COMPUTER METHODS IN APPLIED MECHANICS AND ENGINEERING 19 (1979) 59–98

A STABLE AND ACCURATE CONVECTIVE MODELLING PROCEDURE BASED ON QUADRATIC UPSTREAM INTERPOLATION

B.P. LEONARD *

Sonderforschungsbereich 80, University of Karlsruhe, West Germany

Received 29 December 1977
Revised manuscript received 17 April 1978

A convective modelling procedure is presented which avoids the stability problems of central differencing while remaining free of the inaccuracies of numerical diffusion associated with upstream differencing. For combined convection and diffusion the number of operations at each grid point is comparable to that of standard upstream-plus-central differencing – however, highly accurate solutions can be obtained with a grid spacing much larger than that required by conventional methods for comparable accuracy, with obvious practical advantaged in terms of both speed and storage. The algorithm is based on a conservative control-volume formulation with cell wall values of each field variable written in terms of a quadratic interpolation using in any one coordinate direction the two adjacent nodal values together with the value at the next upstream node. This results in a convective differencing scheme with greater formal accuracy than central differencing while retaining the basic stable convective sensitivity property of upstream-weighted schemes. The consistent treatment of diffusion terms is equivalent to central differencing. With careful modelling, numerical boundary conditions are not troublesome. Some idealized problems are studied, showing the practical advantages of the method over other schemes in comparison with exact solutions. An application to a complex unsteady two-dimensional flow is briefly discussed.

Introduction

Satisfactory numerical modelling of convection presents a well-known dilemma to the computational fluid dynamicist. On the one hand, central differencing may lead to unphysical oscillatory behaviour in an implicit solution or to disastrous nonconvergence in an explicit computation in regions where convection strongly dominates diffusion [1]. On the other hand, computations based on the classical alternative of upstream differencing often suffer from severe inaccuracies due to truncation error [1] or streamline-to-grid skewness [2]. Each of the latter error mechanisms can be associated with equivalent artificial numerical diffusion terms introduced by the one-sided upstream differencing of convection. Although, in principle, grid refinement can alleviate all these problems, the necessary degree of refinement is often totally impracticable for engineering purposes [3], especially if one is attempting to model high-speed unsteady three-dimensional turbulent flows.

Current practice seems to have abandoned hope of accurate modelling. Instead, upstream (or upstream weighted [3]) differencing or similar hybrid schemes [4] are used routinely [5] with grid sizes which greatly exceed those that are theoretically necessary for normal engineering accuracy. While this approach usually results in qualitatively acceptable solutions – and can in fact

* Permanent address: City University of New York, College of Staten Island, 130 Stuyvesant Place, New York, 10301, USA.

often be forced to give global correspondence with measured results – it is clearly disconcerting to allow numerical diffusion to dominate the modelled physical processes. This is especially true when transport models themselves are under investigation [6].

Proponents of upstream differencing would argue that in regions where convection strongly dominates streamwise diffusion the local upstream values of the field variables are swept downstream virtually unchanged, whereas in high-diffusion regions the form of the (then relatively small) convection terms is not important – assuming diffusion is adequately modelled – and in regions where the two mechanisms are comparable one can switch to the more accurate central differencing for convection [4] or use a suitably weighted combination of central and upstream differencing [3]. The usual justification for this argument is based on a study of an extremely specialized model problem, namely the two-point boundary value problem representing a source-free steady one-dimensional balance between convection and diffusion [7]. The fallacy of this argument stems from the fact that conclusions based on this particular model problem simply cannot be generalized to more complex problems involving convection and any combination of streamwise diffusion, cross-stream transport (in two and three dimensions), unsteady boundary conditions, and steady or unsteady source terms [2]. In fact, whenever second differences (representing curvature) of the discrete functional form of the field variables obtained by upstream differencing are nonzero, then, as shown later, in the streamwise coordinate direction there is a local first-order inconsistency ($\sim$ truncation error) in the solution equal to half the difference between the local increment in the function across a cell and the increment at a chosen reference cell. Thus, the only truncation-error-free problems are those whose solutions vary at most linearly with the grid index in the streamwise direction. In the model steady one-dimensional source-free problem referred to above, the high-convection limit upstream difference solution is a constant (equal to the upstream value) except at the downstream boundary, where a sudden jump must occur [7], thus correctly modelling (in this case) the singular physical behaviour [8, 9].

In addition to the often severe truncation error in computations involving field-variable curvature in the streamwise direction there is also the problem of streamline-to-grid skewness in two and three dimensions [2, 4]. Although this latter problem can be alleviated somewhat by (computationally expensive) skew differencing procedures [10], numerical experience seems to indicate that the remaining truncation error is of the same order as the skewness error which has been eliminated. In addition, there is a possibility of convective instability, which makes it inadvisable to use the technique in modelling the convection of transport variables [11]. Thus, the relative cost-benefit ratio of skew differencing is rather unfavourable.

It could be said that skew differencing and the various upstream-central weighting schemes are merely cosmetic approaches to a basically ugly problem. A fresh start is clearly needed. The fundamental task seems to be to devise a scheme which estimates control-volume wall values accurately while avoiding (if possible) the high-convection instability problems that plague linear interpolation (i.e. central differencing). Fortunately, there is a rather simple solution to this problem. Its exposition forms the main subject of this paper; but first the problems associated with central and upstream differencing will be reviewed as a guide in motivating the development of the new method.

1. The problem with central differencing

Consider the differential equation

$$\frac{\partial \phi}{\partial t} = -\frac{\partial (u\phi)}{\partial x} + \frac{\partial}{\partial x}\left(\Gamma \frac{\partial \phi}{\partial x}\right) + S\,, \tag{1}$$

together with appropriate initial and boundary conditions. This can be interpreted as an unsteady one-dimensional convection-diffusion equation for the scalar ϕ with a well-defined source term $S(\phi, x, t)$. Alternatively, it could represent an unsteady convection-diffusion equation in two or three dimensions, where the x-components of convection and diffusion are represented explicitly, and all other transport and source terms are grouped into the term $S(\phi, x, y, z, t)$, which may or may not be known explicitly (e.g. part of the ϕ-dependence may be due to feedback through other dependent variables).

Fig. 1 shows the standard control-volume approach to modelling (1), using linear interpolation for estimating the left and right wall values of ϕ (e.g. $\phi_l = (\phi_L + \phi_C)/2$, etc.). A similar procedure is used for wall values of u and Γ (and all other dependent variables). The consistent estimation of gradient terms is given by

$$\left(\frac{\partial \phi}{\partial x}\right)_l = \frac{\phi_C - \phi_L}{\Delta x_l} \tag{2}$$

and so on, as suggested by the heavy bars in fig. 1. The spatially discretized model of (1) can then be written

$$\overline{\frac{\partial \phi}{\partial t}} = \left[u_l \phi_l - u_r \phi_r + \Gamma_r \left(\frac{\partial \phi}{\partial x}\right)_r - \Gamma_l \left(\frac{\partial \phi}{\partial x}\right)_l\right] / \Delta x_C + \overline{S}\,, \tag{3}$$

where the bars represent control-volume averages. It is usual to speak of (3) as representing central differencing (in x). The terminology is more appropriate for a model problem in which u, Γ and Δx are all constant, giving

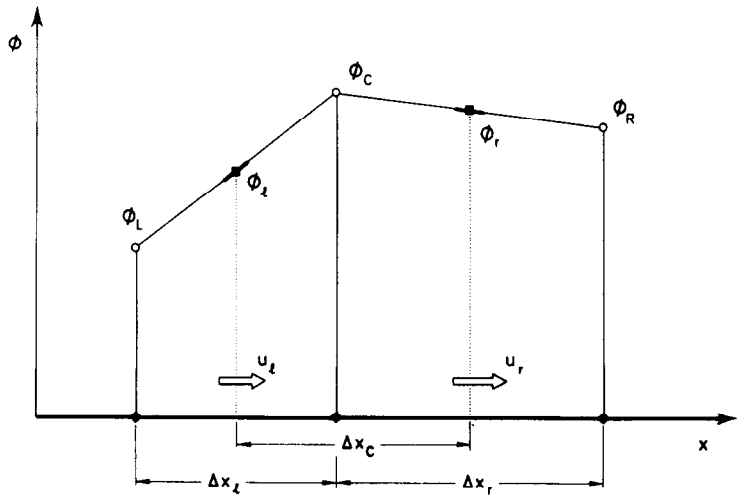

Fig. 1. Central differencing uses linear interpolation for cell wall values and the corresponding gradients.

$$\overline{\frac{\partial \phi}{\partial t}} = -u_0 \left(\frac{\phi_R - \phi_L}{2\Delta x_0} \right) + \Gamma_0 \left(\frac{\phi_L + \phi_R - 2\phi_C}{\Delta x_0^2} \right) + \bar{S} \ . \tag{4}$$

Note that ϕ_C does not appear in the convective term of (4). In the general case of (3) ϕ_C has a coefficient of $(u_l - u_r)/(2\Delta x_C)$, which may be positive, negative or zero, depending on the behaviour of u.

It is instructive to consider the sensitivity of the convective term to variations in ϕ_C. In the case of (4), i.e. central differencing with constant u, this sensitivity is given by the expression

$$\Sigma_C \ (\text{CENTRAL}, u_0) = \frac{\partial(\text{INFLUX})}{\partial \phi_C} \equiv 0 \ , \tag{5}$$

where INFLUX represents the net convective influx per unit volume into the control-volume. In the general central difference case,

$$\Sigma_C (\text{CENTRAL}) = \frac{\partial(\text{INFLUX})}{\partial \phi_C} = \frac{u_l - u_r}{2\Delta x_C} \sim -\frac{1}{2} \frac{\partial u}{\partial x} \ . \tag{6}$$

Convective sensitivity to ϕ_C in this case depends on the local velocity gradient.

If in the course of computation the numerical convective influx happens to be larger than the value needed to balance the transport equation, erroneous accumulation in ϕ_C will occur in numerical time since $\overline{\partial \phi / \partial t}$ is proportional to $\partial \phi_C / \partial t$. The same effect occurs in the iterative solution of a steady flow problem; in this case the iteration number takes the place of computational time [1]. This accumulation will in turn affect the numerical influx depending on the value of Σ_C, as follows:

$$\begin{aligned} &\Sigma_C > 0 \qquad \text{(unstable sensitivity)} \ , \\ &\Sigma_C = 0 \qquad \text{(neutral sensitivity)} \ , \\ &\Sigma_C < 0 \qquad \text{(stable sensitivity)} \ . \end{aligned} \tag{7}$$

In the convectively unstable or neutral cases overall stabilization is dependent on diffusive sensitivity to ϕ_C. The latter is given by

$$\Sigma_D (\text{CENTRAL}, \Gamma_0) = -\frac{2\Gamma_0}{\Delta x_0^2} \qquad \text{(from (4))} \tag{8}$$

or, in the general case,

$$\Sigma_D (\text{CENTRAL}) = -\frac{(\Gamma_l/\Delta x_l + \Gamma_r/\Delta x_r)}{\Delta x_C} \qquad \text{(from (3))} \ . \tag{9}$$

Physically, Γ is always positive, so Σ_D should be stabilizing. However, if the generalized grid Péclet number (see [2])

$$P_\Delta = \frac{u_C \Delta x_C}{\Gamma_C} \tag{10}$$

is large, Σ_D may be too weak to counteract a positive Σ_C in regions of unfavourable velocity gradient (particle deceleration). Even when Σ_C is zero, large transient errors may occur if P_Δ is large. Of course, if Γ is also computed from a transport equation using central differences, the possibility exists of Γ locally taking on erroneous negative values, in which case computational disaster usually follows rapidly.

In a general unsteady multidimensional flow problem involving transport equations for the velocity components, diffusion coefficients and various scalars, there is no way of guaranteeing stable sensitivity of the type discussed above in an explicit calculation using central differencing of the convection terms. Even an implicit solution, although stable for the model problem given by (4) (see [1]), is very hazardous in the general case because of the possibility of negative computed Γ's. This is especially true at large P_Δ's. The problem is intimately associated with the appearance of wiggles – i.e. spatially decaying or growing oscillations of wavelength $2\Delta x$ – typical of central difference solutions of the model two-point boundary value convection-diffusion problem for $P_\Delta > 2$. This phenomenon, especially the role played by boundary conditions, is lucidly described in [1, sec. III-C-8]. Interestingly enough, the criterion for the absence of wiggles, $P_\Delta \leqslant 2$, is not a stability condition in the von Neumann sense for the explicit model equation (4) using forward time differencing, although it is sometimes claimed as such [1, 12]. The confusion apparently arises because of the fact that, from a practical point of view, in a complex flow problem the combination of transient wiggles and neutral or unstable convective sensitivity often results in divergent behaviour of central difference computations in regions where $P_\Delta \gtrsim 2$. It should be stressed however that in some types of problem (as seen later) it is possible to achieve convergent solutions (with or without wiggles, usually depending on outflow numerical boundary conditions) by explicit forward time differencing and central space differencing of the convection-diffusion equation for any finite P_Δ, provided only that the (correct) von Neumann stability conditions are satisfied.

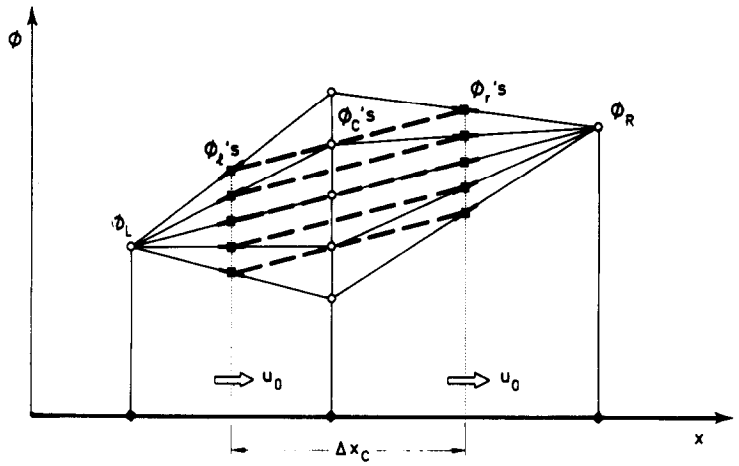

Fig. 2. Convective insensitivity of central differencing (the value of $(\phi_r - \phi_l)$ is independent of variations in ϕ_C).

For reference purposes fig. 2 shows graphically the information contained in (5) for constant u. The convective insensitivity of this case is immediately clear from the parallelism (i.e. identical slope) of the lines joining corresponding ϕ_l and ϕ_r values for different ϕ_C's.

2. Upstream differencing – positive and negative aspects

In fig. 3 the cell wall values corresponding to the control-volume formulation of upstream differencing (called the donor-cell technique [13]) are shown when u_l and u_r are both positive to the right. Clearly, there are three other configurations depending on various combinations of the signs of u_l and u_r. The wall velocity components, u_l and u_r, are the convect*ing* velocities and are usually obtained by linear interpolation between velocity node values. In the transport equation for the u-component of velocity itself (i.e. $\phi \to u$ in (1)) the convect*ed* wall value of the velocity components is treated in the same manner as any other scalar, i.e. its value is taken as that of the adjacent upstream node value, where the upstream direction (as well as the magnitude of the convecting velocity component) is based on the linearly interpolated u-value. This duality of the velocity components in their own transport equations, although conceptually objectionable in terms of consistency of formulation, does not itself lead to problems any worse than those for a velocity-independent scalar. From the point of view of the overall *order* of the truncation error involved it is clearly inconsistent (and slightly more expensive computationally) to use linear interpolation for the convecting velocities in an upstream difference formulation. The fact that it has become almost standard practice is perhaps indicative of the philosophy involved in grasping at a few straws of supposed accuracy (wherever it is not at the obvious expense of stability) in the desperate hope that the overall *size* of the computational errors will somehow be reduced. Under most circumstances a second-order improvement to a first-order error is not likely to meet with much practical success, especially when there is no guarantee that the "improvement" will even be in the right direction. The fallacy of believing that using linearly interpolated convecting velocities necessarily improves accuracy is clearly

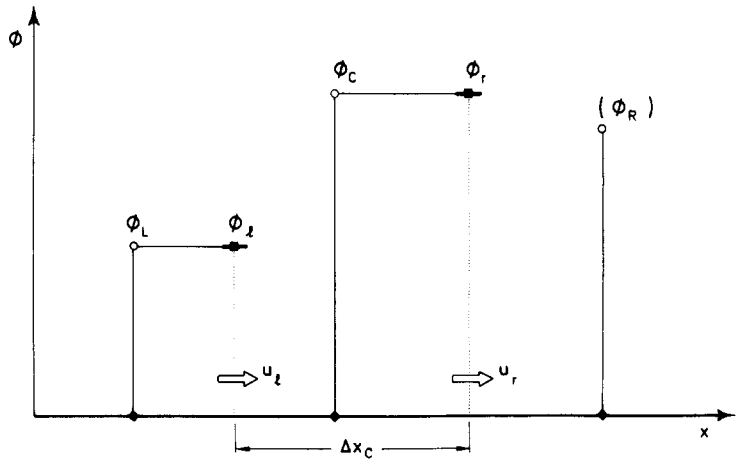

Fig. 3. Cell wall values for upstream (donor-cell) differencing when u_l and u_r are positive to the right, as shown (note the zero wall gradients consistent with this approximation).

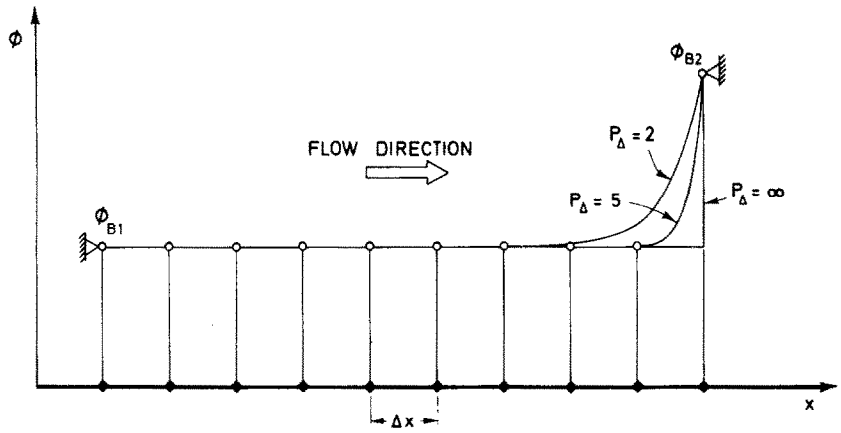

Fig. 4. Singular numerical solution of the two-point boundary value convection-diffusion problem (exact solutions for $P_\Delta = 2, 5$ and ∞ are also shown).

demonstrated by a situation in which u and ϕ both vary but in such a way that $\partial^2(u\phi)/\partial x^2 = 0$ in some finite region. In this case the conventional scheme generates the usual first-order error, whereas the fully upstream method (using upstream values for both convecting velocity and convected scalar) is exact!

The wall gradient terms have been modelled by a variety of methods. The classical method [1] is to assume gradients equivalent to linear interpolation as in the case of central differencing. In this case there is yet another inconsistency between the estimated zeroth-order wall value and the modelled first-order wall gradient. A conceptually more consistent approach assumes all wall gradients, and therefore all diffusion terms, to be zero under conditions where upstream differencing of convection is to be used [4, 7], as suggested in fig. 3. For steady flow, in the absence of source terms and cross-stream transport, field variables are then swept downstream unchanged. This, for example, leads to the singular solution of the two-point boundary value problem mentioned above. As shown in fig. 4, this is the appropriate numerical solution *of this problem* for $P_\Delta \gtrsim 5$. For $2 < P_\Delta < 5$ the singular solution is slightly in error at the last computational grid point. For $P_\Delta < 2$ the method of [4] switches to central differencing (then including diffusion terms). It so happens that at $P_\Delta = 2$ the central difference solution of this problem is identical to the nondiffusive upstream singular solution (above $P_\Delta = 2$ the central difference solution is oscillatory [1]). Thus, the hybrid scheme generates a continuity of solutions across $P_\Delta = 2$. As seen later for a model problem involving a source term, this continuity is due to the fact that upstream differencing of convection without diffusion is identical to central differencing including (artificial) diffusion corresponding to an effective P_Δ of 2.

Other methods have been proposed for modelling the gradient terms [2, 3, 10, 14], usually in conjunction with weighted upstream-central combination schemes for both convection and diffusion. Such schemes can produce slightly greater accuracy than the $P_\Delta \gtrsim 2$ hybrid method for the model problem of fig. 4 in the range $1 \lesssim P_\Delta \lesssim 5$. However, none of the above conclusions regarding accuracy or appropriateness can be generalized with any justification to more complex flow configurations. Such considerations are extremely problem-dependent.

By considering fig. 3 and other configurations for various signs of u_l and u_r, the individual convective flux terms can be written explicitly as

$$u_l \phi_l/\Delta x_C = [(u_l + |u_l|)\phi_L + (u_l - |u_l|)\phi_C]/(2\Delta x_C) \tag{11}$$

and

$$u_r \phi_r/\Delta x_C = [(u_r + |u_r|)\phi_C + (u_r - |u_r|)\phi_R]/(2\Delta x_C)\,. \tag{12}$$

Hence, the net convective sensitivity to ϕ_C is

$$\Sigma_C(\text{UPSTREAM}) = [(u_l - |u_l|) - (u_r + |u_r|)]/(2\Delta x_C)\,. \tag{13}$$

The various combinations of signs of u_l and u_r generate Σ_C-values as shown in table 1. As seen, the sensitivity is stable in all cases except when physical convergence occurs, the latter having neutral sensitivity.

Table 1
Dependence of Σ_C on the signs of u_l and u_r.

u_l	u_r	Σ_C
positive	positive	$-\|u_r\|/\Delta x_C$
positive	negative	0
negative	positive	$-(\|u_l\| + \|u_r\|)/\Delta x_C$
negative	negative	$-\|u_l\|/\Delta x_C$

On a global scale, (13) represents a very stabilizing numerical effect. This is, of course, at first sight, a pleasing result in contrast to the central difference case. However, stability in this case is actually being achieved through the rather heavy-handed introduction of artificial damping (numerical diffusion) at the expense of a faithful response to the physical forcing functions (sources and boundary conditions). Control systems engineers would undoubtedly be astonished at the use of such a brute-force approach to a stability problem. However, many computational fluid dynamicists seem to have been lulled into a state of false security by the docility of upstream differencing – no doubt as an overreaction to troubles and frustations experienced with the wiggles and divergent behaviour of central differencing.

The form of the numerical diffusion can best be demonstrated in the simple case of constant u, Γ and Δx, using central differencing of diffusion,

$$\overline{\frac{\partial \phi}{\partial t}} = -u_0 \frac{(\phi_C - \phi_L)}{\Delta x_0} + \Gamma_0 \frac{(\phi_L + \phi_R - 2\phi_C)}{\Delta x_0^2} + \bar{S}\,, \tag{14}$$

which can be recast into the same form as (4) by writing

$$\overline{\frac{\partial \phi}{\partial t}} = -u_0 \frac{(\phi_R - \phi_L)}{2\Delta x_0} + (\Gamma_0 + \Gamma_{num}) \frac{(\phi_L + \phi_R - 2\phi_C)}{\Delta x_0^2} + \bar{S}\,, \tag{15}$$

where, in comparison with central differencing of convection, the additional numerical diffusion coefficient due to upstream differencing is given by the well-known formula (see [1])

$$\Gamma_{num} = \frac{|u_0|\Delta x_0}{2} \,. \tag{16}$$

Thus, if one wishes to make Γ_{num} insignificant in comparison with the physical diffusion coefficient, the condition

$$P_\Delta = \frac{|u_0|\Delta x_0}{\Gamma_0} \ll 2 \tag{17}$$

is required. This is clearly much more stringent than the practical stability condition $P_\Delta \lesssim 2$ for central differencing. Both conditions are, in fact, highly unrealistic in terms of practical grid requirements for problems of engineering or geophysical interest, in which the global (turbulent) Péclet or Reynolds number, based on a typical macroscopic length scale, can be extremely large.

To estimate the truncation error involved in computations using upstream differencing, consider (14) rewritten as

$$u_0 \frac{(\phi_C - \phi_L)}{\Delta x_0} = \overline{S^*} \,, \tag{18}$$

where the explicit x-diffusion and time-derivative terms of (14) have been absorbed into the source term, which now represents all physical effects other than the explicit (upstream-difference modelled) x-convection term. The corresponding differential equation is

$$u_0 \frac{\partial \phi}{\partial x} = S^* \,. \tag{19}$$

Assume for example that ϕ^E represents an exact solution to some model problem of the type given by (19); then, by definition, $\overline{S^*}$ can be written

$$\overline{S^*} = \frac{1}{\Delta x_0} \int_{-\frac{\Delta x_0}{2}}^{\frac{\Delta x_0}{2}} u_0 \frac{\partial \phi^E}{\partial x} \, dx = u_0 \frac{(\phi_r^E - \phi_l^E)}{\Delta x_0} \,, \tag{20}$$

where ϕ_l^E and ϕ_r^E are the exact left and right wall values, respectively. Thus, (18) and (20) can be combined to give

$$\phi_C - \phi_L = \phi_r^E - \phi_l^E \,. \tag{21}$$

This forms the basis of a simple graphical construction to generate the values that would be obtained by upstream differencing for a problem whose exact solution ϕ^E is known. Fig. 5 shows an example of the construction. Of course, in practice, the exact solution is not known; however, the method can be applied to the computed results as a test of consistency of upstream differencing in a particular case and thus as a quantitative estimate of the (first-order) truncation error involved. It is not difficult to see from geometrical considerations that the global first-order truncation error is given by

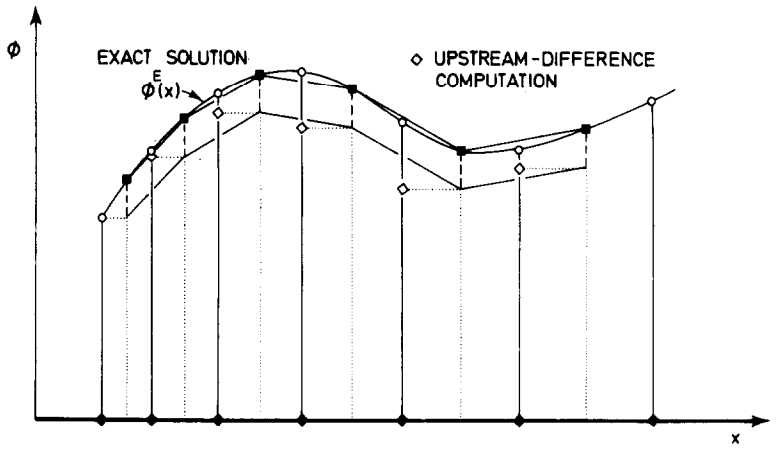

Fig. 5. Graphical construction of the upstream difference solution of (19) from a given exact solution $\phi^E(x)$.

$$\mathcal{C} = \phi^E - \phi \approx \frac{\Delta x_0}{2}\left[\left(\frac{\partial \phi^E}{\partial x}\right)_{Ref} - \left(\frac{\partial \phi^E}{\partial x}\right)\right] = \frac{1}{2}\left(\delta\phi^E_{Ref} - \delta\phi^E\right), \tag{22}$$

where $\partial\phi^E/\partial x$ is the local slope of the exact solution, $(\partial\phi^E/\partial x)_{Ref}$ is the corresponding slope at the common boundary condition reference point, and the $\delta\phi^E$'s are the corresponding algebraic wall-to-wall increases in ϕ^E across a control-volume, as shown in fig. 6. It should be clear from (22) that although the truncation error is formally proportional to Δx (i.e. first order), the actual size depends on the range of cell-increment values $\delta\phi$ throughout the computational region. Thus, a small value of Δx (in comparison with some macroscopic length scale) does not by itself guarantee that the size of the first-order truncation error will be small. A single grid cell for which $\delta\phi$ has a large magnitude may corrupt the entire solution. As mentioned in the introduction, the only up-

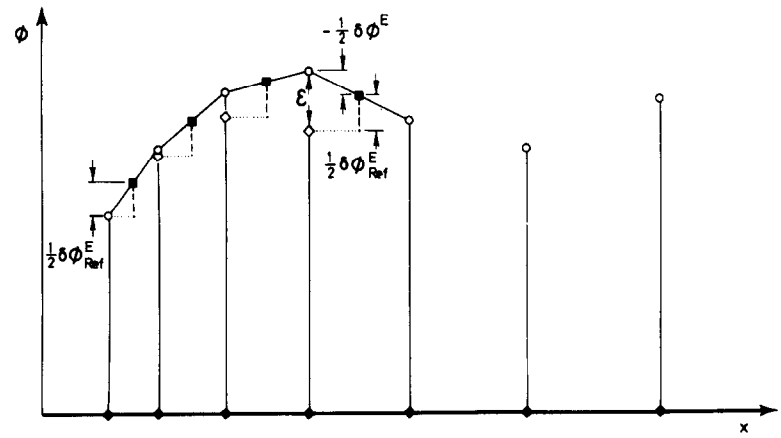

Fig. 6. First-order truncation error $\mathcal{C}$ in terms of cell increments $\delta\phi$ (note the linear interpolation of ϕ^E).

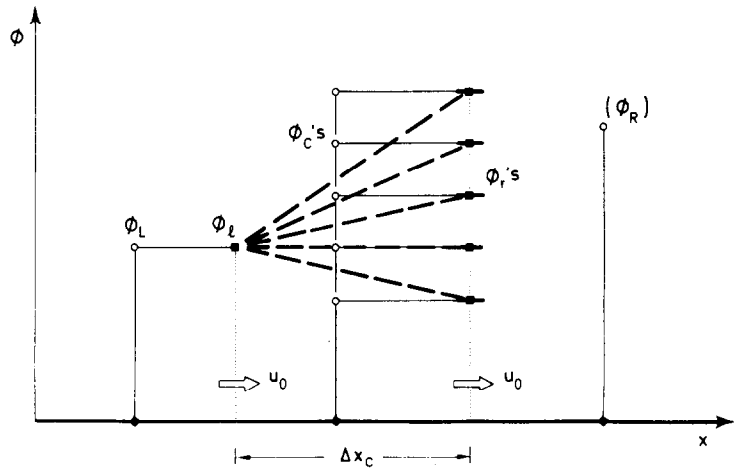

Fig. 7. Stable numerical convective sensitivity of upstream differencing.

stream difference solutions which are free of truncation error are those for which $\delta\phi \equiv$ const. If ϕ is a monotonic function of x, a variable grid could, in principle, be chosen to satisfy this condition. Clearly, any function obtained by upstream differencing which has a local extremum must involve a nonzero first-order truncation error.

In summary, it has been seen that upstream differencing does indeed achieve very stable convective sensitivity. For reference, the phenomenon is portrayed graphically in fig. 7 for the constant velocity case. However, the stability achieved by the introduction of artificial damping (numerical diffusion) is at the direct expense of accuracy. The size of the truncation error may be worse than its order would suggest, especially if the spatial gradients in the computed field variables span a large range of values. Certainly, eq. (22), using computed ϕ-values, or the construction of fig. 5 should always be used as a consistency check (and therefore an error estimate) in any upstream difference solution claiming quantitative agreement with experimental measurements. Perhaps one should not try to match the computed solution to raw experimental data but rather to a transformation of the data based on a construction similar to that of fig. 5 using the computational grid mesh in regions where upstream differencing is used. This concept should at least raise doubts as to the universality of empirical parameters adjusted so as to make upstream difference solutions match (untransformed) data as quantitatively as possible. Except for a rather small class of flow problems (e.g. simple shear flows, unseparated boundary layers, and pipe and channel flows which develop monotonically, assuming optimal grid spacing) one should not expect solutions based on upstream differencing using practical grids to yield anything more than qualitative estimates of the overall flow characteristics.

3. Quadratic Upstream Interpolation for Convective Kinematics (QUICK)

The problems associated with both central and upstream differencing techniques can be interpreted in terms of the method of estimating wall values of the dependent variables in a control-volume formulation. For central differencing the estimation is based on linear (or first-order) inter-

polation (fig. 1), which, although intuitively reasonable and involving only second-order truncation error, has the unfortunate geometrical property that $\phi_r - \phi_l$ is independent of ϕ_C, thus leading to stability problems in the convective sensitivity. Upstream differencing is equivalent to zeroth-order interpolation with the choice of direction depending on the sign of u (fig. 3). The directional dependence gives very stable convective sensitivity; however, this advantage is strongly offset by the first-order truncation error.

A natural question arises as to the possibility of constructing an interpolation scheme which simultaneously possesses good accuracy and the directional properties associated with stable convective sensitivity. First, it should be noted that a centered quadratic fit to the three node values ϕ_L, ϕ_C and ϕ_R, although of good interpolative accuracy, has the same insensitivity of $\phi_r - \phi_l$ to ϕ_C as the linear case. In addition, such a scheme is not conservative, i.e. $\phi_l(i) \neq \phi_r(i-1)$ in general. Higher accuracy schemes involving, in addition to the central node, more than two nodal points per control-volume cell for each spatial coordinate are, of course, possible. When these are centered about the central node (i.e. an equal number on either side), the scheme can be made conservative; but the convective sensitivity of symmetrically centered schemes remains neutrally stable in regions of constant velocity and may be locally stable or unstable depending on the behaviour of the convecting velocity field. For example, higher-even-order schemes still suffer from wiggles in regions of sudden changes of value although the oscillations are usually of smaller amplitude and shorter wavelength than those of the corresponding second-order computation [15]. Special monotonizing techniques can be developed, possibly with the aid of exponential or other functions to resolve local sudden jumps in value. The problem may then become one of computational expense due to the additional number of operations required at each grid-point computation.

Since symmetrically centered schemes possess no inherently stable convective sensitivity properties, asymmetrically placed interpolation schemes suggest themselves as possible candidates. As might be expected, upstream shifting is appropriate. In regions where there are no velocity reversals the simplests schemes (other than conventional upstream differencing) involve a total of four nodal points per cell for one-dimensional convection: the central node plus two upstream and one downstream. Simple cubic interpolation using the four node values leads to a nonconservative scheme although the convective sensitivity is stable. In addition, problems arise in the simple cubic scheme when there are velocity reversals. However, by using a *three-point upstream-weighted quadratic interpolation for each wall value individually* a conservative formulation with stable convective sensitivity can be achieved. If the normal velocity component has the same sign for a given pair of opposite walls, a total of four nodal points per control-volume cell are involved in the respective coordinate direction. Diverging velocity components involve a centered three-point scheme, whereas converging velocity components involve five symmetrically placed points in the corresponding coordinate direction.

The present paper is concerned with primarily one-dimensional convection (in the direction of large P_Δ) – the overall flow may be multidimensional, provided only that the transverse P_Δ values are small. The corresponding formulation for strong convection in two dimensions results in an overall ten-point scheme where there are no velocity component reversals; three-dimensional flow requires twenty grid points per control-volume cell. Of course, more or fewer points will be involved depending on the local velocity field components convergence or divergence, respectively.

The formulation of the QUICK algorithm is conveniently illustrated using constant grid spacing, the variable-grid formulas being given in the appendix. Fig. 8 shows the basic interpolation scheme for ϕ_r when u_r is positive to the right as shown. For constant grid spacing the resulting formula is

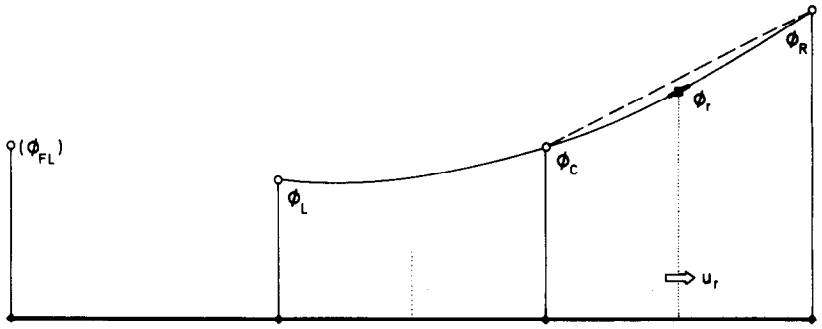

Fig. 8. Quadratic upstream interpolation for ϕ_r and $(\partial\phi/\partial x)_r$.

$$\phi_r = \tfrac{1}{2}(\phi_C + \phi_R) - \tfrac{1}{8}(\phi_L + \phi_R - 2\phi_C)\,, \tag{23}$$

which may be interpreted as a linear interpolation corrected by a term proportional to the upstream-weighted curvature. For modelling the gradient $(\partial\phi/\partial x)_r$, the tangent at the wall has also been shown in fig. 8. It is a geometric property of the parabola that the slope half way between two points is equal to that of the chord joining the points, i.e.

$$\left(\frac{\partial\phi}{\partial x}\right)_r = \frac{\phi_R - \phi_C}{\Delta x_r}\,, \tag{24}$$

which is valid in the variable-grid case as well, as indicated by the notation. This, of course, is identical to the central difference formula.

Similarly, fig. 9 shows the construction for ϕ_l and $(\partial\phi/\partial x)_l$ when u_l is also positive to the right. The corresponding formulas are

$$\phi_l = \tfrac{1}{2}(\phi_L + \phi_C) - \tfrac{1}{8}(\phi_{FL} + \phi_C - 2\phi_L) \tag{25}$$

and

$$\left(\frac{\partial\phi}{\partial x}\right)_l = \frac{\phi_C - \phi_L}{\Delta x_l}\,. \tag{26}$$

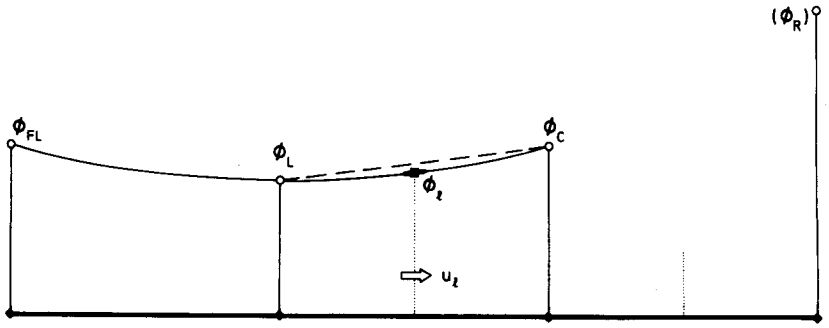

Fig. 9. Quadratic upstream interpolation for ϕ_l and $(\partial\phi/\partial x)_l$.

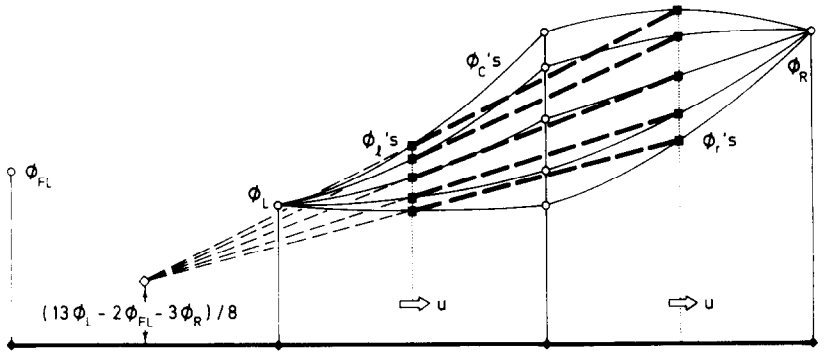

Fig. 10. Stable numerical convective sensitivity of the QUICK method.

The QUICK formulas (23)–(26) are appropriate to steady or quasisteady flow in which P_Δ is large in not more than one direction. The corresponding unsteady-flow (QUICKEST) formulas are developed in the next section.

Because of the consistent quadratic interpolation used in modelling both the convective and diffusive terms, the overall truncation error in a *solution* obtained by the QUICK algorithm is *third-order* in the spatial grid size. Thus, highly accurate solutions can be obtained using practical grid spacing.

Convective sensitivity to ϕ_C is given by

$$\Sigma_C(\text{QUICK}, \Delta x_0) = -\frac{1}{2}\frac{(u_r - u_l)}{\Delta x_0} - \frac{1}{8}\frac{(u_l + 2u_r)}{\Delta x_0}, \tag{27}$$

which is seen to involve a gradient term (the same as that of central differencing) plus a stabilizing term. In the case $u = u_0$ (= a positive or negative constant) the formula becomes

$$\Sigma_C(\text{QUICK}, u_0, \Delta x_0) = -\frac{3}{8}\frac{|u_0|}{\Delta x_0}, \tag{28}$$

which is clearly stable. This behaviour is demonstrated in fig. 10 (with u_0 positive to the right), which should be compared with fig. 2 (central differencing) and fig. 7 (upstream differencing).

In terms of the total number of operations at each grid point the overall QUICK method is relatively inexpensive. Assume a left-to-right explicit computational sweep with increasing index i. For the computation at station i a new value must be computed for the right wall gradient for the cell one grid point ahead ($i + 1$), the old value of this unindexed variable having been used for the current value of $(\partial\phi/\partial x)_r$, whose old value has been in turn previously used for the new value of $(\partial\phi/\partial x)_l$. A branching is then made on the basis of the sign of u_r to compute the appropriate (forward- or backward-weighted) curvature as the difference of two appropriate gradients. The value of ϕ_r is then found directly from

$$\phi_r = \frac{1}{2}(\phi_C + \phi_R) - \frac{\Delta x^2}{8}\,\text{CURV}_r, \tag{29}$$

using an obvious notation, the old value of ϕ_r having been used for ϕ_l, of course. Since the gradient values are needed for the diffusion terms anyway, the total number of operations at each grid point exceeds that of upstream-plus-central differencing by only two subtractions and one multiplication for each spatial coordinate. Implicit, tridiagonal procedures are similarly economical. In that case, one merely uses (29), treating the linear-interpolation term implicitly, with the curvature evaluated as an explicit source term.

The von Neumann stability analysis

It is instructive to investigate the von Neumann stability behaviour of the following model one-dimensional QUICK equation, using explicit forward time-differencing and constant values of u, Γ and Δx:

$$\frac{\phi_C^{n+1} - \phi_C^n}{\Delta t} = -\frac{u_0}{\Delta x_0}(\phi_r^n - \phi_l^n) + \frac{\Gamma_0}{\Delta x_0^2}(\phi_L^n + \phi_R^n - 2\phi_C^n), \tag{30}$$

where

$$\begin{aligned} &\phi_C = \phi_i, \qquad \phi_L = \phi_{i-1}, \qquad \phi_R = \phi_{i+1}, \\ &\phi_r = \tfrac{1}{2}(\phi_i + \phi_{i+1}) - \tfrac{1}{8}(\phi_{i-1} + \phi_{i+1} - 2\phi_i) \end{aligned} \tag{31}$$

and

$$\phi_l = \tfrac{1}{2}(\phi_{i-1} + \phi_i) - \tfrac{1}{8}(\phi_{i-2} + \phi_i - 2\phi_{i-1}). \tag{32}$$

Rewrite (30) as

$$\phi_i^{n+1} = \phi_i^n - c(\tfrac{1}{8}\phi_{i-2}^n - \tfrac{7}{8}\phi_{i-1}^n + \tfrac{3}{8}\phi_i^n + \tfrac{3}{8}\phi_{i+1}^n) + \alpha(\phi_{i-1}^n + \phi_{i+1}^n - 2\phi_i^n), \tag{33}$$

where the Courant number is given by

$$c = \frac{u_0 \Delta t}{\Delta x_0}, \tag{34}$$

and the diffusion parameter is

$$\alpha = \frac{\Gamma_0 \Delta t}{\Delta x_0^2}. \tag{35}$$

Then the Fourier-component amplitude ratio can be written in terms of the phase variable $\theta = k\Delta x_0$ as

$$\mathrm{G(QUICK)} = 1 + 2\alpha(\cos\theta - 1) + [\tfrac{1}{8}c(4\cos\theta - \cos 2\theta - 3)] - \mathrm{j}c\{\sin\theta + [\tfrac{1}{8}(2\sin\theta - \sin 2\theta)]\}, \tag{36}$$

using j for the imaginary unit. This is similar in some respects to the corresponding central difference formula [1]

$$\mathbf{G}(\text{CENTRAL}) = 1 + 2\alpha(\cos\theta - 1) - \mathbf{j}c\sin\theta\ , \tag{37}$$

the difference being given by the terms in square brackets in (36). Since the locus of $\mathbf{G}(\theta)$ must not lie outside the unit circle for formal von Neumann stability, one necessary condition (for $\theta = \pi$ in (36)) is

$$\text{QUICK:} \qquad \alpha + c/4 \leqslant \tfrac{1}{2}\ , \tag{38}$$

which is more restrictive than the corresponding central difference condition $\alpha \leqslant \frac{1}{2}$.

By considering the curvature of the G locus near the point (1, 0) and requiring it to be not less than that of the unit circle, another necessary condition is

$$\text{QUICK:} \qquad c^2 \leqslant 2\alpha\ , \tag{39}$$

which is the same as the corresponding central difference condition, since it turns out that the extra terms in (36) contribute zero curvature at this point.

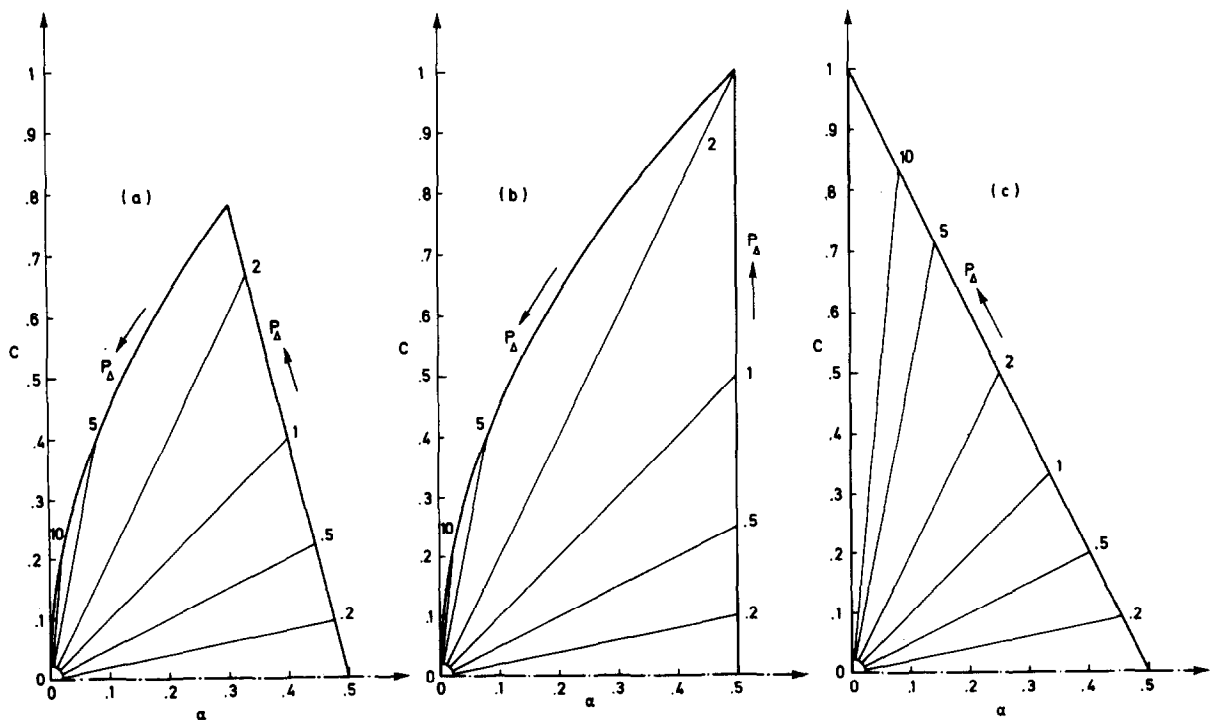

Fig. 11. Stability ranges in the (α, c) pane for: (a) QUICK, (b) central differencing and (c) upstream differencing (central differencing is used for diffusion in each case).

The limiting values of the two conditions (38) and (39) are equivalent if $(2 - 4\alpha)^2 = 2\alpha$; i.e. the cross-over values of α and c are given by

$$\alpha^* = 0.3048 \tag{40}$$

and

$$c^* = 0.7808 , \tag{41}$$

to four significant figures. By considering a number of G loci in the critical region of the (α, c) plane near (α^*, c^*), it is found that the necessary stability conditions (38) and (39) are also sufficient.

Fig. 11a shows the stability region of the (α, c) plane for (36). Lines of constant grid Péclet number $P_\Delta = c/\alpha$ are also shown for reference. Fig. 11b shows the corresponding plot for central differencing. Although the von Neumann stability conditions for the QUICK scheme are more restrictive than those for central differencing, it should be remembered that, for practical purposes, central differencing is often restricted to the range $P_\Delta \lesssim 2$, whereas the QUICK scheme is able to utilize the complete (α, c) stability region, and more. [In this latter respect, well-behaved solutions are attainable for finite c and $\alpha = 0$ ($P_\Delta = \infty$), as seen below.] For reference, fig. 11c shows the (α, c) diagram for upstream differencing.

In fig. 12 some representative QUICK G loci are given for a number of combinations of α and c values. If $\alpha = 0$, the G locus of (36) is as shown in fig. 13. As mentioned previously, this locus has zero curvature at (1, 0) and would therefore require $c \to 0$ for strict satisfaction of von Neumann stability conditions. However, because of the shape of the locus, it is possible to contain most of it within the unit circle for practical c-values, with only a small range of very long wavelength modes unstable in the von Neumann sense. This means that in practice it is possible to obtain (explicitly) infinite P_Δ solutions to problems containing finite length scales, as will be seen below. This is in sharp contrast to the central difference case, for which the diagram corresponding to fig. 13 is a vertical straight-line segment between the points $(1, \pm c)$ for which all wavelengths are unstable for any nonzero c.

4. The QUICKEST * method for unsteady flows

In unsteady flows which are primarily convective (i.e. small diffusion and source terms) field variations are carried along at the local fluid velocity. In fig. 14 this idea is portrayed for one coordinate direction, in which (for simplicity) the velocity component is assumed to be constant. A simple finite difference equation for convection alone can be written

$$(\phi_i^{n+1} - \phi_i^n)/\Delta t = u(\tilde{\phi}_l - \tilde{\phi}_r)/\Delta x , \tag{42}$$

where the $\tilde{\phi}$'s represent the respective average wall values over the time increment Δt. Using linear interpolation between node values (central differencing), the average wall values can be estimated by assuming that the ϕ-profile is swept downstream unchanged. This gives

* QUICK with Estimated Streaming Terms.

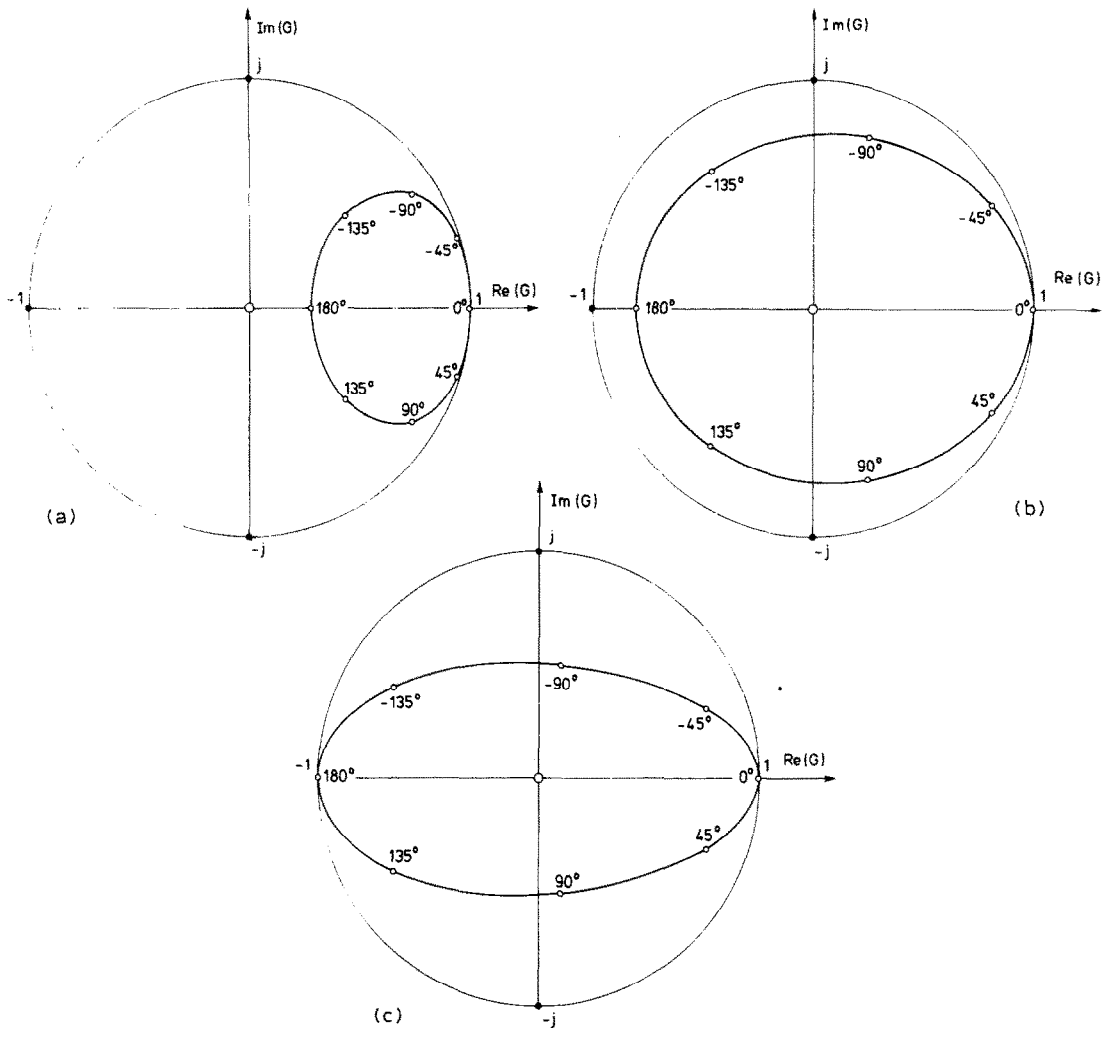

Fig. 12. Typical stable G loci for the **QUICK** method: (a) $c = 0.4$, $\alpha = 0.08$, $P_\Delta = 5$ (stability limited by curvature near + 1); (b) $c = 0.6$, $\alpha = 0.3$, $P_\Delta = 2$ (intermediate locus); (c) $c = 0.4$, $\alpha = 0.4$, $P_\Delta = 1$ (stability limited by −1). Values of the phase parameter $\theta = k\Delta x$ are shown on each curve.

$$\tilde{\phi}_l = \tfrac{1}{2}[(\phi^n_{i-1} + \phi^n_i) - c(\phi^n_i - \phi^n_{i-1})] \tag{43}$$

and

$$\tilde{\phi}_r = \tfrac{1}{2}[(\phi^n_i + \phi^n_{i+1}) - c(\phi^n_{i+1} - \phi^n_i)], \tag{44}$$

as shown in fig. 15. This can be generalized to include a variable convecting velocity field, in which case (42) becomes

$$\phi^{n+1}_i = \phi^n_i + \tfrac{1}{2}c_l[(\phi^n_{i-1} + \phi^n_i) - c_l(\phi^n_i - \phi^n_{i-1})] - \tfrac{1}{2}c_r[(\phi^n_i + \phi^n_{i+1}) - c_r(\phi^n_{i+1} - \phi^n_i)], \tag{45}$$

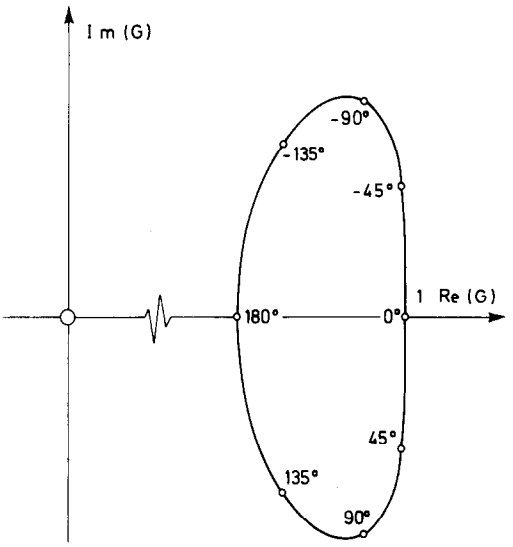

Fig. 13. Complex amplitude locus for the **QUICK** method in the nondiffusive limit $\alpha = 0$, $P_\Delta = \infty$ (note the zero curvature near +1).

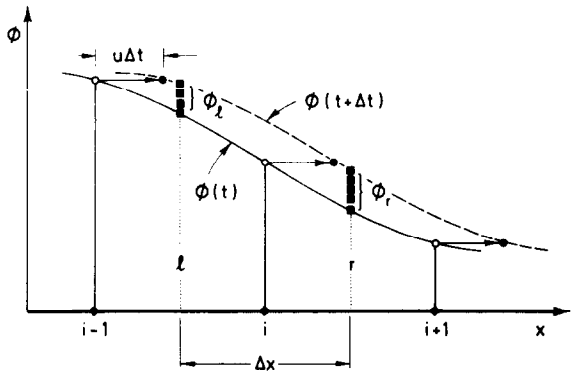

Fig. 14. Estimating cell wall values ϕ_l and ϕ_r assuming pure convection (for constant velocity the curve given by $\phi(t)$ is simply translated to the right by a distance $u\Delta t$).

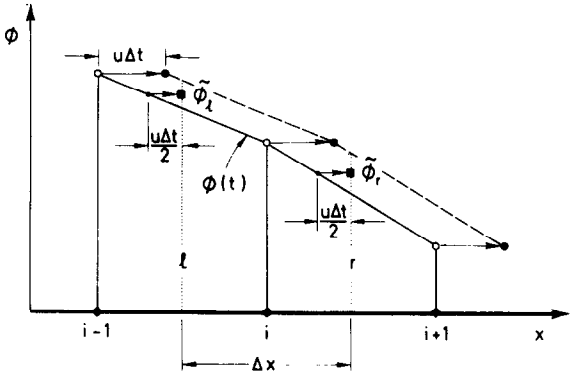

Fig. 15. Convective estimate of average wall values $\tilde{\phi}_l$ and $\tilde{\phi}_r$ assuming linear interpolation between node values for $\phi(t)$.

where c_l and c_r are, respectively, the left and right wall values of the Courant number. This equation represents the one-dimensional conservation form of Leith's method [16, 17] in the absence of physical diffusion. For constant c, (45) can be written

$$\phi_i^{n+1} = \phi_i^n - \frac{c}{2}(\phi_{i+1}^n - \phi_{i-1}^n) + \frac{c^2}{2}(\phi_{i-1}^n + \phi_{i+1}^n - 2\phi_i^n), \tag{46}$$

which is formally equivalent to the forward-time, central difference equation for the case of convection *plus* diffusion, with a numerical diffusion coefficient $u^2\Delta t/2$ [1].

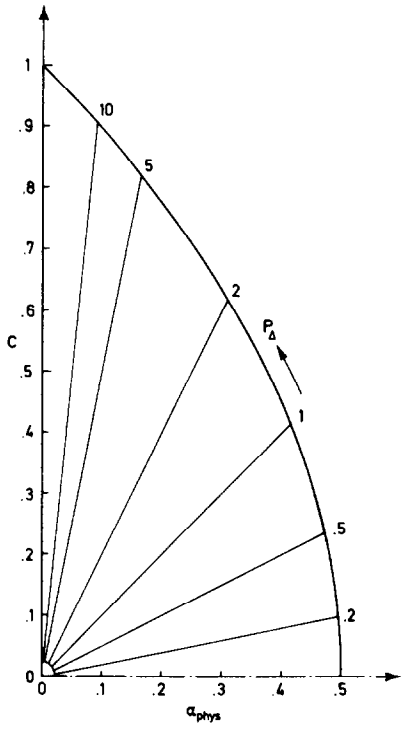

Fig. 16. Stability range in the (α_{phys}, c) plane for Leith's method (47).

If it is assumed that the same streaming estimate of $\tilde{\phi}_l$ and $\tilde{\phi}_r$ can be made when physical diffusion is present, the full convection-diffusion equation becomes (for constant parameters)

$$\phi_i^{n+1} = \phi_i^n - \frac{c}{2}(\phi_{i+1}^n - \phi_{i-1}^n) + \left(\alpha_{phys} + \frac{c^2}{2}\right)(\phi_{i-1}^n + \phi_{i+1}^n - 2\phi_i^n) . \tag{47}$$

The von Neumann stability analysis of (47) can be performed directly by analogy with (37) with α replaced by $\alpha_{phys} + c^2/2$. This results in the necessary and sufficient conditions

$$\text{LEITH:} \qquad 0 \leqslant \alpha_{phys} \leqslant (1 - c^2)/2 . \tag{48}$$

The stability region in the (α_{phys}, c) plane is shown in fig. 16, where again constant-P_Δ radii have been shown for reference.

A similar streaming estimation procedure can be used in conjunction with quadratic upstream interpolation. In this case, since curvature terms are included in the interpolation formula for wall values, it is consistent to include similar curvature terms in the modelling of the time-difference term. Consider the following *exact* integral formulation for purely convective flow:

$$\int_{-\frac{\Delta x_l}{2}}^{\frac{\Delta x_r}{2}} \phi^{n+1}\,\mathrm{d}\xi - \int_{-\frac{\Delta x_l}{2}}^{\frac{\Delta x_r}{2}} \phi^{n}\,\mathrm{d}\xi = \int_0^{\Delta t} u_l \phi_l\,\mathrm{d}\tau - \int_0^{\Delta t} u_r \phi_r\,\mathrm{d}\tau$$

$$= \int_0^{\Delta \zeta_l} \phi_l^n(\zeta_l')\,\mathrm{d}\zeta_l' - \int_0^{\Delta \zeta_r} \phi_r^n(\zeta_r')\,\mathrm{d}\zeta_r' , \tag{49}$$

where

$$\Delta\zeta_l = \int_0^{\Delta t} u_l\,\mathrm{d}\tau = \tilde{u}_l \Delta t = \tilde{c}_l\,\Delta x , \tag{50}$$

and similarly for $\Delta\zeta_r'$, as shown in fig. 17 for constant Δx. Using the appropriate QUICK formulas for $\phi_l^n(\zeta_l')$ and $\phi_r^n(\zeta_r')$, the Lagrangian integrals on the right-hand side of (49) can be approximated by

$$\text{RHS} = \Delta x\,\tilde{c}_l \left(\phi_l^n - \frac{\Delta x}{2}\tilde{c}_l\,\text{GRAD}_l + \frac{\Delta x^2}{6}\tilde{c}_l^2\,\text{CURV}_l\right)$$

$$- \Delta x\,\tilde{c}_r \left(\phi_r^n - \frac{\Delta x}{2}\tilde{c}_r\,\text{GRAD}_r + \frac{\Delta x^2}{6}\tilde{c}_r^2\,\text{CURV}_r\right) , \tag{51}$$

where

$$\text{GRAD}_l = (\phi_i^n - \phi_{i-1}^n)/\Delta x , \qquad \text{CURV}_l = (\phi_{i-2}^n + \phi_i^n - 2\phi_{i-1}^n)/\Delta x^2 ,$$

$$\phi_l^n = \tfrac{1}{2}(\phi_{i-1}^n + \phi_i^n) - \frac{\Delta x^2}{8}\,\text{CURV}_l \tag{52}$$

and

$$\text{GRAD}_r = (\phi_{i+1}^n - \phi_i^n)/\Delta x , \qquad \text{CURV}_r = (\phi_{i-1}^n + \phi_{i+1}^n - 2\phi_i^n)/\Delta x^2 ,$$

$$\phi_r^n = \tfrac{1}{2}(\phi_i^n + \phi_{i+1}^n) - \frac{\Delta x^2}{8}\,\text{CURV}_r , \tag{53}$$

when $\tilde{c}_l$ and $\tilde{c}_r$ are both positive.

The corresponding form of the left-hand side of (49) becomes

$$\text{LHS} = \Delta x \left[\left(\phi_i^{n+1} + \frac{\Delta x^2}{24}\,\text{CURV}_r^{n+1}\right) - \left(\phi_i^n + \frac{\Delta x^2}{24}\,\text{CURV}_r^n\right)\right]$$

$$= \Delta x \left[(\phi_i^{n+1} - \phi_i^n) + \frac{\Delta x^2}{24}(\text{CURV}_r^{n+1} - \text{CURV}_r^n)\right] . \tag{54}$$

Now, the curvature time-difference can be interpreted approximately as

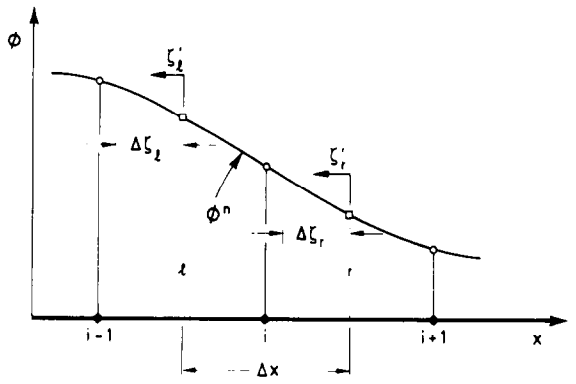

Fig. 17. Lagrangian coordinates ζ'_l and ζ'_r for estimating average wall values under conditions of high convection.

$$\frac{\Delta t}{24}\frac{\partial}{\partial t}\left(\frac{\partial^2\phi}{\partial x^2}\Delta x^2\right) = -\frac{\Delta t\,\Delta x^2}{24}\frac{\partial^2}{\partial x^2}\left[\frac{\partial(u\phi)}{\partial x}\right] \tag{55}$$

from the governing differential equation. For simplicity, assume $u \approx$ const. Then

$$\frac{\Delta t}{24}\frac{\partial}{\partial t}\left(\frac{\partial^2\phi}{\partial x^2}\Delta x^2\right) \approx -\left(\frac{u\,\Delta t}{\Delta x}\right)\frac{\Delta x^2}{24}\,(\mathrm{CURV}_r^n - \mathrm{CURV}_l^n) \tag{56}$$

to a degree of approximation consistent with (51). Combining (51), (54) and (56), the corresponding form of the QUICKEST difference equation incorporating the estimated streaming terms becomes (for constant c and Δx)

$$\phi_i^{n+1} = \phi_i^n - c\left\{\begin{array}{l}\left[\tfrac{1}{2}(\phi_i^n + \phi_{i+1}^n) - \dfrac{\Delta x}{2}\,c\,\mathrm{GRAD}_r - \dfrac{\Delta x^2}{6}(1 - c^2 - 3\alpha)\,\mathrm{CURV}_r\right] \\ -\left[\tfrac{1}{2}(\phi_{i-1}^n + \phi_i^n) - \dfrac{\Delta x}{2}\,c\,\mathrm{GRAD}_l - \dfrac{\Delta x^2}{6}(1 - c^2 - 3\alpha)\,\mathrm{CURV}_l\right]\end{array}\right\}$$

$$+\,\alpha\left[\left(\Delta x\,\mathrm{GRAD}_r - \frac{\Delta x^2}{2}\,c\,\mathrm{CURV}_r\right) - \left(\Delta x\,\mathrm{GRAD}_l - \frac{\Delta x^2}{2}\,c\,\mathrm{CURV}_l\right)\right], \tag{57}$$

where the physical diffusion terms have been included using a convective estimate of the average wall gradients, and, for third-order consistency, the effect of diffusion on the individual wall values has also been included. This equation differs from the central difference (Leith) formula of (47) by the appearance of the quantity

$$\Delta x^2 c\,[\tfrac{1}{6}(1 - c^2) - \alpha]\,(\mathrm{CURV}_r - \mathrm{CURV}_l) \tag{58}$$

on the right-hand side.

A von Neumann analysis of (57) results in an amplitude ratio of

$$G(\theta) = 1 + 2(\alpha + \tfrac{1}{2}c^2)(\cos\theta - 1) + [\tfrac{1}{6}c(1 - c^2 - 6\alpha)(4\cos\theta - \cos 2\theta - 3)]$$

$$- jc\{\sin\theta + [\tfrac{1}{6}(1 - c^2 - 6\alpha)(2\sin\theta - \sin 2\theta)]\}\ , \tag{59}$$

which should be compared with (36) and (37). The corresponding formula for Leith's method is obtained from (59) by deleting the terms in square brackets.

One necessary condition for stability, $G(\pi) \geqslant -1$, results in

QUICKEST :

$$\alpha \leqslant \frac{(3 - 2c)(1 - c^2)}{6(1 - 2c)} \qquad \text{if } c < \tfrac{1}{2}\ , \tag{60}$$

$$\alpha \geqslant \frac{(3 - 2c)(c^2 - 1)}{6(2c - 1)} \qquad \text{if } c > \tfrac{1}{2}\ . \tag{61}$$

As with Leith's method, the curvature condition near $G = 1$ requires only the physically nonrestrictive condition

QUICKEST: $$\alpha \geqslant 0\ . \tag{62}$$

Conditions (60)–(62) are necessary but not sufficient for stability. By parametric scanning throughout a large range of α and c values the complete stability region for the QUICKEST method can be found, as shown in fig. 18. Because of the consistent third-order estimation of the average flux values of ϕ and $(\partial\phi/\partial x)$ the stability range (as well as the magnitude and phase accuracy) of G is considerably improved over other explicit methods. Of particular interest is the significant region above $c = 1$ for finite α.

Note that each of the QUICKEST equation (57), Leith's method (47), and upstream differencing becomes exact in the limit $\alpha = 0$, $c = 1$ since all these equations then degenerate to point-to-point transfer:

$$\phi_i^{n+1} = \phi_{i-1}^n\ . \tag{63}$$

Finally, for the purpose of comparing the computational performance of the methods considered in the next sections, it is instructive to write the upstream difference equation for pure convection

$$\phi_i^{n+1} = \phi_i^n - c(\phi_i^n - \phi_{i-1}^n) \tag{64}$$

in a form corresponding to (47), if possible, by adding and subtracting appropriate terms. This is in fact possible and results in an identification of the effective numerical diffusion of upstream differencing relative to Leith's method by replacing α_{phys} in (47) by $\alpha_{eff} = c(1 - c)/2$, which corresponds to a numerical diffusion coefficient of

$$\Gamma_{num} = u\Delta x(1 - c)/2\ . \tag{65}$$

As pointed out by Roache [1], (65) is the appropriate formula for the numerical diffusion of up-

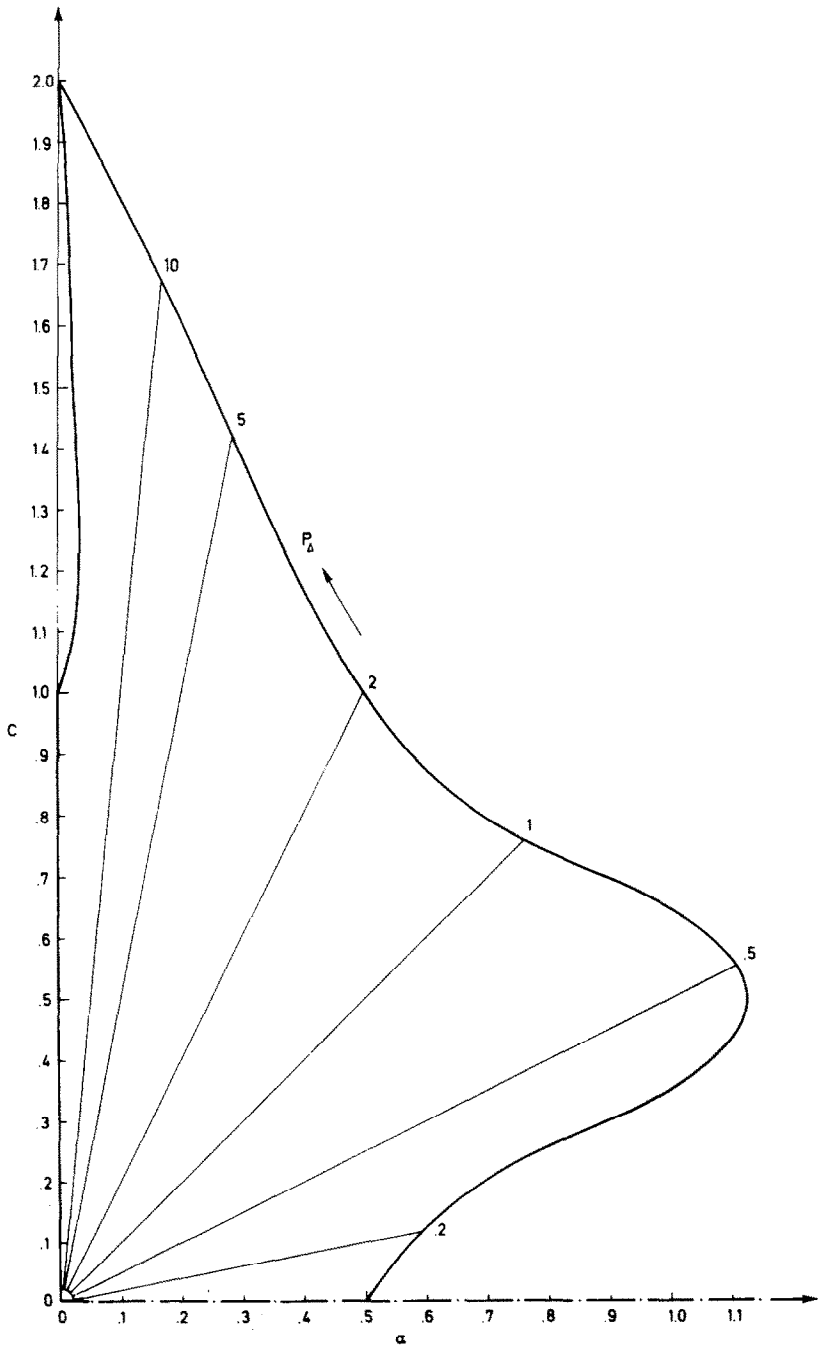

Fig. 18. Stability range in the (α, c) plane for the QUICKEST method (57) (note the extended range, in particular above $c = 1$). The nondiffusive limit $\alpha = 0$ ($P_\Delta = \infty$) is unstable for $1 < c < 2$ and again for $c > 2$.

stream differencing in an unsteady flow which is primarily convective. The previous formula given by (16) is appropriate for a steady flow in which there is a balance between convection, diffusion

and source terms. The effects of the numerical diffusion associated with upstream differencing are demonstrated in the model problems studied in the next sections.

5. Model problem I

The first model problem is a one-dimensional constant-coefficient version of (1), including a prescribed steady source term

$$\frac{\partial \phi}{\partial t} = - u_0 \frac{\partial \phi}{\partial x} + \Gamma_0 \frac{\partial^2 \phi}{\partial x^2} + S(x) \tag{66}$$

with an initial condition

$$\phi(x, 0) = 0 , \tag{67}$$

an upstream boundary condition

$$\phi(0, t) = 0 \tag{68}$$

and various downstream boundary conditions. As will be seen, some care must be taken in prescribing numerical outflow boundary conditions.

Forward time differencing is used together with central differencing for the diffusion term and either QUICK, upstream or central differencing for the convective term, using a constant grid spacing. A relatively simple form is assumed for the source term, allowing an exact steady-state solution to be obtained by standard methods. The various numerical solutions are run until a satisfactory steady state is achieved, and this is compared with the corresponding exact solution. Details of the transient behaviour are not discussed.

Infinite grid Péclet number

The grid Péclet number is defined as

$$P_\Delta = \frac{u_0 \Delta x}{\Gamma_0} . \tag{69}$$

In the nondiffusive case $\Gamma_0 = 0$ and P_Δ is infinite. Mathematically, (66) becomes a first-order differential equation with respect to x, thus requiring only a single boundary condition, such as (68). Numerical boundary conditions depend on the type of convective differencing used.

Explicit steady-state solutions were obtained for both QUICK and upstream convective differencing schemes, using a time step corresponding to $c = 0.1$. Note that very long wavelengths in the QUICK computation are formally unstable in the von Neumann sense for a nonzero c when $P_\Delta = \infty$ (see figs. 11a and 13). However, the growth rate of these instabilities is evidently extremely slow, and a perfectly satisfactory steady state can be achieved for this problem – involving as it does a finite length scale defined by $S(x)$ – with absolutely no sign of unstable modes.

The upstream boundary condition (68) was specified to be a wall value (rather than a node). In order for the *solution* to satisfy this condition to second order (i.e. by linear interpolation), the upstream difference computation requires the specification of an image node value (i.e. a value at a hypothetical node $\Delta x/2$ upstream of the physical boundary) of $\phi_0^{n+1} = -\phi_1^n$ at each time step of the computation. This gives a solution which is consistent (to second order) with a computation using a node at the physical boundary. The alternative of specifying (68) as a wall value in the upstream difference convection simulation term (rather than in the overall solution) is equivalent to specifying $\phi_0^{n+1} \equiv 0$ and leads to a first-order error in the solution at the boundary. As will be seen, this is no worse than the first-order error in the upstream difference solution in the remainder of the computation region. However, since it is possible to insure that the solution at least satisfies the correct boundary condition to second order, the first-mentioned numerical boundary condition has been chosen. It should be noted that many upstream difference schemes in common use often adopt the alternative modelling of upstream boundary conditions [4], thus accepting a first-order error in the solution at the respective boundaries as well as in the remainder of the field. For the case $P_\Delta = \infty$ no downstream boundary condition is required for the upstream difference computation.

In modelling boundary conditions for the QUICK computation, both the wall value and wall gradient are needed at each end of the computational region. Upstream, the wall value is given directly by the physical boondary condition (68). The upstream wall gradient is chosen to be consistent with quadratic interpolation between the given boundary wall value and the first two interior node values ϕ_1 and ϕ_2. This implies

$$\left(\frac{\partial \phi}{\partial x}\right)_B^{n+1} = \frac{(8\phi_B + 3\phi_1^n - \phi_2^n)}{3\Delta x} \tag{70}$$

at the upstream boundary. In this particular case $\phi_B = 0$.

Far downstream, the source term in (66) is zero, and it is consistent to take the downstream wall value equal to the last computed node value and set the downstream gradient to zero. Thus the downstream numerical boundary condition is equivalent to $\partial\phi/\partial x = 0$ for the QUICK computation. An alternative procedure is to require $\partial^2\phi/\partial x^2 = 0$ numerically at the downstream wall. In practice, the two conditions appear to give almost identical results although the zero-curvature condition, being less restrictive than the zero-gradient condition, allows some very weak non-physical transients to damp out more slowly in time.

Fig. 19 shows the steady-state results for $P_\Delta = \infty$ for a source function of the form shown. It is clear that the QUICK solution is almost graphically indistinguishable from the exact result, whereas the upstream difference computation is significantly in error (~43% downstream). This is a situation in which the numerical size of the error is greater than the formal (first) order might suggest. For reference, an exact solution for $P_\Delta = 2$ has also been shown. As will be seen, the upstream difference computation for $P_\Delta = \infty$ is identical to the central difference computation for $P_\Delta = 2$, which of course again emphasizes the fact that numerical diffusion of magnitude $\Gamma_{num} = u_0 \Delta x/2$ is introduced by upstream differencing in steady flow computations.

The simultaneous stability and accuracy of the QUICK solution is impressive when it is realized that it has been obtained by use of a computationally inexpensive explicit algorithm using very practical values for the grid spacing and time step in the absence of any stabilizing effects of diffusion – either physical or numerical.

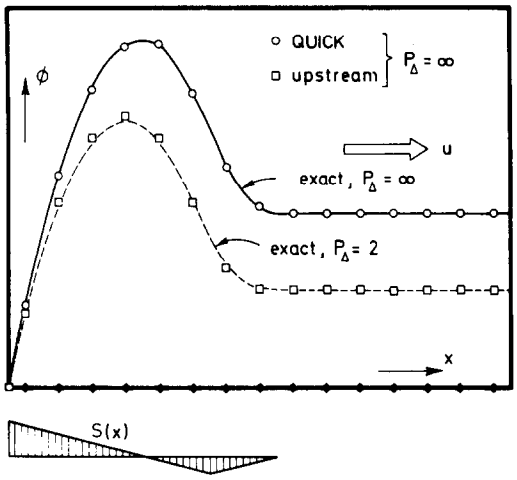

Fig. 19. Steady-state solution of (66) in the nondiffusive case $\Gamma = 0$ ($P_\Delta = \infty$). The QUICK solution (open circles) is almost indistinguishable from the exact curve. By contrast, the upstream difference solution (open squares) is grossly in error, being close to the exact diffusive solution for $P_\Delta = 2$ (dashed curve).

Finite grid Péclet number

Fig. 20 shows QUICK computations and exact solutions for $P_\Delta = 1$, 2 and 5. For reference, the QUICK and exact solutions for $P_\Delta = \infty$ are also shown. In all cases, the QUICK-computed points lie almost indistinguishably close to the corresponding exact curve. In this case, (66) is second order in x, and two physical boundary conditions are appropriate. For the cases shown, either zero-gradient or zero-curvature downstream gave identical results since these are physically equivalent from (66) in the steady state far downstream where $S(x) = 0$. Of course, it is physically possible to specify a downstream condition on the value of ϕ – e.g. $\phi(L) = 0$. The effects of this will be considered in the next subsection.

Fig. 21 shows the results for central differencing by explicit solution of (66) for $P_\Delta = 1$, 2 and 5. In this case, an explicit solution for $P_\Delta = \infty$ is not attainable although in principle any finite value of P_Δ can be treated, provided a small enough time step is used (and provided compatible

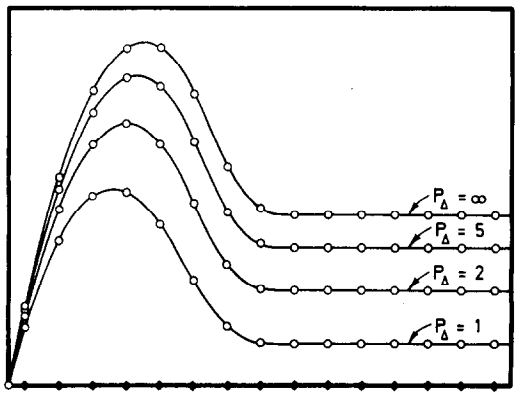

Fig. 20. QUICK computations (open circles) and exact solutions (curves) of (66) for $P_\Delta = 1$, 2, 5 and ∞.

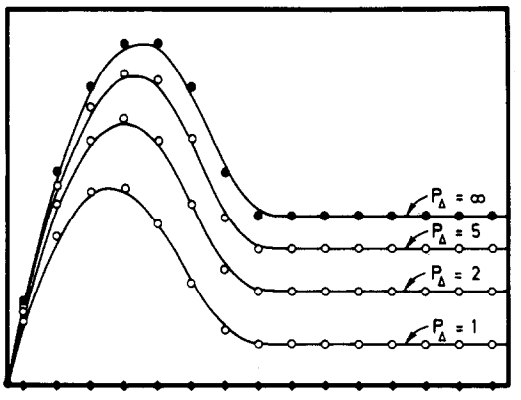

Fig. 21. Central difference computations and exact solutions of (66) for $P_\Delta = 1$, 2 and 5 (open circles) and $P_\Delta = \infty$ (black dots) (see text for explanation of the result for $P_\Delta = \infty$).

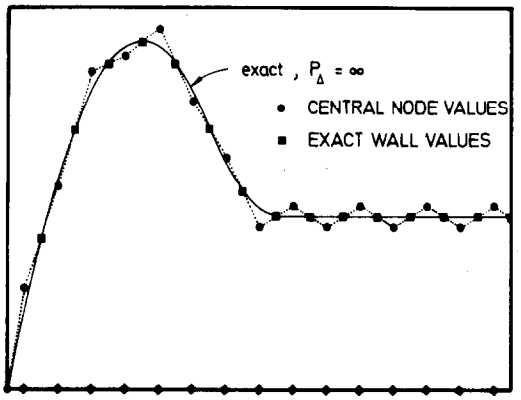

Fig. 22. The effect of perturbing a node value anywhere downstream in a central difference solution for $P_\Delta = \infty$ (the construction is based on the fact that the difference between adjacent wall values must be the same for both the computed and exact solutions).

numerical boundary conditions are specified). The central difference solution for $P_\Delta = \infty$ shown in fig. 21 was obtained by a procedure analogous to that leading to (21) for upstream differencing. The steady-state nondiffusive central difference solution must be such that the linearly interpolated solution wall values are identical to the exact wall values, assuming that at least one computed wall value (in this case upstream) has been fixed to match the corresponding exact value. This gives an algorithm for constructing the central difference solution for $P_\Delta = \infty$ from the exact solution by starting at a downstream location where $\partial\phi/\partial x = 0$ (or $\partial^2\phi/\partial x^2 = 0$) and working back upstream. At any node value i the algorithm becomes

$$\phi_i = 2\phi^E_{i+\frac{1}{2}} - \phi_{i+1} \,, \tag{71}$$

where $\phi^E_{i+\frac{1}{2}}$ is the exact adjacent downstream wall value.

This construction incidentally gives some insight into the strong susceptibility of central differencing to wiggles in cases of large P_Δ. Fig. 22 shows the effect of perturbing one node value *anywhere* downstream for the case $P_\Delta = \infty$. Since the interpolated values must still match the exact wall values (as indicated by the dotted lines), the wiggles propagate throughout the entire solution. Of course, finite P_Δ values would dampen the wiggles, but the damping is not strong for large P_Δ. Also, notice that the specification of an upstream node value (rather than a wall value) would lead to an incompatibility between the upstream and downstream conditions. This type of incompatibility can lead to divergence in an iterative procedure for large P_Δ.

Returning now to fig. 21, it is clear that the second-order convective truncation error causes the computed points to lie off to the locally convex side of the exact curve, with the magnitude of the discrepancy being worse in regions of higher curvature. Of course, the direct error contribution of the diffusive terms (being consistent with quadratic interpolation) is only third order. For this reason, linearly interpolated wall values of the central difference solutions for finite P_Δ are very close to the corresponding exact values (as is identically the case for infinite P_Δ). The numerical boundary conditions for these central difference solutions comprise a given upstream wall value (= 0 in this case), an upstream gradient modelled by $(\partial\phi/\partial x)_B = 2\phi_1/\Delta x$, and zero-gradient or zero-curvature downstream. As previously noted, the central difference solution for $P_\Delta = 2$ is identical to the upstream difference computation for $P_\Delta = \infty$ shown in fig. 19. Upstream difference computations have, of course, been performed for finite P_Δ's as well, but the results are so overwhelmingly artificially diffusive that there is no point in presenting them.

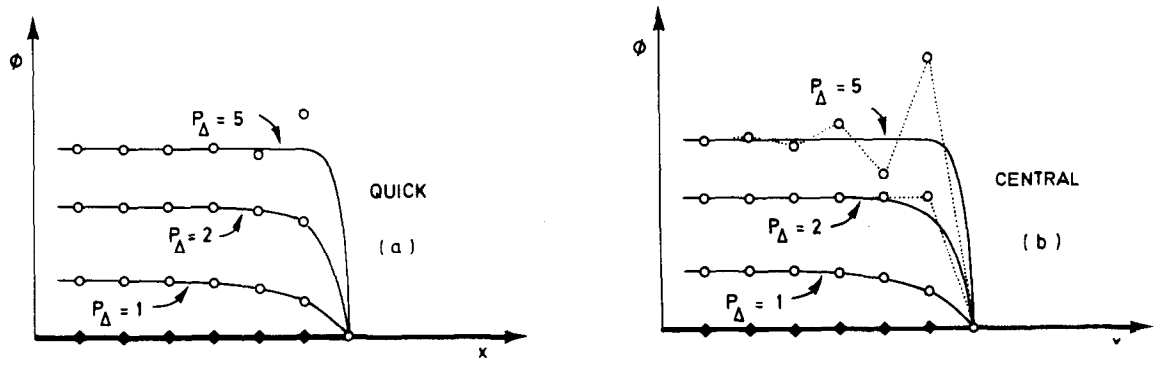

Fig. 23. Effect of imposing a downstream boundary condition on the value of ϕ in the solution of (66) for $P_\Delta = 1$, 2 and 5 using (a) the QUICK method and (b) central differencing.

The downstream boundary condition

So far, zero-gradient or zero-curvature has been assumed as the appropriate numerical boundary condition far downstream. In most cases of practical interest zero-curvature forms an adequate out-flow numerical boondary condition for both QUICK and central difference schemes (for upstream differencing, of course, an outflow boundary condition is not needed for the convective term). However, since (66) is second order for finite P_Δ, it is physically possible to specify a non-zero gradient or value at a given downstream location in addition to the upstream condition. Alternatively, it is mathematically possible to specify both value and gradient at a single point; but, since this is much less likely to occur in a fluid dynamics problem, it need not be considered further here.

Fig. 23 shows the effect of imposing $\phi(L) = 0$ for the problem defined by (66)–(68) for $P_\Delta = 1$, 2 and 5. For simplicity, only the downstream portion of the solutions are shown, the upstream portions being identical to those shown in figs. 18 and 19 (except as noted below for central differencing at high Péclet number). For $P_\Delta = 1$ and 2 the QUICK solution is monotonic, although some slight wiggles appear at $P_\Delta = 5$. It is not difficult to show that the transition between monotonic and oscillatory behaviour occurs at $P_\Delta = 8/3$ for the QUICK scheme. The corresponding cross-over value for central differencing is $P_\Delta = 2$ [1], as seen in fig. 23b. Note that the central differencing wiggles are stronger in both magnitude and extent than the QUICK wiggles. As P_Δ is increased further, an important distinction between the QUICK and central difference schemes becomes apparent: the central difference wiggles continue to grow in both magnitude and extent, ultimately corrupting the entire solution at a finite P_Δ (which depends on the total number of grid points in the solution domain), whereas the QUICK wiggles have a limiting form as $P_\Delta \to \infty$, as shown in fig. 24 (obtained both explicitly with $c = 0.1$, and implicitly using Thomas' tridiagonal algorithm). Note that the practical penetration distance is less than eight grid points, thus confining the anomalous behaviour to twe region close to the singularity. Of course, the correct way to treat this type of problem, should it arise, is to use the approach of singular perturbation analysis [8], i.e. relax the downstream value boundary condition for the outer solution (replacing it by a zero-curvature condition) and then resolve the resulting discontinuity (if necessary) by a fine-grid computation of the inner solution – or the two steps can be combined by using an appropri-

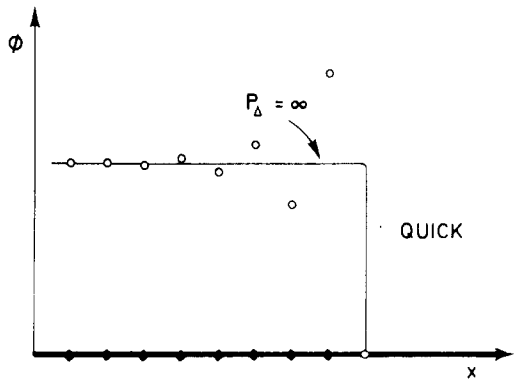

Fig. 24. Effect of imposing a downstream boundary condition on the value of ϕ in the solution of (66) when $P_\Delta \to \infty$ using the QUICK method.

ately adjusted variable grid.

In most fluid dynamic investigations of practical interest a little care in the specification of numerical boundary conditions can avoid the problem of wiggles altogether. However, should a difficulty occur inadvertently, the relatively docile behaviour of the QUICK scheme stands a much better chance of avoiding a catastrophe than the wild oscillations of the corresponding central difference computation for high P_Δ. Subsequent boundary condition adjustment or local grid refinement can then insure a numerically faithful solution to the modelled physical problem over the entire computational region.

6. Model problem II

The second type of model problem to be considered demonstrates the practical advantages of the QUICKEST algorithm, discussed earlier, for unsteady flows which are primarily convective. One of the simpler model problems of this type is the unsteady one-dimensional convection at constant velocity of an initial step in ϕ without physical diffusion. Fig. 25a shows the exact solution, for reference. In fig. 25b the results of using upstream differencing for convection and forward time differencing are shown for a Courant number of 0.5. The highly diffusive nature of the upstream difference solution corresponds closely to the exact (error-function) solution of a physical problem including diffusion of a magnitude $\Gamma_{num} = u\Delta x(1-c)/2$. Clearly, the effect of numerical diffusion is more pronounced at smaller c values. At $c \equiv 1$ the upstream difference method is exact for this problem (as are the other methods discussed in this section). Of course in most practical applications the Courant number is required to be considerably less than unity.

Fig. 25c shows the results of applying Leith's method, given by (46), to this problem, again using $c = 0.5$. The damped oscillations behind the advancing front are typical of this central difference method. The wavelength and damping of the oscillations and, to a lesser extent, the magnitude of the first peak depend on c, as seen in fig. 26.

Fig. 25d shows the results of the QUICKEST computation, given by (57) with $\alpha = 0$. The profile remains comparatively sharp; the small undershoot and overshoot which develop are each about 5% of the step height. The profile shape is very insensitive to the value of the Courant number. For the case shown ($c = 0.5$) the profile is exactly antisymmetrical.

In fig. 27 the effect of including a small amount of physical diffusion ($P_\Delta = 50$) is shown for the same three methods in comparison with the exact error-function solution. The profiles shown

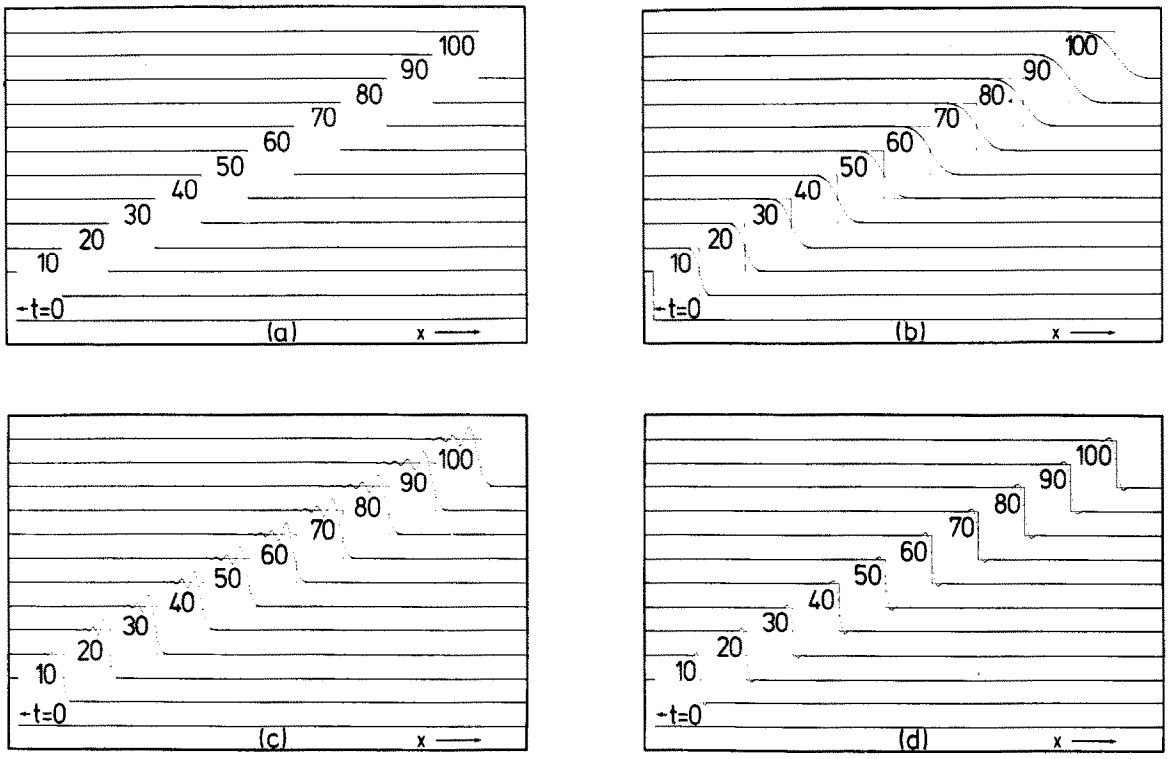

Fig. 25. Pure convection of a step: (a) exact solution, (b) upstream differencing, (c) Leith's method, (d) QUICKEST method (the Courant number is 0.5 in each computation).

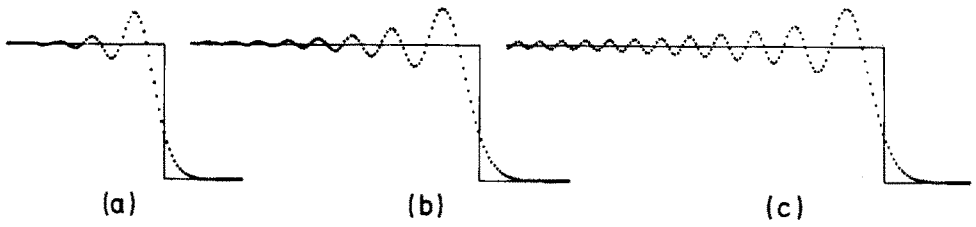

Fig. 26. Dependence of the wavelength and damping of the oscillations of Leith's method on the value of the Courant number: (a) $c = 0.7$, (b) $c = 0.25$, (c) $c = 0.05$.

are taken at a nondimensional time equivalent to $t = 100$ in fig. 25. The Courant number is 0.05 in each case. The practical advantages of the QUICKEST method over the other two techniques are dramatically demonstrated by this example.

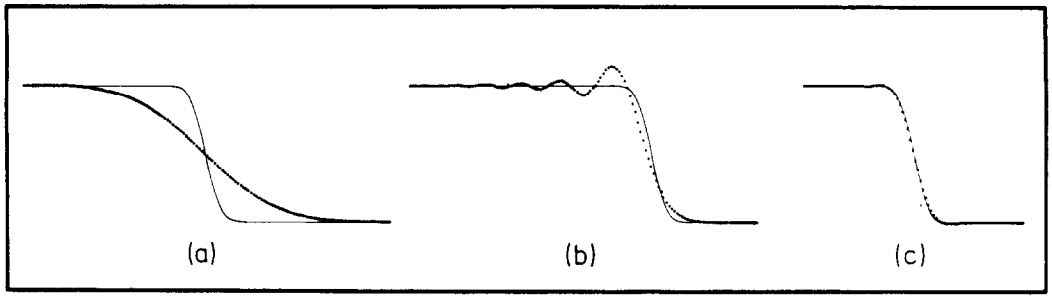

Fig. 27. Convection and diffusion of a step ($P_\Delta = 50$): (a) upstream differencing, (b) Leith's method, (c) QUICKEST method (in each case the exact (error-function) solution is shown for reference).

7. Application to an unsteady two-dimensional flow

The QUICKEST algorithm has been applied in a numerical simulation of the complex hydrodynamics and salinity transport characteristics of a large estuary (Hudson River), using an unsteady laterally-averaged model [18]. In geophysical flows of this type, longitudinal transport is almost entirely advective (in geophysical flows horizontal convection is usually called advection), whereas vertical transport is dominated by turbulent diffusion. This is the classical boundary layer situation. However, because of both local and global flow reversals, the problem is of elliptic type and cannot be treated by parabolic boundary layer methods. The large disparity in the values of the component grid Péclet numbers in the longitudinal ($P_\Delta \to \infty$) and vertical ($P_\Delta < 1$) directions allows the use of the one-dimensional QUICKEST method longitudinally with an appropriate scheme vertically (in this case, five-point).

Advective terms are relatively small in the hydrodynamic equations. For example, in the longitudinal momentum equation the primary balance is between the fluid acceleration, the unsteady pressure gradient (consisting of the driving surface-gradient term and the baroclinic term due to longitudinal density gradients) and the turbulent shear stress. However, in the salinity (or other scalar) transport equation the primary balance is between sheared horizontal advection and vertical turbulent diffusion. It is therefore essential for the numerical scheme to be able to model the advection faithfully without introducing either unphysical oscillations or artificial longitudinal numerical diffusion. Because of the necessarily large longitudinal grid spacing (of the order of kilometers) and instantaneous velocities near a meter per second, models which use upstream differencing for this type of problem [19–21] are doomed to failure because of the overwhelming artificial numerical diffusion $\Gamma_{num} = u\,\Delta x/2$.

Fig. 28 shows a semilogarithmic plot of salinity in the Hudson River against longitudinal distance from New York Harbor. The field data points represent both tidal and cross-sectional averages over a period in which the fresh-water flow and ocean tidal range were fairly constant. A numerical simulation using the QUICKEST algorithm was run for more than 30 tidal cycles using a constant fresh-water inflow at the head and a sinusoidal tidal elevation at the mouth, chosen to match the physical conditions. With this method an excellent periodic steady state is achieved. The results plotted show tidal-average salinity range at each station together with the extreme excursions throughout a single tidal cycle. The detailed behaviour depends strongly on the vertical turbulence model used. However, the physically important fast decay rate above mile-point 65 is well repro-

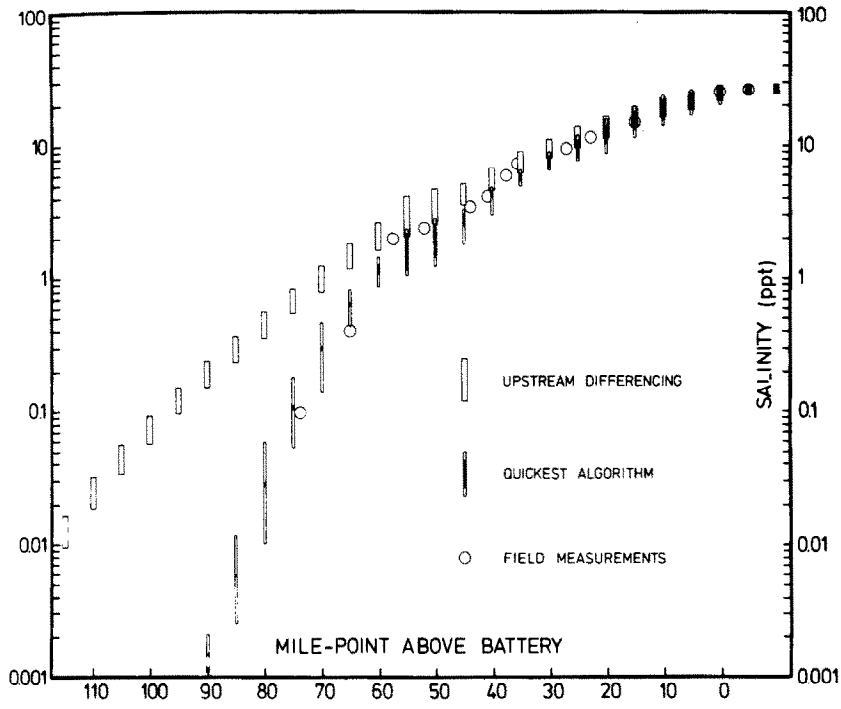

Fig. 28. Semilogarithmic plot of salinity (parts per thousand) in the Hudson River estuary against longitudinal distance from New York Harbor. The QUICKEST simulation mimics the field data quite well – especially the rapid decay rate far upstream. By contrast, the upstream difference solution shows the intolerable effect of artificial numerical diffusion.

duced and reflects the fact that the QUICKEST method does not introduce any appreciable artificial numerical diffusion.

In dramatic contrast, the results of a simulation using upstream differencing for the advection terms in the salinity equation are shown on the same graph. In this case the two-dimensional computation was cut off after only 16 tidal cycles as the numerical diffusion had obviously already swamped the physical processes with no sign of abatement of the relentless artificial upstream penetration of salt. It should be mentioned that after only about 4 or 5 tidal cycles, fair agreement with field data could be achieved with upstream differencing, provided one did not demand a periodic steady state. The pitfall of this should be clear. Unfortunately, some of the results of upstream difference methods reported in the literature [19, 20] are indeed based on only a few tidal cycles of simulation. In any case the physical settling time of a large estuary is of the order of 20 tidal cycles (roughly the time needed for a fluid particle to traverse the large gravitational recirculation cell). Clearly, any numerical model claiming quantitative predictive capability should be able to reach a periodic steady state for periodic boundary conditions. Indeed, the QUICKEST simulation reaches a satisfactory periodic steady state (from more-or-less arbitrary initial conditions) after about 25 tidal cycles.

In fig. 29, results of using central differencing (Leith's method) for the salinity advection are shown. In this case instantaneous salinity ranges are shown in order to demonstrate the unphysical wiggles which appear randomly in both space and time. Although the global features conform to field data, the continued presence of the numerical oscillations is enough to corrupt the predictive

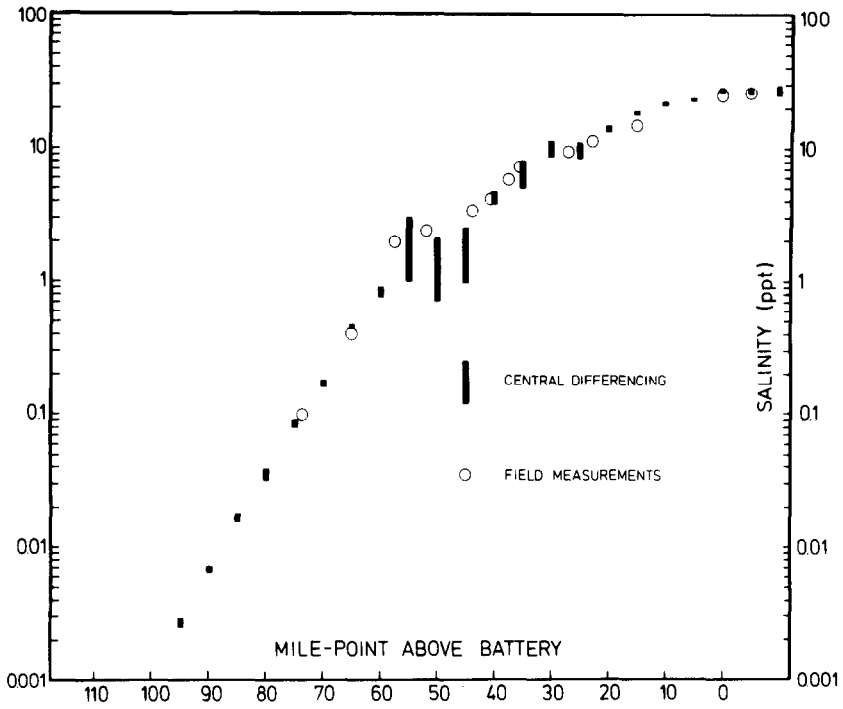

Fig. 29. In this case central differencing is used for longitudinal salinity advection. Although the global features conform to the (monotonic) field data, the continued presence of unphysical numerical oscillations destroys predictive reliability.

reliability of this method. Artificial smoothing techniques have sometimes been used with centrally differenced advection terms [22], but it is difficult to estimate the penalty involved in such procedures. Clearly, the inherent numerical stability properties of the explicit forward-time QUICKEST method give this technique a significant advantage over the potentially wiggly central difference methods when applied to complex multidimensional unsteady flows of this type.

8. Conclusion

Using quadratic upstream interpolation for convective kinematics, explicit numerical procedures have been developed for quasi-steady (QUICK) and unsteady (QUICKEST) flow situations which involve strong convection in one coordinate direction. The methods have the desirable simultaneous properties of high accuracy (third-order spatial truncation error and third-order in time for the QUICKEST method, assuming high convection conditions), inherent numerical convective stability (no wiggles, provided boundary conditions are treated properly), algorithmic simplicity (using a single forward time step explicitly) and a physically appealing foundation (based on a conservative control-volume integral formulation). Compared with upstream differencing or Leith's method, the QUICKEST method can produce a solution of comparable accuracy with a reduction in total computer time of typically a factor of about 100.

There do not seem to be any serious drawbacks to offset these advantageous features, although

the methods are clearly limited in their ability to resolve sudden jumps in value within a small number of grid points. Modifications in this respect are undoubtedly possible – at an increased cost in complexity, of course. And care must be taken to ensure that a model with good discontinuity resolution properties [23] does not grossly distort more gently peaked distributions (such as converting a Gaussian into a box, for example [15]).

Simply going to higher-order schemes does not necessarily produce a proportionate increase in accuracy. For example, ad hoc even-ordered methods [15] possess no inherent numerical convective stability and will always be plagued by wiggles in regions of strong gradients when P_Δ is large. By contrast, upstream-shifted odd-ordered methods possess inherent stability in this respect. Of course, the (all too) common first-order upstream differencing method is the simplest – and most inaccurate – of this type. Unfortunately, many workers have adopted a philosophy of falling *back* on first-order upstream differencing after experiencing trouble with second-order central differencing [24, 25] rather than going *forward* to an inherently stable – and exceedingly more accurate – third-order scheme. It is rather disconcerting to see that the recent discovery of how to incorporate first-order upstream difference techniques into finite element applications to fluid dynamics is being hailed as a break-through in that discipline [26]. On the contrary, this is a serious regression! It is to be hoped that it will not take another decade of artificially diffuse "solutions" – in this case obtained by finite element methods – to realize the gross quantitative inadequacies of first-order upstream differencing. This is particularly pertinent to potential progress in the development of turbulent transport modelling. Certainly, it makes little sense to test a sophisticated transport theory under high-P_Δ conditions with a computer code which always adds artificial diffusion corresponding to (16) to the physical diffusion terms [27].

The explicit third-order methods presented here can claim a high degree of optimality – in terms of both stability and accuracy without significant increase in complexity. There seems little motivation to go to higher order. Of course, the potentially wiggly ad hoc fourth-order schemes can be discounted from further consideration. Justification of the need for a consistent higher-order scheme must await demonstration of some particularly attractive advantage over third-order.

The one-dimensional QUICKEST method in the absence of diffusive terms ($P_\Delta \to \infty$) has an interesting interpretation when u and Δx are constant. At time $t = n\Delta t$, let $\phi^n(\xi)$ represent a cubic interpolant through the four values ϕ_{i-2}, ϕ_{i-1}, ϕ_i and ϕ_{i+1}. The space coordinate ξ is measured downstream from the ith node (in the direction of increasing index). The nondiffusive constant-velocity QUICKEST algorithm is then equivalent to

$$\phi_i^{n+1} = \phi^n(-u\Delta t)\,. \tag{72}$$

This is the four-point (cubic) analogue of Leith's three-point (quadratic) method. As pointed out by Roache [1], standard upstream and central convective differencing can also be interpreted as two-point (linear) versions of this process when u = const – the two points being either center-and-upstream or upstream-and-downstream, respectively. Higher-order methods are also possible.

Fromm's so-called second-order method of zero-average-phase error [28] can also be interpreted in terms of (72) when u = const. As with QUICKEST this is an upstream-shifted four-point method in one dimension, but instead of interpolating a cubic through the four points Fromm effectively averages two parabolas – one through $(\phi_{i-2}, \phi_{i-1}, \phi_i)$ and the other through $(\phi_{i-1}, \phi_i, \phi_{i+1})$ – in order to produce $\phi^n(\xi)$ (in the interval between $i-1$ and i). In this respect Fromm's second-order method must be considered to be actually third order for all practical purposes. In fact it is identi-

cal to QUICKEST for a Courant number of $c = 0.5$ and differs insignificantly for $0.5 < c \leqslant 1$. As $c \to 0$ Fromm's method has slightly more damping than QUICKEST (Fromm's third-difference coefficient approaches 1/4 as opposed to the value of 1/6 in the corresponding QUICKEST term). The same reasoning would indicate that Fromm's modified fourth-order method [29] is effectively fifth order.

Finally, the question of the extension of the QUICK and QUICKEST methods to flow situations in which convection dominates diffusion in more than one coordinate direction will be briefly addressed. The estuary example of sec. 7 is a case for which the one-dimensional unsteady version is adequate even through the flow is two dimensional and elliptic in nature. When convection is dominant in two or all three dimensions, the corresponding algorithm is not just the sum of the individual one-dimensional contributions. In the analogous development of Leith's method for two dimensions the simple addition of individual x and y contributions without appropriate cross terms can lead to a basic instability [16]. It is not immediately clear whether a similar effect would result from the ad hoc extension of the one-dimensional QUICKEST method to two and three dimensions. Numerical experience with the steady-state QUICK method extended to two dimensions indicates that stability can be achieved with a simple-minded seven-point scheme. However, accuracy may be impaired if transverse curvature terms are not included.

One guiding principle in developing the multidimensional methods will be that of consistency with the constant-velocity pure-convection case for constant grid spacing. In one dimension, quadratic interpolation between three points for wall values is consistent with the pure convection of a cubic curve through four points according to (72). In two dimensions a six-point quadratic interpolating surface is appropriate for computing the average flux through an individual control-volume wall, whereas the consistent cubic surface for the complete cell requires a total of ten points. These configurations are sketched in fig. 30. The three-dimensional algorithms will involve ten points (quadratic density) for each wall flux, corresponding to twenty points overall for the cubic density function. In the general case of varying convecting velocities the individual wall fluxes are computed from the respective six-point (for two dimensions) or ten-point (for three dimensions)

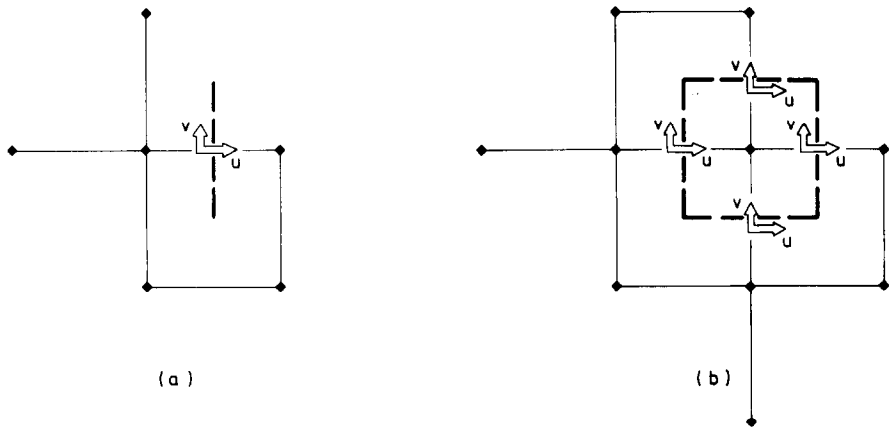

Fig. 30. Two-dimensional QUICKEST computation lattice. In (a) the six points necessary for a quadratic interpolation surface are shown for one control-volume wall; the distribution of nodes depends on the signs of the velocity components. In (b) the corresponding ten-point scheme is shown for the complete control volume, assuming the velocity components do not change sign.

lattice, but of course the individual grid points involved will depend on the local velocity directions, as will the total number of grid points involved in any particular control volume cell. An alternative to the above type of consistent (quadratic-flux, cubic-convection) formulation is the procedure of time splitting, using successive one-dimensional formulas [16, 28]. Although this preserves second-order accuracy when applied to Leith's method, additional grid points (to take account of transverse curvature) would have to be included to preserve the third-order accuracy of the QUICKEST method, thus negating the effectiveness of time splitting. A consistent conservative formulation of QUICK and QUICKEST methods for both two and three dimensions forms the subject of a future publication.

Appendix. Summary of the QUICK and QUICKEST algorithms

Fig. A.1 shows the definition of terms under conditions of variable grid spacing in one dimension. A left-to-right computational sweep is assumed. The velocity components are defined as positive to the right (as shown); their actual values may be positive or negative, independently. When u_r is negative, ϕ_{FR} will be involved in the computation. Thus the computation of the new ϕ_C value (at station i) begins with the computation of

$$\text{GRAD}_{fr} = \frac{1}{\Delta x_{fr}} (\phi^n_{FR} - \phi^n_R) = \text{GRAD}_{fr}(i) , \tag{A.1}$$

having previously set

$$\text{GRAD}_r = \text{GRAD}_r(i) = \text{GRAD}_{fr}(i-1) . \tag{A.2}$$

Then a new value can be computed for

$$\text{CURV}_R = \frac{1}{\Delta x_R} (\text{GRAD}_{fr} - \text{GRAD}_r) = \text{CURV}_R(i) , \tag{A.3}$$

having previously set

$$\text{CURV}_C = \text{CURV}_C(i) = \text{CURV}_R(i-1) . \tag{A.4}$$

Now, the actual curvature term used depends on the sign of the control-volume wall velocity component, as follows:

$$\text{CURV}_r = \begin{cases} \text{CURV}_C & \text{if } u_r \geqslant 0 , \\ \text{CURV}_R & \text{if } u_r < 0 . \end{cases} \tag{A.5}$$

The local Courant numbers and physical diffusion parameters are given by

$$c_r = u_r \Delta t / \Delta x_r = c_r(i) \tag{A.6}$$

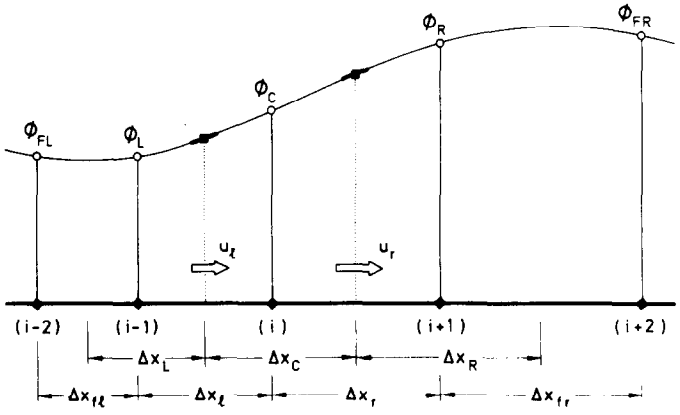

Fig. A.1. Definition of terms under conditions of variable grid spacing.

and

$$\alpha_r = \Gamma_r \Delta t / \Delta x_r^2 = \alpha_r(i) , \tag{A.7}$$

with

$$c_l = c_l(i) = c_r(i-1) \tag{A.8}$$

and

$$\alpha_l = \alpha_l(i) = \alpha_r(i-1) . \tag{A.9}$$

The new ϕ_C value is found from

$$\phi_C^{n+1} = \phi_C^n + \frac{\Delta t}{\Delta x_C}\left[u_l \phi_l^* - u_r \phi_r^* - \Gamma_l \left(\frac{\partial \phi}{\partial x}\right)_l^* + \Gamma_r \left(\frac{\partial \phi}{\partial x}\right)_r^*\right] + \Delta t \bar{S} , \tag{A.10}$$

where $\bar{S}$ is the average source term.

The individual terms in (A.10) are defined as follows:

$$\text{QUICK:} \qquad \phi_r^* = \tfrac{1}{2}(\phi_C^n + \phi_R^n) - \frac{\Delta x_r^2}{8} \text{CURV}_r = \phi_r^*(i) \tag{A.11}$$

and

$$\left(\frac{\partial \phi}{\partial x}\right)_r^* = \text{GRAD}_r(i) , \tag{A.12}$$

with

$$\phi_l^* = \phi_r^*(i-1) \quad) \tag{A.13}$$

and

$$\left(\frac{\partial\phi}{\partial x}\right)_l^* = \mathrm{GRAD}_r(i-1)\ ; \tag{A.14}$$

$$\text{QUICKEST:}\quad \phi_r^* = \tfrac{1}{2}(\phi_C^n + \phi_R^n) - \frac{\Delta x_r}{2}\, c_r \mathrm{GRAD}_r + \frac{\Delta x_r^2}{2}\,[\alpha_r - \tfrac{1}{3}(1-c_r^2)]\,\mathrm{CURV}_r = \phi_r^*(i) \tag{A.15}$$

and

$$\left(\frac{\partial\phi}{\partial x}\right)_r^* = \mathrm{GRAD}_r - \frac{\Delta x_r}{2}\, c_r \mathrm{CURV}_r = \mathrm{GRAD}_r^*(i) \tag{A.16}$$

with

$$\phi_l^* = \phi_r^*(i-1) \tag{A.17}$$

and

$$\left(\frac{\partial\phi}{\partial x}\right)_l^* = \mathrm{GRAD}_r^*(i-1)\ . \tag{A.18}$$

References

[1] P.J. Roache, Computational fluid dynamics (Hermosa, Albuquerque, NM, 1972).
[2] G.D. Raithby, A critical evaluation of upstream differencing applied to problems involving fluid flow, Comp. Meths. Appl. Mech. Eng. 9 (1976) 75–103.
[3] G.D. Raithby and K.E. Torrance, Upstream-weighted schemes and their application to elliptic problems involving fluid flow, Computers and Fluids 2 (1974) 191–206.
[4] A.D. Gosman, W.M. Pun, A.K. Runchal, D.B. Spalding and M. Wolfstein, Heat and mass transfer in recirculating flows (Academic Press, London, 1969).
[5] Proceedings of the Symposium on Turbulent Shear Flows, The Pennsylvania State University (Apr. 1977).
[6] B.E. Launder and D.B. Spalding, Mathematical models of turbulence (Academic Press, London, 1972).
[7] D.B. Spalding, A novel finite difference formulation for differential expressions involving both first and second derivatives, Int. J. Numer. Meth. Eng. 4 (1972) 551–559.
[8] M. van Dyke, Perturbation methods in fluid mechanics (Academic Press, New York, 1964).
[9] S.V. Patankar and D.B. Spalding, Heat and mass transfer in boundary layers, 2nd ed. (Intertext Books, London, 1970).
[10] G.D. Raithby, Skew-upstream differencing schemes for problems involving fluid flow, Comp. Meths. Appl. Mech. Eng. 9 (1976) 151–162.
[11] M.A. Leschziner, On the problem of numerical diffusion in first-order finite-difference schemes applied to free recirculating flows, Proceedings of the Second GAMM Conference on Numerical Methods in Fluid Mechanics, Cologne (Oct. 1977) 105–112.
[12] J.E. Fromm, The time-dependent flow of an incompressible viscous fluid, Meths. Comp. Phys. 3 (1964) 345–382.
[13] R.A. Gentry, R.E. Martin and B.J. Daly, An Eulerian differencing method for unsteady compressible flow problems, J. Comp. Phys. 1 (1966) 87–118.
[14] N.C. Steele and K.E. Barrett, A 2nd order numerical method for laminar flow at moderate to high Reynolds numbers: entrance flow in a duct, Int. J. Numer. Meths. Eng. 12 (1978) 405–414.
[15] C.K. Forester, Higher order monotonic convective differencing schemes, J. Comp. Phys. 23 (1977) 1–22.
[16] C.E. Leith, Numerical simulation of the earth's atmosphere, Meths. Comp. Phys. 4 (1965) 1–28.
[17] M.C. MacCracken and R.D. Bornstein, On the treatment of advection in flux formulations for variable grid models, with application to two models of the atmosphere, J. Comp. Phys. 23 (1977) 135–149.

[18] B.P. Leonard, G.J. Vachtsevanos and K.A. Abood, Unsteady, two-dimensional salinity intrusion model for an estuary, in: C. Brebbia (ed.), Applied numerical modelling (Pen Tech. Press, London, 1978) 113–123.
[19] K.F. Bowden and P. Hamilton, Some experiments with a numerical model of circulation and mixing in a tidal estuary, Estuarine and Coastal Marine Science 3 (1975) 281–301.
[20] R.R. Boericke and J.M. Hogan, An x–z hydraulic/thermal model for estuaries, A.S.C.E. J. Hyd. Div. 103 (1977) 19–37.
[21] E.A. Caponi, A three-dimensional model for the numerical simulation of estuaries, in: H.E. Landsberg (ed.), Advances in Geophysics 19 (Academic Press, New York, 1976).
[22] A.F. Blumberg, Numerical model of estuarine circulation, A.S.C.E. J. Hyd. Div. 103 (1977) 295–310.
[23] J.P. Boris and D.L. Book, Flux-corrected transport. I. SHASTA, a fluid transport algorithm that works, J. Comp. Phys. 11 (1973).
[24] R.L. Hanley and J.M. Wright, Jr., The relationship between the grid size and the coefficient of nonlinear lateral eddy viscosity in numerical ocean circulation models, J. Comp. Phys. 19 (1975) 257–266.
[25] F.H. Harlow and A.A. Amsden, Numerical calculation of multiphase fluid flow, J. Comp. Phys. 17 (1975) 19–52.
[26] J.C. Heinrich, P.S. Huyakorn, O.C. Zienkiewicz and A.R. Mitchell, An "upwind" finite element scheme for two-dimensional convective transport equation, Int. J. Numer. Meth. Eng. 11 (1977) 131–143.
[27] D. Skiadaressis, The stabler code: a draft manual, CHAM TR/35 (1977).
[28] J.E. Fromm, A method for reducing dispersion in convective difference schemes, J. Comp. Phys. 3 (1968) 176–189.
[29] J.E. Fromm, Practical investigation of convective difference approximations of reduced dispersion, Phys. Fluids Suppl. II (1969) 3–12.