt + 1)) + (Vx/D)} \, \text{Erfc} \left(\frac{(x + Vt)}{2\sqrt{Dt(4Dt + 1)}} \right) \right] \tag{24}$$

For our tests we assume $V = 0.1$ and $D = 0.001$ and in Figure 7 we display the exact solution (24) for these values at the first instants of time.

Consider the vectors $u_x = (u(x, t_1), \ldots, u(x, t_n))$, where u is the exact solution and $U_x = (U(x, t_1), \ldots, U(x, t_n))$, where U is the approximated solution. The error is defined by

$$\text{Error}(x) = \|u(x, \cdot) - U(x, \cdot)\|_\infty \tag{25}$$
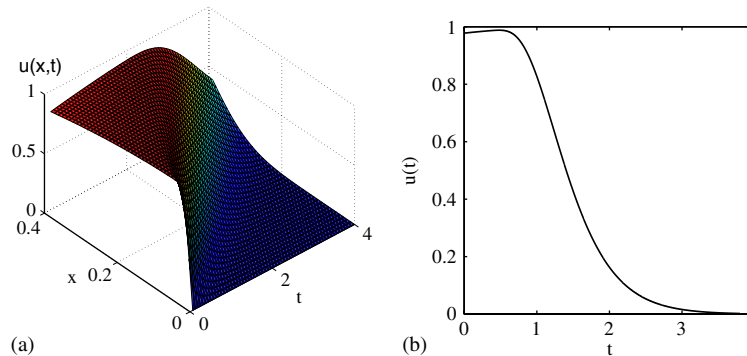
where $\| \cdot \|_\infty$ is the l_∞ norm.

Figure 7. Exact solution (24) for $V = 0.1$ and $D = 0.001$: (a) $0 \leqslant x \leqslant 0.4$, $0 < t \leqslant 4$; and (b) $x = 0.15$.
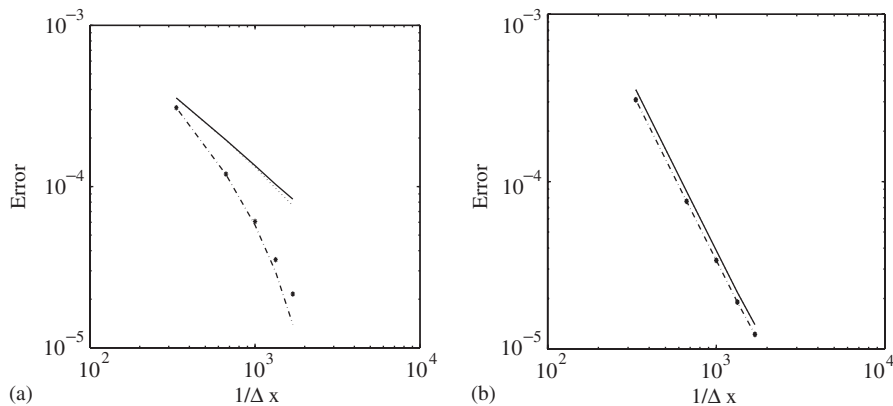

Figure 8. Error function as mesh is refined for: Modified Quickest $(- \cdot -)$, Modified Quickest with Lax–Wendroff $(\cdots)$; Quickest $(—)$; Mixed scheme $(**)$: (a) $v = 0.03$; and (b) $\mu = 0.1$.

In Figure 8 we plot the error at $x = 0.15$, considering different mesh sizes, for the schemes Modified Quickest with Lax–Wendroff numerical boundary condition, the Modified Quickest scheme, the Quickest scheme and the Mixed scheme. Note that when referring to the Modified Quickest scheme we mean the Modified Quickest scheme with numerical boundary condition (14).

In Figure 8(a) we consider the refined path for v fixed, that is $\Delta t = O(\Delta x)$. This refined path cannot be continued indefinitely since $\mu \to \infty$ as $\Delta x \to 0$ and at some point we jump out of the stability region. In our example as we refine the path, μ changes between 0.1 and 0.4. It seems that the Modified Quickest and the Mixed scheme are the best options concerning accuracy. We remind that they were also good options concerning stability.

Figure 8(b) shows the refined path μ fixed, $\Delta t = O(\Delta x^2)$. This path corresponds to $v \to 0$ as $\Delta x \to 0$ and as we refine the mesh v decreases from 0.03 to 0.006. For this case all schemes considered have a similar performance. A disadvantage of this path is that we need to consider a very small time step and therefore it takes longer to reach the given time.

The order of accuracy in Figure 8(a) seems close to first-order for the Quickest and Modified Quickest with Lax–Wendroff and higher for the Modified Quickest and Mixed schemes. In Figure 8(b) they seem closer to second-order. Although the order of convergence is highly dependent on the refinement path overall the Modified Quickest scheme and the Mixed scheme seems to show a better performance.

Since we are in the presence of numerical boundary conditions, some differences between the order of accuracy of the truncation error and the global error may occur. Nevertheless, the truncation error is a good indicator of the global order of convergence. Let us then have a look at the theoretical truncation error described in a previous section. The truncation error for the Modified Quickest is dominated by the leading term

$$\Delta t\, T_j^n \approx [(12\mu^2 - 2\mu - 12\mu v(1-v) + v(1-v^2)(2-v))a(j)]U_{x^4}^n(x_j) \tag{26}$$

where $a(j)$ is given by (12). Note that when $a(j)=1$ this is the truncation error for the Quickest scheme. Since $0 \leqslant a(j) \leqslant 1$ we expect the Modified Quickest scheme to have a slightly smaller truncation error.

On the refined path for v fixed, $\Delta t = O(\Delta x)$, the term that dominates the truncation error for Quickest and Modified Quickest is approximately given by

$$\Delta t\, T_j^n \approx \left[\frac{1}{2}\Delta x^2\frac{D^2 v^2}{V^2} - \frac{1}{12}\Delta x^3\left(\frac{Dv}{V} + 6\frac{D}{V}v^2(1-v)\right) + \frac{1}{24}\Delta x^4 v(1-v^2)(2-v)\right]U_{x^4}^n(x_j) \tag{27}$$

and the leading term is

$$\Delta t\, T_j^n \approx \frac{1}{2}\Delta x^2\frac{D^2 v^2}{V^2}U_{x^4}^n(x_j) \tag{28}$$

Since $\Delta t = O(\Delta x)$, the convergence rate can be close to first-order. It also depends on the value of v we have considered.

For the refined path for μ fixed, $\Delta t = O(\Delta x^2)$, the leading term for the truncation error for Quickest and Modified Quickest is approximately given by

$$\Delta t\, T_j^n \approx \left(\frac{1}{24}\Delta x^4(12\mu^2 - 2\mu) - \frac{1}{2}\Delta x^5\frac{V\mu^2}{D}\left(1 - \frac{V\mu}{D}\Delta x\right)\right.$$

$$\left. + \frac{1}{24}\Delta x^5\frac{V\mu}{D}\left(1 - \frac{V^2\mu^2}{D^2}\Delta x^2\right)\left(2 - \frac{V\mu}{D}\Delta x\right)\right)U_{x^4}^n(x_j) \tag{29}$$

which is dominated by

$$\Delta t\, T_j^n \approx \frac{1}{24}\Delta x^4(12\mu^2 - 2\mu)U_{x^4}^n(x_j) \tag{30}$$

The convergence is close to second-order but also depends on the value of μ we have chosen. In Figure 8(b) we have chosen $\mu = 0.1$. For instance, if instead of $\mu = 0.1$, we have chosen $\mu = 0.001$ the order of accuracy could be increased.

We end the experiments for this test problem by showing in Figure 9 the effect of running numerical experiments outside the stability region. We consider the Modified Quickest scheme and Quickest scheme for $v = 0.2$ and $\mu = 0.6667$ ($\Delta t = 0.006$ and $\Delta x = 0.003$). The latter diverges (check Figure 6 for the stability regions).
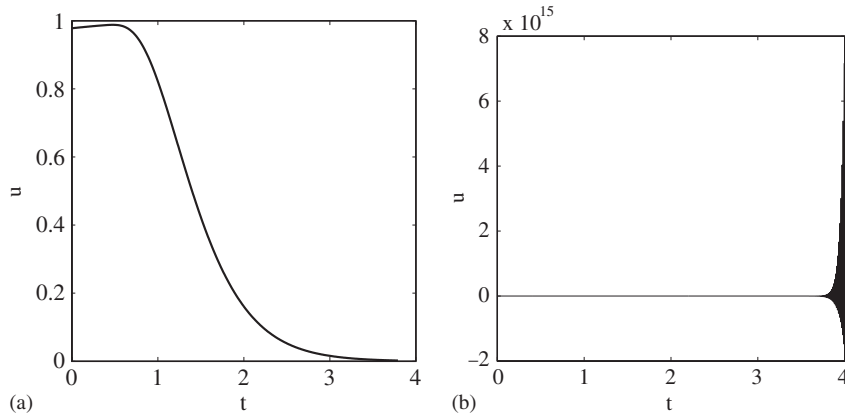
Figure 9. $v = 0.2$ and $\mu = 0.6667$: (a) Modified Quickest scheme; and (b) Quickest scheme.

The second problem, considers the inflow boundary condition, $u(0, t) = g(t)$, such that, $g(t) \neq 0$. Therefore, the first integral that appears in (9), given by

$$\frac{1}{\sqrt{\pi}} \int_0^{\Delta t} g(t_{n+1} - \tau) G^*(x, \tau) \, d\tau \tag{31}$$

where G^* is defined by (5), is added to the right side of the general formula of the finite difference schemes (10). The Modified Quickest scheme includes, for this problem, the computation of (31) using Gaussian quadrature formulas [11].

We consider the test problem with initial and inflow boundary conditions

$$u(x, 0) = 0, \quad x \geqslant 0, \quad u(0, t) = C_0$$

The analytical solution is given by

$$u(x, t) = \frac{C_0}{2} \left[\text{Erfc} \left(\frac{x - Vt}{2\sqrt{Dt}} \right) + e^{Vx/D} \, \text{Erfc} \left(\frac{x + Vt}{2\sqrt{Dt}} \right) \right] \tag{32}$$

and displayed in Figure 10.

Note that, for this example, the integral (31) can be written as

$$\frac{C_0}{\sqrt{\pi}} \int_0^{\Delta t} G^*(x, \tau) \, d\tau \tag{33}$$

In Figure 11 we plot the error at $x = 0.15$, for $V = 0.1$, $D = 0.001$, considering different mesh sizes, for the schemes Modified Quickest with Lax–Wendroff numerical boundary condition, the Modified Quickest, the Quickest scheme and Mixed scheme. We take $v = 0.03$, that is, $\Delta t = O(\Delta x)$.

For the Mixed scheme, the numerical boundary condition (14) is now

$$U_1^{n+1} = \frac{C_0}{\sqrt{\pi}} \int_0^{\Delta t} G^*(x, \tau) \, d\tau + a(1) \left[1 - v\Delta_0 + \left(\frac{v^2}{2} + \mu \right) \delta^2 + \frac{v}{6} (1 - v^2 - 6\mu) \delta^2 \Delta_+ \right] U_1^n$$

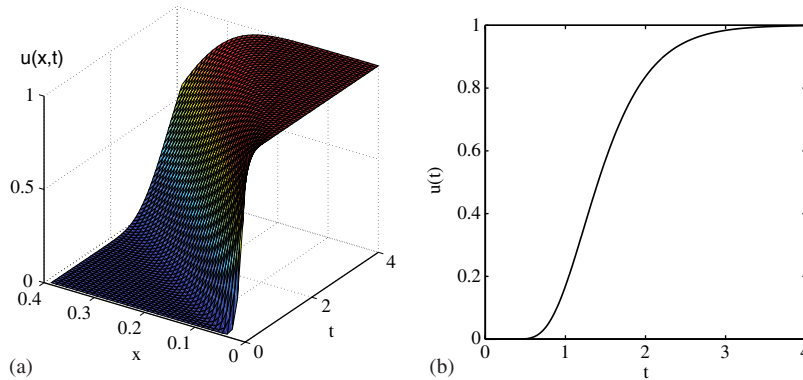$$+ b(1)\Delta_0 U_1^n + c(1)\delta^2 U_1^n + d(1)\delta^2 \Delta_+ U_1^n \tag{34}$$

Figure 10. Exact solution (32) for $C_0 = 1$, $V = 0.1$ and $D = 0.001$: (a) $0 \leqslant x \leqslant 0.4$, $0 < t \leqslant 4$; and (b) $x = 0.15$.
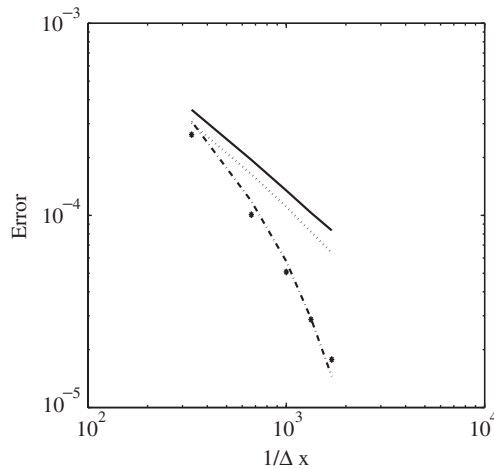

Figure 11. Error function as mesh is refined for: Modified Quickest $(- \cdot -)$, Modified Quickest with Lax–Wendroff $(\cdots)$; Quickest $(—)$; Mixed scheme $(\ast\ast)$.

The Modified Quickest and the Mixed schemes in Figure 11 present the best performances. This was also observed, in Figure 8, for the previous problem.

To close this section we consider a test problem representing a geophysical flow [12]. The initial and inflow boundary conditions are given by

$$u(x, 0) = \sin(2\pi x), \quad u(0, t) = e^{-4\pi^2 Dt} \sin(-2\pi Vt)$$

The exact solution

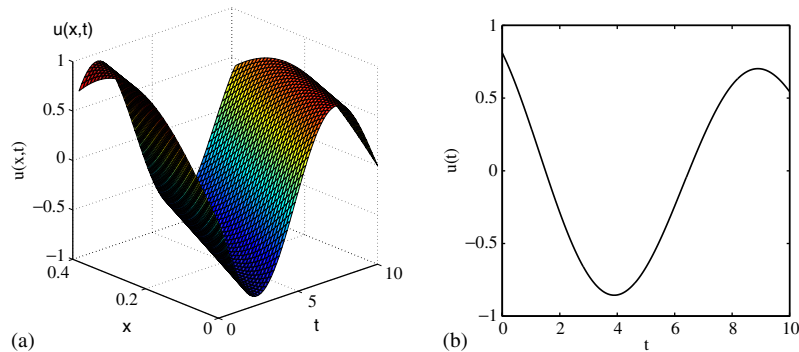$$u(x, t) = e^{-4\pi^2 Dt} \sin(2\pi x - 2\pi Vt) \tag{35}$$

is shown in Figure 12.

Figure 12. Exact solution (32) for $V = 0.1$ and $D = 0.001$: (a) $0 \leqslant x \leqslant 0.4$,
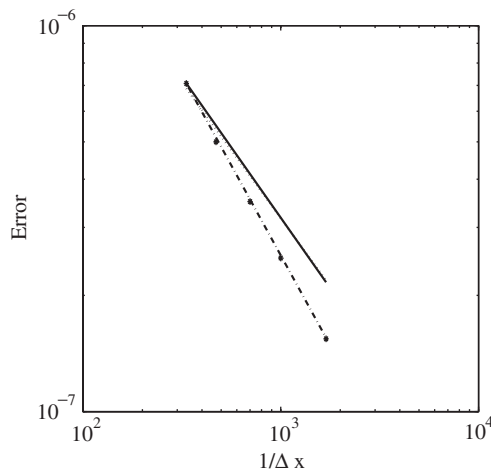$0 < t \leqslant 10$; and (b) $x = 0.15$.

Figure 13. Error function as mesh is refined for: Modified Quickest $(- \cdot -)$, Modified Quickest with
Lax–Wendroff$(\cdot \cdot \cdot)$; Quickest $(—)$; Mixed scheme $(**)$.

Note that the exact solution is continuous at $x = 0$ and $t = 0$ and this is the main difference
between this problem and the previous problems.

In Figure 13, we plot the error at $x = 0.15$, for $V = 0.1$ and $D = 0.001$. We consider different
mesh sizes and $\Delta t = O(\Delta x)$. Although, for this test problem, the Modified Quickest and Mixed
schemes are still the schemes that show best performances, the advantage is less prominent.
The information that comes from the inflow boundary does not have the same importance as it
has in the previous test problems, in the sense that now we have an inflow boundary condition
more predictable.

5. CONCLUDING REMARKS

The new schemes are theoretically interesting because of the general framework in which they
are derived. We have focused our attention on the Modified Quickest scheme which seems to

provide an advantage both in terms of stability and in terms of the error near to the inflow boundary.

The gain in stability seems strongly associated with the choice of the numerical boundary condition at the first point of the mesh. The Modified Quickest scheme for $j \to \infty$ becomes identical to the Quickest scheme and consequently for most practical purposes it is in the first discretization points that we could expect some significant difference. Consequently, the mixed scheme seems also a very good choice. The extension to the two-dimensional case is straightforward.

APPENDIX A: ERROR ANALYSIS FOR THE NEW SCHEMES

Following Morton and Sobey [2], the truncation error is given by

$$T^n = \frac{1}{\Delta t} RE(\Delta t)(u^n - I_p R u^n)$$

where $E(\Delta t)$ is the evolution operator $u(\cdot, t + \Delta t) = E(\Delta t) u(\cdot, t)$, for our problem in the semi line $x \geqslant 0$, R is the restriction operator onto the nodes and I_p is the local approximation based on nodal values.

Let us define the interpolation error $Lu^n = u^n - I_p u^n$ and define for $a \geqslant 0$, the integrals of the form

$$E_m(a; \mu) = \int_0^\infty \xi^m e^{-(\xi+a)^2/4\mu} \frac{d\xi}{2\sqrt{\pi\mu}} \tag{A1}$$

In what follows we denote $s = (x - x_j)/\Delta x$ and we omit the subscript n, referring only to $u(x)$ and its evolution over one time step.

A.1. Quadratic interpolation

We calculate the error at the point $x_j = j\Delta x$. We have, for $j \geqslant 1$

$$Lu = u(x) - \tfrac{1}{2}[-s + s^2]u(x_{j-1}) - [1 - s^2]u(x_j) - \tfrac{1}{2}[s^2 + s]u(x_{j+1})$$

Then using the Peano kernel theorem we can write

$$Lu = \int_0^\infty K(x, p)u^{(3)}(p)\, dp$$

where $K(x, p) = (1/2)L_x[(x - p)_+^2]$, L_x refers to L acting in x, and $(x - p)_+^2 = (x - p)^2$ if $x - p \geqslant 0$, and zero otherwise. We calculate the Peano kernel function $K(s\Delta x + j\Delta x, \xi\Delta x)$

$$K = \frac{1}{2}\Delta x^2 \begin{cases} (s + j - \xi)_+^2 - (j - 1 - \xi)^2(-s/2 + s^2/2) \\ \quad - (j - \xi)^2(1 - s^2) - (j + 1 - \xi)^2(s^2/2 + s/2), & 0 < \xi < j - 1 \\ (s + j - \xi)_+^2 - (j - \xi)^2(1 - s^2) \\ \quad - (j + 1 - \xi)^2(s^2/2 + s/2), & j - 1 < \xi < j \\ (s + j - \xi)_+^2 - (j + 1 - \xi)^2(s^2/2 + s/2), & j < \xi < j + 1 \\ (s + j - \xi)_+^2, & \xi > j + 1 \end{cases}$$

The local error is given by

$$\Delta t T = RE_x \int_0^\infty K(x, p) u^{(3)}(p) \, \mathrm{d}p = \int_0^\infty RE_x K(x, p) u^{(3)}(p) \, \mathrm{d}p$$

where we use the notation E_x to describe $E(\Delta t)$ acting on x. After some manipulation to calculate $RE_x K(x, p) u^{(3)}(p)$ we have

$$RE_x K = \frac{1}{2} \Delta x^2 \begin{cases} E_2(\xi + v - j; \mu) - \mathrm{e}^{vj/\mu} E_2(\xi + v + j; \mu) \\ \quad - (j - 1 - \xi)^2 (-f_1 + f_2)/2 \\ \quad - (j - \xi)^2 (f_3 - f_2) - (j + 1 - \xi)^2 (f_2 + f_1)/2, \quad 0 < \xi < j - 1 \\ E_2(\xi + v - j; \mu) - \mathrm{e}^{vj/\mu} E_2(\xi + v + j; \mu) \\ \quad - (j - \xi)^2 (f_3 - f_2) - (j + 1 - \xi)^2 (f_2 + f_1)/2, \quad j - 1 < \xi < j \\ E_2(\xi + v - j; \mu) - \mathrm{e}^{vj/\mu} E_2(\xi + v + j; \mu) \\ \quad - (j + 1 - \xi)^2 (f_2 + f_1)/2, \qquad\qquad\qquad j < \xi < j + 1 \\ E_2(\xi + v - j; \mu) - \mathrm{e}^{vj/\mu} E_2(\xi + v + j; \mu), \qquad \xi > j + 1 \end{cases}$$

where

$$f_1 = -v a(j) + b(j), \quad f_2 = (v^2 + 2\mu) a(j) + 2c(j)$$

$$f_3 = a(j), \quad f_4 = -(v^3 + 6\mu v) a(j) + e(j)$$

The exact error at x_j is given by

$$\Delta t T_{|x_j} = \Delta x^3 \left[\int_0^{+\infty} K_1(\xi, v, \mu) u^{(3)}(\xi \Delta x) \, \mathrm{d}\xi + \int_0^{j+1} K_2(\xi, v, \mu) u^{(3)}(\xi \Delta x) \, \mathrm{d}\xi \right]$$

where we have introduced two functions

$$K_1(\xi, v, \mu) = \tfrac{1}{2} [E_2(-j + v + \xi; \mu) - \mathrm{e}^{vj/\mu} E_2(j + v + \xi; \mu)]$$

$$K_2(\xi, v, \mu) = \begin{cases} -(f_2 - f_1)(j - 1 - \xi)^2/4 \\ \quad + (f_2 - f_3)(j - \xi)^2/2 \\ \quad - (f_2 + f_1)(j + 1 - \xi)^2/4 \qquad 0 \leqslant \xi < j - 1 \\ (f_2 - f_3)(j - \xi)^2/2 \\ \quad - (f_2 + f_1)(j + 1 - \xi)^2/4 \qquad j - 1 \leqslant \xi < j \\ \quad - (f_2 + f_1)(j + 1 - \xi - j)^2/4 \quad j \leqslant \xi \leqslant j + 1 \end{cases}$$

Although the exact error expression appears complicated, it can be used to examine the detailed structure of the error and to obtain overall bounds for the local error.

Considering first the structure of the error, if we assume $u \in C^\infty(\mathbb{R})$ and use a Taylor series expansion for $u^{(3)}$ around x_j, then after some algebra

$$\Delta t T_{|x_j} = \tfrac{1}{6}\Delta x^3 u_{xxx} g_j^2(v, \mu) + O(\Delta x^4 u_{x^4})$$

where

$$g_j^2(v, \mu) = v(1 - v^2 - 6\mu)a(j) - b(j) + e(j)$$

Secondly, we can obtain an overall bound for the local error since

$$|\Delta t T_{|x_j}| \leqslant \Delta x^3 |u|_{3,\infty} \left[\int_0^{+\infty} |K_1(\xi, v, \mu)| \, \mathrm{d}\xi + \int_0^{j+1} |K_2(\xi, v, \mu)| \, \mathrm{d}\xi \right]$$

A.2. Cubic interpolation

We have

$$Lu = u(x) - \tfrac{1}{6}[s - s^3]u(x_{j-2}) - \tfrac{1}{2}[-2s + s^2 + s^3]u(x_{j-1})$$
$$- \tfrac{1}{2}[2 + s - 2s^2 - s^3]u(x_j) - \tfrac{1}{6}[2s + 3s^2 + s^3]u(x_{j+1})$$

The Peano kernel function $K(x, p)$ for $x = s\Delta x + j\Delta x$ and $p = \Delta x \xi$ is given by

$$K = \tfrac{1}{6}\Delta x^3[(s + j - \xi)_+^3 - \tfrac{1}{6}(j - 2 - \xi)_+^3(s - s^3)$$
$$- \tfrac{1}{2}(j - 1 - \xi)_+^3(-2s + s^2 + s^3) - \tfrac{1}{2}(j - \xi)_+^3(2 + s - 2s^2 - s^3)$$
$$- \tfrac{1}{6}(j + 1 - \xi)_+^3(2s + 3s^2 + s^3)]$$

Summarizing the calculations in this case, the exact error for $x_j, j \geqslant 2$ is given by

$$\Delta t T_{|x_j} = \frac{\Delta x^4}{6} \left[\int_0^{+\infty} W_1^2(\xi, v, \mu)u^{(4)}(\xi\Delta x) \, \mathrm{d}\xi + \int_0^{j+1} W_2^2(\xi, v, \mu)u^{(4)}(\xi\Delta x) \, \mathrm{d}\xi \right]$$

where the functions

$$W_1^2(\xi, v, \mu) = [E_3(-j + v + \xi; \mu) - \mathrm{e}^{vj/\mu}E_3(j + v + \xi; \mu)]$$

$$W_2^2(\xi, v, \mu) = \begin{cases} q_1(\xi - j + 2)^3/6 \\ \quad + q_2(\xi - j + 1)^3/2 \\ \quad + q_3(\xi - j)^3/2 + q_4(j)(\xi - j - 1)^3/6, & 0 \leqslant \xi < j - 2 \\ q_2(\xi - j + 1)^3/2 + q_3(j)(\xi - j)^3/2 \\ \quad + q_4(\xi - j - 1)^3/6, & j - 2 \leqslant \xi < j - 1 \\ q_3(\xi - j)^3/2 + q_4(j)(\xi - j - 1)^3/6, & j - 1 \leqslant \xi \leqslant j \\ q_4(\xi - j - 1)^3/6, & j \leqslant \xi \leqslant j + 1 \end{cases}$$

where

$$q_1 = f_1 - f_4$$

$$q_2 = f_4 + f_2 - 2f_1$$

$$q_3 = 2f_3 - f_4 + f_1 - 2f_2$$

$$q_4 = 2f_1 + 3f_2 + f_4$$

If we assume $u \in C^{\infty}(\mathbb{R})$ and use a Taylor expansion for $u^{(4)}$ then the truncation error is given by the cumbersome expression

$$\Delta t T_{|x_j} = \tfrac{1}{24} \Delta x^4 u_{xxxx} g_j^3(v, \mu) + \cdots$$

with

$$g_j^3(v, \mu) = (12\mu^2 - 2\mu - 12\mu v(1 - v) + v(1 - v^2)(2 - v))a(j)$$

$$- 2b(j)(12\mu v + 2v^3 + 2jv + 1 + 2j^3)$$

$$+ 2c(j)(1 + 6j^2) + 2(1 - 2j)e(j) + 2Z(j)(3v^2 + j^2 + 10\mu)$$

REFERENCES

1. Smith GD. *Numerical Solution of Partial Differential Equations*: *Finite Difference Methods*. Oxford University Press: Oxford, 1985.
2. Morton KW, Sobey IJ. Discretization of a convection–diffusion equation. *IMA Journal of Numerical Analysis* 1993; **13**:141–160.
3. Xu HY, Matovic MD, Pollard A. Finite difference schemes for three-dimensional time-dependent convection–diffusion equation using full global discretization. *Journal of Computational Physics* 1997; **130**:109–122.
4. Lax PD, Wendroff B. Systems of conservations laws. *Communications on Pure and Applied Mathematics* 1960; **13**:217–237.
5. Lax PD, Wendroff B. Difference schemes for hyperbolic equations with high order of accuracy. *Communications on Pure and Applied Mathematics* 1964; **17**:381–398.
6. Leonard BP. A stable and accurate convective modelling procedure based on quadratic upstream interpolation. *Computer Methods in Applied Mechanics and Engineering* 1979; **19**:59–98.
7. Richtmyer RD, Morton KW. *Difference Methods for Initial-value Problems*. Wiley: New York, 1967.
8. Warming FF, Hyett BJ. The modified equation approach to the stability and accuracy analysis of finite difference methods. *Journal of Computational Physics* 1974; **14**:159–179.
9. Powell MJD. *Approximation Theory and Method*. Cambridge University Press: Cambridge, 1981.
10. Sousa E, Sobey IJ. On the influence of numerical boundary conditions. *Applied Numerical Mathematics* 2002; **41**:325–344.
11. Davis PJ, Rabinowitz P. *Methods of Numerical Integration*. Academic Press: New York, 1975.
12. Clancy RM. A note on finite differencing of the advection–diffusion equation. *Monthly Weather Review* 1981; **109**:1807–1809.