INT. J. NUM. METH. HEAT FLUID FLOW, VOL. 5, 341–377 (1995)

THE NIRVANA SCHEME APPLIED TO ONE-DIMENSIONAL ADVECTION

B. P. LEONARD*, A. P. LOCK† AND M. K. MACVEAN†

*Center for Computational Mechanics, The University of Akron, Akron, Ohio 44325-3903, USA
†Meteorological Office, Atmospheric Processes Research Division, Bracknell, Berkshire RG12 2SZ, UK

ABSTRACT

The NIRVANA project is concerned with the development of a nonoscillatory, integrally reconstructed, volume-averaged numerical advection scheme. The conservative, flux-based finite-volume algorithm is built on an explicit, single-step, forward-in-time update of the cell-average variable, without restrictions on the size of the time-step. There are similarities with semi-Lagrangian schemes; a major difference is the introduction of a discrete integral variable, guaranteeing conservation. The crucial step is the interpolation of this variable, which is used in the calculation of the fluxes; the (analytic) derivative of the interpolant then gives sub-cell behaviour of the advected variable. In this paper, basic principles are described, using the simplest possible conditions: pure one-dimensional advection at constant velocity on a uniform grid. Piecewise Nth-degree polynomial interpolation of the discrete integral variable leads to an Nth-order advection scheme, in both space and time. Nonoscillatory results correspond to convexity preservation in the integrated variable, leading naturally to a large-Δt generalisation of the universal limited. More restrictive TVD constraints are also extended to large Δt. Automatic compressive enhancement of step-like profiles can be achieved without exciting 'stair-casing'. One-dimensional simulations are shown for a number of different interpolations. In particular, convexity-limited cubic-spline and higher-order polynomial schemes give very sharp, nonoscillatory results at any Courant number, without clipping of extrema. Some practical generalisations are briefly discussed.

KEY WORDS NIRVAVA scheme Large time-step Advection-dominated flow Shape-preservation Higher-order methods Conservative scheme

INTRODUCTION

Numerical simulation of highly advective flows presents one of the most striking challenges of computational mechanics. Even the apparently simplest of physical problems—pure one-dimensional advection of a scalar profile at constant velocity ($u = \text{const} > 0$), described by,

$$\frac{\partial \phi}{\partial t} + u\frac{\partial \phi}{\partial x} = 0 \tag{1}$$

is plagued with difficulties. Standard methods for simulating the evolution of the scalar from an initial profile, $\phi(x,0)$, typically show strong distortion due to numerical smearing, unphysical oscillations, or both. By contrast, the analytic solution consists of a simple translation of the initial profile,

$$\phi(x,t) = \phi(x - ut, 0) \tag{2}$$

0961–5539/95/040341–37$2.00

Received December 1993
Revised October 1994

This solution is sometimes used as the basis for a numerical advection scheme. For example, a hierarchy of explicit advection schemes based on transient interpolation modelling[1] use (2), written for a single time step,

$$\phi_i^{n+1} = \phi(x_i, \Delta t) = \phi(x_i - u\Delta t, 0) = \phi^n(x_i - u\Delta t) \tag{3}$$

In general, $x_i - u\Delta t$ does not lie at a grid point, so spatial interpolation of ϕ^n is used to evaluate the right-hand side. For a Courant number,

$$c = u\Delta t/h \tag{4}$$

less than one, where h is the mesh spacing, (3) can easily be rewritten in a conservative, pseudoflux-difference form. Recursive formulae for (in principle, arbitrarily) high-order advective fluxes were derived in Reference 1; these, together with universal limiter constraints, give highly accurate, nonoscillatory results. However, in that particular formulation, the time step is restricted by the condition $c \leqslant 1$. One of the main aims of the present paper is to extend these techniques to large Courant number.

Semi-Lagrangian schemes[2] are also based on (2). They are designed to operate at large Δt ($c \gtrless 1$), but are not inherently conservative, although approximate conservation can be enforced using a recently developed technique[3]. The so-called modified method of characteristics[4], MMOC, is closely related to semi-Lagrangian schemes, but rewritten (for one dimension) in a strictly conservative flux-based form. In its published form, MMOC is based on a number of time- and space-centred assumptions, effectively restricting it to second-order accuracy, although it appears that higher-order extensions could be developed. Similar comments apply to the large-time-step methods devised by LeVeque[5].

Finite-volume methods[6] are constructed by integrating (1) spatially across a control-volume cell, and in time from t to $t + \Delta t$, giving (for a uniform grid-spacing),

$$\bar{\phi}_i^{n+1} = \bar{\phi}_i^n - c(\phi_r - \phi_l) \tag{5}$$

where the bars represent *spatially-averaged* cell values across cell i, and right and left *time-averaged* face values have been introduced. This is still an exact representation of the physics (for constant c), equivalent to (1).

Ideally, a practical advection scheme should possess the following properties:

(i) *Single time-step explicit update.* This is potentially very efficient, both in terms of storage (only one time-level is stored—and then overwritten) and computation (because of vectorizability and the possibility of parallel processing).

(ii) *No Courant number restriction.* According to the so-called CFL condition for hyperbolic equations, the numerical domain of dependence (at time-level n) should include the physical domain of dependence (as a necessary stability constraint)[7]. But, for any numerical scheme, the discrete solution is known everywhere at time-level n; therefore, there should be no CFL stability restriction on the Courant number. The reason that many explicit schemes do require a Courant-number constraint stems from a *range restriction* on local interpolants (see later).

(iii) *Conservative, flux-based, finite volume formulation.* In a finite-volume formulation, the advective (plus diffusive) flux at any cell face is simultaneously an outflux from one cell and an influx into the respective adjacent cell sharing the face. 'Mass' is therefore conserved, to machine accuracy. The most straight-forward finite-volume formulation involves cell-average values as dependent variables. In this case, sub-cell interpolation is needed in order to calculate fluxes.

(iv) *Highly accurate, with shape-preserving features.* If the initial profile involves short-wavelength components (especially near-discontinuities), the evolving profile in highly advective flow should maintain high resolution of these features, without unphysical overshoots or oscillations, and without numerical smearing. This implies good phase accuracy (low numerical dispersion) and small numerical dissipation. This is intimately connected with achieving high-order, shape-preserving sub-cell interpolation.

The larger NIRVANA project is concerned with the development of a nonoscillatory, integrally reconstructed, volume-averaged numerical advection scheme. The conservative, flux-based, finite-volume formulation is based on a single-time-step explicit update, without stability restrictions on the time-step. The following sections outline the basic principles in the simplest case of one-dimensional pure advection at a constant velocity on a uniform grid. Some suggestions for generalisation are given in a later section.

BASIC PRINCIPLES

The one-dimensional NIRVANA explicit update is based on (5). In order to calculate the time-averaged face fluxes on the right-hand side, it is necessary to have a good estimate of the sub-cell behaviour of $\phi(x)$. This introduces a basic numerical problem: given a set of values of cell-average data, $\bar{\phi}_i^n$, at the earlier time-level, find an appropriate sub-cell interpolation to give $\phi(x)$, while satisfying the cell-average constraint across each cell,

$$\bar{\phi}_i^n = \frac{1}{h}\int_{x_i-h/2}^{x_i+h/2} \phi(x)\mathrm{d}x \qquad \text{for all } i \tag{6}$$

where x_i is the coordinate at the centre of cell i. Then, knowing the sub-cell behaviour of $\phi(x)$, estimate the time-averaged face fluxes based on a local solution of the continuous problem near each face. For purely advective flow at constant velocity, the latter step is particularly simple. For example, the left-face flux is found by converting the time integral into a spatial integral (noting that $u\Delta t = ch$ and $u\mathrm{d}t = \mathrm{d}x$):

$$c\phi_l(i) = \frac{c}{\Delta t}\int_0^{\Delta t} \phi(x_i - h/2, t)\mathrm{d}t = \frac{1}{h}\int_{x_i-h/2-ch}^{x_i-h/2} \phi(x)\mathrm{d}x \tag{7}$$

where conservation is guaranteed by,

$$c\phi_r(i) = c\phi_l(i+1) \tag{8}$$

Equation (5) can now be updated, and the process begins all over again.

Integral reconstruction

In one dimension, the simplest way to find a sub-cell variable $\phi(x)$ satisfying (6) is to define a *discrete integral variable* in terms of local cell-average values by,

$$\psi_i = \psi(x_i + h/2) = \psi_{i-1} + h\bar{\phi}_i^n \tag{9}$$

By recursion, ψ_i is the cumulative sum of cell-average values,

$$\psi_i = \psi_0 + h\sum_{p=1}^{i} \bar{\phi}_p^n \tag{10}$$

for constant h. Thus, given a set of cell-average values, $\bar{\phi}_i^n$, a corresponding set of discrete ψ_i values can be constructed recursively from (9). Note that, from the definition, ψ_i occurs at the right-hand face of cell i (and, correspondingly, ψ_{i-1} is at the left face of cell i).

Now the crucial step in the NIRVANA strategy is to perform an appropriate interpolation of ψ_i, collocated at all ψ_i values, to give the sub-cell integral variable,

$$\psi(x) = \text{INTERPOLATION OF } (\psi_i) \tag{11}$$

satisfying the collocation conditions,

$$\psi(x_i+h/2)=\psi_i, \qquad \psi(x_i-h/2)=\psi_{i-1}, \qquad \text{etc.} \tag{12}$$

Note that $\psi(x)$ is necessarily continuous, although its first and higher derivatives may not be. The sub-cell interpolation of $\phi(x)$ is given by differentiation,

$$\phi(x)=\frac{d\psi(x)}{dx} \tag{13}$$

thus automatically satisfying the cell-average constraint, (6); i.e.,

$$\frac{1}{h}\int_{x_i-h/2}^{x_i+h/2}\phi(x)\,dx=\frac{\psi_i-\psi_{i-1}}{h}\equiv\bar{\phi}_i^n \tag{14}$$

This is the same procedure as that recently advocated by Hyman *et al.*[8]. The basic idea was originally introduced by Colella and Woodward[9]. From the definition, (13), it should be clear that $\phi(x)$ may involve discontinuities in value. For example, (nonphysical) discontinuities in $\phi(x)$ across cell faces correspond to discontinuities in the gradient of ψ when piecewise interpolants are used. Jumps in $\phi(x)$ may also occur anywhere, representing sub-cell resolution of physical discontinuities.

For pure advection, the face fluxes can be calculated directly from the integral variable, $\psi(x)$, as shown below. The sub-cell scalar itself, $\phi(x)$, is not actually needed for the update, but it can be retrieved at any time (e.g., as a post-processing operation) from (13). Rewriting (7) in terms of $\psi(x)$ gives, for the left-face flux integral,

$$c\phi_l(i)=\frac{1}{h}\int_{x_i-h/2-ch}^{x_i-h/2}\frac{d\psi}{dx}dx=\frac{\psi_{i-1}-\psi(x_i-h/2-ch)}{h}\equiv\frac{\psi_{i-1}-\psi^*}{h} \tag{15}$$

with $c\phi_r(i)$ given by the usual conservation requirement, (8). Note that there is no restriction on the Courant number, c. The crucial step is the evaluation of ψ^* in (15). Thus, for pure one-dimensional advection at constant velocity, the update of (5) proceeds as follows:

(i) Given the current set of $\bar{\phi}^n$ values, construct ψ_i from (9).
(ii) Perform an appropriate interpolation to give $\psi(x)$.
(iii) Compute $c\phi_l(i)$ from (15) and store.
(iv) Then update explicitly, using (8) for $c\phi_r(i)$:

$$\bar{\phi}_i^+=\bar{\phi}_i-[c\phi_l(i+1)-c\phi_l(i)] \tag{16}$$

where superscript '+' represents the new time-level, and the superscript '*n*' has been dropped, for convenience. Note that steps (i) and (iv) involve very little computational effort, and step (iii) requires only a single evaluation of ψ, at $(x_i-h/2-ch)$, denoted as ψ^*. The main bulk of the work is in the interpolation of $\psi(x)$, step (ii). As will become clear, the overall algorithm intimately involves the 'remnant Courant number', Δc, defined by,

$$c=N+\Delta c \tag{17}$$

where the integer part of c is given by,

$$N=\text{INT}(c) \tag{18}$$

Computationally, the update cost is essentially independent of the value of N; thus, a large-Δt update costs the same as a small-Δt update having the same Δc value. Clearly, an explicit update with unrestricted Courant number can lead to substantial efficiencies, in comparison with schemes requiring $c<1$.

INTERPOLATION OF $\psi(x)$

In this section, several forms of interpolation are used to estimate $\psi(x)$. Some of these lead naturally to a number of well-known advection schemes for $|c| \leqslant 1$, and automatically generate their large-Δt extensions, as well. Thus, for example, piecewise linear interpolation of ψ leads to first-order upwinding[6] (and its large-Δt extension); piecewise parabolic interpolation of ψ leads to second-order methods such as those of Lax and Wendroff[10] or Leith[11], second-order upwinding[6], and Fromm's method[12] (together with their generalisations to arbitrarily large c), Piecewise cubic interpolation of ψ leads to a scheme equivalent to QUICKEST[13] and its large-Δt extension.

Continuing with higher-order piecewise interpolants leads to a hierarchy of advection schemes defined recursively. For $|c| \leqslant 1$, these are equivalent to the higher-order (unlimited) transient-interpolation-modelling schemes described in Reference 1, but NIRVANA also automatically gives the $|c| > 1$ generalisations, as well. Cubic-spline interpolation of ψ (corresponding to parabolic-spline interpolation of ϕ) is also considered. Cell-centred interpolants (e.g., the odd-order methods considered here) are independent of the direction of the convecting velocity (i.e., the sign of c), By definition, however, upwind- or downwind-biased interpolants depend on the sign of c. In this respect, one should distinguish between the 'natural upwinding' involved in the flux integral, (15), and the (upwind or downwind) bias in the sub-cell interpolation of ψ (and ϕ). Thus, for example, first-order upwinding is an 'upwind' method due to the natural upwinding; the sub-cell interpolation of ψ (and ϕ) in this case is cell-centred.

Except for the highly diffusive first-order method, none of the interpolations described is shape preserving; i.e., they may introduce extraneous extrema in ϕ (or, equivalently, extraneous inflection-points in ψ). In some cases—in particular, velocity-direction-dependent, even-order interpolations of ψ—serious numerical dispersion occurs, leading to disruptive unphysical oscillations. Odd-order velocity-direction-independent interpolation schemes are much less dispersive, but still generate small overshoots or undershoots near sudden changes in the gradient of ϕ.

This type of distortion can be eliminated by constructing simple local convexity-preservation constraints on the ψ interpolant, which amount to ensuring that there are no extraneous inflection-points. This turns out to be entirely equivalent to the universal limiter[1] for $c \leqslant 1$, but automatically extended to large c, as well. A somewhat more restrictive convexity constraint on ψ leads to the large-Δt generalisation of the TVD limiter[14].

In order to evaluate and compare a number of schemes based on various interpolations of ψ, a simple test problem has been set up. The set of test profiles is shown as an initial condition in *Figure 1*. The computational domain consists of a uniform grid of 150 mesh-widths. The initial profiles include a rectangular pulse, a single wavelength of a sine-squared profile, a semi-ellipse, and a triangle, all $20h$ wide at the base, together with a Gaussian pulse with standard deviation $\sigma = 2.5h$. These are shown by the dashed curves. Corresponding cell-average values are shown as horizontal lines—i.e. as a bar-graph or histogram. The flow is from left to right for positive c, and periodic boundary conditions (on ϕ) are used. Length and time scales are normalised so that the only parameters are the Courant number and the total number of time-steps. Since ψ is an integral variable, periodicity in ϕ implies a continual increase in ψ (equal to the total 'mass' of all profiles) for each 'wrap-around' of the domain. This is seen in the lower half of *Figure 1*, which also shows first-differences of $\bar{\phi}$ (= second-differences of ψ). Details of this, together with formulae for the initial conditions for $\bar{\phi}_i$ and ψ_i, and the exact interpolants, $\phi_{ex}(x)$ and $\psi_{ex}(x)$, are given in the Appendix.

Unsophisticated interpolation methods

As mentioned before, the crucial step in the NIRVANA procedure is the interpolation of the discrete ψ_i values to give $\psi(x)$. For pure, one-dimensional advection at constant velocity, this is

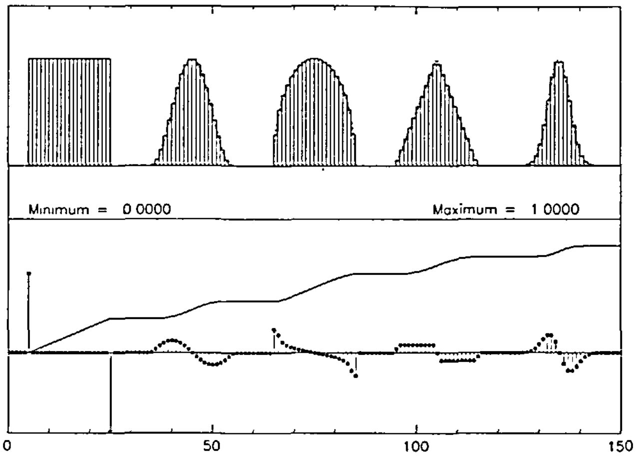

Figure 1 Initial conditions for the test problem. The top portion of the figure shows $\phi(x)$ and ϕ values; the lower portion shows $\psi(x)$ and discrete b values

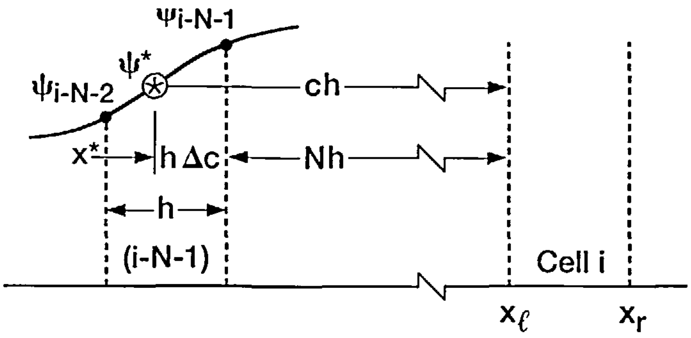

Figure 2 Definition of terms involved in the left-face flux calculation. Note that $c=N+\Delta c$

the *only approximation*; all other steps in the process are exact. In this sub-section, a number of simple interpolation schemes are considered; these lead to some well-known explicit advection schemes and their large-Δt extensions. *Figure 2* shows a schematic diagram of an interpolation of ψ in cell $(i-N-1)$; this is used in the evaluation of the left-face flux of cell i, according to (15). The interpolated value is denoted by $\psi^*=\psi(x^*)$, where $x^*=x_l-ch$ and $c=N+\Delta c$, according to (17) and (18).

The simplest possible scheme is shown in *Figure 3*. This is piecewise linear interpolation of ψ (which happens to be shape preserving—but, as will be seen, very inaccurate). In terms of the local spatial coordinate, $\xi=x-(x_{i-N-1}+h/2)$, the first-order formula for ψ, across cell $(i-N-1)$, is,

$$\psi^{(1C)}(\xi)=\psi_{i-N-1}+\left(\frac{\psi_{i-N-1}-\psi_{i-N-2}}{h}\right)\xi \tag{19}$$

or, equivalently, from (14) written for cell $(i-N-1)$,

$$\psi^{(1C)}(\xi)=\psi_{i-N-1}+\bar{\phi}_{i-N-1}\xi \tag{20}$$

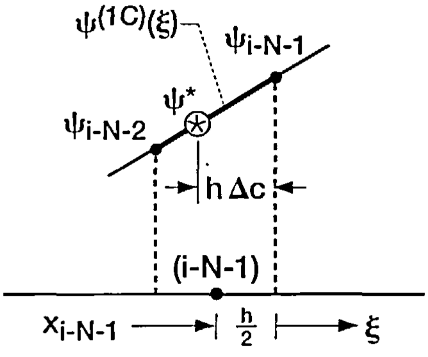

Figure 3 Piecewise linear interpolant across cell $(i-N-1)$

where the superscript indicates that the interpolation is cell-centred. This gives a corresponding piecewise constant interpolation for ϕ,

$$\phi^{(1C)}(\xi)=\frac{d\psi^{(1C)}}{d\xi}\equiv\bar{\phi}_{i-N-1} \tag{21}$$

across cell $(i-N-1)$. The interpolated ψ^* value is obtained by setting $\xi=-h\Delta c$, giving,

$$\psi^{*(1C)}=\psi^{(1C)}(-h\Delta c)=\psi_{i-N-1}-\bar{\phi}_{i-N-1}h\Delta c \tag{22}$$

Substitution into (15) gives, for the left-face flux,

$$c\phi_l^{(1C)}(i)=\frac{\psi_{i-1}-\psi_{i-N-1}+\bar{\phi}_{i-N-1}h\Delta c}{h} \tag{23}$$

which can be rewritten, using (10), as

$$c\phi_l^{(1C)}(i)=\bar{\phi}_{i-1}+\bar{\phi}_{i-2}+\dots+\bar{\phi}_{i-N}+\bar{\phi}_{i-N-1}\Delta c \tag{24}$$

And, by conservation, the right-face flux for cell i is,

$$c\phi_r^{(1C)}(i)=\bar{\phi}_i+\bar{\phi}_{i-1}+\dots+\bar{\phi}_{i-N+1}+\bar{\phi}_{i-N}\Delta c \tag{25}$$

This gives the first-order update formula,

$$\bar{\phi}_i^+(1C)=\bar{\phi}_i-\left(\sum_{p=1}^{i-N+1}\bar{\phi}_p+\bar{\phi}_{i-N}\Delta c\right)+\left(\sum_{p=i-1}^{i-N}\bar{\phi}_p+\bar{\phi}_{i-N-1}\Delta c\right) \tag{26}$$

equivalent to,

$$\bar{\phi}_i^+(1C)=\bar{\phi}_{i-N}-\Delta c(\bar{\phi}_{i-N}-\bar{\phi}_{i-N-1}) \tag{27}$$

in which the right-hand side can be interpreted as a first-order update of cell $(i-N)$ using the remnant Courant number, Δc. Of course, if $c<1$, then $N=0$ and $\Delta c=c$; this gives the well-known update formula for first-order upwinding[6]. Equations (24)–(27) have been written out in order to understand the basic structure of the scheme; the NIRVANA update itself is based entirely on (23) and (16).

Results of using the first-order method on the test problem are shown in *Figure 4*, after 100 time-steps. The Courant number is 10.2 ($N=10$, $\Delta c=0.2$). This means that the exact profiles would have been translated 1020 mesh-widths to the right; i.e., six times through the periodic domain, which is 150h in length, plus an additional 120h. The exact profiles are shown by the dashed lines. The computed values of ϕ are shown by horizontal bars within each cell; because

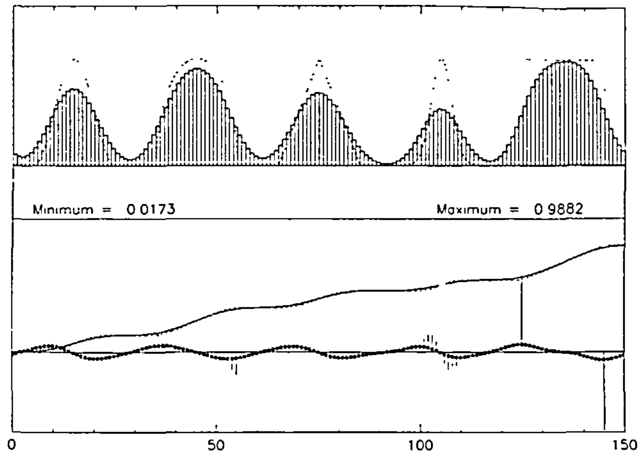

Figure 4 First-order results after 100 time-steps; $c=10.2$. In this case, there is no distinction between the histogram of computed ϕ values and the (piecewise constant) sub-cell interpolation. Note the difference between the exact (dashed) and computed (solid) ψ values

of the piecewise constant 'interpolation' in this case, these also represent $\phi(x)$. The most obvious feature of these results is the gross numerical diffusion, typical of the first-order method. Even after a small number of time-steps, the various profiles lose their individual features and degenerate into blobs resembling spreading Gaussians. As will be seen, the artificial diffusion depends on Δc. If c happens to be an integer ($\Delta c=0$), exact cell-to-cell transfer is obtained over N mesh-widths. This is true for all NIRVANA methods, since, in this case, $\psi^*\equiv\psi_{i-N-1}$, regardless of the interpolation formula.

The lower part of *Figure 4* shows the exact $\psi(x)$ (dashed), together with the piecewise linear interpolation of the computer ψ_i values (solid lines). For the higher-order methods discussed below, the computed values of ψ are almost indistinguishable from the exact values. The fact that there are obvious differences in this case is indicative of the first-order method's very poor performance. The remaining diagnostic in the figure (shown by heavy dots) represents the first-difference of computed $\bar\phi$ values (or second-difference of ψ values),

$$b_i=\frac{\bar\phi_{i+1}-\bar\phi_i}{h}=\frac{\psi_{i+1}-2\psi_i+\psi_{i-1}}{h^2} \tag{28}$$

Note that b_i (along with ψ_i) occurs at the right face of cell i. The small vertical lines represent the exact b_i values. This is a particularly sensitive diagnostic. As seen from the figure, the b_i values calculated from this first-order scheme bear almost no resemblance to the exact values.

If parabolic interpolation of ψ is used instead of the linear interpolation so far considered, there are two distinct cases, both leading to second-order methods. An additional second-order method is obtained by averaging the first two. *Figure 5* shows a second-order *downwind*-weighted interpolation; the formula (for positive c) is,

$$\psi^{(2D)}(\xi)=\psi_{i-N-1}+\left(\frac{\bar\phi_{i-N}+\bar\phi_{i-N-1}}{2}\right)\xi+\left(\frac{\bar\phi_{i-N}-\bar\phi_{i-N-1}}{2h}\right)\xi^2 \tag{29}$$

across cell $(i-N-1)$. From (13), the corresponding downwind-weighted linear interpolant for ϕ across cell $(i-N-1)$ is,

$$\phi^{(2D)}(\xi)=\left(\frac{\bar\phi_{i-N}+\bar\phi_{i-N-1}}{2}\right)+\left(\frac{\bar\phi_{i-N}-\bar\phi_{i-N-1}}{h}\right)\xi \tag{30}$$

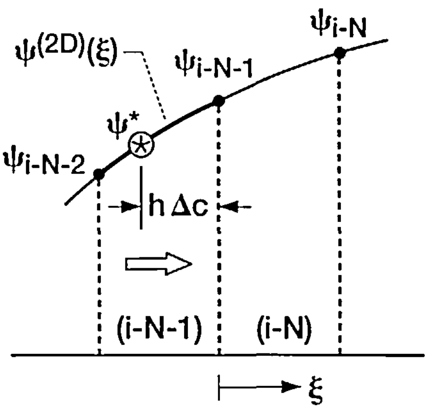

Figure 5 Downwind quadratic interpolation across cell $(i-N-1)$

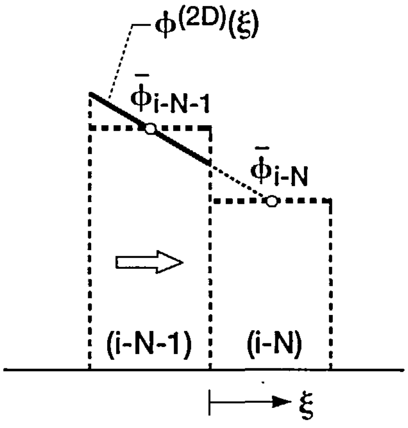

Figure 6 Downwind-weighted linear interpolation of ϕ

This is shown in *Figure 6*. The left-face flux is then obtained from (15) and (29), setting $\xi=-h\Delta c$. This gives,

$$c\phi_l^{(2D)}(i)=\sum_{p=i-1}^{i-N}\bar{\phi}_p-\frac{\Delta c}{2}(\bar{\phi}_{i-N}+\bar{\phi}_{i-N-1})+\frac{\Delta c^2}{2}(\bar{\phi}_{i-N}-\bar{\phi}_{i-N-1}) \tag{31}$$

Using the corresponding formula for the right-face flux, the update equation is equivalent to,

$$\bar{\phi}_i^+(2D)=\bar{\phi}_{i-N}-\frac{\Delta c}{2}(\bar{\phi}_{i-N+1}-\bar{\phi}_{i-N-1})+\frac{\Delta c^2}{2}(\bar{\phi}_{i-N+1}-2\bar{\phi}_{i-N}+\bar{\phi}_{i-N-1}) \tag{32}$$

This is the large-Δt extension of the Lax-Wendroff or Leith method, which is recovered in its standard form when $c<1$.

The results of the test problem (after 100 time-steps, with $c=10.2$) are shown in *Figure 7*. Now the (downwind-weighted, piecewise linear) sub-cell interpolation, $\phi(x)$, can be distinguished from the $\bar{\phi}$ values. In this case, the most obvious feature is the set of trailing numerical oscillations

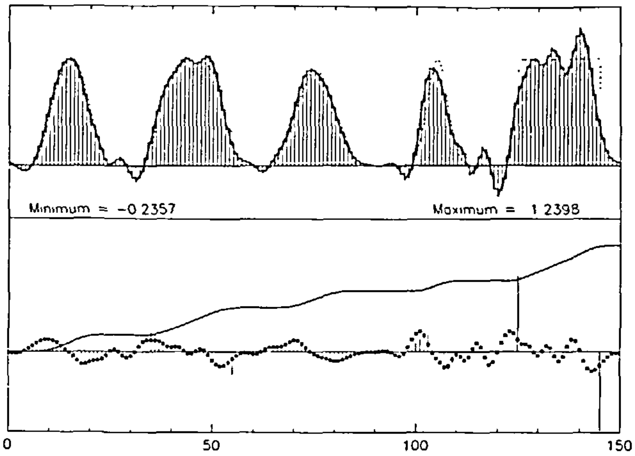

Figure 7 Downwind-weighted second-order results. In this case, the spiky sub-cell ϕ interpolant is clearly distinguishable from the cell-average h. Note the phase lag

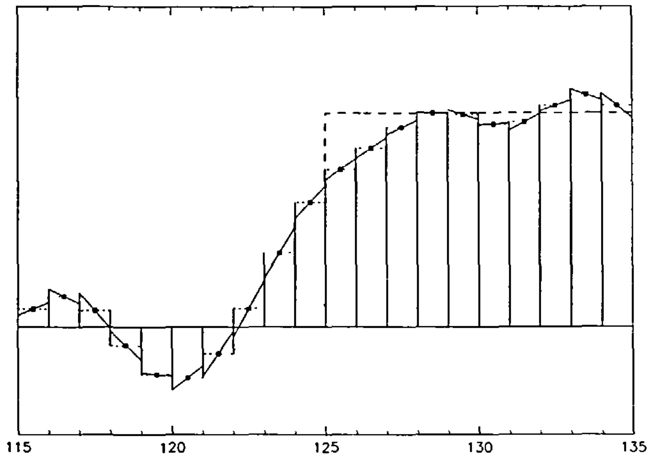

Figure 8 Close-up of the downwind-weighted second-order results in the vicinity of the step. Cell-average values are represented by dashed horizontal bars

(phase-*lag* dispersion), typical of this method. This is accentuated in the b_i data. In this case, numerical dispersion is a function of Δc. *Figure 8* shows a close-up of the oscillating behaviour in the region behind the trailing edge of the rectangular pulse. The spiky nature of the downwind-weighted sub-cell linear interpolation of ϕ is evident.

Now that the (downwind) second-order update algorithm is available, it is of interest to rewrite (27) for the first-order method in the form of (32), together with a 'correction' term. Specifically, (27) becomes,

$$\bar{\phi}_i^+(1C)=\bar{\phi}_{i-N}-\frac{\Delta c}{2}(\bar{\phi}_{i-N+1}-\bar{\phi}_{i-N-1})+\left[\frac{\Delta c^2}{2}+\frac{\Delta c(1-\Delta c)}{2}\right](\bar{\phi}_{i-N+1}-2\bar{\phi}_{i-N}+\bar{\phi}_{i-N-1}) \quad (33)$$

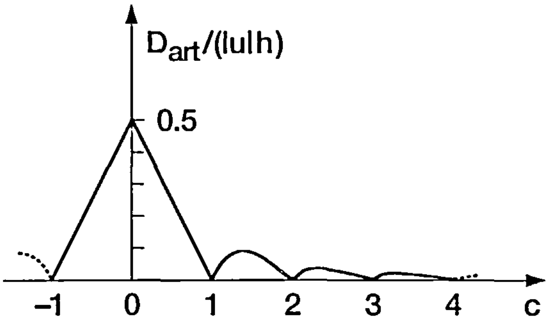

Figure 9 Dependence of first-order artificial diffusion on c for fixed u and h

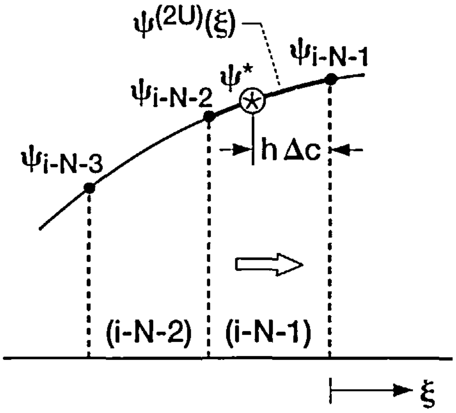

Figure 10 Upwind quadratic interpolant across cell $(i-N-1)$

The term $\Delta c(1-\Delta c)/2$ plays the role of an 'artificial diffusion' term. The artificial diffusion coefficient can be defined as,

$$D_{art}=\left(\frac{h^2}{2\Delta t}\right)\Delta c(1-\Delta c) \tag{34}$$

For given h and Δt, the right-hand side is a function of $u(=ch/\Delta t)$. In terms of c, this is a series of parabolic arcs. It is more instructive to rewrite this in the following way,

$$D_{art}=\left(\frac{|u|h}{2}\right)\left[\frac{\Delta c(1-\Delta c)}{|c|}\right] \tag{35}$$

For given u and h, the first factor on the right-hand side is fixed (this is the 'steady-state' artificial diffusion coefficient) and the term in square brackets is a function of c, or equivalently, $\Delta t(=ch/u)$. *Figure 9* shows the behaviour as a function of c. From this it should be clear that, for the first-order method, a larger c value corresponds to less overall distortion due to artificial diffusion[4]. In practice, however, other considerations may preclude the use of a very large Courant number. Thus, first-order methods should be avoided—especially as third- and higher-order methods (discussed below) are so much more accurate and cost-effective.

Another type of parabolic interpolaion of ψ involves an *upwind* bias. This is shown in *Figure 10* for positive c. Although the interpolation depends (only) on ψ_{i-N-1}, ψ_{i-N-2}, and ψ_{i-N-3}, it is

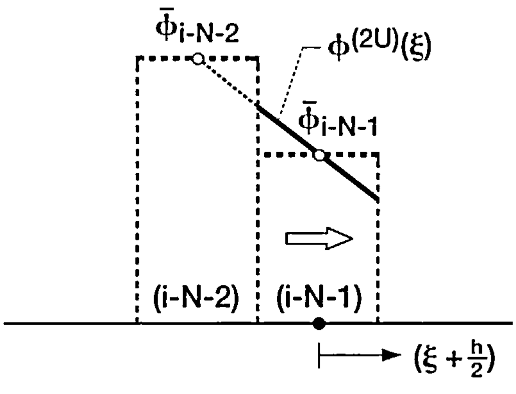

Figure 11 Upwind-weighted linear interpolation of ϕ

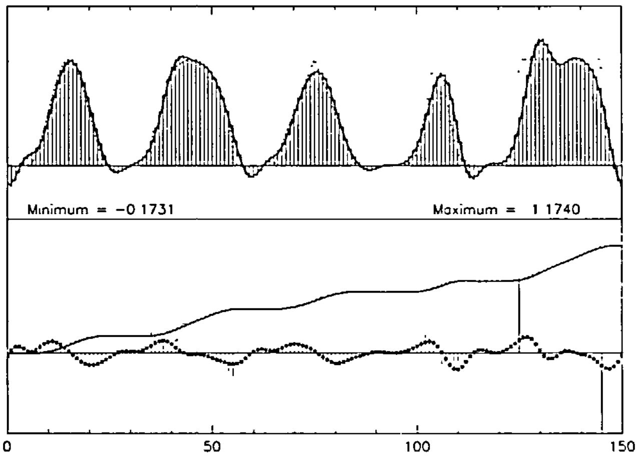

Figure 12 Upwind-weighted second-order results. Note the phase lead

more convenient to rewrite it as a modification of the second-order downwind formula. Across cell $(i-N-1)$, for positive c, this gives,

$$\psi^{(2U)}(\xi)=\psi^{(2D)}(\xi)-\left(\frac{\bar\phi_{i-N}-2\bar\phi_{i-N-1}+\bar\phi_{i-N-2}}{2}\right)\xi\left(1+\frac{\xi}{h}\right) \qquad (36)$$

The corresponding formula for ϕ is more conveniently written in terms of $(\xi+h/2)$, as,

$$\phi^{(2U)}(\xi)=\bar\phi_{i-N-1}+\left(\frac{\bar\phi_{i-N-1}-\bar\phi_{i-N-2}}{h}\right)\left(\xi+\frac{h}{2}\right) \qquad (37)$$

across cell $(i-N-1)$, clearly showing the upwind-biassed linear interpolation, as seen in *Figure 11*.

Results for this second-order upwind method are shown in *Figure 12*, using the usual parameter values. In this case, there is significant phase-*lead* numerical dispersion, manifested in the unphysical oscillations ahead of sharp features in the profiles. In particular, the computed b_i

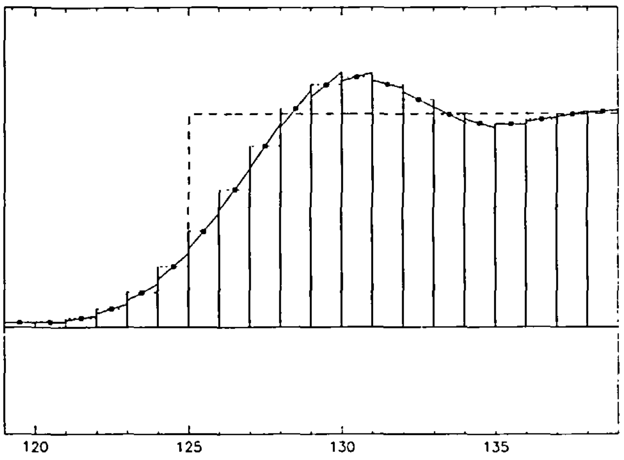

Figure 13 Close-up of the upwind-weighted second-order results in the vicinity of the step

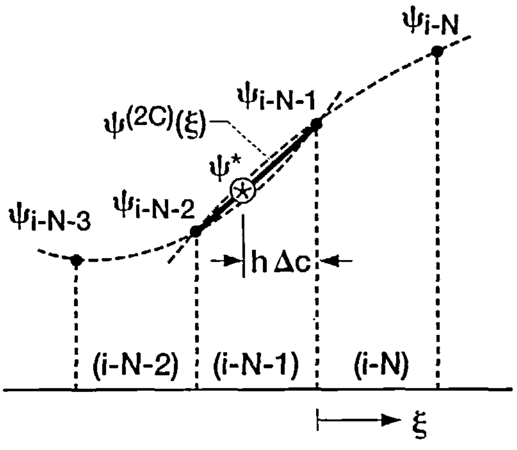

Figure 14 Cell-centred second-order ψ interpolation; average of two parabolas

values at many points bear little resemblance to the exact values. *Figure 13* shows a close-up of the behaviour ahead of the trailing-edge of the rectangular pulse. Compare this with *Figure 8*.

Fromm's method[12] was an attempt to reduce phase error by taking a simple arithmetic mean of the Lax-Wendroff (phase-lag) method and the second-order upwind (phase-lead) method, for $c < 1$. The same averaging can be used for $c > 1$, as well, giving the large-Δt generalisation of Fromm's method. From (29) and (36), this *cell-centred* second-order ψ interpolant takes the form,

$$\psi^{(2C)}(\xi) = \psi^{(2D)}(\xi) - \left(\frac{\bar{\phi}_{i-N} - 2\bar{\phi}_{i-N-1} + \bar{\phi}_{i-N-2}}{4}\right)\xi\left(1 + \frac{\xi}{h}\right) \qquad (38)$$

across cell $(i - N - 1)$. This is shown in *Figure 14*. The corresponding ϕ interpolant is again more

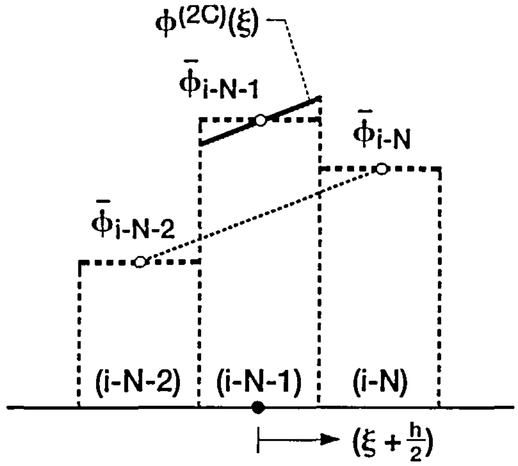

Figure 15 Cell-centred linear sub-cell interpolation of ϕ. The interpolant is parallel to the chord joining adjacent values spanning the cell

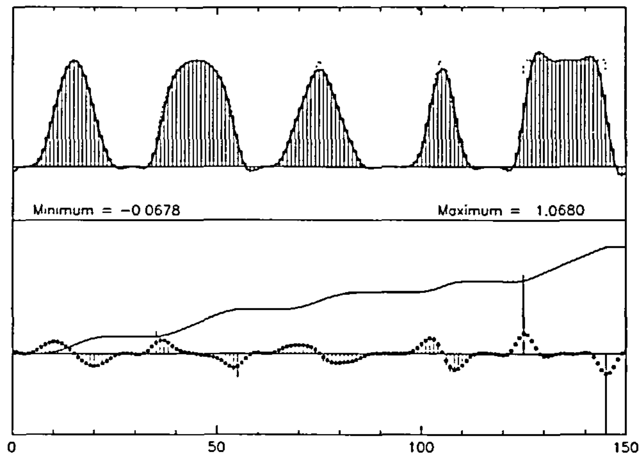

Figure 16 Cell-centred second-order results

conveniently written in terms of $(\xi+h/2)$, as,

$$\phi^{(2C)}(\xi)=\bar{\phi}_{i-N-1}+\left(\frac{\bar{\phi}_{i-N}-\bar{\phi}_{i-N-2}}{2h}\right)\left(\xi+\frac{h}{2}\right) \tag{39}$$

across cell $(i-N-1)$. This is shown in *Figure 15*. Note how both the ψ and the ϕ interpolants are centred with respect to cell $(i-N-1)$, and therefore independent of the sign of c (in contrast to the two previous methods).

Cell-centred interpolants lead to schemes with better phase behaviour than those based on upwind- or downwind-biassed interpolants. In fact, phase error of all schemes based on cell-centred interpolants is zero at half-interger values of c (i.e., $\Delta c=0.5$) as well as at interger

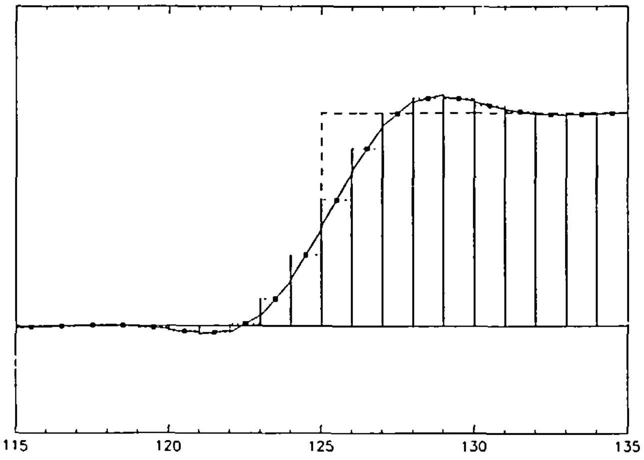

Figure 17 Close-up of the step region for the cell-centred second-order method

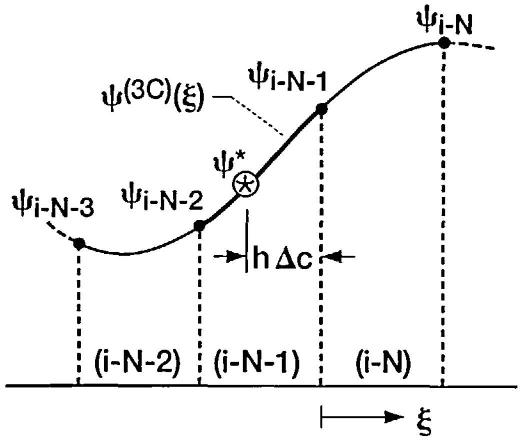

Figure 18 Cell-centred cubic interpolation of ψ

values. The improved phase behaviour is seen immediately in the cell-centred second-order case from the results of the usual test problem, shown in *Figure 16*. In contrast to the downwind- and upwind-weighted second-order methods (which show large regions of oscillatory behaviour), the cell-centred second-order method exhibits only relatively small overshoots and undershoots near regions involving sudden changes in gradient of ϕ (i.e., large b values). Note, particularly, the much better agreement, in general, between the computed and exact bs. *Figure 17* gives a close-up of the region near the step-down discontinuity.

Higher-order piecewise polynomial interpolation

Piecewise polynomial interpolation of ψ across each cell can be extended to higher and higher order, recursively, following a process analogous to that outlined in Reference 1. For example, a centred third-order ψ interpolant across cell $(i-N-1)$ is shown in *Figure 18*. The formula is

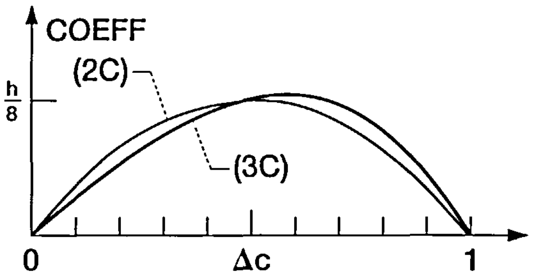

Figure 19 Comparison of COEFF(2C) and COEFF(3C)

again conveniently written as a modification of the downwind second-order method, as,

$$\psi^{(3C)}(\xi)=\psi^{(2D)}(\xi)-\left(\frac{\bar{\phi}_{i-N}-2\bar{\phi}_{i-N-1}+\bar{\phi}_{i-N-2}}{6}\right)\xi\left[1-\left(\frac{\xi}{h}\right)^2\right] \tag{40}$$

across cell $(i-N-1)$. Note the similarity in structure between this and the cell-centred second-order interpolant, (38). It is, therefore, interesting to compare the coefficient of the second-difference term, for $\xi=-h\Delta c$. For the cell-centred second-order interpolant,

$$\mathrm{COEFF(2C)}=\frac{\Delta c(1-\Delta c)}{4}h \tag{41}$$

And for the cell-centred third-order interpolant,

$$\mathrm{COEFF(3C)}=\frac{\Delta c(1-\Delta c^2)}{6}h \tag{42}$$

These are plotted in *Figure 19*; note that the difference between the two curves is always small and, in fact, the curves coincide at $\Delta c=0$, $\Delta c=0.5$, and $\Delta c\rightarrow 1$.

With piecewise parabolic interpolation of ϕ, a new phenomenon occurs. For (first- and) second-order methods, the (linear) ϕ interpolant equals $\bar{\phi}$ at the centre of the cell; i.e., the cell node value, $\phi(x_i)$, is the same as the cell average. At third order and above, this is no longer the case; the node value is not, in general, equal to the cell average,

$$\phi_i=\phi(x_i)\neq\bar{\phi}_i \tag{43}$$

Differentiating (40) and writing in terms of $(\xi+h/2)$, gives, across cell $(i-N-1)$,

$$\phi^{(3C)}(\xi)=\bar{\phi}_{i-N-1}-\frac{1}{24}(\bar{\phi}_{i-N}-2\bar{\phi}_{i-N-1}+\bar{\phi}_{i-N-2})+\left(\frac{\bar{\phi}_{i-N}-\bar{\phi}_{i-N-2}}{2h}\right)\left(\xi+\frac{h}{2}\right)$$
$$+\left(\frac{\bar{\phi}_{i-N}-2\bar{\phi}_{i-N-1}+\bar{\phi}_{i-N-2}}{2h^2}\right)\left(\xi+\frac{h}{2}\right)^2 \tag{44}$$

This is shown in *Figure 20*. The dashed curve is a parabola passing through $\bar{\phi}_{i-N-2}$, $\bar{\phi}_{i-N-1}$, and $\bar{\phi}_{i-N}$, at the centres of each cell. The vertical shift in the parabolic interpolant (heavy curve) is required to satisfy the cell-average constraint; but, of course, this results naturally from the collocation of ψ values. From (44), one sees that the node value at any cell i is related to neighbouring cell averages by,

$$\phi_i^{(3C)}=\phi^{(3C)}(\xi=-h/2)=\bar{\phi}_i-\frac{1}{24}(\bar{\phi}_{i+1}-2\bar{\phi}_i+\bar{\phi}_{i-1}) \tag{45}$$

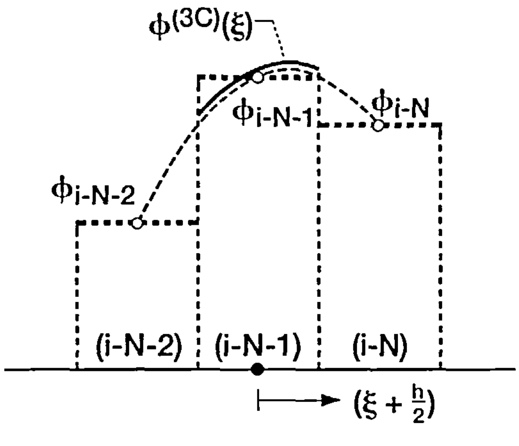

Figure 20 Quadratic sub-cell interpolation of ϕ

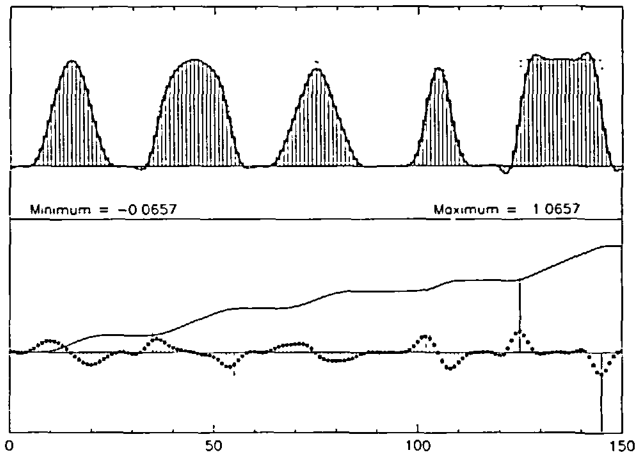

Figure 21 Cell-centred third-order results

Third-order results are shown in *Figure 21*. As perhaps expected from the velocity-direction-independent interpolation, phase behaviour is very good. Numerical distortion consists primarily of small overshoots and undershoots in the vicinity of large b values. A close-up view of the step-down discontinuity region, *Figure 22*, shows the relatively smooth behaviour of the sub-cell interpolants. But, of course, small discontinuities in the value of ϕ (and the slope of ψ) occur at cell faces; this is true of all piecewise interpolations, although the discrepancies become smaller and smaller as the order is increased. For $c < 1$, the third-order update algorithm is identical in structure to the QUICKEST scheme[13], except that QUICKEST uses node values rather than cell averages. However, it should be clear that if one applies the update algorithm to each of the terms on the right-hand side of (45), then (because of the linearity) the same algorithm applies to ϕ_i, as well.

Higher order piecewise polynomial methods can easily be constructed. The results are identical to those of Reference 1, except that c is replaced by Δc, and node values by cell averages.

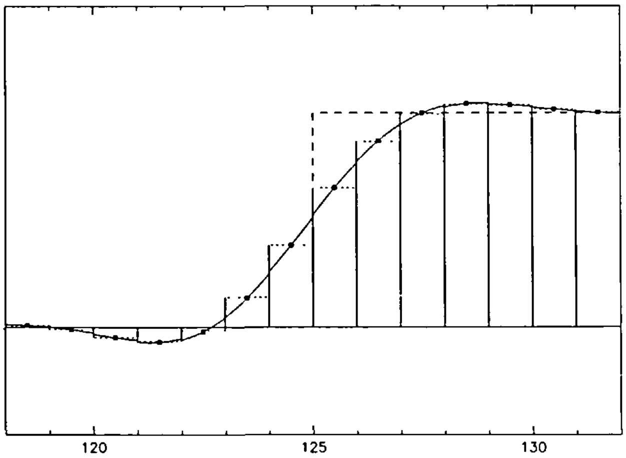

Figure 22 Close-up of the third-order results in the vicinity of the step

Cubic-spline interpolation of ψ

A cubic-spline interpolation of ψ corresponds to a parabolic-spline interpolation of ϕ. Because of the way in which cubic splines are constructed—matching value, gradient, and curvature at both ends of each cell interval[15]—the interpolation of ψ is inherently cell-centred. This means that the parabolic-spline interpolation of ϕ is cell-centred, as well. Remember that $\phi(x)$ is a sub-cell interpolation of cell-average data, satisfying (6). In the more common procedure of interpolating point data, parabolic splines have a directional bias[15]; i.e., a left-to-right sweep in general gives a different answer from a right-to-left sweep.

Usually, a cubic-spline interpolation requires the solution of a tridiagonal matrix equation stemming from the matching-conditions across nodes. In the case of periodic boundary conditions, the matrix has additional elements in the top-right and bottom-left corners. This 'periodic-tridiagonal' system can be easily solved using a modification of the Thomas algorithm[16]. In the present case, there is a slight complication with the 'periodic' nature of ψ. In fact, ψ is not strictly periodic (since it must increase across the domain by a constant equal to the total 'mass' of all profiles); however, its first- and second-derivatives are periodic, and these are the quantities involved in the matrix structure. More details of this and the related 'reach-back' problem—when cell $(i-N-1)$ is outside of the computational domain (so that ψ^* must be adjusted by a multiple of the total 'mass')—are sketched in the Appendix.

The cubic-spline results are shown in *Figures 23* and *24*. Although this is formally only a third-order method, the overall features of the simulation appear to be considerably better than the piecewise third-order method, *Figure 21*. Phase accuracy and smooth-function resolution are excellent. However, as with all (cell-centred) higher-order polynomial methods, a few oscillations occur in the vicinity of large b values. Note, from the close-up of *Figure 24*, the continuity in both value and gradient in the parabolic-spline interpolation of ϕ.

Shape-preserving constraints

Piecewise polynomial or spline interpolation schemes do not, in general, guarantee convexity preservation of $\psi(x)$. This means that extraneous inflection-points in ψ (or, equivalently, extraneous extrema in ϕ) may occur, thereby exciting unphysical numerical oscillations—which grow and disperse as the profile evolves. *Figure 25* shows two alternative interpolations through

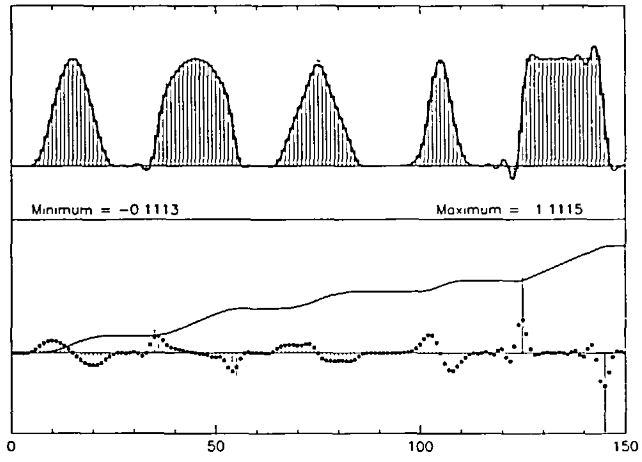

Figure 23 Cubic-spline results

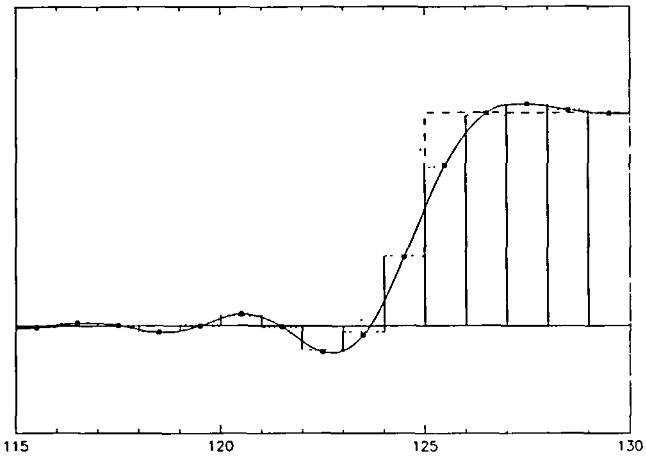

Figure 24 Close-up of cubic-spline results

a set of four ψ values in the vicinity of cell $(i-N-1)$. Note that the discrete second-differences of ψ at each side of the cell, b_{i-N-2} and b_{i-N-1}, are both positive in the case shown; convexity preservation means that the curvature of the interpolant should also be positive throughout cell $(i-N-1)$. An inflection-point in the ψ interpolant within a cell is allowed only if there is a change in sign between b_{i-N-2} and b_{i-N-1}, across cell $(i-N-1)$. *Figure 25a* depicts an extraneous inflection-point (EIP), whereas *25b* shows convexity preservation.

Figure 26 shows two different, very simple, geometric constraints to enforce convexity locally, across cell $(i-N-1)$. In part *a*, the interpolant is constrained to lie within the small triangular region bounded by the chord joining ψ values across the cell and the two tangents joining ψ values across the adjacent cells. In part *b* of the figure, the lower constraint is a parabola, tangent to (at least) one of the constraints in part *a*, but not violating the other. *Figure 27* defines some notation for analysing these constraints. For convenience, j replaces $(i-N-1)$. Points 1, 2, and 3

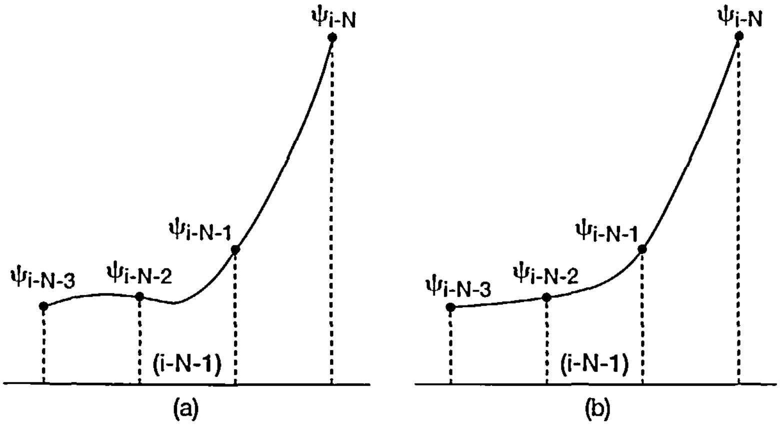

Figure 25 Two different interpolations through the same set of four ψ points. (a) Extraneous inflection-point; (b) convexity preservation

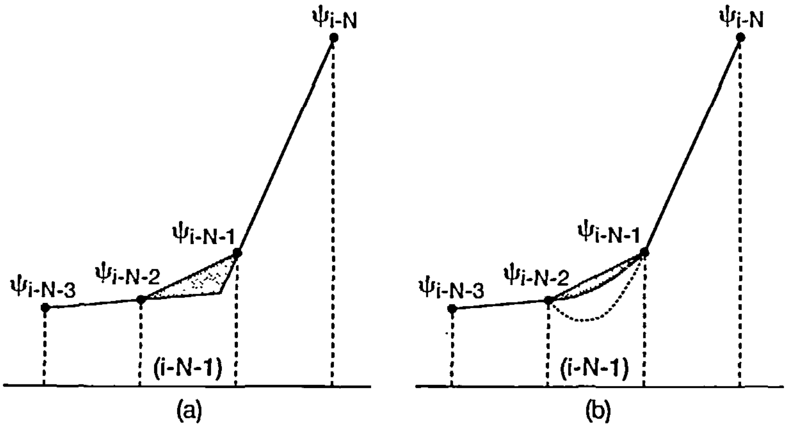

Figure 26 Geometric constraints for local convexity preservation. (a) Triangular region; (b) parabolic sector

represent possible ψ^* values on the respective curves, a distance $h\Delta c$ to the left (upstream for $c>0$) of the right face of cell j. Corresponding 'reference' ϕ values within cell j are defined by,

$$\phi_R=\frac{\psi_j-\psi^*}{h\Delta c} \tag{46}$$

At point 1, in both cases, the reference value is,

$$\phi_{R1}=\frac{\psi_j-\psi_1^*}{h\Delta c}=\frac{\psi_j-\psi_{j-1}}{h}=\bar{\phi}_j \tag{47}$$

At point 2 in *Figure 27a*,

$$\phi_{R2}^{(a)}=\frac{\psi_j-\psi_2^*}{h\Delta c}=\frac{\psi_j-[\psi_{j-1}+\bar{\phi}_{j-1}(h-h\Delta c)]}{h\Delta c} \tag{48}$$

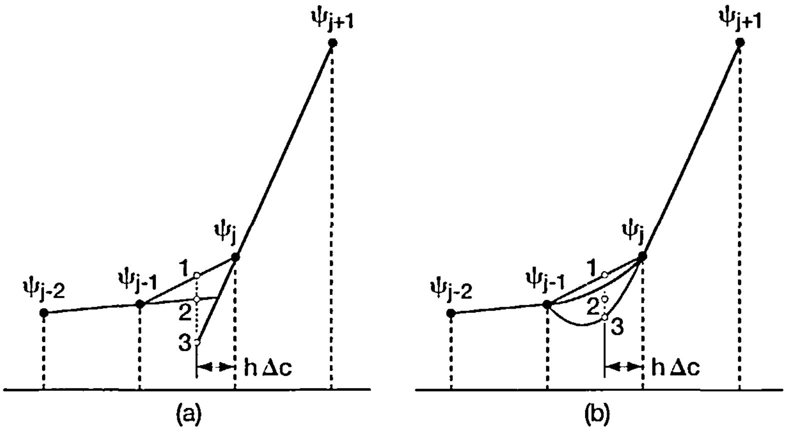

Figure 27 Notation for analysing the limiters. (a) Universal limiter; (b) TVD limiter

or, equivalently,

$$\phi_{R2}^{(a)}=\bar{\phi}_{j-1}+\frac{1}{\Delta c}(\bar{\phi}_j-\bar{\phi}_{j-1}) \tag{49}$$

And at point 3 in *Figure 27a,*

$$\phi_{R3}^{(a)}=\frac{\psi_j-\psi_3^*}{h\Delta c}=\frac{\psi_{j+1}-\psi_j}{h}=\bar{\phi}_{j+1} \tag{50}$$

At point 2 in *Figure 27b*, it is not hard to show that,

$$\phi_{R2}^{(b)}=(1-\Delta c)\phi_{R1}+\Delta c\phi_{R2}^{(a)} \tag{51}$$

whereas, at point 3,

$$\phi_{R3}^{(b)}=\Delta c\phi_{R1}+(1-\Delta c)\phi_{R3}^{(a)} \tag{52}$$

or, in other words, the parabolic constraints in part b are Δc-weighted averages of the constraints in part a.

It is instructive to rewrite these constraint equations in terms of normalised variables[1]. Define a normalised variable with respect to cell j $(=i-N-1)$ as,

$$\tilde{\phi}=\frac{\phi-\bar{\phi}_{j-1}}{\bar{\phi}_{j+1}-\bar{\phi}_{j-1}} \tag{53}$$

Then the normalised reference values in part a of *Figure 28* can be written,

$$\tilde{\phi}_{R1}=\frac{\bar{\phi}_j-\bar{\phi}_{j-1}}{\bar{\phi}_{j+1}-\bar{\phi}_{j-1}}=\tilde{\bar{\phi}}_j \tag{54}$$

$$\tilde{\phi}_{R2}^{(a)}=\frac{1}{\Delta c}\tilde{\bar{\phi}}_j \tag{55}$$

and

$$\tilde{\phi}_{R3}=\tilde{\bar{\phi}}_{j+1}=1 \tag{56}$$

These may be recognised as the universal limiter constraints described in Reference 1 (using $\bar{\phi}$

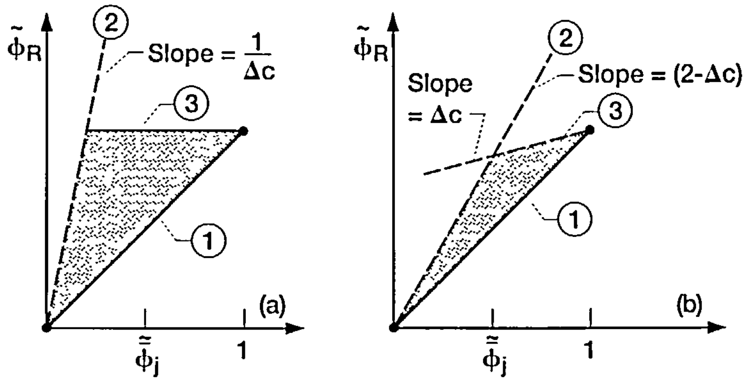

Figure 28 Normalised variable diagrams. (a) Universal limiter; (b) TVD limiter

values instead of ϕ values)—with c replaced by Δc in (55). For part b of the figure, the normalised values are,

$$\tilde{\phi}_{R2}^{(b)} = (1-\Delta c)\tilde{\bar{\phi}}_j + \Delta c \tilde{\phi}_{R2}^{(a)} \tag{57}$$

giving, from (49) and (54),

$$\tilde{\phi}_{R2}^{(b)} = (2-\Delta c)\tilde{\bar{\phi}}_j \tag{58}$$

whereas,

$$\tilde{\phi}_{R3}^{(b)} = \Delta c \tilde{\bar{\phi}}_j + (1-\Delta c) \tag{59}$$

or

$$\tilde{\phi}_{R3}^{(b)} = 1 + \Delta c(\tilde{\bar{\phi}}_j - 1) \tag{60}$$

These are, in fact, the so-called TVD limiter constraints[14], extended to large Courant number and written in terms of normalised variables[17].

Both sets of constraints are summarised in normalised-variable diagrams in *Figures 28a* and *b*, respectively. In these diagrams, the ordinate is $\tilde{\phi}_l^{\#}(i-N)$, normalised with respect to cell $(i-N-1)$, where $\phi_l^{\#}(i-N)$ is defined as follows:

$$c\phi_l(i) = \sum_{p=i-1}^{i-N} \bar{\phi}_p + \Delta c \phi_l^{\#}(i-N) \tag{61}$$

the notation implying that $\phi_l^{\#}(i-N)$ is what the left-face value of cell $(i-N)$ would be if the Courant number were Δc rather than c.

The convexity-preserving constraints of *Figure 26* or, equivalently, the normalised-variable diagrams of *Figure 28* are appropriate only in cases where b_{i-N-1} and b_{i-N-2} are both of the same sign. This is the so-called locally monotonic region,

$$0 < \tilde{\bar{\phi}}_{i-N-1} < 1 \tag{62}$$

If one of the b values is zero or (more importantly) if they change sign across the cell (representing an inflection-point in ψ, i.e., a local extremum in ϕ), special care needs to be taken to avoid 'clipping'. In this case, either

$$\tilde{\bar{\phi}}_{i-N-1} < 0 \tag{63}$$

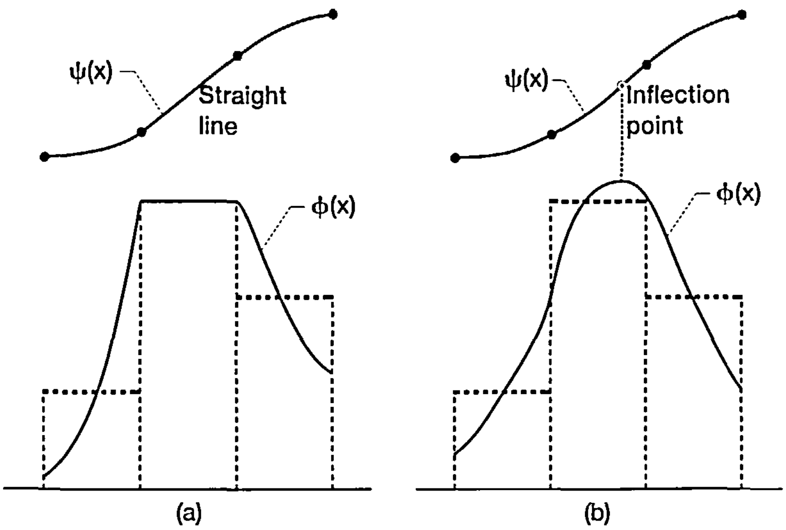

Figure 29 Treatment of local extrema. (a) Clipping; (b) resolution of the inflection-point in ψ

or

$$\bar{\bar{\phi}}_{i-N-1} > 1 \tag{64}$$

Figure 29 shows two possibilities for a cell across which b changes sign. In part a, linear interpolation is used for ψ across the cell. As seen, this leads to a clipped flat top in the ϕ profile; i.e., the maximum local ϕ value can never exceed the local maximum $\bar{\phi}$. This is equivalent to using the first-order scheme within the extremum cell—a common feature of most flux-limiting schemes. By contrast, part b of the figure shows a more realistic representation of the (real) inflection-point in ψ that must occur within the cell. This gives good peak resolution in the ϕ profile. A simple pattern-recognition strategy that resolves real inflection-points in ψ while maintaining monotonic transitions in ϕ, when necessary, is outlined in the appendix.

For reference, *Figure 30* shows results of the test problem using the so-called MUSCL scheme[18], extended to large Δt. In terms of the present analysis, this is the cell-centred second-order method (the large-Δt extension of Fromm's method), (38), with the TVD limiter applied in locally monotonic regions and (the large-c version of) first-order upwinding in non-monotonic regions. Although step-resolution is monotonic, the close-up view in *Figure 31* shows that serious clipping (and concomitant spreading) occurs in the vicinity of the narrow Gaussian peak, since (in keeping with traditional TVD strategy) the inflection-point-resolution algorithm is not activated in this case. Some clipping of other profiles is also noticeable in *Figure 30*.

Figures 32 and *33* show results for the cubic-spline scheme, using the universal limiter of *Figure 36a* applied to ψ, together with the inflection-point-resolution algorithm. Step resolution is relatively sharp and monotonic. Moreover, this is achieved without corrupting the good peak-resolution properties of the cubic-spline scheme. Even sharper—in fact, *exact*—sub-cell resolution of discontinuities can be achieved by a process called automatic compressive enhancement (ACE). Roughly speaking, the ACE strategy consists of a weighting scheme that 'pulls' the interpolation toward limiter R2 or R3 (depending on Δc) in cells containing a near-discontinuity (as determined by a simple pattern recognition algorithm, described in the Appendix). *Figure 34* shows results obtained when this additional feature is applied to the cubic-spline base scheme. Results are shown at 97 Δt with $c = 10.2$, so that the discontinuity occurs within a cell (rather than at a face). Details of the step region are shown in *Figure 35*.

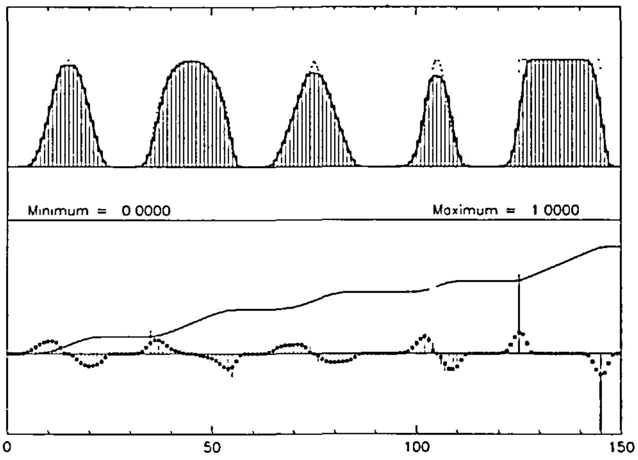

Figure 30 Results using the large-Δt extension of MUSCL

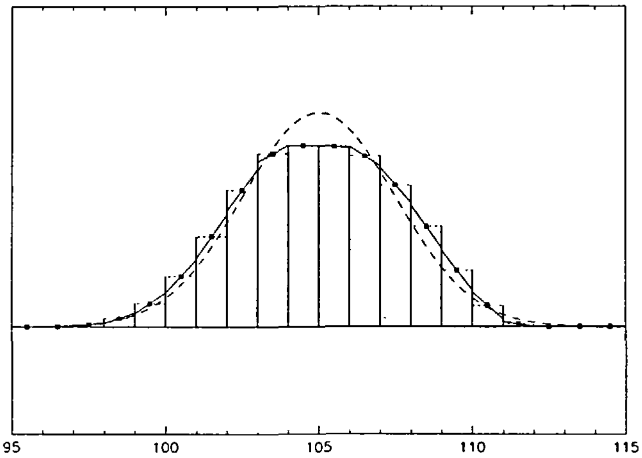

Figure 31 Clipping of the Gaussian peak using MUSCL

These results should be compared with those of other 'super-compressive' schemes, such as Superbee[14], Super-C[1], and Hyper-C[1], where indiscriminate use of compressive limiters leads to strong clipping and 'stair-casing'.

STABILITY ANALYSIS

For pure convection, the exact one-dimensional complex amplitude ratio[19] is,

$$G_{exact} = \frac{\bar{\phi}_i^+}{\bar{\phi}_i} = \exp(\imath c\theta) = \exp(-\imath N\theta)\exp(-\imath \Delta c\theta) \qquad (65)$$

where θ is the nondimensional wavenumber and $\imath = \sqrt{-1}$. Expanding the second factor in a

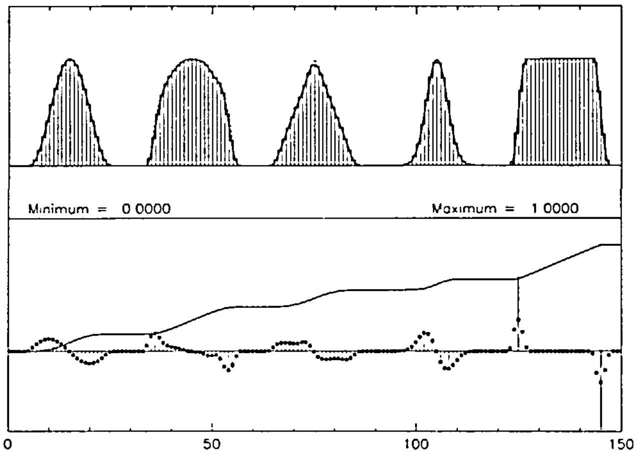

Figure 32 Results after 100 time-steps using the cubic-spline scheme with the universal limiter and automatic inflection-point resolution

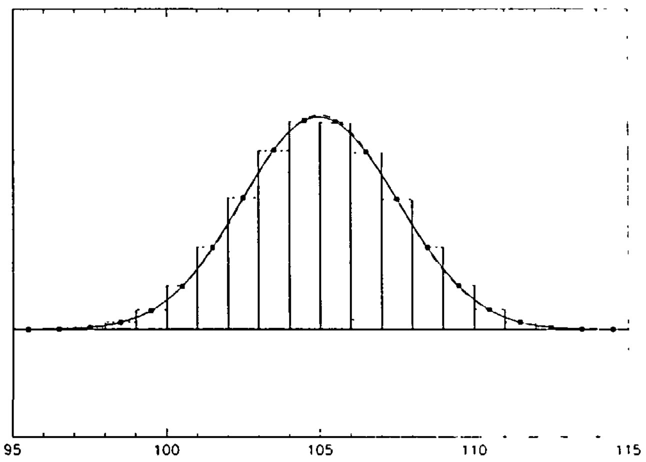

Figure 33 Close-up of the Gaussian peak in *Figure 32*

Taylor series gives,

$$G_{exact}=\exp(-iN\theta)\left\{1-\frac{\Delta c^2}{2}\theta^2+O(\theta^4)-i\left[\Delta c\theta-\frac{\Delta c^3}{6}\theta^3+O(\theta^5)\right]\right\} \tag{66}$$

If a numerical G matches this expression through all terms up to (and including) θ^N, it is an Nth-order method, in both space and time. The scheme is stable provided that $|G|\leqslant 1$ for all θ $(0\leqslant\theta\leqslant\pi)$.

As an example, consider the large-Δt downwind second-order method (the extension of the Lax-Wendroff method). The update equation, (32), is rewritten here for convenience,

$$\bar{\phi}_i^+=\bar{\phi}_{i-N}-\frac{\Delta c}{2}(\bar{\phi}_{i-N+1}-\bar{\phi}_{i-N-1})+\frac{\Delta c^2}{2}(\bar{\phi}_{i-N+1}-2\bar{\phi}_{i-N}+\bar{\phi}_{i-N-1}) \tag{67}$$

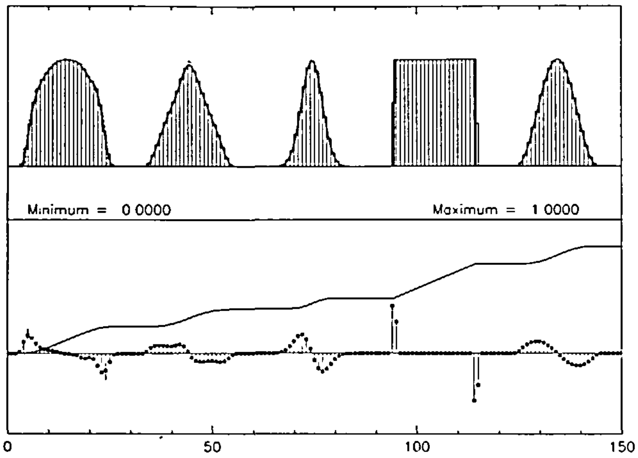

Figure 34 Results after 97 time-steps using the cubic-spline scheme with the universal limiter, automatic inflection-point resolution, and automatic compressive enhancement

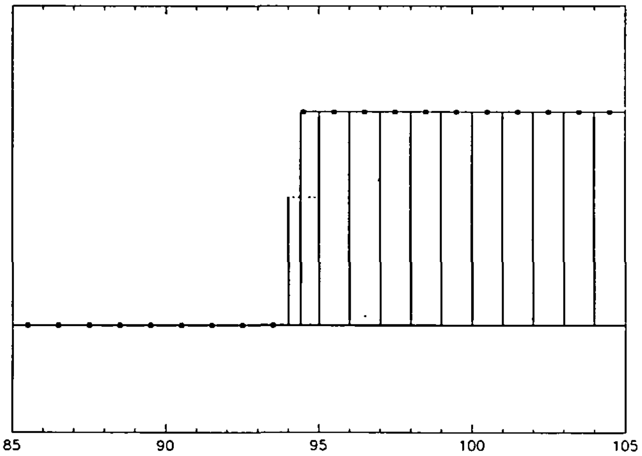

Figure 35 Close up of the step region in *Figure 34*. Note the (exact) sub-cell discontinuity

The corresponding complex amplitude ratio is,

$$G^{(2D)} = \exp(-iN\theta) - \frac{\Delta c}{2}\{\exp[-i(N-1)\theta] - \exp[-i(N+1)\theta]\}$$
$$+\frac{\Delta c^2}{2}\{\exp[-i(N-1)\theta] - 2\exp(-iN\theta) + \exp[-i(N+1)\theta]\} \tag{68}$$

This can be factored into,

$$G^{(2D)} = \exp(-iN\theta)\left\{1 - \frac{\Delta c}{2}[\exp(i\theta) - \exp(-i\theta)] + \frac{\Delta c^2}{2}[\exp(i\theta) - 2 + \exp(-i\theta)]\right\} \tag{69}$$

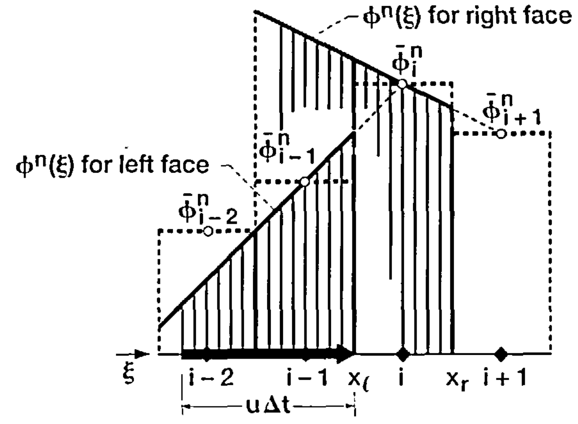

Figure 36 Exceeding the range of validity of piecewise interpolants

which can be written,

$$G^{(2D)} = \exp(-\imath N\theta) G_{LW}(\Delta c) \tag{70}$$

where the notation implies that the term in curly brackets in (69) is the (standard) Lax-Wendroff G—but for Δc rather than c. Since it is well known that,

$$|G_{LW}(c)| \leqslant 1 \qquad \text{for } c < 1 \tag{71}$$

it is clear that $|G_{LW}(\Delta c)| \leqslant 1$, because Δc is less than unity, by definition. The large-Δt second-order downwind scheme is thus stable for any value of c.

In general, one-dimensional, constant-u, constant-h NIRVANA schemes are always equivalent to an update of the form,

$$\bar{\phi}_i^+ = \bar{\phi}_{i-N} - \Delta c[\phi_i^\#(i-N+1) - \phi_i^\#(i-N)] \tag{72}$$

using the notation of (61). This is equivalent to an update of cell $(i-N)$ using the remnant Courant number, Δc, together with a translation across N mesh-widths. The large-time-step G can then be written in terms of the small-time-step G as follows

$$G_{LTS}(c) = \exp(-\imath N\theta) G_{STS}(\Delta c) \tag{73}$$

So, if the small-time-step G is stable (for $\Delta c < 1$), the corresponding large-time-step G is stable for any value of c.

This is the appropriate manifestation of the CFL condition; since the discrete solution is known everywhere at the earlier time-level, there should be no CFL *stability* constraint on the time-step. The reason that many well-known explicit schemes need to satisfy rather stringent Courant-number constraints (often $c \leqslant 1$, or, in the case of second-order upwinding, $c \leqslant 2$) has nothing to do with the CFL condition. Rather, it stems from a *range restriction* on the validity of the local piecewise interpolation used. The restriction may be stated as follows: in any cell in which the interplant is used, it must satisfy the cell-average constraint, (6), for that cell. As an example of a local interpolant extended beyond its valid range, *Figure 36* shows Lax-Wendroff, downwind-weighted interpolants for the two faces of cell i. For the situation shown, i.e.,

$$uBt > \Delta x \tag{74}$$

values of the right-face ϕ interpolant within cell $(i-1)$ are used to extend the flux integral calculation. Clearly these values do not satisfy the cell-average constraint for cell $(i-1)$, and consequently, the right-face flux is (in this case) far too large. Similarly, values of the left-face interpolant within cell $(i-2)$ do not satisfy the cell-average constraint for that cell, and as a result, the left-face flux is (in this case) too small. This quickly leads to numerical divergence.

If only the local information is used in the definition of the piecewise interpolant, then, clearly from *Figure 36*, in this case the *range of validity of the interpolation* must be retricted to,

$$|u|\Delta t \leqslant \Delta x \tag{75}$$

or, equivalently, $|c| \leqslant 1$. But, of course, there is no need to take such a 'short-sighted' view of the interpolation. If one takes a global view of the piecewise interpolation (using different local interpolation equations over each cell, each satisfying the cell-average condition), there is no Courant-number constraint. This is immediately obvious in the case of global interpolants, such as splines. In all cases, the CFL condition is irrelevant, since the numerical domain of dependence is the whole field.

Returning attention to (73), it should be clear that the order of accuracy of a given NIRVANA scheme can be determined from the Taylor expansion of G_{STS}, since the left factor is an exact expression, representing the translation across N mesh-widths. It is not hard to show that Nth-order (piecewise or spline) interpolation of ψ leads to an Nth-order accurate scheme[19] (in both space and time), matching the Taylor expansion of G_{STS} through all terms up to (and including) θ^N.

GENERALISATIONS

The NIRVANA scheme has been described here for the simplest possible conditions—one-dimensional pure advection at constant velocity over a uniform mesh. Although this analysis should be of theoretical interest, several generalisations will need to be made in order to construct a practical algorithm for applied CFD. This section sketches some initial ideas for such generalisations; however, no attempt is made to demonstrate the validity of these proposals here. Practical applications will be reported at a later date.

The most straight-forward generalisation is to the use of a variable one-dimensional mesh. Two things need to be changed. First, the definition of ψ_i given by (10) is easily generalised[8] to include the variable mesh-width inside the summation sign,

$$\psi_i = \psi_0 + \sum_{p=1}^{i} h_p \bar{\phi}_p \tag{76}$$

Secondly, interpolation formulae for $\psi(x)$ will, of course, involve the variable mesh size. This is easily accomplished using Lagrange or Newton interpolants in the piecewise polynomial cases; spline interpolants automatically include a variable mesh in their basic formulation[15]. The universal limiter (or TVD) convexity-preservation constraints require minor modification, although the geometric structure (defined in *Figure 26*, for example) remains the same. Note that, once $\psi(x)$ is found from the interpolation (globally), then ψ^* in (15) for the determination of the (left-face) flux is given by,

$$\psi^* = \psi(x^*) = \psi(x_l - u\Delta t) \tag{77}$$

where x_l is the coordinate the face in question. This is, in fact, the simplest way of computing the constant-h value of ψ^*, as well. Note that N is defined implicitly by the inequalities

$$\sum_{p=i-1}^{i-N-1} h_p < u\Delta t < \sum_{p=i-1}^{i-N} h_p \tag{78}$$

The scheme can also be generalised to handle some forms of spatially varying known advecting velocity fields in one dimension. More specifically, the situation in which the velocity is

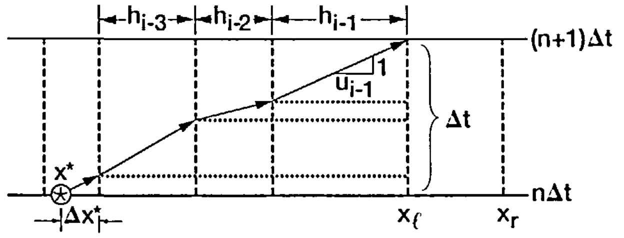

Figure 37 An (x,t) diagram for nonuniform velocity and nonuniform grid

approximated to be piecewise constant over each cell and each time-interval is easily handled[4]. *Figure 37* shows an (x,t) characteristic corresponding to a steady velocity field that is piecewise constant in space (the ideas sketched here cân be extended to curved, simple characteristics—at added expanse). The problem is to determine the departure point, x^*, of the characteristic that just reaches the left face of cell i at time-level $(n+1)$. Once this is known, the advection flux is the intergral of everything swept into the face, from x^* to x_ℓ. From the diagram,

$$\Delta t = \sum_{p=i-1}^{i-N} \left(\frac{h_p}{u_p}\right) + \frac{\Delta x^*}{u_{i-N-1}} \tag{79}$$

or

$$\Delta x^* = u_{i-N-1}\Delta t\left[1 - \sum_{p=i-1}^{i-N} \frac{1}{c_p}\right] \geqslant 0 \tag{80}$$

where $c_p = u_p\Delta t/h_p$ and N is determined from the generalisation of (78),

$$\sum_{p=i-1}^{i-N-1} \frac{1}{c_p} < 1 < \sum_{p=i-1}^{i-N} \frac{1}{c_p} \tag{81}$$

Once x^* is found, the face-values and update can be computed in the usual way.

Nonlinear one-dimensional scalar problems and systems are significantly more difficult—especially in cases where characteristics converge to form shocks. Flux calculations will require a more sophisticated prediction step for local (approximate) characteristics, analogous to the so-called Riemann-solvers used in compressible aerodynamics codes[7].

Multidimensional problems are often handled by the simple expedient of applying a one-dimensional operator successively in each coordinate direction. This could certainly be done with the one-dimensional NIRVANA scheme. However, locally one-dimensional time-splitting of this kind often leads to other errors[20] For example, in a steady, solenoidal, spatially varying advection field, an initially constant scalar does not remain constant! Nevertheless, initial tests using the 1D NIRVANA scheme in a time-split mode suggest that this could be developed into a very powerful technique for nonlinear, multidimensional, advection-dominated problems.

A more interesting (and challenging) procedure is sketched in *Figure 38*, showing the calculation of flux through a single face of a rectangular two-dimensional control-volume cell for a prescribed (time-invariant) velocity field. In principle, this is a conservative, flux-based formulation. The advective flux is given by the spatial *flux integral*.

$$FLUX = \oiint_{\substack{\textit{advection}\\ \textit{area}}} \phi(x,y)\,\mathrm{d}x\,\mathrm{d}y \tag{82}$$

where the 'advection area' is that covered by the streamlines shown in the diagram. There are

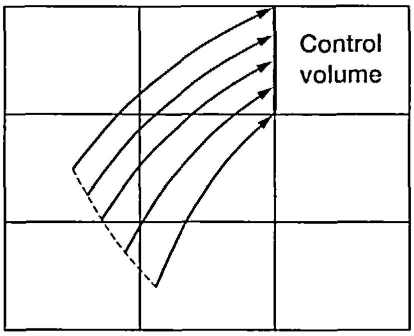

Figure 38 Two-dimensional flux-integral calculation

two problems. First is the problem of calculating the flux-integral in an accurate and inexpensive way. Even if the streamlines' shape and extent were known exactly, this is a nontrivial task. The second (and even more difficult) problem is to estimate $\phi(x,y)$ from the (known) cell-average data—in an *accurate* and *shape-preserving* manner.

In one dimension, both of these problems are solved by the introduction of the integral variable, ψ. One could try a similar strategy in two dimensions[8]. For example, on a uniform square grid of size $(i_{max} \times j_{max})$, define,

$$\psi_{i,j} = \psi_0 + h^2 \sum_{q=1}^{j} \sum_{p=1}^{r} \bar{\phi}_{p,q} \tag{83}$$

where

$$r = i_{max} \quad \text{for} \quad q < j \tag{84}$$

and

$$r = i \quad \text{for} \quad q = j \tag{85}$$

In this case, discrete ψ values occur at vertices of $\bar{\phi}$ cells. Now perform an *accurate, shape-preserving*, two-dimensional interpolation to give $\psi(x,y)$. Then $\phi(x,y)$ is obtained by cross-differentiation,

$$\phi(x,y) = \frac{\partial^2 \psi}{\partial x \partial y} \tag{86}$$

With this definition of ϕ, the area-integral in (82) can be converted into line-integrals of either x- or y-derivatives of ψ along the bounding streamlines (analogous to the point-evaluation of ψ in the one-dimensional case). This idea extends to three dimensions, where volume-integrals of ϕ can be converted to surface-integrals of derivatives of ψ.

As in the one-dimensional case, the crucial step is the accurate, shape-preserving interpolation of ψ. In two dimensions, assume that the advection area for each face can be approximated by a parallelogram defined by 'locally constant' velocity components. This simplifies the flux-integral calculation, provided the sub-cell behaviour is known. Simple bilinear sub-cell interpolation of ψ (analogous to linear interpolation in the one-dimensional case) gives the trivial piecewise constant 'interpolation' for ϕ,

$$\phi(x,y) \equiv \bar{\phi}_{i,j} \tag{87}$$

over cell (i,j). For component cell face Courant numbers *less than unity*, this leads to a generalisation of first-order upwinding, with good isotropy properties[17]—but excessive concomitant artificial diffusion. As in the one-dimensional case, this interpolation is shape-preserving, but very inaccurate. Extension to large Δt values appears to be very complex, geometrically. Various (non-shape-preserving) bidirectional parabolic interpolations of $\psi(x,y)$ lead to two-dimensional generalisations of the Lax-Wendroff, second-order upwind, and Fromm schemes. Two-dimensional cubic interpolation of $\psi(x,y)$ leads to a uniformly third-order polynomial interpolation algorithm, equivalent to UTOPIA[17]. As in the one-dimensional case, this is a cell-centred (velocity-direction-independent) interpolation with good accuracy and excellent phase behaviour—but it is not shape-preserving.

The difficult part seems to be the establishment of inexpensive multidimensional shape-preserving constraints on $\psi(x,y)$ analogous to the simple universal limiter constraints in the one-dimensional case. Some progress on the construction of a genuinely multidimensional shape-preserving limiter has been made, using an alternative approach[17]. Progress on this aspect of the NIRVANA project must await further developments, as must extensions to multi-dimensional nonlinear systems.

Finally, there is the question of diffusion and other terms in the conservation equations. Source terms, for example, should be averaged over the time-step and the spatial cell. To second order, advection and diffusion terms can be treated independently, and simply added together on the right-hand side of the update equation. At third-order and above, however, there is a cross-coupling between the two, even in one dimension[14]: diffusion has an effect on the estimated face value, and advection changes the time-averaged normal gradient[17]. Both of these effects are proportional to second spatial derivatives of the transported variable, and lead to straight-forward multidimensional formulae[17] in the case of $|c_x|, |c_y| \leqslant 1$. Details of the large-Δt extensions need to be worked out. However, the guiding principle is the estimation of terms in the explicit update of the governing conservation equation. For example, in one dimension,

$$\bar{\phi}_i^+ = \bar{\phi}_i - (c_r\phi_r - c_l\phi_l) + h\left[\alpha_r\left(\frac{\partial\phi}{\partial x}\right)_r - \alpha_l\left(\frac{\partial\phi}{\partial x}\right)_l\right] + \Delta t \bar{S}^* \tag{88}$$

where α is the diffusion parameter, $\alpha = D\Delta t/h^2$. Face values and gradients are time-averages, and $\bar{S}^*$ is a space-time-averaged source term.

CONCLUSIONS

The NIRVANA project is an ongoing quest for an expicit, single-time-step, conservative, flux-based, highly accurate, nonoscillatory finite-volume CFD scheme for multidimensional advection-dominated flows, without restrictions on the time-step. In order to demonstrate some of the basic features, the analysis in the present paper has focussed on the simplest possible nontrivial situation: pure one-dimensional advection at constant velocity on a uniform grid. The use of a discrete integral variable, ψ—the cumulative sum of cell-average values—is fundamental to the scheme, in order to estimate sub-cell behaviour of the transported variable, $\phi(x)$, by differentiation of $\psi(x)$. As emphasized many times, the spatial interpolation of discrete ψ_i values to give $\psi(x)$ is the crucial step in constructing the algorithm. Collocation of ψ at cell faces guarantees the cell-average constraint—that cell averages of $\phi(x)$ indeed match the given cell-average data, $\bar{\phi}_i$.

In one dimension, the constant-velocity advection update algorithm is extraordinarily simple. Each advected face value is obtained from a single evaluation of $\psi(x)$ at a point $u\Delta t$ upstream of the particular face. Because of the *global* interpolation of ψ (even in the case of piecewise polynomials over each cell), there is no stability constraint on the time-step. This is, in fact, consistent with the well-known CFL condition, which requires the numerical domain of dependence to include the physical domain of dependence. Since the discrete solution is known everywhere at the earlier time-level, the numerical domain of dependence is the entire physical domain. 'Stability' conditions that apply to a number of well-known explicit schemes (e.g., that the Courant number should be less than or equal to one) are actually not stability conditions at all—and certainly should not be labelled 'CFL conditions'. Rather, the Courant-number restriction of such methods results from a 'short-sighted' view of the piecewise interpolation scheme. The restriction is, in fact, on the range of validity of the local piecewise interpolant. Global interpolants, such as splines, do not have this problem. Similarly, if a global view is taken of the piecewise polynomial interpolants, there is no stability restriction on the time-step: the CFL condition is automatically satisfied.

Unsophisticated interpolation of ψ (such as piecewise linear and parabolic) was seen to lead to large-Δt generalisations of well-known schemes such as the (very diffusive) first-order method, and the (dispersive) Lax-Wendroff and second-order upwind schemes. One needs to be careful to distinguish between the 'natural upwinding' that occurs with all schemes because of the flux integral, and the upwind or downwind bias or cell-centred placement, relative to the cell under consideration, of the collocation points used for the interpolation of ψ. Thus, for example, the first-order scheme is a natural upwind scheme, but its interpolant is *cell-centred*. The Lax-Wendroff scheme and its large-Δt extension involve a *downwind*-weighted parabolic interpolant. Second-order upwinding, by contrast, uses an *upwind*-weighted parabolic interpolation of ψ. By averaging these two second-order schemes, (an unrestricted-Courant-number version of) Fromm's method is derived. This involves a velocity-direction-independent interpolant and, in common with all cell-centred interpolation schemes, has much better phase behaviour, which was, after all, the motivation for Fromm's method in the first place.

Higher-order piecewise polynomial interpolation can be recursively extended to higher and higher order. The third-order NIRVANA scheme reproduces the (large-c generalisation of the) QUICKEST algorithm—a scheme based on velocity-direction-independent interpolation, which has very good phase accuracy. Fourth and higher even-order downwind weighted schemes are all plagued by significant phase-lag dispersion errors, whereas higher odd-order schemes based on cell-centred interpolants have excellent phase accuracy and give progressively more accurate resolution of all profiles—except for Gibbs' phenomena near sharp features[1]. A cubic-spline interpolation of ψ is equivalent to a parabolic-spline sub-cell interpolation of ϕ. This gives very good phase accuracy and high resolution.

Spurious numerical oscillations arise from a lack of shape preservation in the interpolation of ψ (or ϕ). If an inherently convexity-preserving interpolation algorithm were available for ψ, there would be no need for so-called flux limiters. Interestingly enough, a very simple, geometrically based, local-convexity-preservation strategy for ψ turns out to be identical in structure to the previously (and quite independently) derived universal limiter[1]—but in this case, extended to arbitrarily large Courant number. Another, more restrictive, convexity-preservation scheme for ψ is equivalent to the large-Δt generalisation of the TVD limiter[1,14]. Application of the generalised universal limiter convexity-preservation strategy to cubic-spline or higher-order piecewise polynomial base schemes gives excellent (nonoscillatory, high-resolution) results when combined with a simple pattern-recognition algorithm for resolving real inflection-points in ψ (i.e., true local extrema in ϕ). Automatic compressive enhancement leads to *exact* sub-cell resolution of discontinuities without clipping extrema or generating anomalous 'stair-casing' in other profiles.

A brief von Neumann stability analysis has identified two factors in the complex amplitude ratio: one exact factor representing the simple translation over N mesh-widths, the other representing a numerical $\mathbf{G}$ based on Δc. To the extent that the latter is stable for $\Delta c < 1$, the overall $\mathbf{G}$ is always stable for any Courant number.

Finally some ideas were mentioned, outlining possible avenues for generalisation aimed at the development of practical algorithms for highly advective, nonlinear, multidimensional flow problems. These point to a number of possible directions for further research.

ACKNOWLEDGEMENTS

Most of this work was completed while the first author was a Visiting Research Scientist at the U.K. Meterological Office, Atmospheric Processes Division. Portions of the first author's research were supported by the Institute for Computational Mechanics in Propulsion (ICOMP) at the NASA Lewis Research Center under Space Act Agreement NCC 3-233.

APPENDIX

Wrap-around strategy

The periodic boundary conditions employed in the examples in this paper are somewhat complicated by the fact that the discrete function to be interpolated, ψ_i, is in fact the integral of the periodic function $\bar{\phi}_i$, i.e.,

$$\psi_i = h\sum_{p=1}^{i}\bar{\phi}_p \tag{A.1}$$

where $\psi_0 = 0$. Thus ψ_i is not itself periodic, although its first and second derivatives are. For a computational domain of N_h cells, with periodic boundary conditions on $\bar{\phi}_i$, the algorithm to calculate the right-face flux can be written as follows:

- Replace the index of the departure cell upwind from the right face, namely cell $(i-N)$, where $N = INT(c)$, by $i^* = i - N + I \times N_h$, where $I = 1 - INT[(i-N)/N_h]$, so that,

$$1 \leqslant i^* \leqslant N_h \tag{A.2}$$

- Interpolate between ψ_{i^*} and ψ_{i^*-1} to find $\psi(x^*)$, where the wrapped-around departure point, $x^* = [(x_i + h/2 - ch)$, modulo$(hN_h)]$, will now necessarily lie within the departure cell, as calculated above.
- Finally, the 'total mass' part of $\psi(x^*)$, namely $I\psi_{N_h}$, which was implicitly added on by the use of i^* in the interpolation, must be removed for the right-face flux calculation:

$$ch\phi_r(i) = \psi_i - [\psi(x^*) - I\psi_{N_h}] \tag{A.3}$$

Initial conditions

The discrete cell-average function, $\bar{\phi}_i$, is initialised by averaging the exact continuous function $\phi_{ex}(x)$ over the cell i:

$$\bar{\phi}_i = \frac{1}{h}\int_{x_i-h/2}^{x_i+h/2} \phi_{ex}(\xi)\,\mathrm{d}\xi \tag{A.4}$$

For the test profiles employed here (with a domain width of 150h), the initial ϕ is:

$$\phi_{ex}(x)=\begin{cases} 0 & \text{for } 0<x\leqslant 5h \\ 1 & \text{for } 5h<x\leqslant 25h \quad \text{(rectangular box)} \\ 0 & \text{for } 25h<x\leqslant 35h \\ sin^2[\frac{\pi}{20h}(x-35h)] & \text{for } 35h<x\leqslant 55h \quad \text{(sine-squared)} \\ 0 & \text{for } 55h<x\leqslant 65h \\ \sqrt{1-\frac{1}{10h}(\mathrm{x}-75\mathrm{h})} & \text{for } 65h<x\leqslant 85h \quad \text{(semi-ellipse)} \\ 0 & \text{for } 85h<x\leqslant 95h \\ \frac{1}{10h}(x-95h) & \text{for } 95h<x\leqslant 105h \quad \text{(triangle)} \\ 1-\frac{1}{10h}(x-105h) & \text{for } 105h<x\leqslant 115h \\ exp\{-\frac{1}{2\sigma^2}(x-135h)^2\} & \text{for } 115h<x\leqslant 150h \quad \text{(Gaussian)} \end{cases} \tag{A.5}$$

and this may be integrated analytically to give,

$$\psi_{ex}(x)=\int_0^x \phi_{ex}(\xi)\mathrm{d}\xi \tag{A.6}$$

The standard deviation for the Gaussian profile is taken to be,

$$\sigma=2.5h \tag{A.7}$$

From (A.4), the initial conditions for the cell-averages (excluding the Gaussian profile) take the form:

$$\bar{\phi}_i=\begin{cases} 0 & \text{for } 1\leqslant i\leqslant 5 \\ 1 & \text{for } 6\leqslant i\leqslant 25 \\ 0 & \text{for } 26\leqslant i\leqslant 35 \\ \frac{1}{2}\{1+\frac{10}{\pi}[sin(\frac{i-36}{10})-sin(\frac{i-35}{10})]\} & \text{for } 36\leqslant i\leqslant 55 \\ 0 & \text{for } 56\leqslant i\leqslant 65 \\ \frac{5}{2}\{sin[2\,sin^{-1}(\frac{i-75}{10})]+2\,sin^{-1}(\frac{i-75}{10}) & \\ \quad -sin[2\,sin^{-1}(\frac{i-76}{10})]+2\,sin^{-1}(\frac{i-76}{10})\} & \text{for } 66\leqslant i\leqslant 85 \\ 0 & \text{for } 86\leqslant i\leqslant 95 \\ \frac{1}{10}(i-95.5) & \text{for } 96\leqslant i\leqslant 105 \\ 1-\frac{1}{10}(i-105.5) & \text{for } 106\leqslant i\leqslant 115 \end{cases} \tag{A.8}$$

The discrete integral variable is then given, for $i=1$ to 115, by,

$$\psi_i=h\sum_{p=1}^{i}\bar{\phi}_p \tag{A.9}$$

taking $\psi_0=0$.

For the Gaussian profile, $116\leqslant i\leqslant 150$, $\bar{\phi}_i$ is calculated from the exact integral function, ψ_{ex}, by combining (A.4) and (A.6) to give,

$$\bar{\phi}_i=\frac{\psi_{ex}(x_i+{}^h/_2)-\psi_{ex}(x_i-{}^h/_2)}{h} \tag{A.10}$$

The values of $\psi_{ex}(x_i + {}^h/_2)$ and $\psi_{ex}(x_i - {}^h/_2)$ are explicitly calculated from (A.6) which gives, for example,

$$\psi_{ex}(x_i + {}^h/_2) = h\sigma\sqrt{\frac{\pi}{2}}\left[1 + erf\left(\frac{i-135}{\sigma\sqrt{2}}\right)\right] \tag{A.11}$$

Sub-cell interpolation

As a diagnostic for the numerical tests, sub-cell interpolants are calculated in a post-processing operation, using a fine grid of 200 points to every coarse-grid cell. Let *ii* be the index on the fine grid. The sub-cell integral variable values, $\psi(x_{ii}) = \psi_{ii}$, are calculated on this grid by interpolating between the coarse-grid ψ_i values in the same manner as for the departure point, $\psi(x^*)$. The sub-cell function values, ϕ_{ii} (no overbar), are then approximated by a simple backward difference,

$$\phi_{ii} = \frac{\psi_{ii} - \psi_{ii-1}}{h_{fine}} \quad \text{for } 1 \leqslant ii \leqslant 30\,000 \tag{A.12}$$

where $h_{fine} = h/200$. This choice gives adequate resolution of any discontinuities in $\phi(x)$ appearing in the sub-cell structure.

Real inflection point discriminator

Indiscriminate application of the limiter at all cell faces can result in an unnecessary, and sometimes significant, reduction in accuracy in certain regions (e.g., at the peak of the Gaussian profile). These regions can be identified with extrema in the ϕ profile, or inflection points in the integral function, ψ. If these 'real' inflection points can be located and the limiter subsequently not activated locally, significant improvement can result, without compromising the operation of the limiter.

The algorithm adopted identifies inflection points in ψ resulting from extrema in the basic $\bar{\phi}$ profile by a local change in sign of the second differences

$$b_i = \frac{\psi_{i+1} - 2\psi_i + \psi_{i-1}}{h^2} \tag{A.13}$$

'Real' inflection points are distinguished from spurious numerical oscillations by the use of a tolerance, δ, so that (for a maximum in ϕ) the change in sign is actually required to be a change from $b_i > \delta$ to $b_j < -\delta$, for $j = i+1$ or $j = i+2$, as i increases (and *vice versa* for a minimum in ϕ). So, given an integral-variable value $\psi(x^*)$, found by interpolating between ψ_{i^*} and ψ_{i^*-1}, the algorithm can be written:

- First check as usual for nonmonotonicity in ψ; i.e., constrain $\psi(x^*)$ by,

$$\psi_{i^*-1} \leqslant \psi(x^*) \leqslant \psi_{i^*} \tag{A.14}$$

- Identify a 'real' inflection point in ψ within the departure-point as satisfying any one of the following conditions on the second differences (for a local maximum in ϕ):
  (i) $b_{i^*-1} \gg 0 \gg b_{i^*}$
  (ii) $b_{i^*-1} \sim 0$ and $b_{i^*-2} \gg 0 \gg b_{i^*}$
  (iii) $b_{i^*} \sim 0$ and $b_{i^*-1} \gg 0 \gg b_{i^*+1}$ (A.15)
  (iv) $b_{i^*-2} \gg 0 \gg b_{i^*-1} > b_{i^*}$
  (v) $b_{i^*-1} > b_{i^*} \gg 0 \gg b_{i^*+1}$

where,

$$b_i \sim 0 \quad \text{implies} \quad |b_i| < \delta \tag{A.16}$$

$$b_i \gtrless 0 \quad \text{implies} \quad b_i \gtrless \pm\delta \tag{A.17}$$

- If a 'real' inflection point is identified in the ψ field, then the interpolated value, $\psi(x^*)$, is not further limited.
- A similar procedure (with appropriate reversal of inequalities) is used for local minima in ϕ.

A tolerance of $\delta = 0.01$ was empirically found to be small enough to recognise 'real' inflection points, whilst being large enough to suppress numerical oscillations.

Automatic compressive enhancement

In regions of large gradient in ϕ (or large curvature in ψ), the interpolant, $\psi(x^*)$, can be weighted towards the nearer of the two linear constraints, ψ_{R2} and ψ_{R3}, used in the universal limiter of *Figure 27a*, thus tightening the curvature in ψ and consequently sharpening the gradient of ϕ.

Thus for some tolerance, δ_c, if,

$$b_{i^*-1} b_{i^*} > \delta_c \tag{A.18}$$

(using the notation of A.3) then the interpolant is first constrained by the universal limiter (giving $\psi_b = \text{median}\,[\psi_{R1}, \psi_a, \psi(x^*)]$, where $\psi_a = \text{median}\,[\psi_{R1}, \psi_{R2}, \psi_{R3}]$ and the median of three numbers is that number between the other two) and then weighted towards the nearer limiter constraints, ψ_a. Thus, the new 'compressed' interpolant is given by,

$$\psi^* = \gamma\psi_b + (1-\gamma)\psi_a \tag{A.19}$$

The weighting, γ, is itself weighted with Δc to the neighbouring ratios of the local curvatures, constrained to lie between 0 and 1, namely:

$$\alpha = median\left[0, \frac{b_{i^*-2}}{b_{i^*-1}}, 1\right] \quad \text{and} \quad \beta = median\left[0, \frac{b_{i^*+1}}{b_{i^*}}, 1\right] \tag{A.20}$$

so that,

$$\gamma = \Delta c\alpha + (1-\Delta c)\beta \tag{A.21}$$

Thus, for example, for Δc small and (A.18) satisfied with b_{i^*+1} small (such as in the region of a sharp increase in ϕ), then γ is weighted by the small Δc towards β which is itself small and so ψ^* is strongly weighted towards the limiter constraint, ψ_a.

A tolerance of $\delta_c = 0.1$ was empirically found to be small enough to recognise near-discontinuities without introducing 'stair-casing' to otherwise smooth functions.

REFERENCES

1 Leonard, B. P. The ULTIMATE conservative difference scheme applied to unsteady one-dimensional advection, *Computer Meth. in Applied Mech. and Eng.*, **88**, 17–74 (1991)

2 Staniforth, A. and Côté, J. Semi-Lagrangian integration schemes for atmospheric models—a review, *Monthly Weather Review*, **119**, 2206–2223 (1991)

3 Priestly, A. A quasi-conservative version of the semi-Lagrangian advection scheme, *Monthly Weather Review*, **121**, 621–629 (1993)

4 Roache, P. J. A flux-based modified method of characteristics, *Int. J. for Num. Meth. in Fluids*, **15**, 1259–1275 (1992)

5 LeVeque, R. J. A large time step generalization of Godunov's method for systems of conservation laws, *SIAM J. of Num. Analysis*, **22**, No. 6, 1051–1073 (1983)

6 Fletcher, C. A. J. *Comput. Techniques for Fluid Dynamics*, Vols I and II, Springer, New York (1990)

7 LeVeque, R. J. *Num. Meth. for Conservation Laws*, 2nd ed., Birkhäuser, Basel (1992)

8 Hyman, J. M., Knapp, R. J. and Scovel, J. C. High order finite volume approximations of differential operators on nonuniform grids, *Physica D*, **60**, 112–138 (1992)

9 Colella, P. and Woodward, P. R. The piecewise parabolic method (PPM) for gas-dynamical simulations, *J. of Comput. Physics*, **54**, 174–200 (1984)

10 Lax, P. D. and Wendroff, B. Systems of conservation laws, *Communications in Pure and Applied Mathematics*, **13**, 217–237 (1960)

11 Leith, C. E. Numerical simulation of the earth's atmosphere, *Meth. of Comput. Physics*, **4**, 1–28 (1965)

12 Fromm, J. E. A method for reducing dispersion in convective difference schemes, *J. of Comput. Physics*, **3**, 176–189 (1968)

13 Leonard, B. P. A stable and accurate convective modelling procedure based on quadratic upstream interpolation, *Computer Meth. in Applied Mechanics and Eng.*, **19**, 59–98 (1979)

14 Sweby, P. K. High resolution schemes using flux limiters for hyperbolic conservation laws, *SIAM J. of Num. Anal.*, **21**, 995–1011 (1984)

15 Chapra, S. C. and Canale, R. P. *Num. Meth. for Engineers*, 2nd ed., McGraw-Hill, New York (1988)

16 Forsyth, G. and Moler, C. B. *Computer Solution of Linear Algebraic Systems*, Prentice-Hall, Englewood Cliffs (1967)

17 Leonard, B. P., MacVean, M. K. and Lock, A. P. Positivity-Preserving Numerical Schemes for Multidimensional Advection, *NASA TM 106055*, Lewis Research Center, Cleveland (1993)

18 van Leer, B. Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method, *J. of Comput. Physics*, **32**, 101–136 (1979)

19 Leonard, B. P. Note on the von Neumann stability of explicit one-dimensional advection schemes, *Comput. Meth. in Applied Mech. and Eng.*, **91**, in press (1994)

20 Rasch, P. J. Conservative shape-preserving two-dimensional transport on a spherical reduced grid, *Monthly Weather Review*, **122**, No. 6, 1337–1350 (1994)