THE UNIVERSITY OF READING

High Order Advection Schemes: Automatic Generation

by

R. K. Lashley

Numerical Analysis Report 5/99

DEPARTMENT OF MATHEMATICS

High Order Advection Schemes: Automatic Generation

R. K. Lashley [*]

September 17, 1999

1 Introduction

The advection equation is widely encountered in the field of atmospheric and fluid modeling, often as a significant component of a more complex equation. It's significance and regular occurrence in these problems make it an important equation to model accurately and efficiently. Many methods have been proposed and accepted as suitable approximations to the one-dimensional equation [6] but few, if any, of these have simple and equally successful extensions to two or more dimensions. Most of these extensions rely on reducing the two-dimensional equation to two one-dimensional equations which may be formally as accurate as the one dimensional methods under certain conditions but neglect corner effects when flow is at an angle to the grid giving poor shape preservation of the advected profile. Fully two (and higher) dimensional finite volume schemes have been developed but for many applications the method of directional splitting is still used.

One benefit of many of the one-dimensional schemes is the ease with which they may be extended to produce higher order, more accurate schemes. For example, schemes based around the Taylor expansion can often be made more accurate by retaining more terms in the approximation. In a similar scheme in two or more dimensions there are cross terms (e.g. $f_x \times f_y$); it is not so clear which of these should be retained and what effect they have on the scheme. One-dimensional schemes also lend themselves to easy analysis of the 'best' stencil to use, usually centred or upwinded stencils are used. With a two-dimensional scheme the upwind direction is unlikely to align with the grid, making selecting the 'best' stencil a less straightforward task. The 'best' stencil is one which provides the best approximation over all possible flow speeds and directions, not just for a specific case.

A problem that occurs when using high order advection schemes is that the numerical solution will often produce non-physical oscillations close to steep

[*] Supervised by J Thuburn and M Baines

gradients. These oscillations cause spurious maxima and minima in the numerical solution which can in turn cause negative values of the advected quantity which may be physically impossible, such as water vapour amounts in the atmosphere. There have been several methods suggested for removing these oscillations (see [11]) of which the flux-limiting method lends itself most readily to this method as it is separate from the scheme being used and has an extension to higher dimensions [10].

This report discusses the automatic generation of high order schemes in one and two dimensions using Transient Interpolation Modeling [2] in one dimension and an extension of this and Leonard's UTOPIA scheme [3] in two dimensions. Different schemes will be generated for different directions and speeds of the flow across the faces on a regular grid, automation enables all these scheme to be generated and stored should the same flow arise elsewhere in space or time. Along with an assessment of the accuracy of these schemes a cost/accuracy comparison is carried out between higher order schemes and increased grid resolution. In two dimensions a multidimensional flux-limiter is used to prevent spurious oscillations occurring in the numerical solutions, it's effects on error measurements are also noted. It is hoped that this method can be extended to work on unstructured grids on the sphere where the automation will be required.

The results in one-dimension concur with those of Leonard [2] and the two-dimensional results show a similar behaviour. There appears to be a level beyond which there is little increase in accuracy from increasing the order and this is certainly not cost effective. The results also highlight a number of other considerations which must be taken into account, including the representation of the flow-field and the stability of the matrix inversion routine used.

2 One-Dimensional Advection

The one-dimensional advection equation for advection of tracer ϕ with non-uniform, constant velocity u is

$$\phi_t = -u\,(x)\,\phi_x \ . \tag{1}$$

This can be integrated over a time step Δt and a control volume in space from $-\frac{\Delta x}{2}$ to $\frac{\Delta x}{2}$ (on a uniform grid) across cell boundary i. This integration leads to an exact formula for the update equation

$$\phi_i^{n+1} = \phi_i^n - (\alpha_r \phi_r^* - \alpha_l \phi_l^*) \ , \tag{2}$$

where ϕ_i^n is the total amount of tracer in the i^{th} grid box at the n^{th} time step, $\alpha = \frac{u\Delta t}{\Delta x}$ is the Courant number and the fluxes at the left and right volume faces are the $\alpha\phi^*$. The value of ϕ^* is difficult to determine but may be approximated in many different ways, several of which come under the heading of Transient Interpolation Modeling. This involves fitting a polynomial over a region around the boundary in question and integrating it over the length swept across the boundary in one time step (a full description and derivation of the process can be found in [2]). In order to fit an N^{th} order polynomial across a boundary $N+1$ conditions are required; these come from insisting that the integral of the polynomial sought over a control volume is equal to the amount of tracer in that volume ($\phi\Delta x$). If N is odd then this is done for the first $\frac{N+1}{2}$ volumes each side of the face (a centred scheme), while if N is even $\frac{N}{2}$ boxes each side of the face are used plus one more from the upwind side of the face (an upwind scheme). This can be done for a general distribution of ϕ which will lead to the generation of a general set of coefficients which can be applied to the values of α and ϕ in the region of each face.

This process can readily be done by hand for low order polynomials to produce the First-Order Upwind scheme using a zeroth order polynomial, the second order scheme of Lax and Wendroff using a first order polynomial and Leonard's QUICKEST scheme using a second order polynomial. Leonard went on to calculate the coefficients for schemes up to ninth order (eighth order polynomial) using this method (see [2]). The method employed in this work was to write a computer program in FORTRAN90 to perform these integrations, tabulate the results and then use them in a time-stepping routine. The method begins by finding the coefficients a_j of the interpolating polynomial given by insisting that

$$\int_{cell\ i} \sum_{j=0}^{N} a_j x^j \, dx = \phi_i \Delta x_i \ , \tag{3}$$

for each cell in the stencil. The grid need not be uniform, hence the subscript on the Δx. This may be re-written as

$$\phi_i \Delta x_i = \sum_{j=0}^{N} \int_{S_i}^{S_i+\Delta x_i} a_j x^j \, dx \ , \tag{4}$$

where S_i is the distance of the left boundary of the grid box from an arbitrary origin. Taking the origin to be the boundary across which the flux is being approximated, the following $N+1$ equations for the $N+1$ a_j's can be derived;

$$\begin{aligned} \phi_i &= a_0 + \sum_{j=1}^{N} a_j \sum_{k=0}^{j} \frac{j!}{(j-k)!(k+1)!} \Delta x_i^k S_i^{j-k} && , \ S_i \neq 0 \\ \phi_m &= a_0 + \sum_{j=1}^{N} \frac{a_j \Delta x_m^j}{j+1} && , \ S_m = 0 \ . \end{aligned} \tag{5}$$

The solution of these equations for the a_j's is a matrix inversion problem of the form

$$\begin{pmatrix} 1 & P_1(\Delta x_1, S_1) & P_2(\Delta x_1, S_1) & \cdots & \cdots & \cdots & P_N(\Delta x_1, S_1) \\ 1 & P_1(\Delta x_2, S_2) & P_2(\Delta x_2, S_2) & \cdots & \cdots & \cdots & P_N(\Delta x_2, S_2) \\ 1 & P_1(\Delta x_3, S_3) & P_2(\Delta x_3, S_3) & \cdots & \cdots & \cdots & P_N(\Delta x_3, S_3) \\ \vdots & \vdots & \vdots & \ddots & & & \vdots \\ 1 & \hat{P}_1(\Delta x_m) & \hat{P}_2(\Delta x_m) & & \ddots & & \hat{P}_{N-1}(\Delta x_m) \\ \vdots & \vdots & \vdots & & & \ddots & \vdots \\ 1 & P_1(\Delta x_N, S_N) & P_2(\Delta x_N, S_N) & \cdots & \cdots & \cdots & P_N(\Delta x_N, S_N) \end{pmatrix} \underline{a} = \underline{\phi} \ . \tag{6}$$

Assuming S_i is $O(\Delta x_i)$, i.e. N is much less than the number of boxes in the domain, P_k is a polynomial of order $(\Delta x_i)^k$ making the matrix ill-conditioned. This ill-conditioning can be lessened by factoring out a diagonal matrix of general Δx's (found either by averaging or inspired guess), rewriting eq 6 in this way gives

$$Q \ Diag[1, \Delta x, \Delta x^2, \cdots, \Delta x^{N-1}]\underline{a} = \underline{\phi} \ , \tag{7}$$

so

$$a_j = \frac{1}{\Delta x^j} \sum_{i=1}^{N+1} Q_{ij}^{-1} \phi_i \ , \tag{8}$$

where Q_{ij}^{-1} is the $i, j+1^{th}$ element of Q^{-1}. The closer to unity the ratio $\frac{\Delta x_{max}}{\Delta x_{min}}$ is, the better the conditioning of the matrix, Q. This procedure gives the coefficients of the interpolating polynomial in terms of a general tracer distribution, $\phi(x)$. Q must be non-singular if $N+1$ distinct boxes are used in the stencil, (i.e. boxes can only be counted once in each stencil and only have one value of tracer amount) and $\Delta x_i \neq 0$. The flux across the boundary, $\alpha_l \phi_l$, can now be calculated from

$$\alpha_l \phi_l = \frac{1}{v\Delta t} \int_{-v\Delta t}^{0} \sum_{j=0}^{N} a_j x^j \ dx \ . \tag{9}$$

Substituting in the a_j's and integrating gives

$$\alpha_l \phi_l = \sum_{j=0}^{N} \frac{-(-\alpha)^j}{j+1} \sum_{i=0}^{N} Q_{ij}^{-1} \phi_i \ . \tag{10}$$

Note that the general Δx is used in the α on the right hand side of eq 10, which may be different from α_l on a non-uniform grid.

This enables coefficients of α and the ϕ_i's to be calculated for all boxes and time and thus coefficients for ϕ_i to be calculated for all time at each grid box (under constant flow conditions). The coefficients of α depend on the grid size and so need only be calculated once for a regular grid, in which case the coefficients of ϕ depend only on α and need only be calculated once for uniform, constant flow. These coefficients are used to calculate the fluxes for the update equation which are used to approximate the tracer distribution at the next time step. This last step is repeated for each time step.

Unsurprisingly the results from the schemes generated in this way are the same as those found by Leonard [2] for the regular grid case. The main difference is that irregularly sized grid cells have been allowed in this computation due to the process being automated. This means more thought must be given to the choice of a suitable time step. A smaller time step will generally improve the accuracy but in this case appear to have a bigger improvement in the smaller cells than the larger ones. If a limiter is to be used then this will impose a maximum time step dependent on the size of the smallest cell.

3 Two-Dimensional Advection

In two dimensions, the advection equation for a fluid of density ρ and mixing ratio q can be written in two ways, either in the advective form

$$\frac{Dq}{Dt} = 0 \ , \tag{11}$$

where,

$$\frac{D}{Dt} \equiv \frac{\partial}{\partial t} + \vec{v} \cdot \nabla \ , \tag{12}$$

or in the flux form,

$$\frac{\partial}{\partial t}(\rho q) + \nabla \cdot (\rho \vec{v} q) = 0 \ . \tag{13}$$

The flux form is the conservation form of the equation and tells us that the total amount of the advected substance must be conserved under suitable boundary conditions. The advective form tells us that, following a fluid parcel, the mixing ratio does not change and thus that extrema are not amplified. A scheme with these desirable, physical properties as well as good accuracy, is now sought.

The first of these issues examined is good accuracy. One way of generating a scheme would be to apply the high order, one-dimensional schemes already discussed separately in each coordinate direction. This directional splitting approach often gives unsatisfactory results [4, 10], especially in strong deformation flows and will not work on unstructured grids, so a fully two-dimensional approach, extendable to higher dimensions, is sought here. The flux-limiter that is used adjusts only the fluxes across cell boundaries. Other limiting methods are largely 'built in' to the schemes from the beginning and form an integral part of those schemes and so would require simultaneous consideration.

The idea behind this method is the same as for the one-dimensional case, i.e. fit a polynomial of order N over a stencil, then integrate it over the region swept across the face in one time step. In this case, however, a two-dimensional polynomial must be fitted over K grid-boxes where K is given by;

$$K = \sum_{l=1}^{N+1} l = \frac{(N+2)(N+1)}{2} \ . \tag{14}$$

The first problem is to choose a suitable stencil over which to fit the polynomial. The fitting is again done by insisting that the integral of the polynomial over a box is equal to the total tracer amount in that box. This again leads to a matrix which requires inversion, but in this case the matrix will be singular if certain stencils are used and ill-conditioned for many more stencils. Looking at why these stencils give singular matrices suggest some rules for choosing a suitable stencil, although the final restrictions used to select a unique stencil could be viewed as intuitive. The first restriction is that boxes from at least $N+1$ different rows and $N+1$ different columns must be chosen. If this is not fulfilled then row operations on the matrix generated can reduce the problem to a one-dimensional case with fewer equations than unknowns. If more than $N+1$ boxes are used in any row or column then the matrix can again be reduced to a one-dimensional problem, this time with as many equations as unknowns though the equations are not independent.

With this restriction in mind we can now begin to choose a stencil by selecting $N+1$ grid boxes in each of the x and y directions, centred about the face in question or the cell upwind of it. These $2N+1$ boxes are enough to generate the coefficients of the terms x^m and y^n for $0 \leq n, m \leq N$, leaving only the cross terms, $x^m y^n$, where $n+m \leq N$. Without loss of generality, the origin can be defined as the centre of the cell upwind of the face in question, this allows the matrix being generated to be written in the reduced form;

$$B = \left(\begin{array}{c c|c|c} 1 & (\Delta x)^m & (\Delta y)^n & (\Delta x)^k(\Delta y)^l \\ \hline 0 & & & \\ \vdots & \emptyset & N \times N & \\ \vdots & & & \emptyset \\ \cline{2-3} \vdots & N \times N & \emptyset & \\ \vdots & & & \\ \hline \vdots & & & \\ \vdots & \multicolumn{2}{c|}{\emptyset} & K-2N-1 \times K-2N-1 \\ \vdots & & & \\ 0 & & & \end{array}\right) \begin{array}{l} ,\ i=j=0 \\ \\ ,\ i=0 \\ \\ ,\ j=0 \\ \\ \\ \\ \\ \\ \end{array}$$

Where elements are non-zero, the size of the block is given and the first row shows the order of the terms, $k, l, n, m = 1, 2, \ldots, Nk + l \leq N$.

Each of the terms in the lower right block of the matrix can be written as $xyx^iy^j\,(i, j = 0, 1, 2, .., N-1)$ so, as neither x nor y is equal to zero in this block, dividing through by xy shows that this block is similar to the original un-reduced matrix but for an order $N-1$ polynomial. Selecting another cross of N boxes in the x and y directions, not entirely coincident with the first, and

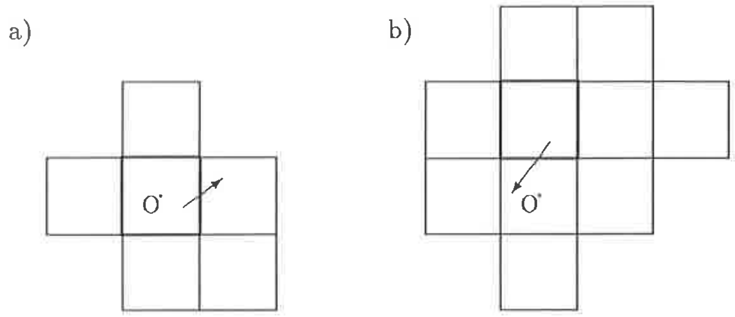


Figure 1: Stencils for the flow directions across the faces shown for a) Third order scheme and b) Fourth order scheme. Upwind boxes are in bold and O is the local origin (see text)

performing row operations to make the upper left block upper triangular as before, transforms the upper right block to a zero matrix. This allows the lower right block of this matrix to be decomposed in the same form as the full matrix. The cross of boxes should be upwinded about the face if the last was centred and downwinded if the last was upwinded. Repeating this process as many times as the matrix will allow yields a square stencil at 45° to the grid with extra boxes along one/three sides if the polynomial is even/odd order. The extra boxes are downwind of the flow across the face and upwind of the flow perpendicular to it for an even order polynomial, and also along the two sides perpendicular to this for an odd order polynomial. Figure 1 shows two examples of such stencils.

This method generates four stencils for flows from each quadrant for faces in each direction, a total of seven stencils since flow from the NW quadrant across a N-S face has the same stencil as flow from the SE quadrant across a E-W face. Three of these are the same shape as three of the others with the centre shifted by one cell due to changes in flow direction. There are other ways of selecting a stencil but they are more likely to lead to ill-conditioned and even singular matrices. One such way which leads to a singular matrix is to select boxes whose indices lie on a polynomial of order N or less, this leads to the loss of independence of the K equations generated.

This argument assumes powers of x and y are being used as basis functions. The conditioning of the matrix being generated shuld be improved if orthogonal basis functions were used. If Legendre polynomials in x and y are used instead of powers of x and y respectively then the structure of the matrix is different but it must still be non-singular as the same unique solution is being sought. In all of what follows, Legendre polynomials (written as $P_k = \sum_{s=0}^{k} p_{k\,s} x^s$) are used along with a LU decomposition method for finding the inverse matrix.

The seven matrices of polynomials associated with these seven stencils are non-singular and can now be inverted. The inversions may be done either by standard techniques or by a solver designed for matrices with the structure given by the method of selection of the stencil. The inverse matrix gives the coefficients of the interpolated polynomial in terms of the values of the tracer in the stencil, i.e. $\underline{a} = B^{-1}\underline{q}$. This allows the polynomial to be integrated over an approximation to the area swept across the face to give the flux across that face in one time-step. This can be done exactly for a general B allowing the equation for the flux;

$$FLUX_x = \frac{1}{u\Delta t\Delta x}\int_{\frac{\Delta x}{2}-u\Delta t}^{\frac{\Delta x}{2}}\int_{wx-\frac{w\Delta x}{2}-\frac{\Delta y}{2}}^{wx-\frac{w\Delta x}{2}+\frac{\Delta y}{2}}\sum_{r=0}^{N}\sum_{k=0}^{r} a_z P_{r-k}(x)P_k(y)dxdy \quad (15)$$

to be evaluated, where $w = \frac{u}{v}$, with a similar equation for $FLUX_y$. $FLUX_x$ and $FLUX_y$ are fluxes across boundaries perpendicular to the x and y directions respectively. By performing these integrations, the flux across a boundary can be found for a general q-field from;

$$FLUX_x = \sum_{g=1}^{K} q_g \sum_{r=0}^{N}\sum_{k=0}^{r} b_{z\ g}^{-1} \sum_{t=0}^{r-k} p_{r-k\ t} \sum_{s=0}^{k} p_{k\ s}\Delta x^{t+s} s! \sum_{l=0}^{\frac{s}{2}} \frac{(\frac{\Delta x}{\Delta y})^{2l} w^{s-2l}}{(2l+1)!}$$
$$\sum_{m=0}^{s-2l} \frac{(m+t)!}{m!(s-2l-m)!} \sum_{n=0}^{m+t} \frac{(\alpha_x)^n(-1)^{s+n+m}}{2^{t+s+n}(n+1)!(m+t-n)!} \quad (16)$$

where $\alpha_x = \frac{u\Delta t}{\Delta x}$ is the Courant number in the x-direction. This depends on the flow velocity and direction as well as the time step and can also be written as

$$FLUX_x = \sum_{g=1}^{K} q_g \sum_{r=0}^{N}\sum_{k=0}^{r} b_{z\ g}^{-1} \sum_{t=0}^{r-k} p_{r-k\ t}\Delta x^t \sum_{s=0}^{k} p_{k\ s}\Delta y^s s! \sum_{l=0}^{\frac{s}{2}} \frac{1}{(2l+1)!}$$
$$\sum_{m=0}^{s-2l} \frac{(m+t)!}{m!(s-2l-m)!} \sum_{n=0}^{m+t} \frac{(-1)^{s+n+m}(\alpha_x)^{n-(s-2l)}(\alpha_y)^{s-2l}}{2^{t+s+n}(n+1)!(m+t-n)!} \quad (17)$$

These equations hold for uniform Δx and a not necessarily equal but still uniform Δy. If different sized boxes are allowed, as in the one-dimensional case, then similar equations will hold if the matrix of polynomials remains invertible.

Equation 16 or 17 can be used to calculate the coefficients of the q_i in the stencil and then the flux across a face. When the fluxes have been calculated the q-field can be updated by;

$$q_i^{n+1} = q_i^n + \sum_{inflow} FLUX - \sum_{outflow} FLUX, \quad (18)$$

where fluxes are summed over the inflow and outflow faces. The flux across each boundary is both inflow into one box and outflow from another, ensuring that the total amount of q is conserved. In general, if the update is done now the result will probably contain oscillations and spurious negative values that we wish to avoid. For this reason, before this update is done the Universal

Limiter [2, 4, 10] is applied to constrain the fluxes and ensure these spurious effects do not occur when the q-distribution is updated.

The idea of the limiter is to prevent oscillations occurring by limiting the outflow fluxes subject to suitable constraint of the inflow fluxes. The bounds on the inflow fluxes are found from the q-distribution at the current time-step and are used to calculate suitable bounds for the flux across the outflow faces. There is always a value that will satisfy these conditions, i.e.the value in the box upwind of a face, so in the worst case the limiter acts like the first order upwind scheme at a face. In practice the limiter rarely imposes such severe conditions and often does not alter the flux given by the scheme. Once the limited fluxes are known they can be used in the update equation 18. The updated q-field is then used with the same coefficients as before to calculate new fluxes for each boundary. These are limited as before and then applied to eq 18. In the following test cases the full limiter together with the refinements of [10] is used.

One other consideration, not unique to this method, is how to store the velocity field. The three most obvious methods are similar to the A, C and E grids of Arakawa [1]. Firstly, the flow may be stored at the centre of each cell and averaged to each face when needed, which is equivalent to the A-grid. The second way is to store the flow in each direction at the centre of the faces perpendicular to the flow, in a similar way to the Arakawa C-grid. The final way considered is to store the flow in both directions at the centre of each face in a similar fashion to the E-grid of Arakawa. Each of these is a valid choice for storing the flow but after a few experiments and consideration of [1] the third/E-grid method was used. For the first two test cases used all three grids are equivalent but the E-grid gives a better approximation to the area swept across each boundary for the third case.

4 Test Cases

Three test cases were used to test the schemes produced by this method, uniform flow, rotational flow and a strong deformational flow. These three cases are a set which is frequently used, though not always together, in the literature to show the strengths and weaknesses of numerical advection schemes. Here all three have been used to show whether this method works well in a range of flows or just one. All the test cases were performed on a square grid in a periodic unit square domain. Error measurements were made along with a measure of the cost of computing the results, the CPU runtime. Generally, higher order schemes are more accurate, as are those run on more refined grids, and these results should show when a higher order scheme is more cost effective than increasing the resolution for this method. The cost of each part of the method was recorded though only the total runtime is used as the measure of cost. A discussion of this breakdown will follow the results.

These test cases give a measure of the success of the scheme which can be compared with other schemes run under the same conditions. The tests in themselves do not give a complete picture of how well the scheme would perform in the atmospheric problem. The uniform flow, section 4.1, has little resemblance to flows found in the real atmosphere but does provide an indication of how well the scheme handles flow at an angle to the grid. The rotational flow, section 4.2, has more in common with rotating systems in the atmosphere but does not stretch parcels in the same way due to it's constant angular momentum. The final test, section 4.3 deformational flow, does include vortices which stretch fluid parcels but these are regular and do not interact with one another as in the real atmosphere. Looking at the results of these tests together should give a good idea of the strengths and weaknesses of the scheme and where it will succeed or have problems in a realistic problem.

For completeness, several measurements were taken of errors and properties of the advected profile. The error measures used were the L_1, L_2 and L_∞ norms and first and second moments of mass, along with a breakdown of the L_2 norm into diffusive and dispersive components. The L_1 norm behaves in the same way as the L_2 norm in these cases and so is not shown, nor is the first moment of mass as this is always unity to machine accuracy because of the conservative nature of this method. Other measures taken were of the final maximum and minimum values of the distribution and of how well the scheme maintained the symmetry of solutions (where it should). The first of these measures is used primarily to check that the limiter is working and not allowing the maximum or minimum values to grow, as was the case in all runs. For symmetric initial conditions in a symmetric flow the solution should also be symmetric, and is to machine accuracy for these test cases. These measures can also give an idea of the amount of dissipation in the scheme but different error measures are used to investigate that here.

In all the cases that follow the errors for 9^{th} & 10^{th} order schemes on higher resolution grids are worse than those for most lower order schemes. This is probably because of the ill-conditioning of the matrix of polynomials which does not allow accurate inversion and subsequent evaluation of coefficients and fluxes. There are methods by which the conditioning of these matrices may be improved but this is not done here as the results for schemes lower than $9th$ order provide enough information. These results do however show the success of the limiter in not only preserving shape but also in keeping errors to a sensible level for a scheme with pseudo-random errors.

4.1 Uniform Flow

For this test case an initial profile was advected by a constant uniform flow at an angle of 45° to the grid. Three grids were used, each double the resolution of the previous one; 20×20 ($\Delta x = 0.05$), 40×40 ($\Delta x = 0.025$), 80×80 ($\Delta x = 0.0125$). The flow speed was 1 unit per second in each direction and a Courant number of 0.25 was used in each direction, which means that each doubling of the resolution requires twice the number of time steps to be used. The profile is advected once around the periodic domain in 1 second requiring 80, 160 or 320 time steps depending on the grid size. Flow at an angle to the grid is difficult to model accurately using directional splitting methods but a fully two-dimensional

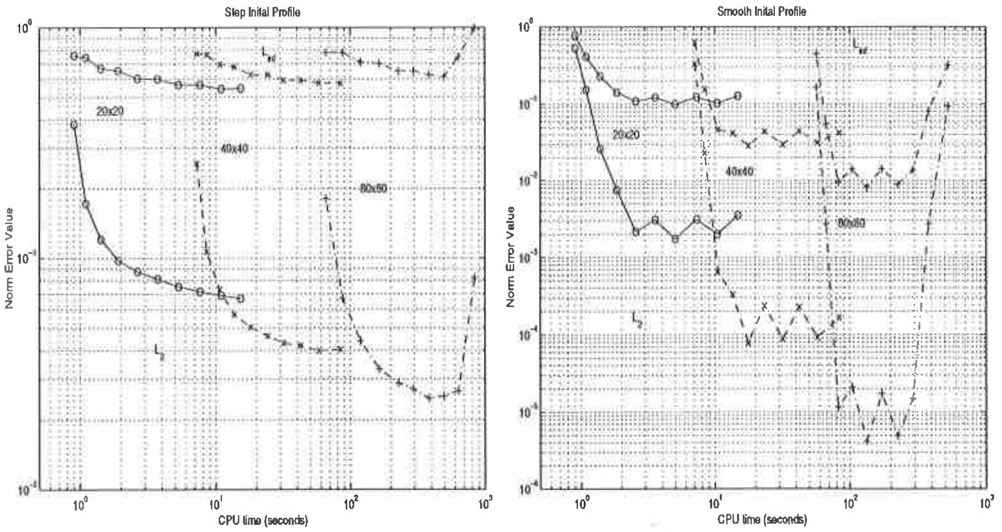


Figure 2: Error norms for step profile (left) and smooth profile (right) advected by uniform constant flow. Grid sizes are as shown.

method such as this should perform equally as well as for flow at any angle to the grid.

Two initial profiles were used for this test case, a square step and a smooth $\cos^2$ function. The square step was unity over $\frac{1}{4}$ of the domain and zero elsewhere giving an idea of how the schemes cope with sharp gradients. The 'smooth' function was also defined on the same $\frac{1}{4}$ of the domain and behaves as $\cos^2 x \cos^2 y$ with a continuous value, but discontinuous derivative at the boundary of this region. The main difficulty for any numerical scheme here being to avoid flattening the peak and maintaining it's value.

Figure 2 shows the cost against the L_2 and L_∞ norms for both profiles, where grid sizes are as shown and each order scheme is marked. The step profile shows continued reduction in these error norms as higher order schemes are used. In the L_2 norm this improvement is more cost effective than an increase in resolution up to about the fifth order. The L_∞ norm, or the maximum error in any cell, shows an increase with increasing resolution, probably due to a better capture of the smoothing of the gradient of the numerical solution which is effectively averaged out on a coarse grid. The smooth profile shows larger improvements for all these errors as the order of the scheme is increased up to the fifth order but little or no improvement as higher order schemes are used. In this case an increase in the resolution does lead to a decrease in the maximum error in one box. These results suggest that using a scheme higher than fifth order will only improve accuracy near sharp gradients whereas increasing resolution would be more cost effective as it would improve accuracy everywhere.

The L_2 error measure may by written in terms of diffusion and dispersion

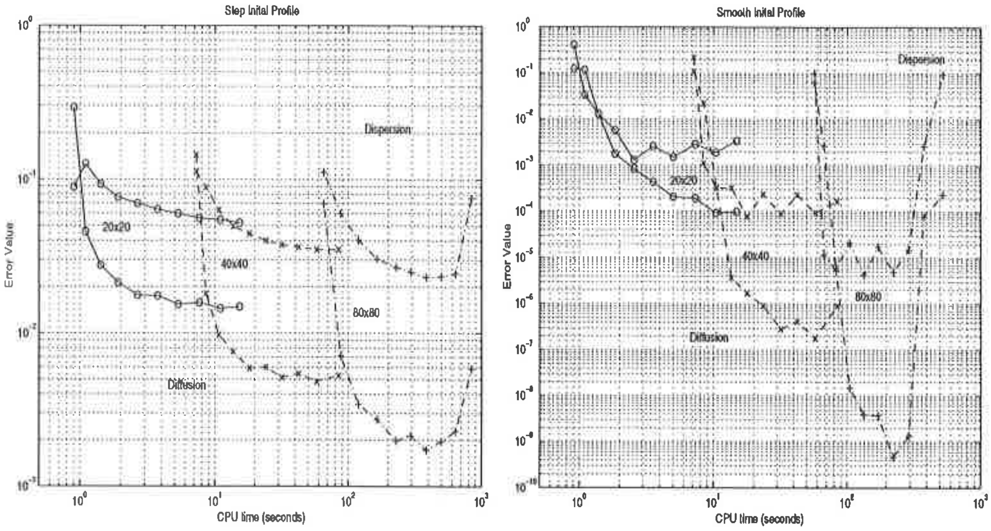


Figure 3: Comparison of Diffusion and Dispersion components of L_2 norm for step profile (left) and smooth profile (right) under uniform constant flow. Grid sizes are as shown.

errors, deriving from writing the L_2 error in terms of mean and variance,

$$L_2 = \frac{\sum(\phi_T - \phi_N)^2}{\sum \phi_T} = \frac{M}{\sum \phi_T} \left[\sigma^2(\phi_T - \phi_N) + (\bar{\phi}_T - \bar{\phi}_N)^2\right] \ , \tag{19}$$

where ϕ_T is the true solution and ϕ_N is the numerical solution. By expanding and simplifying this expression it can be shown that

$$L_2 = \frac{M}{\sum \phi_T} \left([\sigma(\phi_T) - \sigma(\phi_N)]^2 + (\bar{\phi}_T - \bar{\phi}_N)^2 + 2(1-\rho)\sigma(\phi_T)\sigma(\phi_N) \right) \ , \tag{20}$$

where ρ is a coefficient between zero and one, depending on the correlation between the numerical and exact results. Due to the conservative nature of this method, $\bar{\phi}_T - \bar{\phi}_N = 0$ and if there is no diffusion $\sigma(\phi_T) - \sigma(\phi_N) = 0$ also, giving the dispersion error as

$$E_{disp} = 2(1-\rho)\sigma(\phi_T)\sigma(\phi_N) \ . \tag{21}$$

So the diffusion error is

$$E_{diff} = [\sigma(\phi_T) - \sigma(\phi_N)]^2 \ . \tag{22}$$

Figure 3 shows these component parts for the L_2 norm of both profiles. For the step profile the diffusion error is an order of magnitude or two smaller than the dispersion error and both show similar behaviour, the main difference being the greater improvements in the diffusion error when going from even to odd

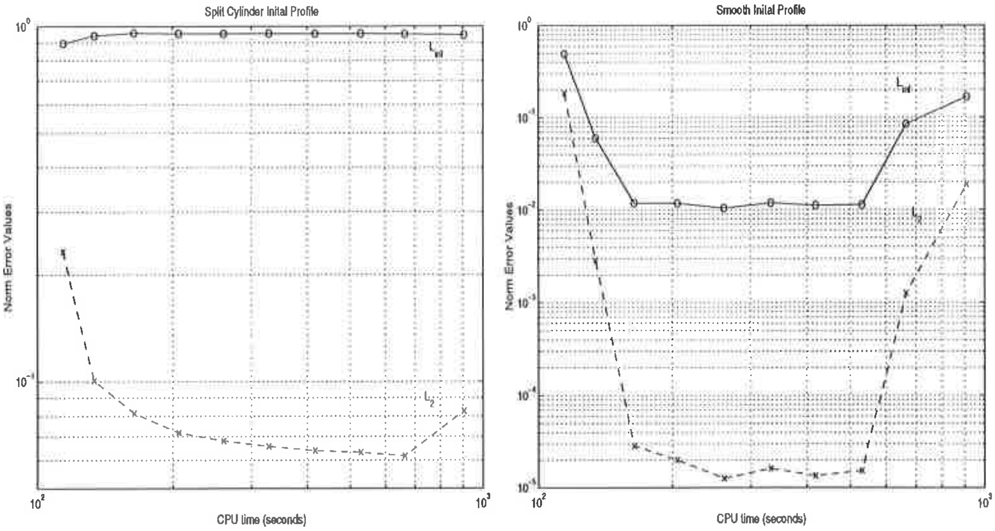


Figure 4: Error norms for the split cylinder (left) and smooth profile (right) for a constant rotational flow on an 80×80 grid

order schemes than from odd to even order schemes. The diffusion error for the smooth profile also shows this stepping and, unlike the dispersion error, continues to decrease beyond fifth order. The other remarkable feature is that the diffusion error is anywhere up to five orders of magnitude smaller than the dispersion error for this smooth case.

Schemes run on this test case without the limiter show the same trends in the error measures but highlight the distinct differences between even and odd order schemes. The stepping seen in the errors of the smooth profile for the limited scheme is present for both cases, seen in this case as oscillations about the general improving trend. The components of the L_2 norm clearly show the improved diffusion or dispersion of the odd and even order schemes.

4.2 Rotational Flow

For this test case a counterclockwise solid body rotation was applied to two initial profiles. The rotation speed was π times the distance from the centre of the rotation so the solution after two seconds was the initial distribution. A square grid of 80×80 boxes was used with a maximum Courant number of $\pi/8$ in each direction so one rotation took 640 time-steps. The first profile used was a split cylinder of radius $15\Delta x$ with a slit $6\Delta x$ wide removed, while the second profile was the same $\cos^2$ function used for the uniform flow case.

The norm error measures for the split cylinder, fig 4, show similar behaviour, as the order increases, to the step profile in a uniform flow, i.e. a continuing decrease. This is again due to the higher order schemes maintaining a steeper

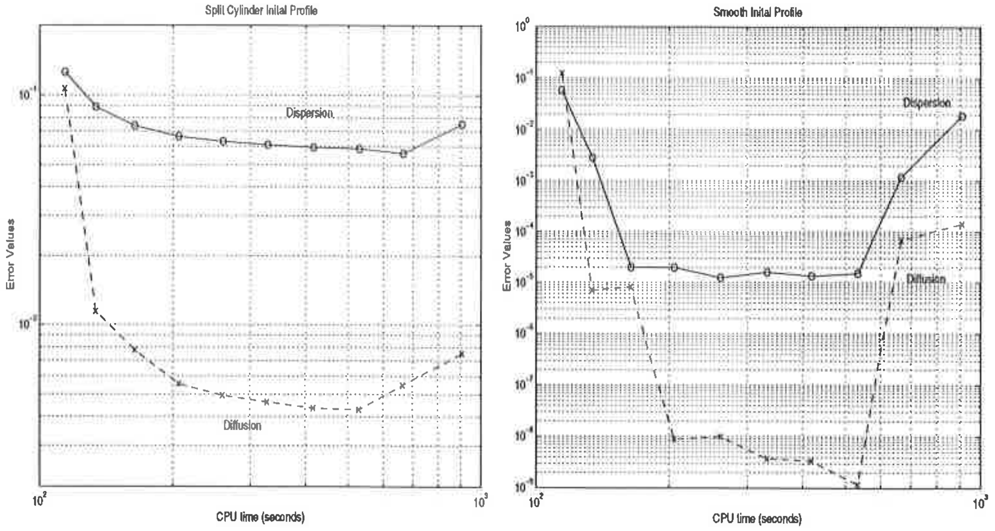


Figure 5: Comparison of Diffusion and Dispersion components of L_2 norm for the split cylinder (left) and smooth profile (right) for a constant rotational flow on an 80×80 grid

gradient, but with the L_∞ norm having a much shallower downward trend than before and increasing slightly between 1^{st} and 3^{rd} order. The same norms for the rotation of the smooth profile show slightly different behaviour, the errors being much smaller but not improving significantly beyond the third order or at all beyond fifth order. This is most likely due to the approximation of the area being swept across a face, which in the case of non-uniform flow does not coincide exactly with the area over which the scheme integrates the polynomial to approximate the flux.

The diffusion and dispersion components of the L_2 norm, shown in figure 5, also behave in a similar fashion to the uniform flow case. Starting as being roughly equal for the first order case, the diffusion error decreases more rapidly with increasing order and for high order schemes is at least an order of magnitude smaller than the dispersion error. For the smooth initial profile the diffusion error continues to decrease with increasing order while the behaviour of the L_2 norm is dominated by that of the dispersion error. Again the un-limited results show similar trends with the same differences between odd and even order schemes as before. An additional problem in this case is that the oscillations produced by the un-limited scheme cross the boundaries of the domain causing extra interferance.

Both the uniform and rotational flows show similar behaviour when considering the second moment of mass. A scheme which conserves the second moment of mass exactly will have a score of 1, anything below this value gives an indication of the diffusivity of the scheme. As the previous plots suggest,

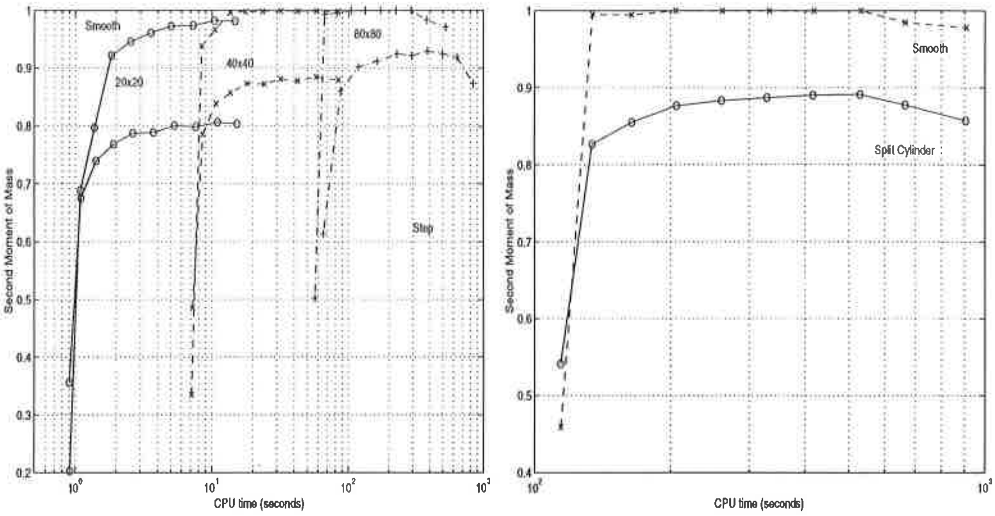


Figure 6: Second moments of mass for uniform flow (left) and rotational flow (right). Grid sizes are a s shown.

and figure 6 shows, the higher order schemes on more refined grids produce a score very close to one for advecting the smooth profile. The profiles with large gradients are not so well handled though with a few percent loss in the second moment at best.

4.3 Deformational Flow

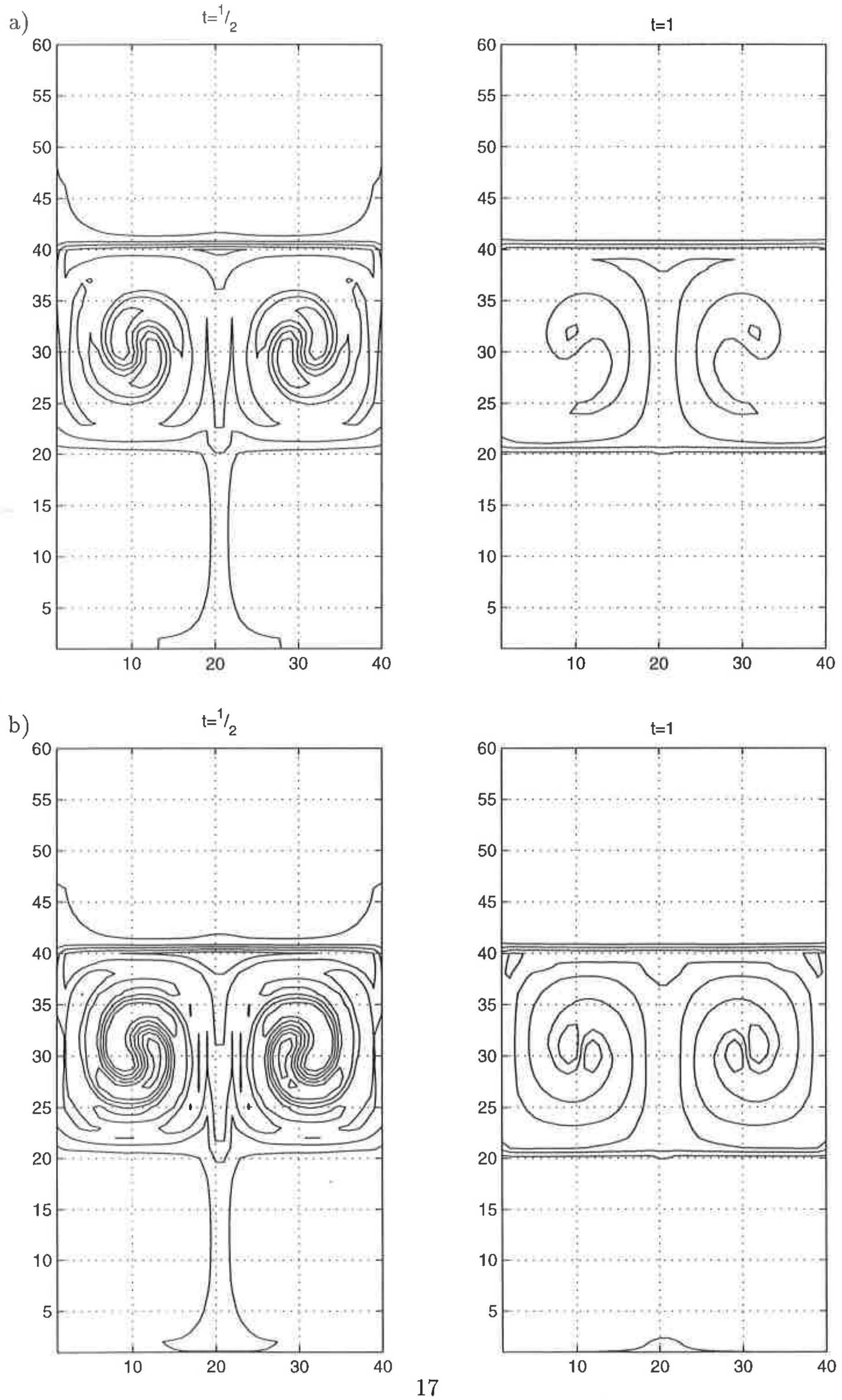
This test case is similar to that devised by Smolarkiewicz [7] in which an initial profile is advected by a number of rotating vortices. In this case an 80 × 80 grid is used on a domain containing eight pairs of vortices rotating in opposite directions, each in a 20 × 20 region. An initial tracer profile is centred on one pair of vortices with a small amount in the pairs above and below. Staniforth et al. [9] (see also [8]) presented a method for solving the problem exactly, providing insights as to how the true solution behaves and desirable properties of any numerical solution. Fluid elements must move along streamlines and so must remain in the vortex in which they start. Moreover, the solution must remain non-zero on the six vortices containing the initial tracer distribution and zero elsewhere. In the centres' of the four outer vortices the solution should also remain zero and the non-zero part is confined to a region close to the boundary. As the tracer field develops the tracer is stretched and becomes pulled out into filaments that become infinitesimally small in infinite time and thinner than any finite grid resolution in a finite time. In practice a numerical scheme would hope to resolve these filaments down to the grid scale where, due to conservation, they would decrease in magnitude. As the true solution cannot be discretised onto the grids used without aliasing it is not sensible to attempt to measure the errors as in the previous tests. The oscillations produced by the un-limited schemes would quickly destroy the filamentary structure being form and so these schemes are ignored here.

The first two plots in figure 7 show the results of a third order scheme on an 80×80 grid at two times. At $t = \frac{1}{2}$ the third order scheme barely shows the filament that should be formed by this time as it is already decaying. At $t = 1$ almost all of the filaments have decayed and spread to form an egg-cup shape typical of numerical solutions of this problem. The fifth order scheme in the next two plots shows how increasing the order has maintained steeper gradients and allowed less decay at $t = \frac{1}{2}$. New maxima and minima can occur due to a stretching flow and these are allowed by the limiter, for this case they should only occur on the boundary. The small oscillations on the central ridges here are caused by the discrete representation of the problem creating the same conditions as at the boundary of the 'real' problem. These should therefore become smaller as $\Delta x \rightarrow \infty$ and are less noticeable on the higher resolution plot. An improvement is also seen at $t = 1$ where the filamentary structure is still apparent but, due to decay of the filaments, is forming the often seen egg-cup shape. These improvements are not as good as those shown in the final two plots, showing a third order scheme on a grid with twice the resolution of the last two cases. Unsurprisingly much steeper gradients are maintained and the filamentary structure is still resolved in the centre of the vortex at $t = 1$. Small scales are being lost by this time at the edge of the vortices where the deformation is stronger. This case also shows the smaller amounts of tracer in the four outer boxes has been confined to the outer parts of the vortex, which is also the case to a lesser extent for the other two cases, but is not captured by the contours shown.

4.4 Cost

The implementation of this method can be viewed in three separate stages. The first is the generation of the seven stencils and polynomial matrices along with their inversion. For the relatively small matrices involved (55×55 for a 10^{th} order scheme) this is not a time consuming operation and need only be done once for each scheme. The second stage is to use the inverted matrices to generate the coefficients of the scheme, the time for which should increase with the fourth power of the scheme order and the square of the number of boxes in one direction. Assuming that the flow is different across each face this procedure takes a significant amount of time for high order schemes or those run on a fine grid but need only be done once if the flow is constant in time. The final stage is the running of the scheme along with the limiter, the time for which increases with the square of the order and the third power of the number of grid boxes. A run of a few hundred time steps will mean that this stage is the most expensive to run.

The cases run here confirm the costs of these sections to be proportional to the powers of the scheme order, O, and number of grid boxes, N, in one


a)
$t={}^1/_2$
t=1
b)
$t={}^1/_2$
t=1
60
55
50
45
40
35
30
25
20
15
10
5
10
20
30
40

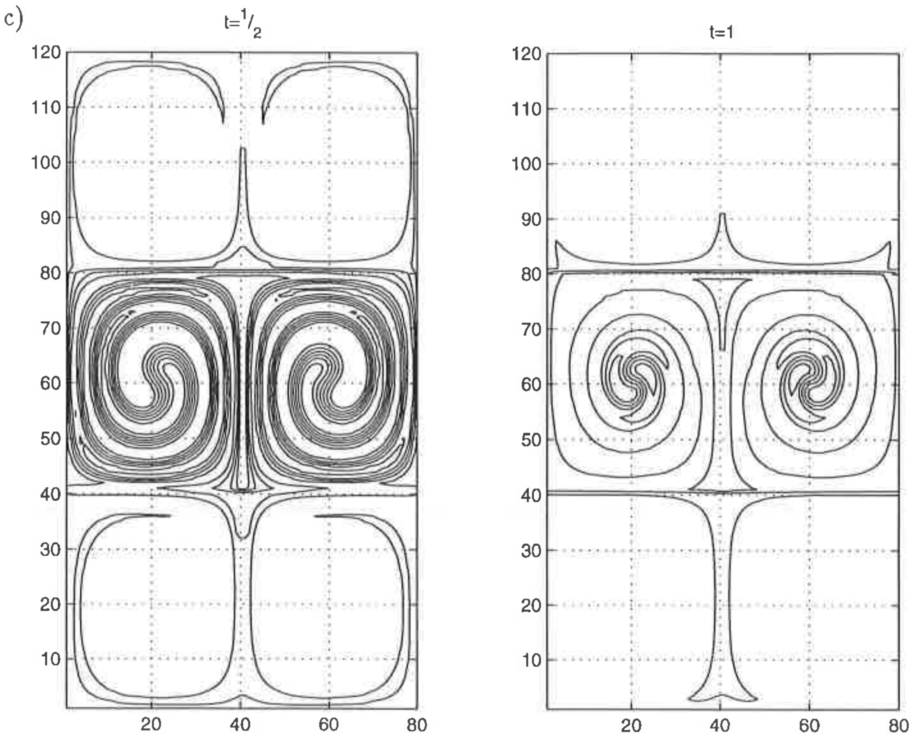


Figure 7: Numerical results of Smolakiewicz's deformational flow problem at (left) $t =^1 /_2$ and (right) $t =$ for a) 3^{rd} order scheme, 80×80 grid, b) 5^{th} order scheme, 80×80 grid and c) 3^{rd} order scheme, 160×160 grid, (only the nonzero regions are shown)

direction as would be expected from consideration of the method. Generating and inverting the seven matrices does not depend on the grid size and is an order O^4 expense. Calculating the coefficients must be done for each face and so is an order N^2O^4. The grid size will effect the Courant number so that more time steps are required for a more refined grid, which makes running the scheme an order N^3O^2 expense. The overall runtime is difficult to calculate but appears to increase with the order of $N^3O^{\frac{5}{2}}$ for higher order schemes.

5 Summary and Discussion

A general method for generating multidimensional advection schemes of any order on a regular rectangular grid has been developed and implemented in two dimensions. The resulting schemes have been run on a number of test problems with a flux limiter to prevent spurious oscillations and negative values being generated. Numerical results show a large improvement in accuracy for a relatively small cost by increasing the order of the scheme from 1^{st} to 3^{rd} order accurate and less of an improvement for a greater increase in cost between 3^{rd} and 5^{th} order schemes. For most cases, increasing the order of a scheme beyond 5 would be less cost effective (in terms of CPU time) than increasing the resolution of the grid on which the scheme is being run. One exception to this may be the climate model where increasing the resolution increases the cost not only of the advection scheme but many other parts of the model as well. It may also be that the errors in other parts of the model are not sensitive to small improvements in the advection scheme making the higher order schemes more cost-ineffective.

The results of these schemes compare well, in terms of accuracy, with many of the schemes in the literature, which is not surprising considering the lower order schemes generated in this way are finite volume versions of known schemes. The first order scheme can only be the diffusive First Order Upwind scheme and the second order scheme turns out to be a two-dimensional generalisation of the Lax-Wendroff scheme. The third order scheme is exactly the same as Leonard's UTOPIA scheme with higher order methods being based on the same methodology. The higher order methods also compare favourably with the results of the Piecewise Parabolic Method given in [5].

These results show that in this case, and quite possibly others, increasing the order of a scheme gives a larger improvement in the diffusion error than in the dispersive part. This would suggest that at some point simply increasing the order of a scheme will have little discernible effect on the error, which is also shown in these results. Increasing the resolution will cause an improvement in the dispersion error and a lower order scheme may be used for a saving in the cost with little extra error. Finding the exact optimum balance between cost and error for any given problem would be difficult and with the many other considerations involved, probably unnecessary. These results should give a rough guide to the optimum levels where this and perhaps other methods are used.

The current implementation of this method has not yet been fully optimised for constant flow conditions and is certainly unsuitable for variable flow problems. On a structured grid it is not too difficult to optimise the code for variable flow. Only speed and direction will vary the coefficients needed, so these can be stored when first generated and re-used when flow at another face or time are near enough identical. On unstructured grids the same idea may be implemented but with few if any identical faces storage of all coefficients may become a problem.

The results that have been presented strongly suggest that there is a level at which using a scheme of a higher order is not cost effective when compared to increasing the resolution of the grid. In practice the advection equation is rarely solved in isolation, particularly in the atmosphere, so the cost of increasing the resolution is greater than that here, making the judgment less straightforward. In this case there will also be other errors which may well outweigh those of the advection part of the equation so reducing the error in this part could be unnecessary. The cost of generating coefficients for such a large model would also impose a very large cost in the runtime which could be transferred to a storage cost in the same way as for an unstructured grid.

References

[1] K. Arakawa. *Methods in Computational Physics: Advances in Research and applications*, volume 17, chapter The UCLA Global Circulation Model, pages 174–264. 1977.

[2] B. Leonard. Universal limiter for transient interpolation modeling of the advective transport equations: The ULTIMATE conservative difference scheme. Technical memorandum, North American Space Agency, 1994. (unpublished).

[3] B. P. Leonard, M. K. MacVean, and A. P. Lock. The flux-integral method for multidimensional convection and diffusion. Technical report, North American Space Agency.

[4] B. P. Leonard, M. K. MacVean, and A. P. Lock. Positivity preserving numerical schemes for multidimensional advection. Technical Report 106055 ICOMP-93-05, North American Space Agency, 1993.

[5] J. Richard L. Carpenter, K. K. Droegemeier, P. R. Woodward, and C. E. Hane. Application of the piecewise parabolic method (PPM) to meteorological modeling. *Monthly Weather Review*, 118:586–612, Mar. 1990.

[6] R. B. Rood. Numerical advection algorythms. *Review of Geophysics*, 25:71–100, 1987.

[7] P. K. Smolarkiewicz. The multi-dimensional Crowley advection scheme. *Monthly Weather Review*, 113:1968–1983, 1982.

[8] P. K. Smolarkiewicz. Reply to comments on "Smolarkiewicz's deformational flow". Monthly Weather Review, 1987. pages 901-902.

[9] A. Staniforth, J. Côté, and J. Pudykiewicz. Comments on "Smolarkiewicz's deformational flow". Monthly Weather Review, 1987. pages 894-900.

[10] J. Thuburn. Multidimensional flux-limited advection schemes. *Journal of Computational Physics*, 123:74–83, 1996.

[11] J. Thuburn. TVD schemes, positive schemes, and the universal limiter. *Monthly Weather Review*, 125:1990–1993, 1997.