WATERCOMP '89 The First Australian Conference on Technical Computing in the Water Industry, Melbourne 30 May-1 June 1989, *pp* 185-189, Retyped in 2011.

Stable numerical methods for diffusive flows

J.D. Fenton

Professor, Fluid Mechanics, Department of Civil Engineering,
University of Auckland, New Zealand

SUMMARY: A numerical method is proposed for one-dimensional problems which involve the transport of substances (salt, sea water, radioactivity, heat, pollutants *etc.*) and/or the diffusion of those substances in river, estuarine, and groundwater problems. The scheme involves Fourier approximation, invoked by standard fast Fourier transform procedures. It is highly accurate in its time stepping, is unconditionally stable for all values of space and time steps and fluid flow parameters, and the time stepping is exact for constant flow velocities and diffusion coefficients. Very large time steps can be used.

1. Introduction

The advection-diffusion equation governs a number of problems of importance to the water industry. Typically, it is used to predict the amount or concentration of water of a different quality, the concentration of pollutants, whether salt, bacteria, oil, radioactivity or heat *etc.*. Flow situations where it arises include the movement of fresh and saline waters in rivers and estuaries, the motion of groundwater pollutants, or the movement of moisture in unsaturated soils. Also, it has been widely used as a model equation for the governing equations of fluid mechanics, as it includes effects due to advection by a local fluid velocity, and diffusion due to molecular or turbulent processes.

In its simplest form, in one space variable x, the equation can be written

$$\frac{\partial \phi}{\partial t} = \kappa \frac{\partial^2 \phi}{\partial x^2} - u \frac{\partial \phi}{\partial x} \tag{1.1}$$

The equation governs the dependence of concentration or temperature, denoted by ϕ, on position x and time t, determined by the local fluid velocity u and the diffusion coefficient κ.

Numerical solution of the equation has always been beset by difficulties. All of the main methods used in fluid mechanics have been applied, including finite differences, finite elements, the method of characteristics, spectral methods, and combinations of these methods. Sometimes the numerical methods fail spectacularly, having very demanding stability criteria, or exhibiting undesirable phenomena such as numerical diffusion or dispersion. Finite difference schemes can usually be simply programmed, but they exhibit the numerical disadvantages more than others. In recent years, finite element schemes have found favour, however they seem to be rather complicated to implement, often involve matrix solutions at each time step, and like the finite difference schemes, no outstanding method seems to exist. What is perhaps least desirable of all is the arbitrariness of choosing a method from those available. Surveys of finite difference schemes may be found in Roache (1976), while Fletcher (1984) gives an extensive discussion of finite element methods.

In this paper a method is presented which attempts to overcome some of the disadvantages of previous methods. The scheme proposed seems to have some advantages: it is capable of very high accuracy, and is exact for equations with constant coefficients (*i.e.* constant velocity and diffusivity), it is non-dispersive, non-diffusive, and unconditionally stable. The computational cost is rather less than that of finite element methods. In particular, the goal of simplicity of coding is sought. Perhaps

the most desirable feature of the method proposed is that it is very simply implemented, most of the operations being able to be performed by one-line calls to standard fast Fourier routines.

2. Development of method

A typical problem involving the advection-diffusion equation (1.1) is where ϕ is known for all x or a finite number of point values of x in some region $0 \leqslant x \leqslant L$ at a certain time t, and it is desired to obtain the same knowledge of ϕ at later times, usually by stepping forward in discrete time steps ϕ. The solution at the later time $t + \Delta$ can be written in terms of the shift operator:

$$\phi\left(x, t+\Delta\right) = e^{\Delta \frac{\partial}{\partial t}} \phi\left(x, t\right), \tag{2.1}$$

for the moment ignoring the effects of boundary conditions. In practice, the shift operator is treated only as its Taylor expansion. Fenton (1983) showed that it is consistent to replace the time differentiation in the exponent with the spatial differential operator, as given by the partial differential equation (1.1), to give the solution scheme

$$\phi\left(x, t+\Delta\right) = e^{\Delta\left(\kappa \frac{\partial^2}{\partial x^2} - u \frac{\partial}{\partial x}\right)} \phi\left(x, t\right) + O\left(\Delta^2\right), \tag{2.2}$$

The neglected terms $O\left(\Delta^2\right)$ contain terms with derivatives of the flow and medium properties κ and u. In the case where κ and u are constant, the neglected terms of second and higher order disappear, and (2.2) is exact.

Within the order of approximation here, the exponential operator can be split, and the scheme (2.2) written in the form

$$\phi\left(x, t+\Delta\right) = e^{\Delta\kappa \frac{\partial^2}{\partial x^2}} e^{-\Delta u \frac{\partial}{\partial x}} \phi\left(x, t\right) + O\left(\Delta^2\right), \tag{2.3}$$

One of these operators is familiar from elementary numerical analysis. The quantity $\exp(-u\Delta\partial/\partial x)$ is the shift operator $E(-u\Delta)$, such that if $f(x)$ is some function of x, then

$$e^{-\Delta u \frac{\partial}{\partial x}} f(x) = f(x - u\Delta), \tag{2.4}$$

which simply has the action of shifting the argument of the function from x to $x - u\Delta$. Combining equations (2.3) and (2.4) we obtain

$$\phi\left(x, t+\Delta\right) = e^{\Delta\kappa \frac{\partial^2}{\partial x^2}} \phi\left(x - u\left(x, t\right)\Delta, t\right) + O\left(\Delta^2\right) \tag{2.5}$$

which can be interpreted as "interpolate to find the value of ϕ upstream at $(x - u\Delta, t)$ as if only the advective term existed, and then allow the solution to diffuse, as if only the diffusive term were included". The interpolation shows the characteristic nature of the scheme.

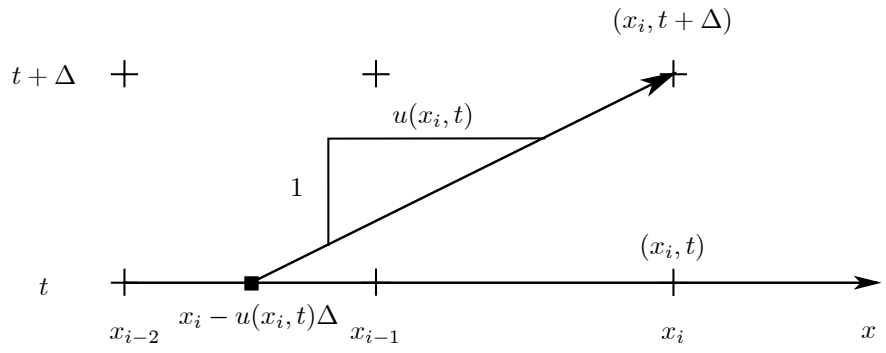

Figure 1. The "quasi-characteristic" nature of the scheme

Figure 1 shows how information proceeds on the (x, t) plane. The advection step consists of taking the value of ϕ at the point $(x_i - \Delta u(x_i, t), t)$, marked by the square on the figure, and transferring its value to $(x_i, t + \Delta)$ at the arrowhead. Having transferred all such point values, each is then modified by the process of diffusion over the travel time Δ. The straight line joining the points, of gradient $1/uu(x_i, t)$ will be termed a *quasi-characteristic.*

Another interpretation is that the method uses full upwinding, in that it uses an actual upstream value of ϕ, even if the actual point at which it samples, given by the quasi-characteristic, is not exact. This can be compared with many previous schemes, which have used low-order approximations to this operation of upstream sampling, often without the physical insight that the full scheme provides. It has been shown by Fenton (1983) that those approximations all have finite accuracy and finite stability criteria. They often require complicated finite difference expressions and program logic to handle changes in sign of the advective velocity. This is in comparison with the marked simplicity (and unconditional stability to be shown below) of the upwind interpolation $\phi(x - u\Delta, t)$. It will be shown in Section 5 how this procedure leads to simple and consistent ways of incorporating boundary conditions.

An unusual feature of the time-stepping scheme generated above is that it separates the procedures of spatial approximation and time stepping, unlike most finite difference schemes. With the scheme as developed above, one is free to use any reasonable method of spatial interpolation. However, it will be seen that Fourier approximation is very much the natural way of implementing the scheme.

3. Fourier approximation

Consider the finite Fourier series representation of ϕ as a function of x at some instant t:

$$\phi(x, t) = \sum_{j=-N/2}^{N/2}{}'' \, \Phi_j(t) \, e^{\mathrm{i}jkx}, \tag{3.1}$$

where $k = 2\pi/L$, and L is the length of the computational domain. The summation over j includes factors of $1/2$ at $j = \pm N/2$, indicated by the double prime superscript. The Fourier coefficients $\Phi_j(t)$ can be easily obtained by a discrete Fourier transform of the N point values $\phi_m(t) = \phi(x_m, t) = \phi(mL/N, t)$ for $m = 0, 1, \ldots, N-1$:

$$\Phi_j(t) = \frac{1}{N} \sum_{m=0}^{N-1}{}'' \, \phi_m(t) \, e^{-\mathrm{i}2\pi mj/N}. \tag{3.2}$$

It can be shown that, with the $\Phi_j(t)$ as given by (3.2), that the values of $\phi(x, t)$ given by (3.1) at values of $kx = kx_m = (2\pi/L)(mL/N) = 2\pi m/N$, are indeed $\phi_m(t)$. That is, the Fourier series (3.1) interpolates the point values $\phi_m(t)$, $m = 0, 1, 2, \ldots, N-1$. It is simple to introduce the notation $\mathbf{D}(..., j)$ for the discrete Fourier transform of the sequence of point values ϕ_m into the sequence of Fourier coefficients Φ_j.. Thus equation (3.2) can be written

$$\Phi_j(t) = \mathbf{D}(\phi_m(t), j). \tag{3.3}$$

This operation can be performed using fast Fourier transform techniques, for which software is widely available, see for example the FFT routine given in Conte and de Boor (1980) or Press *et al.* (1986). The computational effort is $O(N \log_2 N)$, compared with $O\,(N^2)$ of direct evaluation.

Now, implementing the scheme (2.5) for the Fourier representation (3.1) and interchanging the

orders of summation and differentiation:

$$\phi(x,t+\Delta) = \sum_{j=-N/2}^{N/2}{}'' \Phi_j(t) e^{\Delta\kappa\frac{\partial^2}{\partial x^2}} e^{\mathrm{i}jk(x-u\Delta)} + O\left(\Delta^2\right). \tag{3.4}$$

The effect of the diffusion operator on the function $\exp(\mathrm{i}jkx)$ can be shown by expanding the operator, performing the differentiations, and rewriting the easily-recognisable series so obtained, to give

$$\phi(x,t+\Delta) = \sum_{j=-N/2}^{N/2}{}'' \Phi_j(t) e^{-\Delta\kappa j^2k^2} e^{\mathrm{i}jk(x-u\Delta)} + O\left(\Delta^2\right). \tag{3.5}$$

In this form, the physical significance of the present method becomes clearer, and it can be seen that the numerical solution mimics the actual physical process. It can be seen that the combined effect of advection and diffusion is (i) to shift the solution along by an amount $u\Delta$ (the phase in the imaginary part of the exponent has been changed from x to $x-u\Delta$), precisely the expected behaviour of solutions of the equation, and (ii) to reduce the amplitude of the jth Fourier component by a factor $\exp(-\Delta\kappa j^2k^2)$, a result familiar from Fourier's method for solving the heat equation; its use as a numerical tool seems novel.

The scheme represented by (3.5) is straightforward to implement; transform the N discrete values $\phi(x_m,t)$ according to equation (3.2) using a fast Fourier transform, then evaluate the inverse transform represented by (3.5), written here for the point m :

$$\phi(x_m,t+\Delta) = \sum_{j=-N/2}^{N/2}{}'' \Phi_j(t) e^{-\Delta\kappa_m(t)j^2k^2} e^{\mathrm{i}jk(x_m-u_m(t)\Delta)} + O\left(\Delta^2\right), \tag{3.6}$$

for each of the interior points m on which no boundary condition is to be imposed. In this equation the possible variation of κ and u with x and t has been made more explicit: $\kappa_m(t)=\kappa(x_m,t)$ and $u_m(t)=u(x_m,t)$.

In the commonly-encountered situation where κ and u are independent of x, then fast Fourier methods can be used to evaluate (3.6). That is, where $u_m(t)=u$ and $\kappa_m=\kappa$ for all m, equation (3.6) can be written

$$\phi(x_m,t+\Delta) = \mathbf{D}^{-1}\left(\Phi_j(t)e^{-\Delta\kappa(t)j^2k^2}, m\right), \tag{3.7}$$

where $\mathbf{D}^{-1}(\ldots,m)$ is the inverse discrete Fourier transform, which can be performed using methods for the transform itself, see Conte and de Boor (1980) or Press *et al.* (1986).

For equations with coefficients which are functions of x (*i.e.* m), the series in equation (3.6) must be evaluated directly as presented, which involves the evaluation of the N-term series at each of N points, with a total number of operations of order N^2. This might be a major disadvantage if large numbers of points are required, although in many situations the cost of an order N^2 computational scheme might be quite acceptable. Finite element methods usually require a matrix solution at each time step, with a computational cost $O\left(N^3\right)$.

In many physical problems, the diffusion coefficient κ is constant, while the velocity u is a function of x. It is possible to introduce a hybrid scheme such that the operational count is about the same, but where the very high accuracy possible with Fourier interpolation is lost. This may not be a problem, as if the advective velocity is a function of x the fundamental scheme is not exact anyway, hence it is in keeping with the finite accuracy of the numerical scheme to use piecewise polynomial approximation to perform the interpolation. Such a hybrid scheme would be to use piece-wise polynomial approximation, possibly cubic splines or even piecewise linear approximation. for the advection step.

There is, however, one feature of the Fourier interpolation which *can* render it almost useless for computational purposes. This is where the values of ϕ at the ends of the computational domain are different, which is the usual case. The computational domain in x is $[0, L]$, however, the implication of the Fourier representation is that the function is continued periodically outside $[0, L]$. This will, in general, give discontinuities at 0 and L, and might prove catastrophic. The Fourier series is required to describe a function discontinuous at $x = 0$ and $x = L$. In this case, the Fourier coefficients decay like $1/j$ (for small j),the series is slowly convergent, and the accuracy of approximation is poor. This leads to the well-known Gibbs' phenomenon. Figure 2(a) shows the nature of the Fourier approximation, when, for example, a simple cubic $y = x^3$, shown by the dashed line, is continued periodically by the Fourier series, shown by the solid line.

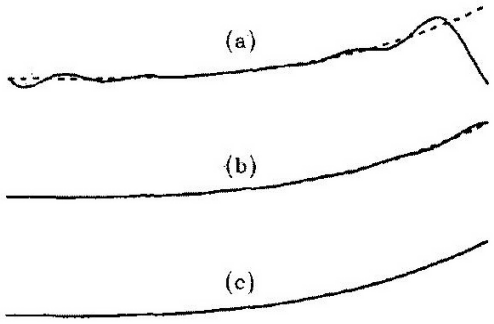

Figure 2. Approximation of a cubic function (shown dashed) for (a) Fourier approximation, (b) linear subtraction, and (c) linear subtraction and sine approximation.

There is a simple artifice by which a more accurate Fourier representation can be constructed for functions discontinuous across the ends. Firstly subtract a linear function from the $\phi(x, t)$,which in the discrete computational implementation is to subtract a uniformly-changing sequence from the ϕ_m, such that the resulting sequence, denoted by ψ_m, has zero values at $m = 0$ and $m = N$, at $x = 0$ and $x = L$ respectively:

$$\psi_m = \phi_m - \phi_0 - \frac{x_m}{L}\left(\phi_N - \phi_0\right), \tag{3.8}$$

for all m.The N-term Fourier series of this would have coefficients of $O\left(j^{-2}\right)$ as there would be gradient discontinuities at the ends, because of the implied periodicity. Such an approximation is shown on Figure 2(b). It is interesting that although the approximation looks accurate on the scale of this figure, the curvature of the Fourier approximation at the right end is very different from the function it is approximating. Computational results based on the suggested scheme and this approximation were poor.

However, a similar artifice yields high accuracy. Instead of the full Fourier approximation, a sine series approximation is used. In this case, the sequence ψ_m to be approximated has zero values at the ends, and continued outside the domain $[0, L]$ in a sine series sense, the interpolating function has, at worst, to describe a function which has discontinuities in the second derivative at the ends. Figure 2(c) shows the nature of the approximation in this case. The approximation is excellent. The coefficients in the series are at worst $O\left(j^{-3}\right)$, and give a considerably more accurate representation for a given number of terms.

A simple way of implementing the sine transform scheme is to supplement the $N + 1$ values with another N values continued for $-m$, such that the resulting sequence is an odd function of m:

$$\psi_{-m} = -\psi_m \quad \text{for } m = 1, ..., N - 1, \tag{3.9}$$

then taking a $2N$ term Fourier transform of these points, replacing N by $2N$ in (3.2), and using $2L$ as the period for the interpolating series. This more than doubles the computational effort, however

the simplicity of coding might be considered to outweigh this disadvantage. This was the method used used to obtain the results described below.

4. Stability and accuracy

The Fourier scheme described above can be simply used to examine the stability of the numerical schemes provided in this work. The case considered is where κ and u are constant, and where the effect of boundary conditions on stability is not considered. Now, (3.5) can be rewritten with the left side also as a Fourier series in x. Considering just one term of the series gives the difference equation in Δ :

$$\Phi_j(t+\Delta) = \Phi_j(t)\, e^{-\Delta\kappa j^2 k^2} e^{-\mathrm{i}jku\Delta}. \tag{4.1}$$

The solution of this difference equation for $\Phi_j(t)$ is

$$\Phi_j(t) = \Phi_j(0)\, e^{-\kappa j^2 k^2 t} e^{-\mathrm{i}jkut}. \tag{4.2}$$

Thus the numerical solution mimics the expected behaviour of the physical solution and analytical solutions: the phase of the Fourier coefficient changes by $-jkut$ in time t, corresponding to a uniform translation with speed u, while the magnitude of this Fourier coefficient is damped by a factor $\exp(-\kappa j^2 k^2 t)$. This is the unique solution of (4.1), there are no parasitic solutions which might grow with time, hence the scheme is unconditionally stable for all values of u and κ, an unusual and highly desirable situation.

Furthermore, the solution for ϕ from equation (4.2) is

$$\phi(x, t+\Delta) = \sum_{j=-N/2}^{N/2\,\prime\prime} \Phi_j(0)\, e^{-\kappa j^2 k^2 t}\, e^{\mathrm{i}jk(x-ut)}, \tag{4.3}$$

can be easily shown to be an exact solution of the advection-diffusion equation with constant coefficients!

These features of the Fourier method combined with the time-stepping scheme proposed in Section 2 seem to suggest the Fourier method as the most natural of all methods for the application of the scheme: it can solve equations with constant coefficients exactly, and is unconditionally stable.

5. Boundary conditions

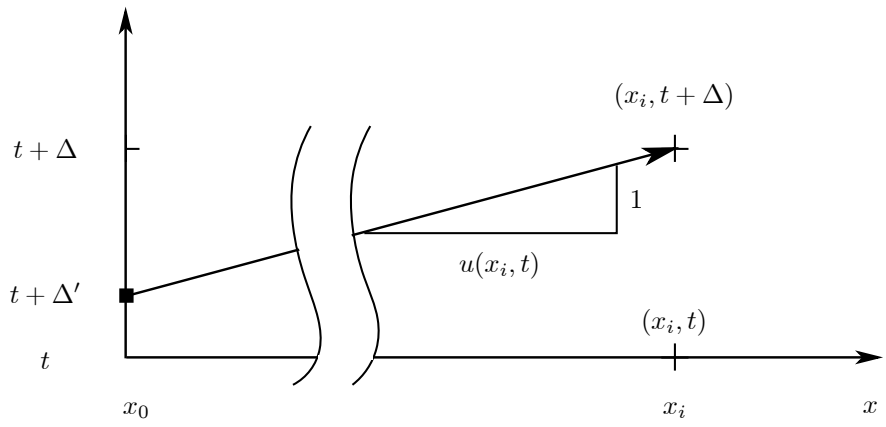

Figure 3. Treatment of boundaries

The basic scheme (2.5) cannot be implemented at those points x where $x - u\Delta$ lies outside $[0, L]$, as can be seen on Figure 3, representing the left end of the computational domain, on the (x, t) plane. In this case the quasi-characteristic intersects the boundary at $(x_0, t + \Delta')$, where Δ' is given by simple geometry from Figure 3:

$$\Delta - \Delta' = \frac{x_i - x_0}{u(x_i, t)}, \tag{5.1}$$

where i is the point considered, u is positive, and the quasi-characteristic emerges from the left boundary. The programming for a reversal of flow is trivial, which is fortunate for the case of simulation of tidal flows in estuaries where this is a common occurrence.

The problem is now to provide a scheme for the situation where the point is dominated by conditions at the boundary. This is provided by an interpretation of the basic scheme (2.5), which was essentially "the value at $(x, t + \Delta)$ is that which was previously at $(x - u\Delta, t)$, modified by the effects of diffusion over the time of travel Δ". This suggests the heuristic modification to (2.5) when the quasi-characteristic at (x_i, t) emerges from the boundary:

$$\phi\left(x_i, t + \Delta\right) = e^{(\Delta - \Delta')\kappa \frac{\partial^2}{\partial x^2}} \phi\left(x_0, t + \Delta'\right) + O\left(\Delta^2\right) \tag{5.2}$$

where Δ' is given by (5.1), and the travel time of information along the characteristic is $\Delta - \Delta'$. It can be shown by using Taylor expansions in Δ that this scheme is consistent with the original partial differential equation.

Usually, boundary information is provided in the form of

$$\phi\left(x_0, t\right) = F_0(t), \tag{5.3}$$

where the form of $F_0(t)$ is given, whether as a supplied function of time or as a sequence of discrete values between which interpolation in time can be used.

At points where boundary information has to be used, the diffusion term in the Fourier series is different for each point considered ($\Delta - \Delta'$ is a function of x, and is different for each point). This does not lead to much more computation, for the series has to be evaluated at only those points which are close enough to the boundary that the methods of this section have to be used.

6. Results

Here the performance of the method is examined for three problems for which analytical solutions are known. The flow conditions used were the same in all three cases. In the x direction 64 computational points were used. The curves plotted correspond to the actual computational steps. It will be seen that the time steps used are very large indeed, sufficient to advect the solution $1/4$ of the way across the computational domain in each step. In fact, rather larger time steps can be used, the value of $1/4$ being chosen so as to show the progress of the solutions. In linear problems, such as those shown here, a single time step could be used for solution. The details of the computational parameters are:

$$\text{Courant Number} = \frac{u\Delta}{\delta} = 16.0,$$

$$\text{Cell Péclet Number} = \frac{u\delta}{\kappa} = 3.125,$$

in which δ is the spacing in x. The Courant number expresses the number of computational points the solution traverses in one time step. In many numerical schemes which are not characteristic-based, this is limited to 1 to ensure stability. The cell Péclet number expresses the relative importance of advection to diffusion in the computational scheme. In typical finite difference schemes

which do not use upwinding this is limited to 2. In the case of the present method, it has been shown above that there are no stability restrictions on the time step, and the method works just as well for zero diffusion as for the cases given below.

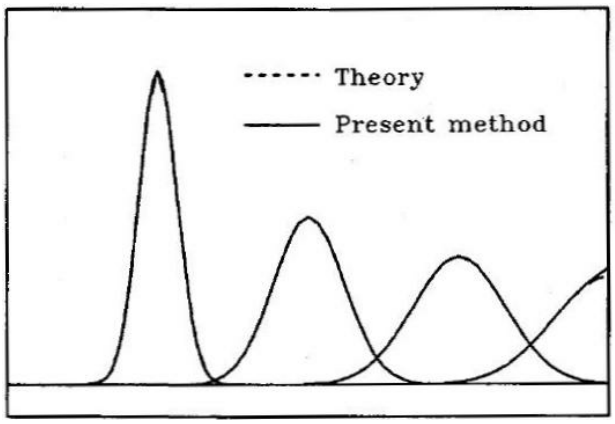

Figure 4. Propagation and diffusion of a single pulse

The first problem solved is that of the instantaneous introduction of a finite mass of material into a uniform flow. The initial conditions are. for an infinitely-large concentration over an infinitely-small region, which cannot be handled by a computational grid. Instead, computations began a short time after the introduction of the mass, as was done by Sobey (1983). Results are given in Figure 4.

It can be seen that almost everywhere the computational and theoretical results coincide on the figure, (they agreed to within several decimal places). At the right end of the domain, however, as the high-curvature crest of the concentration encounters the boundary the method does lose accuracy. This might be a problem in estuarine flow problems where the velocity can reverse, but in unidirectional problems the inaccuracies here are swept out of the flow domain.

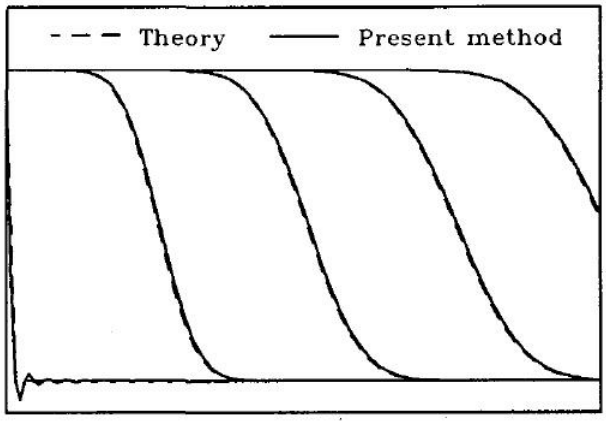

Figure 5. Propagation and diffusion of a step discontinuity

The next problem is that where the concentration initially is everywhere zero, and is then instantaneously raised to a constant value at one end. This is the problem first solved by Fourier for the case of no advection. Results are shown in Figure 5, and are compared with the theoretical solution for an infinite domain given on page 388 of Carslaw and Jaeger (1959). Initially it can be seen that the Fourier series yields the familiar Gibbs' phenomenon of finite oscillations in the vicinity of the discontinuity. The diffusion quickly eliminates these oscillations, and thereafter it can be seen that the computational solution reproduces the theoretical one very accurately. There is no numerical diffusion, and the speed of propagation is precisely that of the analytical solution. The only difference is that the computational solution leads by a small constant amount, proportional to the grid

spacing. This arises because at the initial instant, the theoretical solution has a step discontinuity of infinite gradient, whereas the computational solution based on a finite representation has to assume that the variation from the boundary value to the zero internal value occurs over one space step. At this instant then, the computational solution leads the exact solution by something like half a space step, and this is maintained throughout the computations.

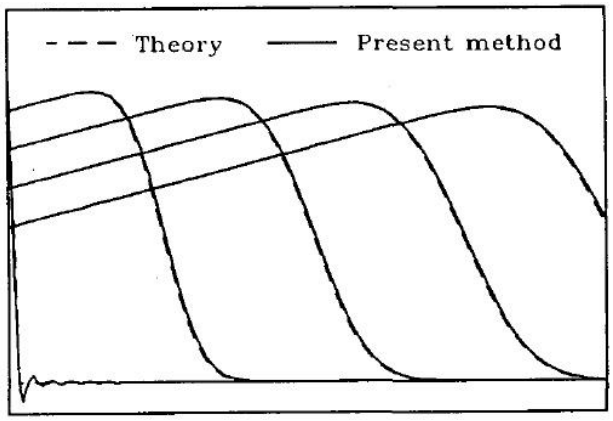

Figure 6. Propagation and diffusion of a step discontinuity

The final problem used for comparison is initially the same as that of Figure 5, but that the boundary value then reduces at a uniform rate. This is something of a test of the boundary condition treatment described in Section 5. Results are shown in Figure 6. It can be seen that the problem is solved with a similar level of accuracy as that of Figure 5, and the boundary treatment seems to be as accurate as the internal computations. In the general case where the variation of the boundary condition is not uniform, the method would not be as accurate for such large time steps. In such cases it would be necessary to take time steps such that the solution did not propagate more than, say, a single space step in a single time step, as with conventional schemes.

6. References

Carslaw, H.S. and Jaeger, J.C. (1959) *Conduction of Heat in Solids, O.U.P.:* Oxford.

Conte, S.D., and de Boor, C. (1980) *Elementary Numerical Analysis, (Third edn),* McGraw-Hill Kogakusha. Tokyo.

Fenton, J.D. (1983) A Taylor series method for numerical fluid mechanics, *Proc. 8th Australasian Fluid Mech. Conf. Newcastle.,* 1C13-1C16.

Fletcher, C.A.J. (1984) *Computational Galerkin Methods,* Springer: New York.

Press, W.H., Flannery, B.P., Teukolsky, S.A., and Vetter-ling, W.T. (1986) *Numerical Recipes, - C.U.P.:* Cambridge.

Roache, P.J. (1976) *Computational Fluid Dynamics,* Hermosa: Albuquerque.

Sobey, R.J. (1983) Fractional step algorithm for estuarine mass transport, *Int. J. Numer. Meth. Fluids* 3, 567-581.