AMD - Vol. 34

Finite Element Methods for Convection Dominated Flows

presented at

THE WINTER ANNUAL MEETING OF
THE AMERICAN SOCIETY OF MECHANICAL ENGINEERS
NEW YORK, NEW YORK
DECEMBER 2-7, 1979

sponsored by

THE APPLIED MECHANICS DIVISION, ASME

edited by

T. J. R. HUGHES
CALIFORNIA INSTITUTE OF TECHNOLOGY
PASADENA, CALIFORNIA

THE AMERICAN SOCIETY OF MECHANICAL ENGINEERS
United Engineering Center 345 East 47th Street New York, N.Y. 10017

A SURVEY OF FINITE DIFFERENCES OF OPINION ON NUMERICAL MUDDLING OF THE INCOMPREHENSIBLE DEFECTIVE CONFUSION EQUATION

B. P. Leonard

City University of New York
College of Staten Island, New York

ABSTRACT

Finite difference methods have been very successful for partial differential equations dominated by the Laplacian operator, such as those for diffusion and wave motion. However, when the first-derivative convection operator becomes important, standard central difference methods lead to incomprehensible wiggles or confusing nonconvergence — and are therefore clearly defective. There have been various opinions on suitable remedies, one of the most popular being to use highly stable one-sided upstream differencing for the convection term. But the artificial diffusion of such methods leads to low accuracy. Nevertheless, proponents have "justified" upstream differencing (or upstream-central hybrids) by a variety of arguments — unfortunately, all fallacious! Much confusion has stemmed from a muddling of the meaning of truncation error. When it is realized that "standard" central differencing for the Laplacian operator is a *third* order method, and that central difference methods (of any order) lack inherent stability for modelling *odd* order derivatives, the consistent third order convective differencing scheme is seen to be optimal in terms of accuracy, stability, and simplicity.

INTRODUCTION

Computational Fluid Dynamics (CFD) is still a relatively young science. And, as with any new endeavour there have been blind-alleys, band-wagons, break-throughs, and battle-grounds. True success often requires as much Art and Intuition (A & I) as Science and Technique (S & T) — not to mention R & D. Modern computer (as opposed to computational) technology, in terms of speed and volume and output hardware, is particularly impressive -- and can easily be abused! Sophisticated graphical or tabular presentations can often disguise a wealth of mindless number crunching — very often achieving nothing better than old-fashioned Back-Of-Envelope (BOE) order-of-magnitude estimates. Undoubtedly there are finite differences of opinion on the relative merits of the new methods as compared with time-tested design techniques which were often set by BOE-ing (particularly in the aircraft industry).

Again perhaps because of the youth of CFD, a good deal of folklore and mysticism has developed to explain away otherwise incomprehensible phenomena —

especially those associated with the imcomprehensible defective confusion equation. Unfortunately the lore is sometimes perceived as law and the mysticism can develop into religious fervor, creating camps, cliques, sects, schools, and even empires. It is fascinating to watch the development of the subject as otherwise entirely rational scientists, perhaps confused by some particularly incomprehensible defect in their early attempts at CFD modelling, latch onto what appears to be a particularly attractive band-wagon ("because it works") and not only don't let go but start blowing their own trumpet as loudly as possible to attract others to their "discovery" (and perhaps to drown out any opposition).

At the risk of dampening the festive spirit, this paper is addressed to workers using the finite *element* technique, warning them to beware of a well-known band wagon (featuring particularly Loud and Overpowering Trumpeters of Unusual Stamina) which has already led many previously wiggle-worried finite-difference workers into an artificially diffuse state of dreamy contentment.

The LOTUS syndrome is strikingly consistent in the way it ensnares the unsuspecting finite difference modeller. In his early studies, the typical victim has received formal indoctrination in the beauty, power, and accuracy of what are usually called "second order central difference" methods. Everything seems terribly easy. He quickly forgets how to manipulate Bessel functions and spherical harmonics (with good riddance, he says, because you couldn't treat "real-life" boundary shapes with them anyhow) as he solves practical problems in unsteady heat conduction, ground-water diffusion, neutron transport, shallow-water wave motion, and a host of other *Laplacian dominated* problems. His success with ground water and tidal flow encourages him to select a CFD topic for his Ph.D. project — perhaps a "relevant" subject like pollutant dispersal by advection. He tools up his central difference code and presses "RUN."

Several months and hundreds of overflow messages later, our would-be CFD modeller is convinced there are no programming bugs in his code. But it still won't give him numbers. He tries some test problems, such as the one-dimensional steady convection and diffusion of a scalar between two fixed boundary conditions. Hooray! — numbers! But when he plots the results, he just gets wiggles, wiggles, wiggles! By changing some parameters, he finds that he gets wiggle-free solutions when he increases the amount of physical diffusion — although he notices that the computed results look something like an exact (exponential) solution with a smaller diffusion coefficient. It's as if central differencing (of the convection term) introduces some effective *negative* diffusion. This might "explain" the instabilities at smaller values of the physical diffusion (he thinks).

In his continuing literature search, our perplexed graduate student notes that other researchers have experienced similar problems with wiggles or nonconvergence, and often get around this by using one-sided upstream-weighted first-order differencing — with or without further justification (other than "it works"). He tries it on the test problem, and it certainly always gives wiggle-free results — but a bit over-diffusive looking. On digging further, he comes across the concept of false or artificial numerical diffusion (or viscosity), which a quick BOE estimate suggests might completely swamp any physical diffusion in the kind of problems he's interested in. Attractive as it may be, he worries about the legitimacy of pure upstream differencing.

On chatting over this problem with his thesis adviser one day — i.e., upstream differencing's positive false diffusion is opposed to central differencing's apparent negative diffusion effect — they hit upon a brilliant idea: why not weight the convection term, partially central and partially upstream, to balance the positive and negative diffusion effects? When tried on the test problem, the technique seems to need an adjustable weighting factor, depending on the physical diffusion-to-convection ratio and the spatial step size. By matching to the exact exponential solution, they can get an optimal albeit rather expensive, weighting which gives exact results (for this problem). An inexpensive suboptimal algorithm is easily devised however, which gives results which are almost as good.

Of course the weighting has been devised for a rather special one-dimensional test case. It is realized that for a two-dimensional problem, the exponential solution may not be quite appropriate. However, the student and his adviser convince themselves that it is better to use a difference formula which reduces to the exact form under some conditions rather than one which hardly ever does (which is the case for pure upstream or pure central differencing, they say). They feel quite content that this is sufficient justification to legitimize the algorithm. They crunch numbers, publish, and tell all their friends about their great discovery. The LOTUS syndrome has set in!

The natural question at this point is: what's so bad about the weighted upstream-central idea? [The justification sounds reasonable. It has in fact been independently (re)discovered by several highly qualified researchers. And it seems to give plausible looking results.] One of the main aims of this paper is to answer this question. Essentially, under (even only moderately) convection-dominated conditions, the algorithm reverts to virtually pure upstream differencing. Even this has been defended as being entirely appropriate; and it is -- *for the particular test problem on which the algorithm was based.* But to the extent that conditions differ from those of the test problem, the technique suffers from the same low accuracy (or an equivalent false diffusivity) as pure first-order upstream differencing. Some other test problems will be suggested to show up the defects. But first, some of the mysticism surrounding the reason for central differencing's wiggliness must be dispelled. It turns out that some folklore concerning truncation error also needs to be repelled. And this leads naturally and logically to a development of finite difference formulas for both diffusion and convection with a consistent (high) accuracy and inherent stability against wiggles.

NEW PERSPECTIVES ON SPATIAL TRUNCATION ERROR

So-called "second" order central differencing has certainly been tremendously successful for Laplacian-dominated problems such as diffusion and wave motion. A skeptic might ask: why? The correct and concise answer is: because it's a *third* order method. The popular use of the term "second" order is a little bit of folkloric jargon which has crept into computational physics due to confusion of the discretization error in the Taylor series expansion of a derivative with the global truncation error in the solution. It is the latter which is of interest, of course.

The classical Taylor series analysis of the accuracy of the central second difference as an approximation to the second derivative at the central point is

$$\frac{\phi_{i+1} - 2\phi_i + \phi_{i-1}}{\Delta x^2} = \left(\frac{\partial^2\phi}{\partial x^2}\right)_i + \frac{1}{12}\,\phi_i^{(iv)}\,\Delta x^2 + \text{HOT} \qquad (1)$$

And as $\Delta x \to 0$, the left hand side differs from the second derivative by terms of order Δx^2. The "second" order accuracy refers to the accuracy of the operator -- not to the accuracy of an algorithm using this operator to solve a differential equation numerically. The "order" of a numerical algorithm is classically defined to be the highest degree (P, say) of a polynomial $\phi_P(x)$ for which the algorithm is exact; local truncation error in an actual solution is then of order P + 1 and global truncation error is again P. Since the discretization error in (1) involves fourth and higher order derivatives, the left hand side is an exact representation of the second derivative for a *third* order polynomial. Hence an algorithm using this difference operator is a *third order algorithm* (as far as this operator is concerned), giving a solution with fourth order local truncation error and third order global truncation error. Of course, errors of lower order may occur due to other modelled terms in the governing equation or to the use of lower order boundary conditions -- the latter being a common mistake for the case under consideration. Naturally, this idea generalizes to two and three dimensions. The well-known five-point Laplacian operator in two dimensions generates a solution with third order accuracy (in terms of global truncation error); it is exact for a general cubic function of x and y.

Compare now a central difference expression for the first derivative. In fact, it is advantageous to form the convection term $-u(\partial\phi/\partial x)$ for comparison with a normalized diffusion term given by the (positive) second derivative (with a diffusion coefficient of unity). Thus,

$$-u\left[\frac{\phi_{i+1}-\phi_{i-1}}{2\Delta x}\right] = -u\left[\left(\frac{\partial\phi}{\partial x}\right)_i + \frac{1}{6}\phi_i''' \Delta x^2 + \mathrm{HOT}\right] \qquad (2)$$

This convective difference operator is usually claimed to be *consistent* with (1) — because of the same power of Δx appearing in the discretization error. But this operator really *is* a second order accurate operator (when used in an algorithm for ϕ) because of the ϕ''' leading-order discretization error term (i.e., it is exact only for a quadratic — not a cubic, as is the case of the second central difference). Thus the use of (1) and (2) in an algorithm to solve the convection-diffusion equation is *not* a consistent method in terms of the order of the overall algorithm. It is third order for diffusion-dominated flows and second order for the high convection regime. Of course there are other problems as well, having to do with numerical stability, as explained in the next section.

For reference, consider the case of the one-sided first difference

$$-u\left[\frac{\phi_i-\phi_{i-1}}{\Delta x}\right] = -u\left[\left(\frac{\partial\phi}{\partial x}\right)_i - \frac{1}{2}\phi_i'' \Delta x + \mathrm{HOT}\right] \qquad (3)$$

or

$$-u\left[\frac{\phi_{i+1}-\phi_i}{\Delta x}\right] = -u\left[\left(\frac{\partial\phi}{\partial x}\right)_i + \frac{1}{2}\phi_i'' \Delta x + \mathrm{HOT}\right] \qquad (4)$$

As is the case for any finite difference approximation to a *first* derivative, the order of the discretization error (in this case, first) is the same as that of an algorithm using the operator. In this case each of (3) and (4) is exact when ϕ is a linear function of x. These are truly first order operators. Note that the leading-order discretization term is equivalent to a physical diffusion term with an effective diffusion coefficient of

$$\Gamma_{num} = \pm\frac{u\Delta x}{2} \qquad (5)$$

This, of course, is one explanation of the stability of upstream (first-order) differencing. By choosing (3) when $u > 0$ and (4) when $u < 0$, the discretization error terms are always stabilizing. Unfortunately, they also interfere with, corrupt, and may dominate the physical diffusion terms. The appropriate parameter is the generalized cell Péclet number (or Reynolds number, etc.) given by the ratio of discretized convection terms to discretized diffusion terms:

$$P_\Delta = \frac{u\Delta x}{\Gamma} \qquad (6)$$

where Γ is the effective (laminar or turbulent) diffusion coefficient. When $P_\Delta = 2$, the additional effective artificial numerical diffusion of first-order upstream differencing is the same size as the physical diffusion.

Roughly speaking, if $P_\Delta \lesssim 1$ the flow is diffusion dominated for computational purposes. In that case, the form of the convective differencing is not important — first, second, and higher order forms give about the same results. But for large values of u/Γ, it is clearly impracticable to require a small enough Δx to keep P_Δ less than 1 throughout the entire computational domain. This is where trouble begins. From a practical view-point, second-order central differencing is potentially wiggly for $P_\Delta > 2$; but this is the regime in which first-order upstream differencing becomes seriously artificially diffusive.

Computationally, a flow can be considered convection dominated if $P_\Delta \gtrsim 5$. In practical calculations of flows of engineering interest, P_Δ may need to be $O(100)$; and in geophysical flows, the necessarily large spatial grid may mandate much larger P_Δ values. Clearly, it's important to have a convective differencing scheme which is both stable and accurate under these high convection conditions. Interestingly enough, a simple third order scheme has both these properties, and of course remains uniformly third order (using (1) for the diffusion terms) throughout the entire P_Δ range.

Consider the following:

$$- u\left[\frac{\phi_{i+1} - \phi_{i-1}}{2\Delta x} - \frac{\phi_{i+1} - 3\phi_i + 3\phi_{i-1} - \phi_{i-2}}{6\Delta x}\right]$$

$$= - u\left[\left(\frac{\partial\phi}{\partial x}\right)_i + \frac{1}{12}\phi_i^{(iv)}\Delta x^3 + \text{HOT}\right] \tag{7}$$

$$- u\left[\frac{\phi_{i+1} - \phi_{i-1}}{2\Delta x} - \frac{\phi_{i+2} - 3\phi_{i+1} + 3\phi_i - \phi_{i-1}}{6\Delta x}\right]$$

$$= - u\left[\left(\frac{\partial\phi}{\partial x}\right)_i - \frac{1}{12}\phi_i^{(iv)}\Delta x^3 + \text{HOT}\right] \tag{8}$$

As seen in the next section, (7) is appropriate if $u > 0$ and (8) when $u < 0$. This is the basis of third order upstream-weighted convective differencing. An algorithm using this convective differencing scheme together with (1) for diffusion represents the consistently third order method — it is exact for a cubic, for any combination of convection and diffusion.

For reference, the fourth order centrally distributed convection scheme is

$$- u\left[\frac{\phi_{i+1} - \phi_{i-1}}{2\Delta x} - \frac{\phi_{i+2} - 2\phi_{i+1} + 2\phi_{i-1} - \phi_{i-2}}{12\Delta x}\right]$$

$$= - u\left[\left(\frac{\partial\phi}{\partial x}\right)_i - \frac{1}{4}\phi_i^{(v)}\Delta x^4 + \text{HOT}\right] \tag{9}$$

which turns out to have stability problems similar to those of (2).

It should be clear that any particular finite difference approximation to an N-th derivative will always have a Taylor series expansion of the form

$$\frac{[\text{FDT}]}{\Delta x^N} = \left(\frac{\partial^N\phi}{\partial x^N}\right)_i + \text{const}\left(\frac{\partial^{M+N}\phi}{\partial x^{M+N}}\right)_i \Delta x^M + \text{HOT} \tag{10}$$

The global truncation error corresponding to this operator is then

$$\mathcal{E} - \mathcal{E}_{\text{Ref}} = \text{const}\left[\left(\frac{\partial^P\phi}{\partial x^P}\right) - \left(\frac{\partial^P\phi}{\partial x^P}\right)_{\text{Ref}}\right]\Delta x^P + \text{HOT} \tag{11}$$

where

$$P = M + N - 1 \tag{12}$$

And of course $\mathcal{E} - \mathcal{E}_{\text{Ref}}$ is identically zero for a P-th degree polynomial. In any specific case, the constant in (11) can be evaluated, and the expression forms a good estimate for the global truncation error in the solution, using any convenient P-th difference operators for the right hand side, i.e.,

$$\mathcal{E} - \mathcal{E}_{Ref} = \text{const}\left[\Delta^P\phi - (\Delta^P\phi)_{Ref}\right] + \text{HOT} \tag{13}$$

In the case of first order upstream differencing, for example, the constant is 0.5 and P = 1. This represents very severe truncation error if there is any deviation in the solution away from a linear function of the spatial coordinates — unless the grid is so fine that the (first) differences in (13) are everywhere quite small. With the consistent third order method, (13) indicates that extremely accurate solutions can be obtained with very practical grids.

NUMERICAL STABILITY

The confusion about truncation error has led many people to expect that since central differencing for the Laplacian operator is so accurate, then central differencing of convection as given by (2) should be similarly accurate — based on the (erroneous) concept of "second-order consistency" between the two operators. Of course, as explained in the previous section, central differencing for the first derivative is formally less accurate (in terms of the solution) than that for the second derivative. But this is not the main reason for the poor performance of central differencing for the first derivative. The problem is that *central* differencing (of any order) has no inherent "numerical stability" when applied to derivatives of *odd* order.

To develop the concept of numerical stability, consider the model incompressible convective diffusion equation

$$\frac{\partial\phi}{\partial t} = -u\frac{\partial\phi}{\partial x} + \Gamma\frac{\partial^2\phi}{\partial x^2} + S \tag{14}$$

where S represents a catch-all term to account for anything else not explicitly shown in the other terms. Focus attention on a particular grid point, i. If the left hand side (LHS) of (14) represents $\partial\phi_i/\partial t$, then the modelled RHS represents net influx and source terms creating the change in ϕ_i. If there were numerical errors in the RHS, then from (14) ϕ_i would increase or decrease erroneously. In a stable algorithm this change in ϕ_i due to errors in the RHS should feed back negatively into the RHS as a self correction device. Clearly, for this kind of numerical stability it is necessary that

$$\frac{\partial(\text{RHS})}{\partial\phi_i} < 0 \tag{15}$$

i.e., that the sensitivity to ϕ_i of the combination of modelled terms on the right hand side of (14) be negative. Now, look back at the LHS of equation (1). The sensitivity of the centrally modelled second derivative is $-2/\Delta x^2$; and since Γ is (supposed to be) always positive, the numerical stability of this operator is good. [By the way, if Γ itself is computed from a wiggly algorithm, it must never be allowed to take on unphysical negative values — for obvious computational reasons!] But now look at the LHS of equation (2). Where's ϕ_i? Nowhere! Sensitivity is zero. Stability is neutral. In fact, ϕ_i can drift all over the place and central differencing of convection can't do anything to correct it. This is the basic cause of wiggles. In a control-volume formulation, the sensitivity actually becomes positive (i.e., strictly unstable) in regions of decelerating flow.

$$\text{INFLUX} = u_\ell\,\frac{(\phi_{i-1} + \phi_i)}{2} - u_r\,\frac{(\phi_i + \phi_{i+1})}{2} \tag{16}$$

In a general flow, such regions act as wiggle sources and can easily lead to total numerical catastrophe.

Note that higher order *central* difference methods are no better in this regard — where is ϕ_i in equation (9)? Again, this fourth order operator has neutral numerical stability. The same is true for the fourth order model of the third derivative. In fact, as stated before, finite difference models of *odd* order derivatives using *central* differencing of any order lead to formally neutral numerical stability; and in practical flows using control-volume formulations, there will always be destabilizing regions.

There have of course been many finite differences of opinion on the cause of wiggles. The term "nonlinear instability" completely misses the boat. Nonconstant coefficients have been blamed; but these can either stabilize or destabilize an essentially neutral operator. [Of course the destabilizing is always more noticeable.] Roache [1] came close to identifying the basic cause of wiggles. Unfortunately, in currently available editions of his popular and influential book, he tends to confuse the practical criterion for the absence of wiggles (which in the present terminology is $P_\Delta \leq 2$) with a supposed von Neumann stability condition for the explicit forward-time-central-space (FTCS) convective diffusion equation. This condition (in Roache's terminology $R_c \leq 2$, emphasized several times throughout the book) is based on an erroneous analysis of the FTCS equation by Fromm [2]. The correct von Neumann condition is the right-hand inequality in the following

$$2c \leq P_\Delta \leq \frac{2}{c} \tag{17}$$

where $c = u\Delta t/\Delta x$ is the Courant number. [The left-hand inequality is the diffusive restriction, $\Gamma\Delta t/\Delta x^2 \leq 0.5$.] Clearly, (17) is much less restrictive than the erroneous $2c \leq P_\Delta \leq 2$ condition. For one thing, (17) allows arbitrarily large P_Δ values -- provided the time step is reduced appropriately. From a practical point of view however, the widely believed incorrect "von Neumann condition" makes better folklore, and is appropriately defective for the IDC equation, and certainly adds mystique to an otherwise rather mundane topic.

A Slight Digression

[While on the subject of folklore, it is worth mentioning two other topics which have faithful believers in opposing camps. One involves the idea of "conservative" or "nonconservative" formulation. The other concerns whether or not the algorithm should be explicit or implicit. Very often one hears that "the problem was solved by a fully implicit conservative finite difference method," stated with a conviction that could only mean that the appropriate gods would look favourably on such a "safe" scheme. Actually, if there are no physical source terms in the equation to be modelled, then a strictly conservative scheme (no "effective" source terms in the modelled equation) is appropriate. With physical source terms (which will have their own modelling errors), it doesn't matter — provided the overall errors are within tolerable bounds of course. The explicit/implicit question is a little different. There are good and bad (stable and unstable, accurate and inaccurate) possibilities within both categories. Certainly, there is no guarantee that an implicit method is going to be safer or more economical than an explicit formulation which solves the same problem. In some cases the matrix equations of an implicit formulation are solved by an explicit iterative method — in which case the moot is mute.

Concerning economics (i.e., computer usage), there's a particularly unfortunate bit of folklore involving the misconception that "higher order methods are exhorbitantly more expensive than lower order." This, of course, is utter hogwash. Although, almost by definition, a higher order method requires more operations per space-time grid point, the increased accuracy allows a grid coarsening which greatly decreases overall usage (time and storage) while still increasing global accuracy. Roughly speaking, individual grid-point computations increase arithmetically (actually rather slowly) while the corresponding reduction in the total number of space-time grid points reduces geometrically. There are limits, of course — primarily human. Algorithmic complexity is a serious limitation. Any method which simultaneously possesses accuracy, stability, algorithmic simplicity, and an easily comprehended physical interpretation would seem to be optimal. Interestingly enough, the present author's third order methods [3] just happen to possess all these properties.]

Wiggle Remedies

Central differencing's penchant for wiggles has led to various finite differences of opinion on alternatives. Roache analyzed wiggles in some detail and pointed out that they *could* be avoided by using one-sided upstream-weighted differencing for the convection terms, but suggested that this "remedy" is actually fictitious because the stability is achieved by the effective introduction of false numerical diffusion — i.e., one might just as well use central differencing while arbitrarily adding artificial diffusion, as given by (5), to the physical diffusion. In convection dominated flows, this constitutes extremely heavy handed stabilization — somewhat akin to stabilizing a delicate instrument servomechanism by packing the whole thing in heavy axle grease. In this interpretation, most reasonable people find first order differencing hard to justify. In terms of numerical stability, it's easy to see why first order differencing is so stable. Both (3) and (4) have strong sensitivity to ϕ_i, and clearly to satisfy (15), one needs to choose (3) when $u > 0$ and (4) when $u < 0$.

If wiggles result from a downstream boundary condition, Roache has suggested the possibility of relaxing the boundary condition (in the spirit of singular perturbation techniques). This will work in certain model problems. But the essential difficulty comes from the requirement of a sudden change in value of the computed function. And this can occur within the computational domain when the physical diffusion is small. This is the favorite realm of the unphysical wiggles of central differencing of convection-dominated flows.

Many others have had finite differences of opinion on what to do about wiggles and nonconvergence. In nearly all cases however, the remedy has been to revert to some form of (essentially first order) upstream differencing under high convection conditions. The variations are in the "justification," which ranges from the pseudo-physical argument of the Los Alamos group's "donor cell" concept [4], and the "locally one-dimensional" fallacy which has been the basis of the Imperial College Heat Transfer Section's work [5] (and is rapidly gaining acceptance in finite element techniques [6-10]) to simple expedients which might be paraphrased as "it was *necessary* in order to insure diagonal dominance" [11] or, even more frankly, "it was *necessary* for convergent 'results'" [12] (i.e., numbers as opposed to overflow messages). The confusion of "necessary" with "sufficient" is bad enough, but since when have matrix equations needed to be diagonally dominant for a solution? They may be easier to solve (e.g., iteratively); but there is nothing particularly nasty about

$$\begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \tag{18}$$

And yet some people would undoubtedly want to dump this equation into a can of (artificial) axle grease.

Accuracy criteria have been correspondingly fuzzy, involving marginal grid refinement, or fiddling with specified inlet conditions or other adjustable parameters to give global correspondence with measurements. The most successful applications of first order schemes often involve flows which are governed primarily by the kinematic constraint of incompressibility (through an accurate pressure solver) together with empirical modelling of boundary shear stresses (such as log-law boundary layers, for example). Because of upstream differencing's inherent false diffusion, it is clearly illogical and scientifically unsound to use it for testing delicate and highly sophisticated turbulence models. But this doesn't seem to have deterred many people from trying. Could it be that poor numerics has hindered progress in turbulence modelling? There must be finite differences of opinion on this.

Other options have been explored by a number of fairly isolated workers who form a sort of Anti-Upstream-Differencing Intelligentsia Encouraging Numerical Computational Excellence [13-17]. But so far this loosely knit group has not been able to attract a wide audience. Some techniques remain based on second order central differencing with carefully controlled wiggle filters. Others involve second order upstream differencing [18], which for $u > 0$ can be written

$$- u \left[\frac{\phi_i - \phi_{i-1}}{\Delta x} + \frac{\phi_i - 2\phi_{i-1} + \phi_{i-2}}{2\Delta x} \right]$$

$$= - u \left[\left(\frac{\partial \phi}{\partial x} \right)_i - \phi_i''' \Delta x^2 + \mathrm{HOT} \right] \tag{19}$$

and which is clearly highly stable. Somewhat incomprehensibly, there has been a good deal of interest in *fourth* order *central* difference methods [19], which however are almost as convectively defective as second order, as explained earlier. Perhaps because of the confusion regarding truncation error, consistently *third* order methods have not been explored very much (until recently [3]). From (7) and (8) it can be seen that an upstream weighted third order convection scheme is simultaneously stable and accurate. This is by far the most logical and scientifically sound way of extending the success of standard central differencing for diffusion-dominated problems into the realm of highly convective flows. It is hoped that this paper will help to expand the audience for this philosophy.

UPWINDING LEGITIMIZED?

The concept of artificial diffusion introduced by first order upstream differencing is based on the leading term in the Taylor series expansion of (3) or (4). Now it may be thought that higher order terms might not always be negligible and could possibly compensate for the dominant term under some conditions. Figure 1 shows the steady one-dimensional convection and diffusion of a scalar between two fixed boundary conditions. The exact solution is an exponential with a length constant depending on the Péclet number. The pure upstream difference solution of this problem (with the physical diffusion term neglected) is ϕ = const (equal to the upstream boundary condition). Clearly, this is an entirely appropriate solution for $P_\Delta \gtrsim 5$ -- i.e., the high convection regime. It must be that conclusions derived from the Taylor series expansion of an exponential can be misleading when the length constant is small. In fact, from Figure 1 it looks as though upstream differencing is more accurate for higher and higher P_Δ's. This appears to be a legitimate justification of upstream differencing under high convection conditions. For brevity this will be referred to as the SPALD concept, since it was introduced in 1972 by a Self-Proclaimed Amateur in Legitimate Differencing [5].

It was natural for professional computational fluid dynamicists to want to adopt any philosophy with a claim to legitimacy which would give bounded results without wiggles, and for this the SPALD philosophy was admirably suited. In fact, SPALD-ing quickly became almost universally accepted over a far-flung empire, and remains very influential to this day.

At lower values of P_Δ, SPALD-ing requires a change-over from upstream to central (including diffusion). The simple hybrid scheme of the well-known TEACH code [20] simply switches as $P_\Delta \gtrless 2$. More sophisticated weighting techniques can generate very accurate solutions of the test problem of Figure 1. [But as shown later, in more realistic problems, accuracy is severely impaired -- even with "optimal" weighting.] Recently, SPALD-ing has had a strong influence on the development of finite *element* techniques, spear-headed by a highly respected group at a well known institution. When translated into finite element terms, the basic philosophy has the effect of Suppressing Wiggles and Adding Numerical Stability at the Expense of Accuracy.

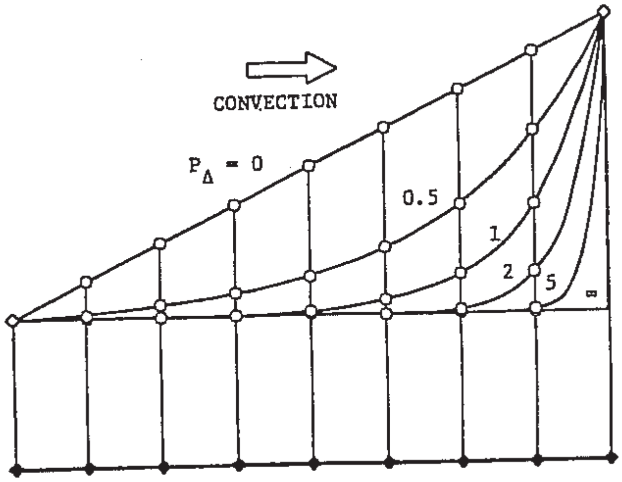


Figure 1. Steady convection and diffusion.

From Hogs to Herrings

Under high convection conditions ($P_\Delta \gtrsim 5$) all first/second-order upstream/central-weighted schemes essentially revert to pure upstream differencing (with physical diffusion neglected). Defenders argue that under such conditions, the dependent variable is swept downstream virtually unchanged, (except very close to the next grid-point), and should have almost no upstream effect, invoking the oft-quoted "pigpen" analogy that farm-yard olfactory information is more strongly carried downwind than up. For example, given the series of ϕ values shown in Figure 2a, pigpen enthusiasts would interpolate a curve such as that shown in Figure 2b. But, to this

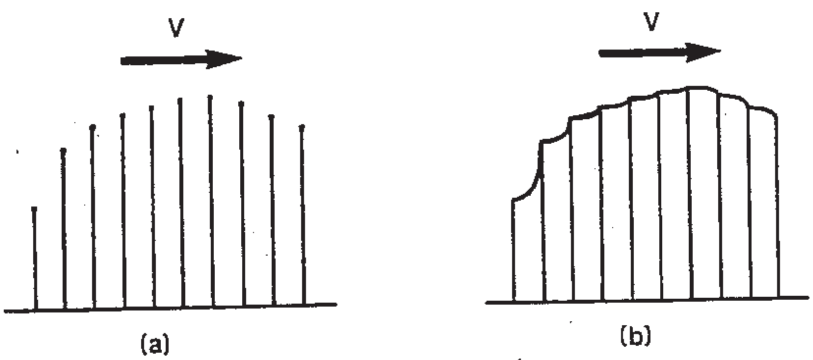
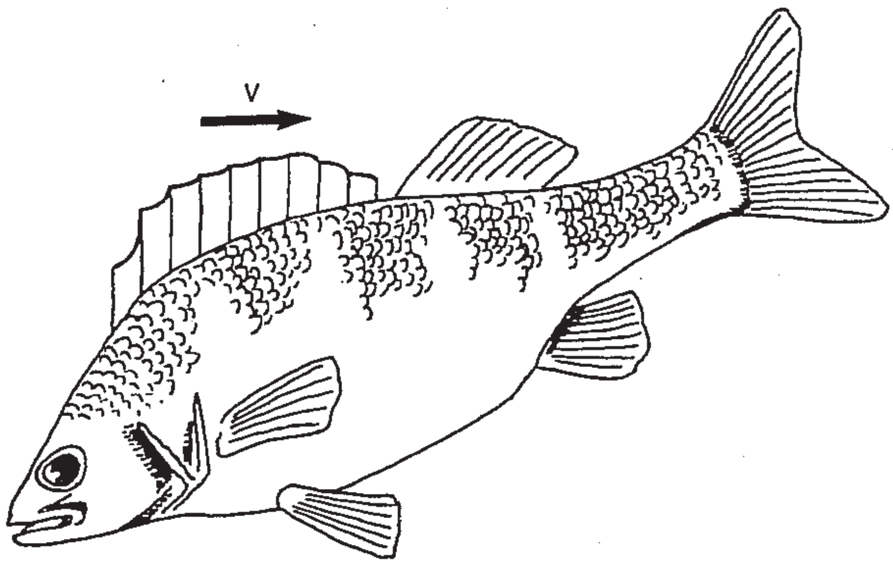


Figure 2. Pigpen interpolation.

author's knowledge, very few physical systems would show this kind of behaviour -- a possible exception being that shown in Figure 3. It seems fairly clear that pigpen interpolation, and in fact everything to do with the "justification" of upwinding, is really a red herring.



Fugure 3. Example of upwind philosophy.

THE PROBLEM WITH FIRST-ORDER METHODS

If one were to ask any professional draftsman to put an interpolation curve through the points in Figure 2a, the result would undoubtedly be a good bit more continuous than that in Figure 2b. If the indicated function were supposed to be the solution of a transport equation of the form of (14) — with S representing transverse transport as well as actual source terms — it would be of interest to see if upstream differencing of the convection term would be consistent with the solution. Specifically, rearrange (14) to give

$$u\frac{\partial\phi}{\partial x} = S^* \qquad (20)$$

where S^* simply represents all the other terms. For simplicity, assume u = const (the idea is easily generalized). Then by integrating (20) across a control volume from left to right

$$S^* = \frac{1}{\Delta x}\int_{-\Delta x/2}^{\Delta x/2} u\frac{\partial\phi}{\partial x}\,dx = u\,\frac{(\phi_r - \phi_\ell)}{\Delta x} \qquad (21)$$

and this can actually be estimated by using some reasonable interpolation for ϕ_r and ϕ_ℓ. But the upstream difference algorithm requires

$$u\,\frac{(\phi_i - \phi_{i-1})}{\Delta x} = S^* \qquad (22)$$

Hence

$$\phi_i = \phi_{i-1} + \frac{\Delta x}{u}S^* \qquad (23)$$

Clearly if ϕ is recomputed from (23), using S* estimated from (21), any discrepancy between the new and old ϕ values is a direct estimate of the truncation error of the upstream differencing. The result is

$$\phi_i^{NEW} = \phi_{i-1}^{NEW} + (\phi_r^{OLD} - \phi_\ell^{OLD}) \tag{24}$$

which is portrayed graphically in Figure 4.

It should be obvious that if $\partial\phi/\partial x$ varies throughout the computational domain, then truncation error may be severe. In fact

$$\mathcal{E} - \mathcal{E}_{Ref} = \frac{1}{2}\left[\left(\frac{\partial\phi}{\partial x}\right)\Delta x - \left(\frac{\partial\phi}{\partial x}\right)_{Ref}\Delta x_{Ref}\right]$$

$$= \frac{1}{2}\left[\delta\phi - \delta\phi_{Ref}\right] \tag{25}$$

where $\delta\phi$ is the algebraic increase in ϕ across a control volume [16].

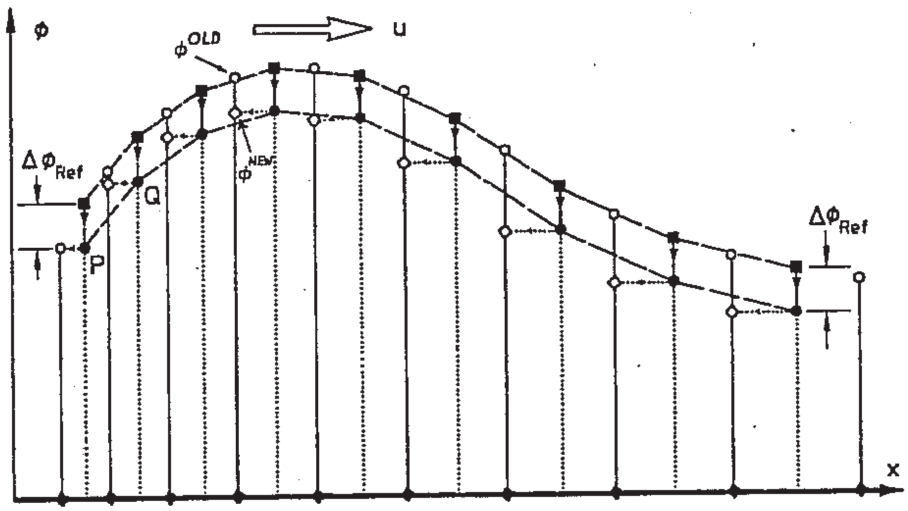


Figure 4. Construction of ϕ^{NEW} from ϕ^{OLD}.

The important fact to remember is that in most problems of practical interest, S^* in (20) will be nonzero and variable — due to sources, transients, and transport terms other than the explicit x-convection. Note that in the red herring test problem of Figure 1, $S^* \to 0$ for $P_\Delta \gtrsim 5$. Upstream differencing can be accurate — if $S^* \approx$ const. But this is obviously too restrictive to be generally useful.

Figure 5 shows a more meaningful test problem representing pure convection ($P_\Delta = \infty$) at constant velocity across a stationary specified source term. Of course, under these conditions, all weighting schemes (even optimal) are identically equivalent to pure upstream differencing. Note that the upstream difference solution (for $P_\Delta = \infty$) is close to an exact solution for $P_\Delta = 2$ (actually, it's identical to the central difference result for $P_\Delta = 2$), corresponding to the *classical* Taylor series estimate given by (5) — a rather refreshing little breeze within an otherwise stifling atmosphere of pigpens and red herrings. Note that the third order algorithm based on Quadratic Upstream Interpolation for Convective Kinematics [3] (QUICK) gives results which are graphically almost indistinguishable from the exact solution (and it's also quite stable, even under these pure convection conditions).

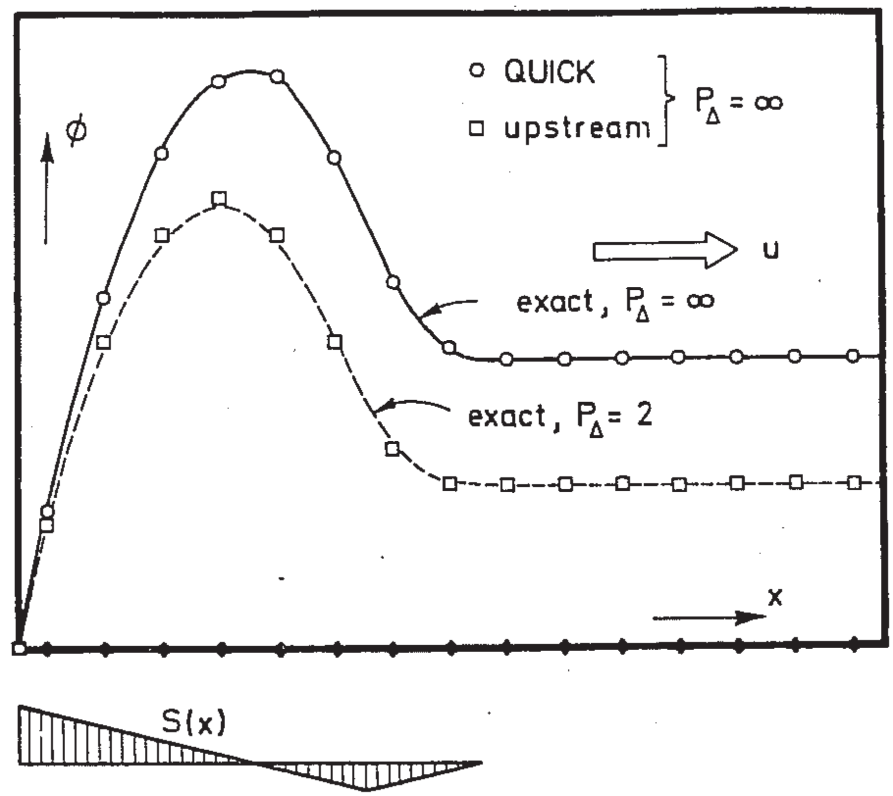


Figure 5. Pure convection with source term.

Another standard test problem for which first/second-order upstream-weighted schemes fail to give satisfactory results is shown in Figures 6, 7, and 8. This is two-dimensional steady convection at an angle to the grid, with a transverse step along an upstream boundary. In this problem, S*

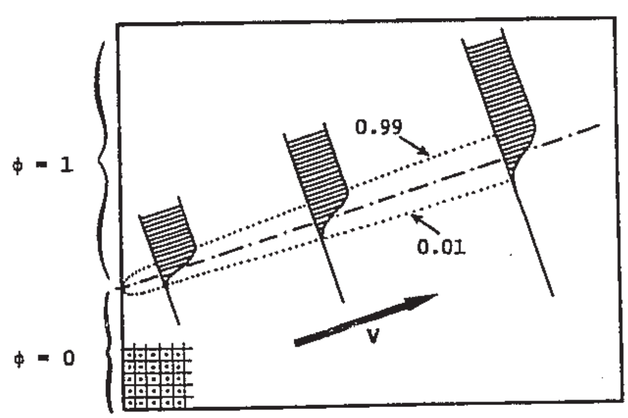


Figure 6. Oblique convection: exact solution, $P_\Delta = 100$.

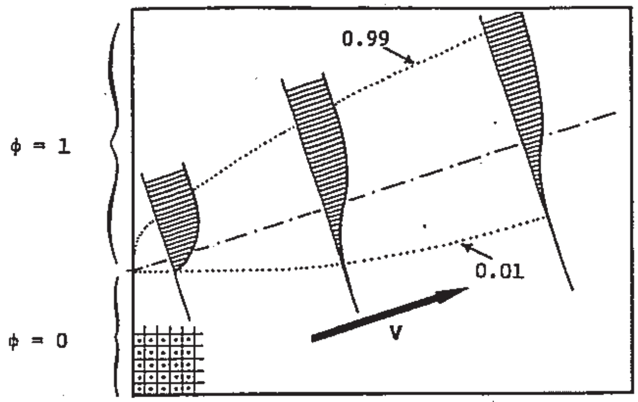


Figure 7. Oblique convection: optimal upwinding, $P_\Delta = 100$.

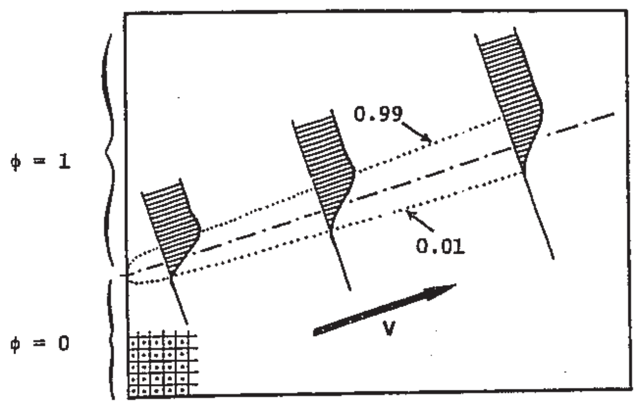


Figure 8. Oblique convection: QUICK method, $P_\Delta = 100$.

represents transverse convection, and the truncation error manifests itself as gross false transverse diffusion [15]. By contrast the third-order QUICK method is much less artificially diffusive and models the true (error function) profile fairly well.

In case anyone is still not convinced, Figures 9, 10, and 11 show convective-diffusive heat transfer in a cavity with a specified recirculating velocity and constant diffusivity corresponding to a nominal Péclet number ($Pe = u_{top} L/\Gamma$) of 1000 (giving P_Δ's of order 10 or more throughout the field for a 13 x 13 grid). The typical wiggles of central differencing are clear in Figure 10. The false diffusion of the hybrid TEACH scheme has grossly corrupted the computation in Figure 11 by contrast with the results of the highly accurate QUICK algorithm shown in Figure 9.

ADVICE TO FEM DEVELOPERS

In conclusion then, what can be learnt from the mistakes of the past?
First of all:

There's no need to tolerate the unphysical wiggles of centered schemes.

Secondly:

Don't suppress the wiggles merely by the arbitrary introduction of artificial diffusion of first/second-order upstream-weighted schemes.

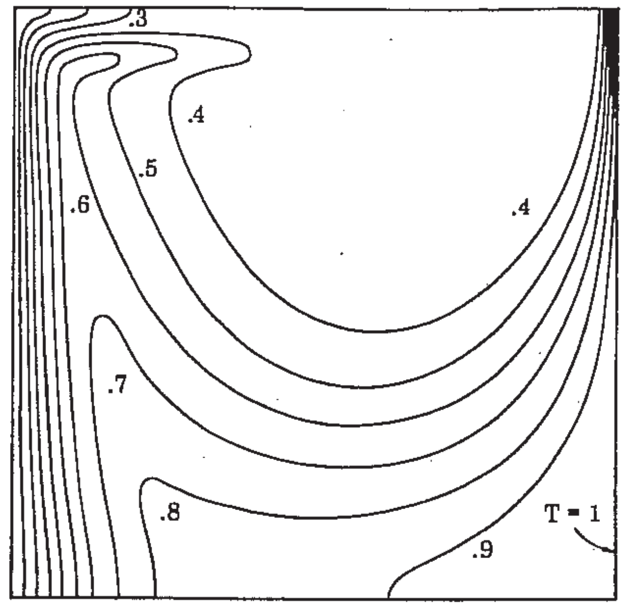


Figure 9. QUICK method, Pe = 1000.

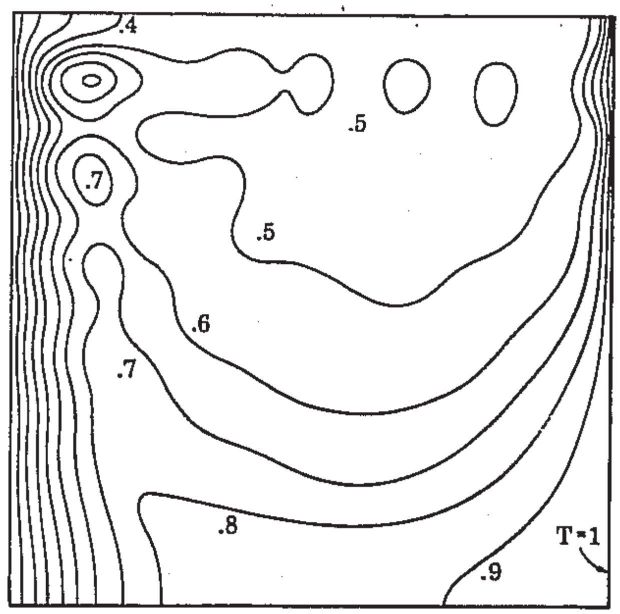


Figure 10. Central differencing, Pe = 1000.

Thirdly:

Remember that even optimally weighted schemes revert to the pure upstream form for $P_\Delta \gtrsim 5$; i.e., the high convection regime. And don't be thrown off the scent by the old SWANSEA red herring — Taylor series is just as refreshing as ever (probably more so).

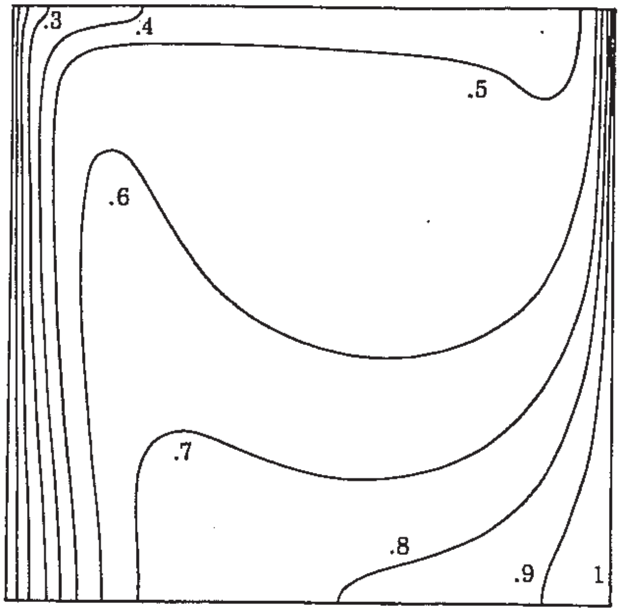


Figure 11. TEACH code, Pe = 1000.

And finally:

Try to extend the accuracy of the *third* order Laplacian operator to the convective term, thereby coming up with a new SWANSEA philosophy based on Sending Wiggles Away with Numerically Stable Extended Accuracy.

Clearly it's time to write-off the wiggle riddles, wind-up unwinding, and get on with some QUICK solutions to some real problems. [Of course, there may still be some finite differences of opinion on this.]

REFERENCES

1. Roache, P.J., *Computational Fluid Dynamics*, Hermosa Publishers, Albuquerque, New Mexico, 1976.
2. Fromm, J., "The Time Dependent Flow of an Incompressible Viscous Fluid," *Methods in Computational Physics*, Vol. 3, 1964, pp. 345-382.
3. Leonard, B.P., "A Stable and Accurate Convective Modelling Procedure Based on Quadratic Upstream Interpolation," *Computer Methods in Applied Mechanics and Engineering*, Vol. 19, June 1979.
4. Gentry, R.A., Martin, R.E., and Daly, B.J., "An Eulerian Differencing Method for Unsteady Compressible Flow Problems," *Journal of Computational Physics*, Vol. 1, 1966, pp. 87-118.
5. Spalding, D.B., "A Novel Finite Difference Formulation for Differential Expressions Involving Both First and Second Derivatives," *International Journal for Numerical Methods in Engineering*, Vol. 4, 1972, pp. 551-559.
6. Christie, I., Griffiths, D.F., and Mitchell, A.R., "Finite Element Methods for Second Order Differential Equations with Significant First Derivatives," *International Journal for Numerical Methods in Engineering*, Vol. 10, 1976, pp. 1389-1396.
7. Heinrich, J.C., Huyakorn, P.S. and Zienkiewicz, O.C., "An 'Upwind' Finite Element Scheme for Two-Dimensional Convective Transport Equation," *International Journal for Numerical Methods in Engineering*, Vol. 11, 1977, 131-143.
8. Huyakorn, P.S., "Solution of the Steady State Convective Transport Equation Using an Upwind Finite Element Scheme," *Applied Mathematical Modelling*, Vol. 1, 1977, p. 187.

9. Ikegawa, M., "A New Finite Element Technique for the Analysis of Steady Viscous Flow Problems," *International Journal for Numerical Methods in Engineering*, Vol. 14, 1979, pp. 104-113.
10. Hughes, T.J.R., Liu, W.K., and Brooks, A., "Finite Element Analysis of Incompressible Viscous Flows by the Penalty Function Formulation," *Journal of Computational Physics*, Vol. 30, 1979, pp. 1-60.
11. Moult, A., Burley, D., and Rawson, H., "The Numerical Solution of 2D Steady Flow Problems by the Finite Element Method," *International Journal for Numerical Methods in Engineering*, Vol. 14, 1979, pp. 11-35.
12. See, for example, several papers (which will remain anonymous) in the *Proceedings of the Symposium on Turbulent Shear Flows*, The Pennsylvania State University, April 1977.
13. de Vahl Davis, G. and Mallinson, G.D., "An Evaluation of Upwind and Central Difference Approximations by a Study of Recirculating Flow," *Computers and Fluids*, Vol. 4, 1976, p. 29.
14. Raithby, G.D. and Torrance, K.E., "Upstream-Weighted Schemes and their Application to Elliptic Problems Involving Fluid Flow," *Computers and Fluids*, Vol. 2, 1974, pp. 191-206.
15. Raithby, G.D., "A Critical Evaluation of Upstream Differencing Applied to Problems Involving Fluid Flow," *Computer Methods in Applied Mechanics and Engineering*, Vol. 9, 1976, 75-103.
16. Leonard, B.P., "A Consistency Check for Estimating Truncation Error Due to Upstream Differencing," *Applied Mathematical Modelling*, Vol. 2, December 1978, pp. 239-244.
17. Tuann, S.Y. and Olson, M., "Review of Computing Methods for Recirculating Flows," *Journal of Computational Physics*, Vol. 29, October 1978, pp. 1-19.
18. Hodge, J.K., Stone, A.L., and Miller, T.E., "Numerical Solution for Airfoils near Stall in Optimized Boundary-Fitted Curvilinear Coordinates," *AIAA Journal*, Vol. 17, May 1979, pp. 458-464.
19. Roberts, K.V. and Weiss, N.O., "Convective Difference Schemes," *Mathematics of Computation*, Vol. 20, 1966, pp. 272-299.
20. Gosman, A.D., Pun, W.M., Runchal, A.K., Spalding, D.B. and Wolfshtein, M., *Heat and Mass Transfer in Recirculating Flows*, Academic Press, London, 1969.

Computational Techniques & Applications: CTAC-83
J. Noye & C. Fletcher (Editors)

THIRD-ORDER UPWINDING AS A RATIONAL BASIS FOR COMPUTATIONAL FLUID DYNAMICS

B.P. Leonard

City University of New York
Staten Island, N.Y.
U.S.A.

Classical numerical methods applied to problems in diffusion, wave-motion, and elasticity, have been highly successful -- due to inherent feedback stability of central differencing in modelling even-order spatial derivatives. However, central methods lack this stability when applied to odd-order derivatives -- the major cause of problems in modelling the convection in fluid flow. Irrational "remedies" have compounded the difficulties; but rational design of stable and accurate algorithms shows that third-order upwinding provides a clean and robust foundation for computational fluid dynamics.

INTRODUCTION

Computational fluid dynamics is distinguished from the other branches of computat-
ional physics by the importance of the convection term -- a first-order spatial
derivative. "Classical" numerical techniques (essentially second-order central
differencing) evolved in response to problems dominated by diffusion, wave-motion,
or elasticity -- all involving even-order spatial derivatives. When applied to the
first-derivative convection term of fluid dynamics, central-difference methods gen-
erate a variety of "difficulties", ranging from bizarre and clearly unphysical osc-
illations to catastrophic numerical divergence. Historically, various characteris-
tics of the convection term have been blamed; e.g., the problem is sometimes known
as the "nonlinear instability" or put down to the variable coefficient (convecting
velocity). But the primary cause of the difficulty is much more basic, stemming
from an habitual use of the wrong old tools for the "new" problem: it occurs when-
ever central-difference methods (of any order of accuracy) are used to model odd-
order spatial derivatives.

Some insight can be gained by studying the "feedback sensitivity" of difference op-
erators; i.e., the sensitivity to perturbations of the central node value. If this
is negative, stable negative feedback occurs and numerical noise is damped out. If
positive, of course, divergence will follow quickly. In the case of the classical
central difference models of even-order derivatives, the feedback sensitivity hap-
pens to be strictly negative. This is the primary (and somewhat fortuitous) reason
for central difference methods' notable success for problems dominated by even-
order derivatives. For these problems, the classical techniques are accurate and
stable; or, in popular jargon: clean and robust. In hindsight, one could say that,
for successful computation it is quite sensible to require that, in general:

> The inherent feedback sensitivity of finite-difference operators should be strictly negative.

However, for odd-order derivatives, central differencing results in essentially
neutral sensitivity, and undamped parasitic oscillations may pervade the computat-
ions. In a control-volume formulation of convection, the feedback sensitivity may
actually be positive (in decelerating regions); and in coupled equations, a comp-
uted diffusion coefficient may take on an unphysical negative value. In either

case, computational disaster usually follows rapidly!

One of the central aims of this paper is to drive home the fact that, because of a
lack of inherent feedback sensitivity:

> Central difference methods (of any order of accuracy) are not appropriate tools for modelling odd-order spatial derivatives, in particular the first-order convection term.

Non-centered schemes offer a possible improvement; e.g., over the past decade or
so, first-order upwinding techniques have become very popular. But there is now a
widening realization that, by using these methods, stability is obtained at too
high a cost in accuracy. The difficulty is best portrayed in terms of the artific-
ial numerical diffusion introduced by the truncation error in the usual form of
first-order upwinding. This is most pronounced in problems involving (what should
be) thin shear layers at an oblique angle to the grid mesh, and in problems involv-
ing sources or unsteady terms. The problem is that, in the convection-diffusion
equation, modelling high convection is equivalent to correctly simulating very
small (or negligible) physical diffusion. The second-derivative truncation error
inherent in first-order upwinding totally corrupts this intention. Again in hind-
sight (and on making reasonable smoothness assumptions), it seems clear that one
should require that:

> The order of the spatial derivatives introduced by the truncation error in finite-difference operators should be higher than that of any modelled physical term.

This simple and rational criterion (quite properly) excludes first-order upwinding
as a technique for modelling the convection-diffusion equation. The same conclus-
ion has been reached in several studies concerned with directly measuring the ef-
fects of the artificial diffusion introduced by first-order upwinding [1-14].

A number of techniques have been developed in an attempt to model high convection,
which could be called Lagrangian in nature [15]. For one-dimensional pure convect-
ion of a scalar ϕ at constant velocity, the equation

$$\phi(x,\ t+\Delta t) \ = \ \phi(x-u\Delta t,\ t) \tag{1}$$

is, of course, exact in principle. Given discrete spatial values of ϕ at time t,
the numerical problem becomes one of interpolation in estimating the right-hand
side of Equation (1). Various interpolation schemes will be briefly described. It
will be found, for example, that centered polynomial interpolation schemes of even
degree are much more oscillatory than odd-degree upwind-biassed methods. Once
again, this distinction can be related to feedback sensitivity and spatial truncat-
ion error. The following summarizes the behaviour of Lagrangian methods of this
kind:

> Methods involving odd-order spatial derivatives in the leading truncation error term are much more oscillatory (dispersive) than those involving even-order derivatives.

This is true even in the case of upwinded even-degree polynomial methods; e.g.,
second-order upwinding involves a dispersive third-derivative in the truncation
error. In the test problem of a uniformly convected step profile, for example,
one sees this distinction in dispersive characteristics quite clearly. In addit-
ion, central methods become much more dispersive as the time step is decreased,
whereas the dispersion of upwinded odd-degree methods is practically insensitive
to Δt.

When considering the attributes of higher-order methods, of course, one must be
concerned with the computational costs in comparison with the benefits gained.
For a given method, accuracy improves in principle with spatial grid refinement;

for a given grid, accuracy may or may not improve with an increase in the formal order of the algorithm (e.g., for pure convection, Nth-order central methods are less accurate than (N-1)th-order upwinding). However, for methods in which fixed-grid accuracy improves with order, the following is true:

> For a prescribed global accuracy, higher-order methods are computationally more economical.

The cost per space-time grid point certainly increases with order (roughly arithmetically); but, to achieve the same global accuracy, a lower-order method will need the number of spatial grid points to be increased by a factor in each coordinate direction, and the number of time (or iteration) steps must be increased by the same factor. The cost of this geometric increase in the number of space-time grid points of the lower-order method far outweighs the arithmetic increase in local cost of the higher-order algorithm. In terms of machine running costs, there is thus a strong motivation toward higher-order methods. However, one must also consider developmental costs: above fourth-order, algorithmic complexity begins to escalate rapidly, and soon outstrips the normal powers of human comprehension and associated debugging skills.

Since central-difference methods of any order are not appropriate for convection-dominated flows, and since first-order upwinding must be excluded on the basis of its artificial diffusion, and since second- (and fourth-) order upwind methods are too dispersive, and since fifth- and higher-order upwind methods are excessively complex, one is drawn to the logical conclusion that:

> THIRD-ORDER UPWINDING FORMS A RATIONAL BASIS FOR THE DEVELOPMENT OF CLEAN AND ROBUST ALGORITHMS FOR COMPUTATIONAL FLUID DYNAMICS.

This is not to say that third-order upwinding has absolutely no problems of its own. For example, under pure convection conditions, third-order upwinding may generate small (~5%) overshoots in modelling a step, or a few wiggles near a downstream boundary [7]; but these problems can be overcome by using alternate interpolation techniques locally. By monitoring local curvature of the convected variables, an algorithm can easily be designed which changes over automatically from third-order upwinding in the "smooth" regions (i.e., most of the flow domain) to a more appropriate interpolation (e.g., exponential upwinding) in local regions of rapidly varying gradients. In this way, sudden jumps can be negotiated monotonically. The EULER-QUICK shock-capturing compressible Euler code, for example, produces shocks at the correct location, with the proper strength, and with a numerical shock structure only two grid points wider than that obtainable with exact solutions [16]. In isentropic regions, third-order upwinding generates results which are graphically indistinguishable from theoretically exact; in particular:

> Third-order upwinding conserves stagnation pressure in isentropic regions.

This inherent property of conserving stagnation pressure in inviscid regions is a strong indication of the algorithm's non-dissipative nature, and has been confirmed for incompressible flows, as well [5]. Static pressure behaviour is correspondingly good [4, 16].

Finally, some comments on computational grids are in order. For one-dimensional flow with non-reversing convecting velocities at control-volume cell faces, second-order central differencing requires three points for updating the central node value due to convection and diffusion. By contrast, first-order upwinding requires only two for convection (physical diffusion usually being ignored), whereas third-order upwinding requires four. These are shown in Figure 1a. When reversing velocities are allowed for, the situation is as shown in Figure 1b. Note that, in this case, first- and second-order methods both require three points, whereas third-order upwinding needs four (as does fourth-order central). In two dimensions, allowing for velocity reversals, first- and second-order methods involve the well-known five-point "star" (as does the Laplacian operator for diffusion).

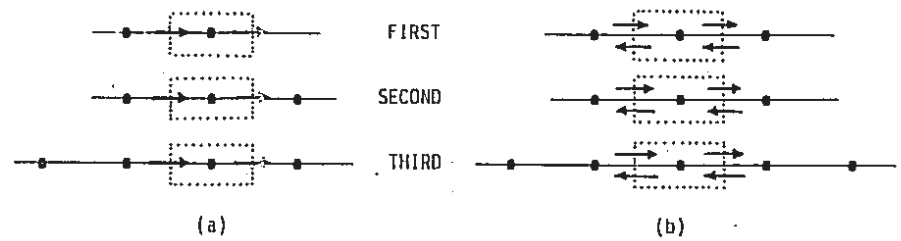


Figure 1. One-dimensional spatial grids.

Third-order upwinding requires an additional point in the normal direction for each face (i.e., four more, making a total of nine); this is enough to guarantee the essential convective stability of third-order upwinding, and from a practical point of view generates results which are essentially third-order accurate. However, for strict formal third-order accuracy, four additional corner points are required, to take account of transverse curvature effects. The complete star is shown in Figure 2 (with the latter corner points shown by hollow circles).

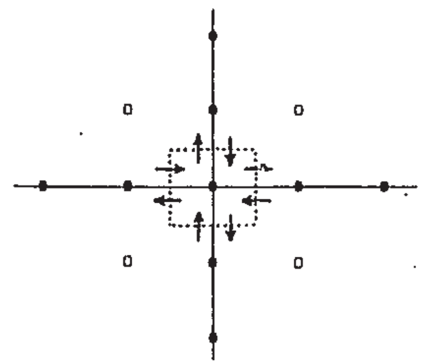


Figure 2. Computational star for third-order upwinding in two dimensions.

By an obvious extension of these ideas, it can be seen that in three dimensions, first- and second-order require a seven-point configuration; quasi-third-order-upwinding requires one more for each of the six faces (i.e., thirteen in all), whereas a formally fully third-order algorithm would require a total of twenty-five. The effects of including or deleting transverse curvature terms have not yet been fully tested; however, if one makes a worst-case assumption (i.e., only second-order accurate if deleted), the inherent stability of the retained normal curvature terms is enough to justify the additional normal-direction grid points.

FEEDBACK SENSITIVITY AND TRUNCATION ERROR

Consider the unsteady convection and diffusion of a scalar in the presence of source terms:

$$\frac{\partial \phi}{\partial t} = -\nabla\cdot(\underline{u}\phi) + \nabla\cdot(\Gamma\nabla\phi) + S \qquad (2)$$

Assume that a computational algorithm for a node value ϕ_i can be written

$$\frac{\partial \phi_i}{\partial t} = RHS \tag{3}$$

where RHS represents the modelled terms on the right-hand side. In general, RHS will involve some dependence on ϕ_i; thus the evolution of numerical perturbations in ϕ_i can be studied by taking the variation of Equation (3) with respect to ϕ_i, giving

$$\frac{\partial \delta \phi_i}{\partial t} = \frac{\partial RHS}{\partial \phi_i} \delta \phi_i \tag{4}$$

which has a formal solution

$$\delta \phi_i = \exp[\, \int \Sigma \, dt \,] \tag{5}$$

where the feedback sensitivity, Σ, is given by

$$\Sigma = \frac{\partial RHS}{\partial \phi_i} \tag{6}$$

Clearly, a positive Σ would lead to explosive growth of numerical perturbations -- a highly undesirable occurrence. If $\Sigma = 0$, there is no inherent feedback and certain types of perturbations (e.g., wiggles) can be superimposed on the solution without affecting the RHS, thus generating no automatic corrective action. A negative Σ assures that the algorithm will provide inherent damping of random fluctuations -- a highly desirable property.

In a model one-dimensional version of Equation (2) involving constant u and Γ, the "classical" (second-order) finite-difference model of diffusion is

$$\Gamma \left[\frac{\phi_{i+1} - 2\phi_i + \phi_{i-1}}{\Delta x^2} \right] = \Gamma \left[\left(\frac{\partial^2 \phi}{\partial x^2} \right)_i + \frac{1}{12} \phi_i^{(iv)} \Delta x^2 + \ldots \right] \tag{7}$$

Note that this operator is exact for a cubic polynomial interpolated through (i-1), i, (i+1), and either (i-2) or (i+2). The feedback sensitivity is $-2\Gamma/\Delta x^2$; i.e., strictly negative for finite (physical) diffusion. The fourth-order model for diffusion has a similar structure (as does the second-order model of the biharmonic operator).

Second-order central differencing of the model convection term leads to

$$- u \left[\frac{\phi_{i+1} - \phi_{i-1}}{2\Delta x} \right] = - u \left[\left(\frac{\partial \phi}{\partial x} \right)_i + \frac{1}{6} \phi_i''' \Delta x^2 + \ldots \right] \tag{8}$$

which has the same formal order of accuracy as Equation (7), although it is exact only for quadratic interpolation. However, a major problem exists for this operator: $\Sigma \equiv 0$. Thus, in a second-order central-difference model of the convection-diffusion equation, feedback stabilization depends on the physical diffusion term. Sudden jumps in the streamwise direction lead to upstream-penetrating oscillations (the penetration distance is linearly proportional to the cell Reynolds or Peclet number [10]).

Going to higher-order central methods does not correct this inherent neutral sensitivity in modelling odd-order derivatives. For example, the fourth-order central-difference model for convection is

$$- u \left[\frac{- \phi_{i+2} + 8\phi_{i+1} - 8\phi_{i-1} + \phi_{i-2}}{12 \Delta x} \right]$$
$$= - u \left[\left(\frac{\partial \phi}{\partial x} \right)_i - \frac{1}{4} \phi_i^{(v)} \Delta x^4 + \ldots \right] \tag{9}$$

and the corresponding second-order version of the third derivative is

$$\frac{\phi_{i+2} - 2\phi_{i+1} + 2\phi_{i-1} - \phi_{i-2}}{2 \Delta x^3} = \left(\frac{\partial^3 \phi}{\partial x^3} \right)_i + \frac{1}{4} \phi_i^{(v)} \Delta x^2 + \ldots \tag{10}$$

both of which have neutral sensitivity. In practical flow situations (variable u), the convective feedback sensitivity becomes positive in decelerating regions; this is very often the cause of numerical divergence.

First-order upwinding of convection results in

$$- u \left[\frac{\phi_i - \phi_{i-1}}{\Delta x} \right] = - u \left[\left(\frac{\partial \phi}{\partial x} \right)_i - \frac{1}{2} \phi_i'' \Delta x + \ldots \right] \qquad u > 0 \tag{11}$$

and

$$- u \left[\frac{\phi_{i+1} - \phi_i}{\Delta x} \right] = - u \left[\left(\frac{\partial \phi}{\partial x} \right)_i + \frac{1}{2} \phi_i'' \Delta x + \ldots \right] \qquad u < 0 \tag{12}$$

The feedback sensitivity is strictly negative: $\Sigma = -|u|/\Delta x$. But note that the leading truncation error term is equivalent to a numerical diffusion term with an artificial diffusion coefficient

$$\Gamma_{num} = \frac{|u| \Delta x}{2} \tag{13}$$

which, of course, corrupts (or dominates) the physical diffusion unless the grid is fine enough so that the cell peclet number, $P_\Delta = |u| \Delta x/\Gamma$ is very much smaller than 2.

The basic algorithm of the well-known TEACH code [17] uses central differencing for both convection and diffusion if $P_\Delta \le 2$, and first-order upwinding (with physical diffusion neglected) if $P_\Delta > 2$. In this hybrid strategy, the effective cell Peclet number is

$$\left. \begin{aligned} P_\Delta^* &= P_\Delta \qquad \text{for } P_\Delta \le 2 \\ P_\Delta^* &\equiv 2 \qquad \text{for } P_\Delta > 2 \end{aligned} \right\} \tag{14}$$

A slight modification of this procedure is advocated by the "optimal upwinding" fallacy [18]:

$$P_\Delta^* = 2 \tanh (P_\Delta /2) \tag{15}$$

A recent text-book on numerical heat transfer and fluid flow [19] has introduced a further red herring by going to considerable lengths to approximate Equation (15) by (less expensive) quintic polynomials. The futility of such exercises should, by now, be obvious. For high convection ($P_\Delta > 2$ for hybrid or $P_\Delta \geq 5$ for optimal), these schemes consist of pure first-order upwinding for convection with all physical diffusion terms totally neglected. Very often, these methods are used to test quite sophisticated turbulence models under highly convective conditions. One goes to the expense of solving (at least) two additional transport equations for turbulence properties and several auxiliary variables, and then proceeds to totally ignore them in modelling the momentum and thermal or species transport equations! The unacceptable effects of the artificial diffusion associated with first-order methods are now well documented in the fundamental research literature. Unfortunately, because of natural time lags, these methods are currently becoming even more firmly established as standard "handbook" techniques in the heat-transfer industry and related areas.

Second-order upwinding can be written, for $u > 0$,

$$- u \left[\frac{3\phi_i - 4\phi_{i-1} + \phi_{i-2}}{2 \Delta x} \right] = - u \left[\left(\frac{\partial \phi}{\partial x}\right)_i - \frac{1}{3}\phi_i''' \Delta x^2 + \frac{1}{4}\phi_i^{(iv)} \Delta x^3 + \dots \right] \tag{16}$$

with a similar expression for $u < 0$. Regardless of velocity direction, the feedback sensitivity is $\Sigma = -3|u|/2\Delta x$, which is encouraging. However, the truncation error, being dominated by the third-derivative term, is responsible for oscillatory behaviour (although much less than that of central methods).

The troublesome third-derivative term can be entirely eliminated by going to third-order upwinding, which can be written in symmetrical form (for $u \gtrless 0$) as:

$$- u \left[\frac{-\phi_{i+2} + 8\phi_{i+1} - 8\phi_{i-1} + \phi_{i-2}}{12 \Delta x} \right] - |u| \left[\frac{\phi_{i+2} - 4\phi_{i+1} + 6\phi_i - 4\phi_{i-1} + \phi_{i-2}}{12 \Delta x} \right]$$

$$= - u \left(\frac{\partial \phi}{\partial x}\right)_i - \frac{|u|}{12} \phi_i^{(iv)} \Delta x^3 + \frac{u}{4} \phi_i^{(v)} \Delta x^4 + \dots \tag{17}$$

in which the left-hand side is recognized as the fourth-order operator of Equation (9), modified by a fourth-difference term. The particular combination is consistent with interpolating a cubic polynomial through (i-1), i, (i+1), and (i-2) if $u > 0$, or (i+2) if $u < 0$. As noted in relation to Equation (7), this is also consistent with the second-order operator for diffusion. The corresponding combined model convection-diffusion equation can be written

$$\frac{\partial \phi_i}{\partial t} = S - u\left(\frac{\partial \phi}{\partial x}\right)_i + \Gamma \left(\frac{\partial^2 \phi}{\partial x^2}\right)_i - \frac{|u|}{12}\left(1 - \frac{1}{|P_\Delta|}\right) \phi_i^{(iv)} \Delta x^3 + \dots \tag{18}$$

and the combined feedback sensitivity is

$$\Sigma = -\frac{|u|}{2\Delta x} - \frac{2\Gamma}{\Delta x^2} = -\frac{|u|}{2\Delta x} \left(1 + \frac{4}{|P_\Delta|}\right) \tag{19}$$

which, of course, is always stabilizing.

LAGRANGIAN CONVECTION SCHEMES

Figure 3 shows the basis of several Lagrangian schemes for pure convection based on Equation (1).

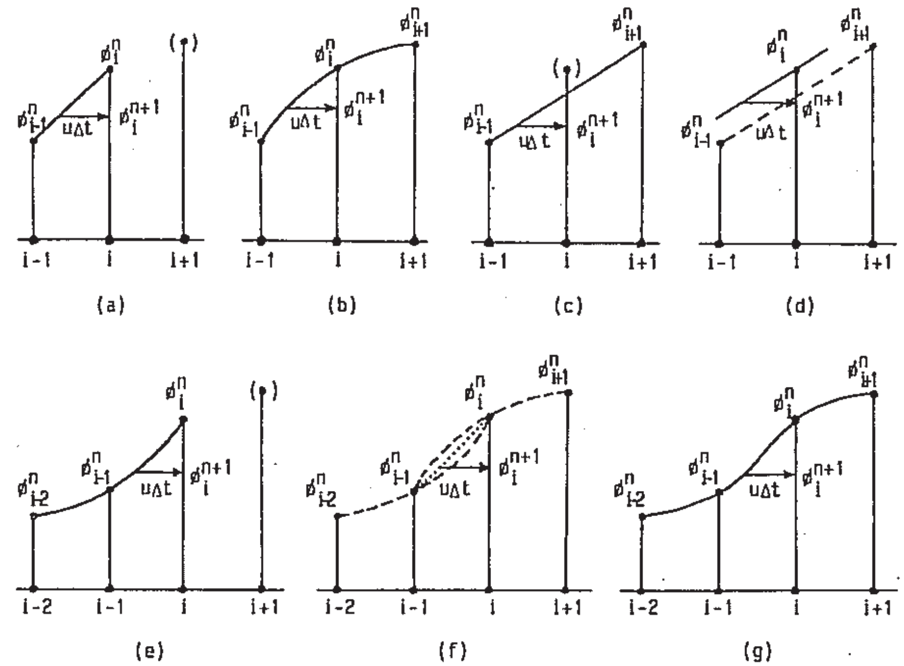


Figure 3. Lagrangian schemes for pure convection at constant velocity. (a) First-order upwinding; (b) second-order central (Leith's method); (c) the Lax scheme; (d) forward-time-central-space (FTCS); (e) second-order upwinding; (f) Fromm's method; (g) third-order upwinding (QUICKEST in inviscid limit).

In the case of upwind linear interpolation, for example, it is clear that the update algorithm is

$$\phi_i^{n+1} = \phi_i^n - u\Delta t \left(\frac{\phi_i^n - \phi_{i-1}^n}{\Delta x} \right) \tag{20}$$

which, of course, is the well-known first-order upwind method (with second-derivative truncation error).

All methods of this type can be written in conservative control-volume form as:

$$\frac{\phi_i^{n+1} - \phi_i^n}{\Delta t} = \frac{u}{\Delta x} (\phi_\ell - \phi_r) \tag{21}$$

where $\phi_\ell(i) = \phi_r(i-1)$. The notation refers to the left and right control-volume face values (half-way between nodes). For constant u and Δx, a measure of feedback sensitivity in this case is

$$\sigma = \frac{\partial}{\partial \phi_i} (\phi_\ell - \phi_r) \tag{22}$$

For example, for first-order upwinding ($\phi_\ell = \phi^n_{i-1}$, $\phi_r = \phi^n_i$),

$$\sigma_1 = -1 \tag{23}$$

Note, in particular, that this is independent of Δt.

The centered parabolic interpolation scheme, shown in Figure 3(b), results in

$$\phi_i^{n+1} = \phi_i^n - \frac{c}{2}(\phi^n_{i+1} - \phi^n_{i-1}) + \frac{c^2}{2}(\phi^n_{i+1} - 2\phi^n_i + \phi^n_{i-1}) \tag{24}$$

from which, using Equation (21), one can identify the left face convective flux value as

$$\phi_\ell = \frac{1}{2}(\phi^n_i + \phi^n_{i-1}) - \frac{c}{2}(\phi^n_i - \phi^n_{i-1}) \tag{25}$$

introducing the Courant number, $c = u\Delta t/\Delta x$. Leading truncation error is proportional to $\phi_i'''\Delta x^2$, implying second-order accuracy and (as seen later) oscillatory behaviour. The corresponding feedback sensitivity is

$$\sigma_2 = -c \tag{26}$$

Thus, as Δt (i.e., c) is reduced, the feedback sensitivity of the second-order method decreases in linear proportion. This scheme is also known as Leith's method [20] or (in the variable velocity form) the Lax-Wendroff method [21].

Another method due to Lax [22] is shown in Figure 3(c). In this case, the update algorithm is

$$\phi_i^{n+1} = \phi_i^n - \frac{c}{2}(\phi^n_{i+1} - \phi^n_{i-1}) + \frac{1}{2}(\phi^n_{i+1} - 2\phi^n_i + \phi^n_{i-1}) \tag{27}$$

which is Leith's method together with an added artificial diffusion equivalent to $\Gamma_{LAX} = (1-c^2)u\Delta x/2c$ (!).

The so-called forward-time-central-space (FTCS) method is portrayed in Figure 3(d); in this case, instead of interpolation, one approximates ϕ in the vicinity of node i by a first-order Taylor series about i, using central-differencing to approximate the first derivative. This gives

$$\phi_\ell = \frac{1}{2}(\phi^n_i + \phi^n_{i-1}) \tag{28}$$

and, in particular,

$$\sigma_{FTCS} = 0 \tag{29}$$

[As is well known, this algorithm is unconditionally unstable.]

In the case of second-order upwinding,

$$\phi_\ell = \frac{1}{2}(\phi^n_i + \phi^n_{i-1}) - \frac{c}{2}(\phi^n_i - \phi^n_{i-1}) - \frac{(1-c)}{2}(\phi^n_i - 2\phi^n_{i-1} + \phi^n_{i-2}) \tag{30}$$

and

$$\sigma_{2UP} = -c - \frac{3}{2}(1-c) = -\frac{3}{2} + \frac{c}{2} \tag{31}$$

Truncation error again involves ϕ_i''' (an oscillatory term).

Fromm's so-called second-order method of zero-average-phase-error [23] averages two quadratics (the central and upwind) to give

$$\phi_\ell = \frac{1}{2}(\phi^n_i + \phi^n_{i-1}) - \frac{c}{2}(\phi^n_i - \phi^n_{i-1}) - \frac{(1-c)}{4}(\phi^n_i - 2\phi^n_{i-1} + \phi^n_{i-2}) \tag{32}$$

and

$$\sigma_{FROMM} = -c - \frac{3}{4}(1-c) = -\frac{3}{4} - \frac{c}{4} \tag{33}$$

approaching $-3/4$ as $c \to 0$.

Finally, Figure 3(g) shows the pure-convection form of the consistent third-order upwind algorithm known as QUICKEST -- Quadratic Upstream Interpolation for Convective Kinematics with Estimated Streaming Terms -- [7]. The quadratic upstream interpolation is used in the control-volume formulation; for the Lagrangian form, the appropriate interpolation is an upwind-biassed cubic through the four points shown:

$$\phi_\ell = \frac{1}{2}(\phi^n_i + \phi^n_{i-1}) - \frac{c}{2}(\phi^n_i - \phi^n_{i-1}) - \frac{(1-c^2)}{6}(\phi^n_i - 2\phi^n_{i-1} + \phi^n_{i-2}) \tag{34}$$

gives the convected left face value, and the feedback sensitivity is

$$\sigma_3 = -c - \frac{(1-c^2)}{2} \tag{35}$$

which, of course, approaches $-1/2$ as $c \to 0$. Leading truncation error involves, in this case, $\phi_i^{(iv)}\Delta x^3$ (a non-oscillatory term).

Fourier von Neumann Analysis of Lagrangian Schemes

By studying the evolution of Fourier components of the form

$$\phi = A(t)\, e^{ikx} \tag{36}$$

for a range of the wave number k, one can determine the complex amplitude ratio (or "amplification factor") corresponding to a time increment Δt, i.e.,

$$G = \frac{\phi(x, t+\Delta t)}{\phi(x,t)} = \frac{A(t+\Delta t)}{A(t)} \tag{37}$$

For the exact (constant u) result, in the absence of diffusion, $A = e^{-iukt}$, and

$$G_{EXACT} = e^{-ic\theta} = \cos(c\theta) - i\sin(c\theta) \tag{38}$$

where $\theta = k\Delta x$. By computing G for the various polynomial convection schemes, one can get a measure of the accuracy of a given algorithm. If the expansion of the particular method's G in Cartesian form agrees with that of Equation (38) through θ^N terms, the algorithm is correctly denoted as being Nth-order accurate in both space and time (in terms of modelling unsteady, purely convective flow). Perhaps not surprisingly, Nth-degree polynomial Lagrangian methods generate Nth-order accurate G's. For example, it is not difficult to derive the following for first-order upwinding:

$$G_1(\theta) = 1 + c(\cos\theta - 1) - i\,c\sin\theta \qquad (39)$$

$$= 1 - \frac{c}{2}\theta^2 + O(\theta^4) - i\,[c\,\theta - \frac{c}{6}\theta^3 + O(\theta^5)] \qquad (40)$$

which, of course agrees with Equation (38) only through the first-order term in θ.

Leith's method has

$$G_2(\theta) = 1 + c^2(\cos\theta - 1) - i\,c\sin\theta \qquad (41)$$

$$= 1 - \frac{1}{2}(c\,\theta)^2 + O(\theta^4) - i\,[c\,\theta - \frac{c}{6}\theta^3 + O(\theta^5)] \qquad (42)$$

which has the correct second-order term, but is deficient at third-order. It is this third-order deficiency in the imaginary part which is reflected in poor phase behaviour of Leith's method.

For a particular algorithm, the solution for each wave component can be written

$$\phi = \phi_0 e^{i\,k\,(x - u_{ph}t)} \qquad (43)$$

where the phase velocity is given by

$$u_{ph} = -\frac{ph(G)}{c\,\theta}\,u \qquad (44)$$

[the exact phase velocity being the convecting velocity u, of course]. The fact that u_{ph} in general depends on θ means that in the computational simulation, different wavelengths move at different speeds, instead of the correct constant speed, u. This is the phenomenon known as dispersion. Centrally interpolated methods tend to have far greater dispersion than upwinded methods, the effect being worse for smaller c values (i.e., smaller time-step).

Third-order upwinding has a complex amplitude ratio given by

$$G_3(\theta) = 1 + c^2(\cos\theta - 1) - \frac{c}{6}(1-c^2)(\cos 2\theta - 4\cos\theta + 3) - i\,[c\sin\theta - \frac{c}{6}(1-c^2)(\sin 2\theta - 2\sin\theta)] \qquad (45)$$

$$= 1 - \frac{1}{2}(c\,\theta)^2 + \frac{c}{24}(2c^2+c-2)\,\theta^4 + O(\theta^6) - i\,[c\,\theta - \frac{1}{6}(c\,\theta)^3 + \frac{c}{120}(5c^2-4)\,\theta^5 + O(\theta^7)] \qquad (46)$$

which agrees with the expansion of G in Equation (38) through the critical third-order term. It might be noted at this point that the full QUICKEST [7] algorithm (including both convection and diffusion) has

$$G_{QUICKEST} = 1 - (2\alpha + c^2)(1 - \cos\theta) - 2c\,[\frac{1}{6}(1-c^2) - \alpha]\,(1 - \cos\theta)^2 - i\,\{c\sin\theta + 2c\sin\theta\,[\frac{1}{6}(1-c^2) - \alpha]\,(1 - \cos\theta)\} \qquad (47)$$

where α is the diffusion parameter $\Gamma\,\Delta t/\Delta x^2$. This agrees with the expansion of the exact complex amplitude ratio for the model convection-diffusion equation -- the Gaussian spiral [24]:

$$G_{EX} = e^{-\alpha\theta^2} e^{-ic\theta} \qquad (48)$$

through θ^3 terms involving both c and α ! In this interpretation, the QUICKEST algorithm is the "canonical explicit third-order" method for fluid transport for any combination of convection and diffusion (within the appropriate von Neumann stability boundary [7], of course).

The process of interpolating higher-degree polynomials can clearly be continued indefinitely, leading to correspondingly higher difference terms in the expressions for convected face values. Even-degree, centrally distributed schemes always result in highly dispersive algorithms (the leading truncation-error derivative is of odd order and the feedback sensitivity goes to zero linearly with c). Upwinded odd-degree methods involve even-order leading truncation-error derivatives and the feedback sensitivity is much less dependent on c (and approaches a constant as c goes to zero). Figure 4 shows the results of simulating the pure convection (at constant velocity) of an initial step profile in a convected passive scalar, for polynomial Lagrangian schemes from first through fifth degree; the Courant number is 0.5 in each case (for smaller c values, the shape of the third and higher odd-order profiles remains virtually insensitive to c, whereas the extent of the even methods' trailing oscillations increases rapidly).

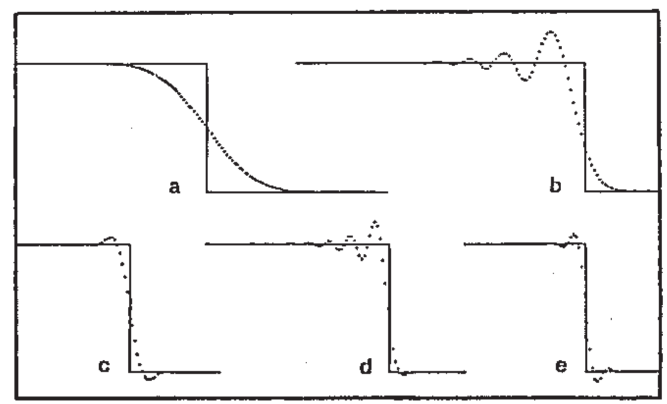


Figure 4. Details of model step convection at a nondimensional time of $t = 100$. (a) first-order upwinding; (b) Leith's method; (c) third-order upwinding; (d) fourth-order; (e) fifth-order.

Note the gross artificial diffusion of the first-order scheme giving typical error-function profiles. The trailing oscillations of the even order methods reflect the poor phase behaviour of these schemes: shorter wavelengths have sharply lagging phase velocities and not very much damping. [Upwind even-order methods (not shown) generate leading oscillations.] By contrast, upwind odd-order methods generally have excellent phase behaviour and, except for first-order (in which all wavelengths are strongly damped), they have good amplitude characteristics, as well.

From consideration of Figure 4 alone, it can be seen that third-order upwinding is clearly preferable to first-, second-, and fourth-order methods. Although sharper profiles can be obtained with fifth- and higher odd-order schemes, the complexity of the algorithms effectively eliminates such methods from serious consideration as practical simulation codes for multidimensional problems. Again, third-order upwinding emerges as the most rational basis for further development of computational techniques for unsteady highly convective flow. For example, the 5% overshoots in the third-order profile in Figure 4(c) can be entirely eliminated by using alternate interpolation (such as exponential upwinding) in the strong-curvature region. A similar strategy is used in the Exponential-Upwinding or Linear-Extrapolation Refinement of QUICK for solving the compressible Euler equations [16].

SUMMARY

Space does not permit a comprehensive survey outlining details of all the advantages of third-order upwinding over other techniques. The brief one-dimensional considerations discussed above indicate the essential features. Other attributes are apparent from the results of comparative tests and practical applications reported elsewhere. For example, an unsteady two-dimensional QUICKEST scheme has been applied to the simulation of vortex shedding from bluff bodies [25], showing excellent resolution of the vortex dynamics and agreement with experiments on shedding frequency; the same study showed that lower-order hybrid schemes generate anomolous results.

Several workers have successfully QUICKened 2D and 3D versions of TEACH-type codes. This is a relatively straight-forward procedure for any code based on a control-volume flux formulation [4, 5, 11, 12]. There have now been a number of carefully chosen numerical tests of QUICK against hybrid and other (e.g., skew-differencing) first-order schemes, involving linear problems (e.g., oblique convection of a scalar to test for cross-grid false diffusion) and nonlinear (laminar and turbulent) transport (such as driven-cavity flows, back steps, and impinging jets). Although there are clearly many problems which will need further development, it is now quite firmly established that third-order upwinding provides a firm foundation for building further refinements in specific applications. To quote from one of the most recent reports [5]:

> "QUICK emerged, overall, as decisively the most successful of the schemes. ... In our view it is the best of the simple interpolation schemes currently available and is well suited for incorporation in a robust, general-purpose solver for laminar or turbulent flows."

REFERENCES

[1] Brooks, A. and Hughes, T.J.R., Streamline-Upwind/Petrov-Galerkin Fomulation for Convection-Dominated Flows with Particular Emphasis on the Incompressible Navier-Stokes Equations, Computer Methods in Applied Mechanics and Engineering 32 (1982) 199.

[2] deVahl Davis, G. and Mallinson, G.D., An Evaluation of Upwind and Central Difference Approximations by a Study of Recirculating Flow, Computers and Fluids 4 (1976) 29.

[3] Hackman, L.P., Raithby, G.D., and Strong, A.B., Numerical Predictions of Flows Over Backward-Facing Steps, International Journal for Numerical Methods in Fluids (to appear).

[4] Han, T., Humphrey, J.A.C., and Launder, B.E., A Comparison of Hybrid and Quadratic-Upstream Differencing in High Reynolds Number Elliptic Flows, Computer Methods in Applied Mechanics and Engineering 29 (1981) 81.

[5] Huang, P.G., Launder, B.E., and Leschziner, M.A., Discretization of Nonlinear Convection Processes: A Broad Range Comparison of Four Schemes, Mechanical Engineering Department Report TDF/83/1, University of Manchester Institute of Technology (1983).

[6] Leonard, B.P., A Consistency Check for Estimating Truncation Error due to Upstream Differencing, Applied Mathematical Modelling 2 (1978) 239.

[7] Leonard, B.P., A Stable and Accurate Convective Modelling Procedure Based on Quadratic Upstream Interpolation, Computer Methods in Applied Mechanics and Engineering 19 (1979) 59.

[8] Leonard, B.P., The QUICK Finite Difference Method for the Convection-Diffusion Equation, in: Vichnevetsky, R. and Stepleman, R.S. (eds.), Advances in Computer Methods for Partial Differential Equations--III (IMACS, 1979) 292.

[9] Leonard, B.P., A Survey of Finite Differences of Opinion on Numerical Muddling of the Incomprehensible Defective Confusion Equation, in: Hughes, T.J.R. (ed.), Finite Element Methods for Convection Dominated Flows (AMD-34, ASME, New York, 1979) 1.

[10] Leonard, B.P., A Survey of Finite Differences with Upwinding for Numerical Modelling of the Incompressible Convective Diffusion Equation, in: Taylor, C. (ed.), Recent Advances in Numerical Methods in Fluids--Volume 2 (Pineridge Press, 1981).

[11] Leschziner, M.A., Practical Evaluation of Three Finite Difference Schemes for the Computation of Steady-State Recirculating Flows, Computer Methods in Applied Mechanics and Engineering 23 (1980) 293.

[12] Leschziner, M.A. and Rodi, W., Calculation of Annular and Twin Parallel Jets Using Various Discretization Schemes and Turbulence Model Variants, ASME Journal of Fluids Engineering 103 (1981) 352.

[13] Raithby, G.D., A Critical Evaluation of Upstream Differencing Applied to Problems Involving Fluid Flow, Computer Methods in Applied Mechanics and Engineering 9 (1976) 75.

[14] Raithby, G.D., Skew-Upstream Differencing for Problems Involving Fluid Flow, Computer Methods in Applied Mechanics and Engineering 9 (1976) 151.

[15] Forester, C.K., Higher Order Monotonic Convective Differencing Schemes, Journal of Computational Physics 23 (1977) 1.

[16] Leonard, B.P., The EULER-QUICK Code, in: Taylor, C. (ed.), Numerical Methods in Laminar and Turbulent Flow--III (Seattle, 1983).

[17] Gosman, A.D., Pun, W.M., Runchal, A.K., and Wolfshtein, M., Heat and Mass Transfer in Recirculating Flows (Academic Press, 1969).

[18] Christie, I., Griffiths, D.F., and Mitchell, A.R., Finite Element Methods for Second Order Differential Equations with Significant First Derivatives, International Journal for Numerical Methods in Engineering 10 (1976) 1389.

[19] Patankar, S.V., Numerical Heat Transfer and Fluid Flow (McGraw-Hill, 1980).

[20] Leith, C.E., Numerical Simulation of the Earth's Atmosphere, Methods in Computational Physics 4 (1965) 1.

[21] Lax, P.D. and Wendroff, B., Difference Schemes with High Order of Accuracy for Solving Hyperbolic Equations, Communications of Pure and Applied Mathematics 17 (1964) 381.

[22] Lax, P.D., Weak Solutions of Nonlinear Hyperbolic Equations and Their Numerical Computation, Communications on Pure and Applied Mathematics 7 (1954) 159.

[23] Fromm, J.E., A Method for Reducing Dispersion in Convective Difference Schemes, Journal of Computational Physics 3 (1968) 176.

[24] Leonard, B.P., Note on the von Neumann Stability of the Explicit FTCS Convective Diffusion Equation, Applied Mathematical Modelling 4 (1980) 401.

[25] Davis, R.W. and Moore, E.F., A Numerical Study of Vortex Shedding from Rectangles, Journal of Fluid Mechanics 116 (1982) 475.