INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2008; **56**:469–484
Published online 6 June 2007 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.1536

A simple high-resolution advection scheme

C. G. Mingham[1,*,†] and D. M. Causon[2]

[1]*Centre for Mathematical Modelling and Flow Analysis, Manchester Metropolitan University, Chester Street, Manchester M1 5GD, U.K.*
[2]*Department of Computing and Mathematics, Manchester Metropolitan University, Chester Street, Manchester M1 5GD, U.K.*

SUMMARY

A simple, robust, mass-conserving numerical scheme for solving the linear advection equation is described. The scheme can estimate peak solution values accurately even in regions where spatial gradients are high. Such situations present a severe challenge to classical numerical algorithms. Attention is restricted to the case of pure advection in one and two dimensions since this is where past numerical problems have arisen. The authors' scheme is of the Godunov type and is second-order in space and time. The required cell interface fluxes are obtained by MUSCL interpolation and the exact solution of a degenerate Riemann problem. Second-order accuracy in time is achieved *via* a Runge–Kutta predictor–corrector sequence. The scheme is explicit and expressed in finite volume form for ease of implementation on a boundary-conforming grid. Benchmark test problems in one and two dimensions are used to illustrate the high-spatial accuracy of the method and its applicability to non-uniform grids. Copyright © 2007 John Wiley & Sons, Ltd.

Received 5 June 2006; Revised 18 April 2007; Accepted 19 April 2007

KEY WORDS: time marching; advection; numerical; finite volume; Riemann

1. INTRODUCTION

The advection equation plays an important role in many numerical models of fluid flow, both directly as a description of a physical process, or indirectly as part of a solution method. In either case the advection equation describes the temporal and spatial variation of a scalar quantity in response to a velocity field. Examples of direct and indirect applications of the advection equation include pollutant transport modelling and interface tracking, respectively. In pollutant transport modelling, the scalar is pollutant concentration and is described by a partial differential equation

*Correspondence to: C. G. Mingham, Centre for Mathematical Modelling and Flow Analysis, Manchester Metropolitan University, Chester Street, Manchester M1 5GD, U.K.
†E-mail: c.mingham@mmu.ac.uk

at the heart of which is the advection equation, e.g. [1]. There are two common methods for interface tracking: the volume of fluid (VoF) method [2] and the level set method [3] where the advection equation is used to describe the transport of a volume fraction function and a height function, respectively.

When modelling pollutant transport around chimney stacks or outfalls, concentrations can vary rapidly over small distances. These high spatial gradients present a severe challenge to classical numerical schemes; indeed many well-known schemes either introduce large dissipative and dispersive errors [1] or fail completely. Furthermore, flow features like sharp fronts and shear layers, caused by sudden variations in topography, introduce high-spatial gradients in velocities which also cause many existing numerical schemes to fail. A review of finite difference schemes for solving the linear advection equation can be found in [4]. The Bott scheme, used in atmospheric physics and based on polynomial interpolation of fluxes is an example of a modern scheme and shows promising results [5]. This scheme contains adjustable parameters, is more mathematically complicated than the authors' scheme, lacks the flexibility of the finite volume approach as it is essentially one-dimensional and relies on dimensional splitting which has been shown to produce inaccuracies for some problems [6]. In recent years classical schemes for pollutant transport have been put into a semi-Lagrangian framework to improve their computational efficiency and increase the allowable time step [7–9]. This depends on particle tracking along characteristic lines to determine the location of the new computational points for the next iteration. While these methods are relatively straightforward in one dimension, their extension to higher dimensions may become very complicated and computationally expensive. Other approaches include finite element schemes [10, 11].

It should be noted that pollutant transport is described, in general, by the advection–diffusion reaction equation. Some numerical schemes which perform poorly on purely advective problems *seem* to give good results when diffusion is included. We view these with suspicion since it is generally possible to choose diffusion coefficients to mask poor advective performance in *specific* problems.

An explicit finite volume algorithm of the Godunov type is described that accurately resolves high spatial gradients. The method employs cell-centred collocated data extrapolated to cell interfaces using MUSCL reconstruction [12] to provide second-order spatial accuracy in smooth regions dropping locally to first-order around discontinuities. Slope limiting is used to avoid spurious oscillatory behaviour around discontinuities during reconstruction which give rise to non-physical undershoots or overshoots. The resulting degenerate Riemann problems have trivial exact solutions. These in turn provide accurate cell interface data for the evaluation of fluxes around each cell. Second-order accuracy in time is achieved by using a two-step Runge–Kutta integrator. The scheme is presented in finite volume form which allows complex geometries to be fitted accurately with a boundary-conforming grid without incurring the complexity of an explicit co-ordinate transformation [13, 14], or the staircase-like inaccuracies of overlaying an irregular flow domain with a Cartesian mesh. Boundary-conforming grids can automatically be produced using the Cartesian cut cell method [15] which can also cope with moving solid regions [16].

The paper is organized as follows: Section 2 presents the equation describing the advection equation, Section 3 outlines the finite volume discretization, the authors' scheme, referred to as second-order limited (SOL) from now on, is described in Section 4, and Section 5 compares the scheme to published schemes for a number of one- and two-dimensional benchmark test cases. Finally, in Section 6 we make some concluding remarks and comments concerning future work.

DOI: 10.1002/fld

2. DEFINING EQUATION

The pure advection equation is

$$\frac{\partial c}{\partial t} + \frac{\partial (cu)}{\partial x} + \frac{\partial (cv)}{\partial y} = \frac{\partial (cw)}{\partial z} = 0 \tag{1}$$

where $c = c(x, y, z, t)$ is the scalar to be advected and $u = u(x, y, z, t)$, $v = v(x, y, z, t)$, $w = w(x, y, z, t)$ are velocities in the x, y and z directions, respectively. This equation is of hyperbolic type and admits *discontinuous* solutions such as sharp fronts that are difficult to resolve numerically. To simplify the analysis equation (1) is restricted to two spatial dimensions and the last term is omitted from all subsequent work. Extension of the SOL scheme to three spatial dimensions is straightforward.

3. FINITE VOLUME METHOD

The computational domain may contain geometrically complicated interior or exterior (boundary) solid regions and a classical finite difference approach based on a Cartesian grid using a staircase approximation to boundaries may lead to unnecessary solution errors. The advantage of the finite volume method is that it permits the use of boundary-fitted grids of either structured or unstructured type. This flexibility allows the computational grid to conform to irregular boundaries. At the same time, the cell size can be varied locally to provide enhanced resolution in areas of particular interest like plumes and fronts or the free surface in the popular VoF or level set methods. Furthermore, no explicit co-ordinate transformation is needed to treat complicated geometries. The dependent variables and flux terms remain referenced to the Cartesian frame. Cell areas and side vectors, easily computed by simple vector operations, take the place of the transformation Jacobian and metric terms; thus, the possibility of geometrically induced errors arising from the numerical discretization of the metric derivative terms is avoided.

Integrating (1) over an arbitrary planar region of area A with boundary S and using the Gauss divergence theorem gives

$$\frac{\partial}{\partial t} \iint_A c \, \mathrm{d}A + \oint_s \mathbf{H} \, \mathrm{d}\mathbf{s} = 0 \tag{2}$$

where $\mathbf{H} = c\mathbf{q}$ is the advective flux tensor, $\mathbf{q} = u\mathbf{i} + v\mathbf{j}$ is the velocity vector and, $\mathbf{i}$, $\mathbf{j}$ are the usual Cartesian basis vectors.

The physical region over which the equation is to be solved is tessellated by a structured mesh of cells indexed by i, j. The cell in row i and column j has area $A_{i,j}$. Note that a structured mesh is not necessary if an unsplit scheme is employed but is used here to simplify labelling. Let $c_{i,j}$ be the integral average of c over cell i, j (stored at the centre of the cell). Since (2) holds for an arbitrary region it can be approximated over each cell by

$$\frac{\partial c_{i,j}}{\partial t} = -\frac{1}{A_{i,j}} \sum_{k=1}^{4} \mathbf{H}_k \cdot \mathbf{s}_k \tag{3}$$

DOI: 10.1002/fld

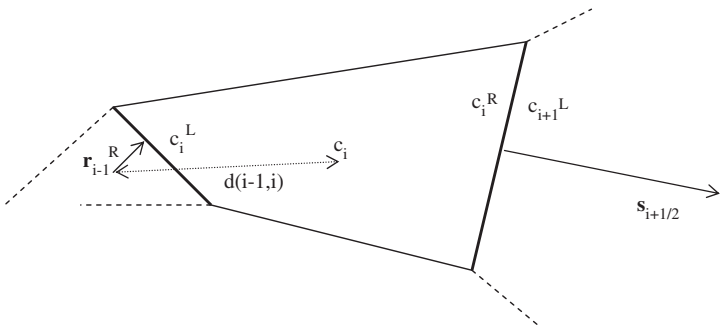

Figure 1. Cell i showing side and normal vectors, distance between cell centres and interface values of c.

where the summation is taken over the sides, k, of cell i, j and $\mathbf{s}_k$ is the outward-pointing normal vector to side k whose magnitude is the length of side k. $\mathbf{s}_k$ are called side vectors (see Figure 1). Note that a structured mesh implies that each cell has four sides but this need not be the case in a more general mesh; however, the principle of summing fluxes across cell sides remains the same.

4. DISCRETIZATION

The numerical scheme for solving (3) is denoted

$$c_{i,j}^{n+1} = L(\Delta t)c_{i,j}^{n} \tag{4a}$$

where $L(\Delta t)$ is an unsplit operator, Δt is the time step and the superscript denotes the number of time steps. Although the purpose of this paper is to present an unsplit scheme, for clarity of exposition and comparison with essentially one dimensional schemes the operator in (4a) is initially dimensionally split using the following operator sequence [17] to give:

$$c_{i,j}^{n+1} = L_j(\Delta t)L_i(\Delta t)c_{i,j}^{n} \tag{4b}$$

where $L_i(\Delta t)$ and $L_j(\Delta t)$ are one-dimensional operators in the i (row) and j (column) directions, respectively. Note that it has been reported [18] that dimensional splitting may lead to solution inaccuracies since only the first operator acts on the correct data. These errors may become more serious on a non-uniform mesh and so, to overcome this potential problem and to illustrate the flexibility of the scheme on a distorted mesh, the unsplit version is presented later.

Since the scheme has been reduced to two exactly similar one-dimensional operators it suffices to consider $L_i(\Delta t)$ and, since attention is restricted to a single row i, the subscript j will be dropped for clarity. Essentially the algorithm for $L_i(\Delta t)$ depends on estimating interface fluxes at either side of the computational cell. Since we are restricted to a row of cells each cell has two sides. Left and right cell interfaces are denoted by subtraction and addition, respectively, of $\frac{1}{2}$ to subscripts.

DOI: 10.1002/fld

4.1. Numerical scheme

The numerical algorithm for $L_i(\Delta t)$ is a two-stage spatially and temporally second-order accurate scheme of the Godunov type based on work attributed to Hancock [19] given by

Predictor:

$$c_i^{n+1/2} = c_i^n - \frac{\Delta t}{2A_i}(\mathbf{H}_i^{\mathrm{R}} \cdot \mathbf{s}_{i+1/2} + \mathbf{H}_i^{\mathrm{L}} \cdot \mathbf{s}_{i-1/2}) \tag{5a}$$

Corrector:

$$c_i^{n+1} = c_i^n - \frac{\Delta t}{A_i}(\mathbf{H}_{i+1/2}^{*} \cdot \mathbf{s}_{i+1/2} + \mathbf{H}_{i-1/2}^{*} \cdot \mathbf{s}_{i-1/2}) \tag{5b}$$

In the predictor stage (5a), $\mathbf{H}$ must be evaluated at each of the two interfaces of each cell and, since it depends on c, it is necessary to estimate c at these points. Let $c_i^{\mathrm{L}}, c_i^{\mathrm{R}}$ be the c values at the left and right interfaces of cell i, respectively, found by extrapolating c using the stored cell centre data and an estimate of the gradient of c across the cell (see Section 4.2). The corresponding interface fluxes are denoted $\mathbf{H}_i^{\mathrm{L}}$ and $\mathbf{H}_i^{\mathrm{R}}$, respectively.

The corrector stage (5b), requires $\mathbf{H}$ to be estimated as follows. At interface $i + \frac{1}{2}$ two values of c may be defined, one to the left and one to the right of the interface, namely c_i^{R} and c_{i+1}^{L}, respectively. These are obtained by linearly extrapolating the data from the centres of cells i and $i + 1$ to the interface *using the gradients calculated in the predictor stage*. This defines a local Riemann problem [20] at the interface where the left and right fluxes may be regarded as being constant states with the interface marking the point of discontinuity between them. The required interface flux, $\mathbf{H}_{i+1/2}^{*}$, in the corrector stage is the solution to this problem. In the case of simple advection this has the trivial exact solution $\mathbf{H}_i^{\mathrm{R}}$ if $\mathbf{q} \cdot \mathbf{s}_i > 0$ and $\mathbf{H}_{i+1}^{\mathrm{L}}$ otherwise. The advantage of this method is that the unique interface values are obtained by respecting the underlying physics once left and right states have been defined.

The overall allowable time step in Equation (4b) is the minimum of the time steps in the i and j directions and is defined by the usual Courant–Friedrichs–Lewy condition as, $\Delta t = v \min(\Delta t_i, \Delta t_j)$, where

$$\Delta t_i = \min_i \frac{A_i}{|\mathbf{q}_i \cdot \mathbf{s}_{i+1/2}|} \tag{6}$$

with Δt_j being defined similarly. Theoretically, the maximum value of the Courant number, v, for this (split) scheme to be stable is 1.0.

4.2. Gradient calculation and MUSCL interpolation

To obtain the interface values c_i^{L} and c_i^{R}, c is assumed to have a piecewise linear variation within cell i. Its gradient is obtained by estimating the (left) gradient between cell centre data in cells $i - 1$ and i, and the (right) gradient between cell centre data i and $i + 1$. To avoid non-physical interpolated values these gradients must be limited using a slope limiter function $f(x, y)$ so that estimated cell interface data lie within adjacent stored cell centre values. Since the mesh is, in general, non-uniform, gradients must be regarded as vectors and c extrapolated to cell interfaces in the required direction using directional derivatives. c_i^{R} and c_i^{L} are given by

$$c_i^{\mathrm{R}} = c_i + \mathbf{r}_i^{\mathrm{R}} \cdot \mathbf{g}_i, \quad c_i^{\mathrm{L}} = c_i + \mathbf{r}_i^{\mathrm{L}} \cdot \mathbf{g}_i \tag{7a}$$

DOI: 10.1002/fld

where $\mathbf{r}_i^{\mathrm{R}}$, $\mathbf{r}_i^{\mathrm{R}}$ are the normal distance vectors from the centre of cell i to the right and left interfaces, respectively (Figure 1) and $\mathbf{g}_i$ is the gradient vector in cell i. The left and right scalar gradients g_i^{L} and g_i^{R}, respectively, are defined by

$$g_i^{\mathrm{L}} = \frac{c_i - c_{i-1}}{d(i-1,i)}, \quad g_i^{\mathrm{R}} = \frac{c_{i+1} - c_i}{d(i,i+1)} \tag{7b}$$

where $d(n,m)$ is the distance between cell centres n and m (Figure 1). These scalar gradients are limited by the limiter function f and converted to a vector to give

$$\mathbf{g}_i = f(g_i^{\mathrm{L}}, g_i^{\mathrm{R}})\mathbf{n}_{i-1,i+1} \tag{8}$$

where $\mathbf{n}_{i-1,i+1}$ is the unit vector from the centre of cell $i-1$ to the centre of cell $i+1$ and f is given by

$$f(x,y) = \begin{cases} \max\{0, \min(kx, y), \min(x, ky)\}, & x>0 \\ \min\{0, \max(kx, y), \max(x, ky)\} & \text{otherwise} \end{cases} \tag{9}$$

In all reported numerical experiments k was set to 2.0 which corresponds to the Superbee limiter.

The unsplit version of the scheme requires flux calculations at each cell interface. For a structured mesh each cell has four interfaces and the required fluxes are found in the same way as above. The scheme is

Predictor:

$$c_{i,j}^{n+1/2} = c_{i,j}^n - \frac{\Delta t}{2A_{i,j}}(\mathbf{H}_{i,j}^{\mathrm{R}} \cdot \mathbf{s}_{i+1/2,j} + \mathbf{H}_{i,j}^{\mathrm{L}} \cdot \mathbf{s}_{i-1/2,j} + \mathbf{H}_{i,j}^{\mathrm{U}} \cdot \mathbf{s}_{i,j+1/2} + \mathbf{H}_{i,j}^{\mathrm{D}} \cdot \mathbf{s}_{i,j-1/2}) \tag{10a}$$

Corrector:

$$c_{i,j}^{n+1} = c_{i,j}^n - \frac{\Delta t}{A_{i,j}}(\mathbf{H}_{i+1/2,j}^* \cdot \mathbf{s}_{i+1/2,j} + \mathbf{H}_{i-1/2,j}^* \cdot \mathbf{s}_{i-1/2,j} + \mathbf{H}_{i,j+1/2}^* \cdot \mathbf{s}_{i+1/2,j} + \mathbf{H}_{i,j-1/2}^* \cdot \mathbf{s}_{i,j-1/2}) \tag{10b}$$

Where the superscripts R, L, U and D denote right, left, upper and lower cell interfaces, respectively. The time step is given by $\Delta t = v \min_{i,j}(\Delta t_{i,j})$ where

$$\Delta t_{i,j} = \frac{A_{i,j}}{|\mathbf{q}_{i,j} \cdot \mathbf{s}_{i+1/2,j}| + |\mathbf{q}_{i,j} \cdot \mathbf{s}_{i,j+1/2}|} \tag{11}$$

v was set to 0.7 for all unsplit computations.

4.3. Boundary conditions

Transmissive boundary conditions were used throughout. These were obtained by assuming a zero gradient for c at the boundary so that its value is taken to be that at the neighbouring cell centre. If required, boundary conditions appropriate to a solid boundary can easily be imposed since the flow tangency condition implies that $\mathbf{q} \cdot \mathbf{s} = 0$ and therefore, at a solid interface, the fluxes are zero.

DOI: 10.1002/fld

5. NUMERICAL RESULTS AND DISCUSSION

The following one- and two-dimensional benchmark test problems present a severe challenge to numerical schemes because of discontinuities in the concentration gradient which occur in each case. Any flow solver applied to single species or multispecies pollutant transport would need to be able to provide high accuracy for these test problems. The SOL scheme and several published schemes were applied to each test case where the numerical results can be compared to exact solutions. In each case the schemes were run at around 95% of their maximum allowable time steps so that a fair comparison could be made. The final time step was reduced so that stated run times were achieved exactly.

5.1. Test 1: linear advection of one-dimensional composite profile

A domain of length 20 m was uniformly discretized using 100 cells. The initial scalar profile, $c(x, 0)$, which contains both smooth and discontinuous data was given by

$$C(x,0) = \begin{cases} 1, & 1 \leqslant x \leqslant 3 \\ x-4, & 4 \leqslant x \leqslant 5 \\ -x+6, & 5 \leqslant x \leqslant 6 \\ \cos(0.5\pi(x-8)), & 7 \leqslant x \leqslant 9 \\ \exp(-4.5(x-11)^2), & 10 \leqslant x \leqslant 12 \\ 0 & \text{elsewhere} \end{cases}$$

The initial profile was advected for 7 s at a speed of $u = 1\,\mathrm{m\,s^{-1}}$. Figure 2 compares numerical and exact solutions. Of the two first-order schemes, the Lax–Friedrichs (LXF) scheme [21] exhibits significant smearing of the square profile and loss of peak amplitude in the other profiles. The first-order upwind (FOU) scheme [21] is better but still smears the discontinuities in the square profile over five mesh points and overestimates the zero concentrations between profiles in addition to underestimating peak concentrations. These two schemes will be omitted from further tests as the failure of first-order schemes to resolve high spatial gradients due to the presence of strong numerical diffusion is a well-known phenomenon.

Two of the classical second-order schemes: the MacCormack scheme (MAC) [22] and the same scheme with an added total variation diminishing (TVD) correction term (MACTVD) [23], show considerable inaccuracies both in terms of their predictions of extrema and resolution of high spatial gradients. The MAC scheme produces negative concentrations and a significant overshoot on the square profile although it does resolve the continuous profiles fairly well. The addition of the TVD corrective term improves the resolution of the square profile at the expense significant loss of peak amplitude on the continuous profiles and overshooting of the zero concentration background between profiles. The third-order SHARP [24] scheme, the final classical scheme in the test, produces significant under- and overshoots especially for the square component.

Results from both the SOL scheme and the Bott scheme [5] show considerable improvement over the classical schemes. Each of these schemes resolves the discontinuities in the square part of the profile over only three mesh points and accurately approximates the maximum concentration. The continuous components of the initial profile are resolved with similar accuracy and there is

DOI: 10.1002/fld

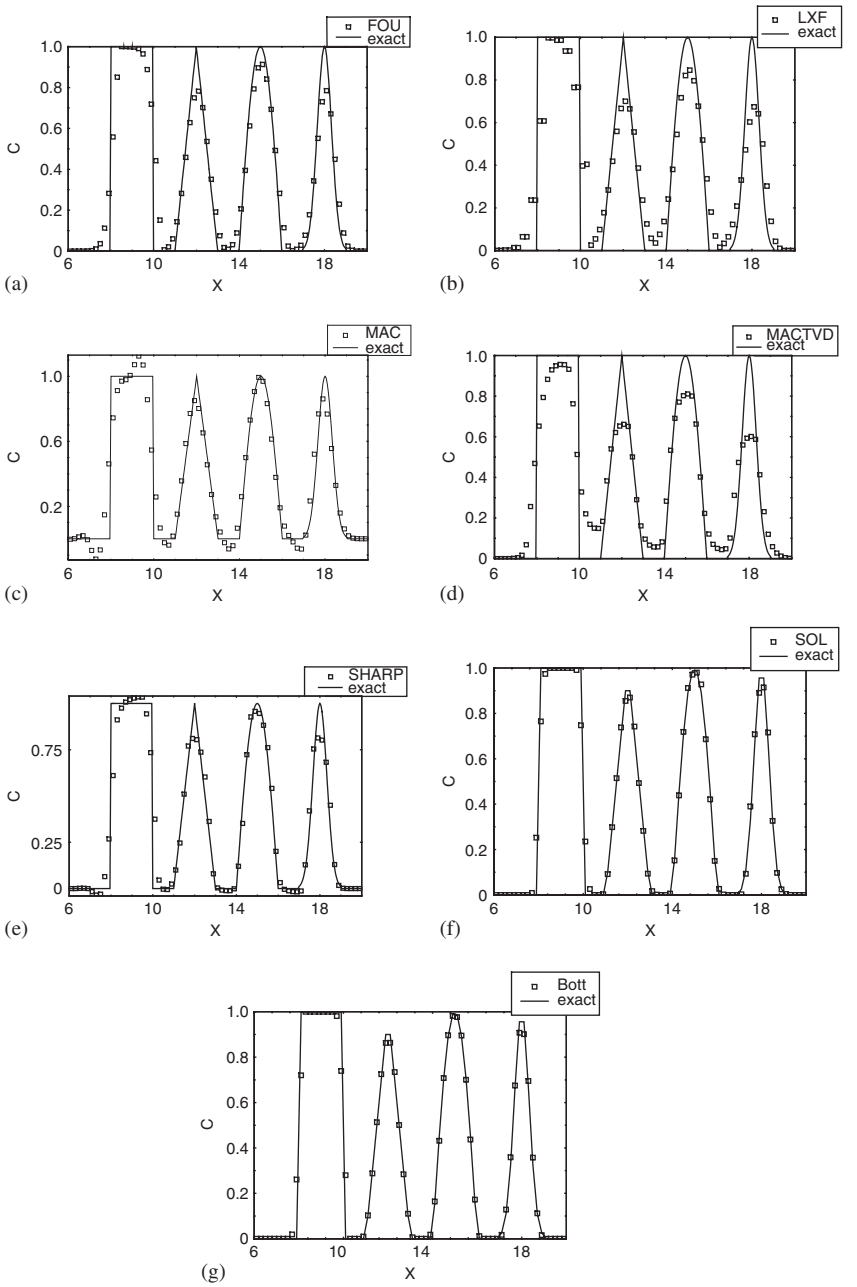

Figure 2. Test 1: linear advection of a composite profile; comparisons of numerical and exact solutions: (a) FOU; (b) LXF; (c) MAC; (d) MACTVD; (e) SHARP; (f) SOL; and (g) BOTT.

DOI: 10.1002/fld

Table I. Comparison of accuracy of the schemes for test cases 1–3: (a) Test 1: one-dimensional advection of a composite profile; (b) Test 2: two-dimensional rotation of a notched cylinder; (c) Test 3: two-dimensional rotation of a cone; (d) Test 4a: Cyclogenisis on a uniform mesh; and (e) Test 4b: Cyclogenisis on a non-uniform mesh.

(a) Test 1	LXF	FOU	MAC	MACTVD	SHARP	SOL	BOTT
Min c	0.00	0.00	−0.12	0.00	−0.03	0.00	0.00
Max c	0.99	0.99	1.12	0.96	1.04	1.00	1.00
Mean abs(error)*	0.05	0.05	0.05	0.08	0.04	0.01811	0.0002

(b) Test 2	MAC	MACTVD	SHARP	SOL	BOTT
Min c	0.48	0.99	0.99	1.00	1.00
Max c	1.86	1.52	1.57	1.96	2.00
Mean abs(error)	0.0514	0.0250	0.0242	0.0083	0.0162

(c) Test 3	SOL	BOTT
Min c	1.00	1.00
Max c	1.84	2.00
Mean abs(error)	0.0040	0.0016

(d) Test 4(a)	SOL $t=1$	SOL $t=3$	SOL $t=8$	BOTT $t=1$	BOTT $t=3$	BOTT $t=8$
Min c	−0.964	−0.964	−0.964	−0.964	−0.964	−0.964
Max c	0.964	0.964	0.964	0.964	0.964	0.964
Mean abs(error)	0.0009	0.0025	0.0067	0.0008	0.0017	0.0041

(e) Test 4(b)	$t=1$	$t=3$	$t=8$
Min c	−0.974	−0.990	−1.013
Max c	0.973	0.981	0.987
Mean abs(error)	0.0036	0.0061	0.0133

*Mean abs(error)$=(\sum_{i=1}^{N}|c_i - e_i|)/N$ where c_i , e_i are the ith numerical and exact solution values, respectively, and N is the total number of mesh points.

no appreciable overall phase error or tendency to under- or overshoot. Table I(a) summarizes the accuracy of the schemes. It can be seen that classical schemes are considerably less accurate than modern schemes with the Bott scheme being the most accurate.

In the next two test cases the SOL scheme is applied in its split form for comparison purposes.

5.2. Test 2: rotation of a notched cylinder

This test problem involves high spatial gradients which are not aligned with the grid. A 100 m square region was discretized using 100 square cells in each of the x and y directions. The initial background scalar value was unity outside a notched circle of radius 10 m centred on $x=50$ m, $y=25$ m and two inside. The notch was defined as the region inside the circle such that $x<50$ and $20<y<30$. A spatially varying velocity field was used to produce a uniform angular rotation rate of one cycle per second about the centre of the region. Figure 3 shows surface plots of the advected scalar after six anti-clockwise rotations for each of the schemes and Table I(b) summarizes the accuracy of the schemes.

DOI: 10.1002/fld

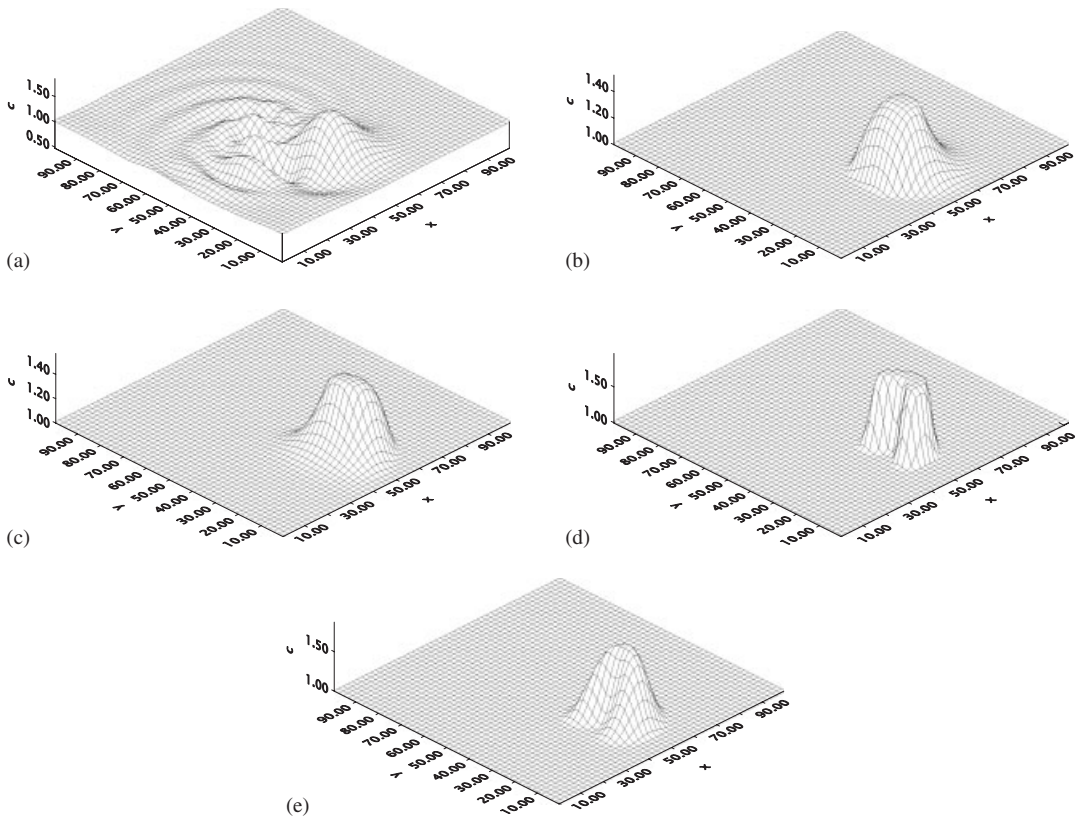

Figure 3. Test 2: rotating notched cylinder; comparison of numerical solutions: (a) MAC; (b) MACTVD; (c) SHARP; (d) SOL; and (e) BOTT.

The surface plots show that the classical schemes give poor results and in each case the notch more-or-less vanishes and there is a significant reduction of peak scalar value. The oscillatory nature of the MAC scheme produces a trailing wake and a corruption of the original profile whilst the MACTVD and SHARP schemes smear the vertical cylinder sides. The Bott scheme preserves extreme scalar values but exhibits some diffusion. In contrast, the SOL scheme performs well and resolves the vertical cylinder sides and the notch. Table I(b) shows that the SOL and Bott schemes are by far the most accurate with the latter being marginally the best. As the classical schemes produce poor results they were omitted from further tests.

5.3. Test 3: rotation of a two-dimensional cone

The initial scalar profile was a right circular cone having a peak value of two centred on $x = 25$ m, $y = 50$ m and radius 15 m. The background value was unity outside the cone's base. The mesh and rotational velocity were as in the previous test. Figure 4 shows surface and contour plots of the advected scalar after six anti-clockwise rotations for the SOL and Bott schemes. Table I(c) summarizes the accuracy of the schemes.

DOI: 10.1002/fld

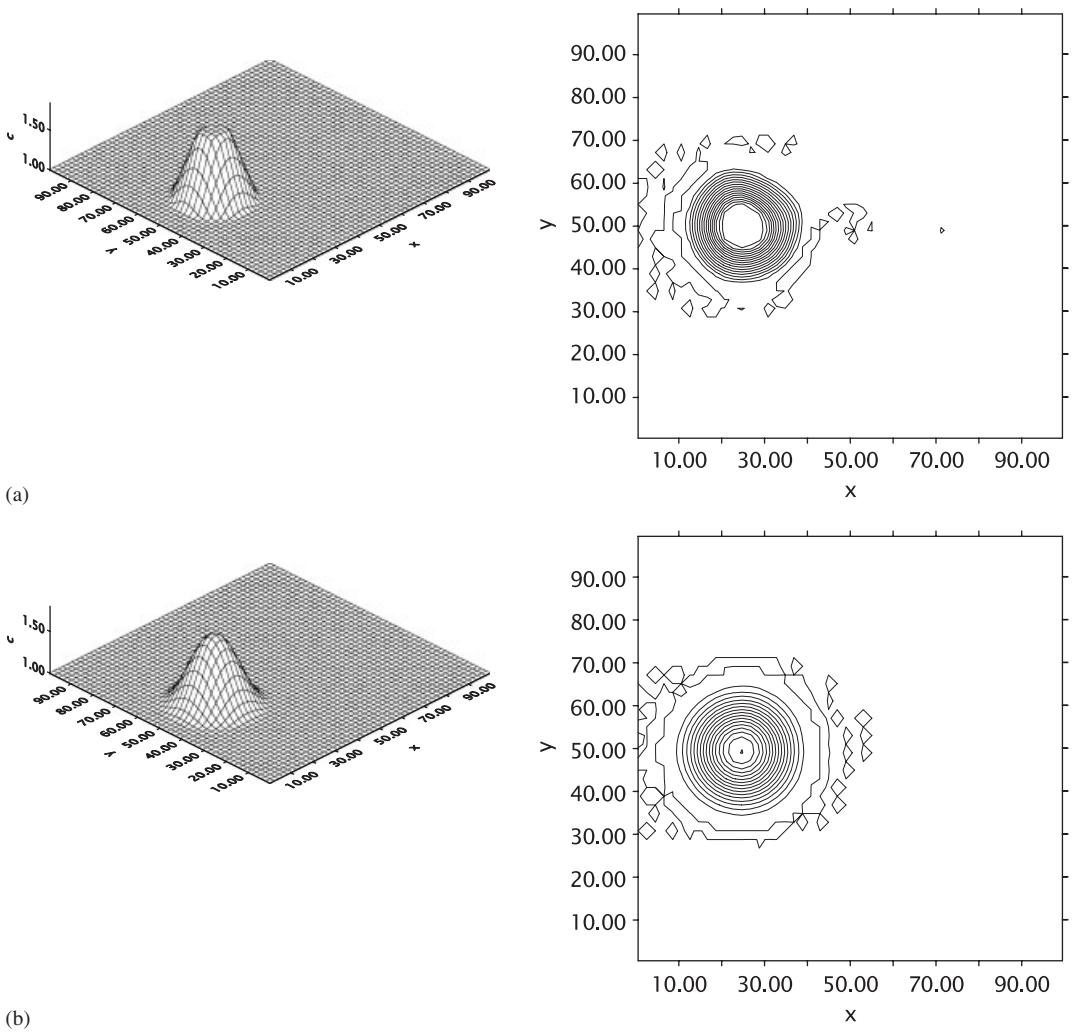

Figure 4. Test 3: rotating cone; problem; comparison of numerical solutions: (a) SOL and (b) BOTT.

Both schemes perform well and results suggest minimal phase error and no distortion of the background unit scalar field away from the cone. The Bott scheme captures the peak value but rounds off the vertex slightly. The SOL scheme underestimates peak value slightly and produces a small plateau instead of a sharp vertex. However, an inspection of the contour plots shows that the SOL scheme resolves the circular base of the cone better than the Bott scheme which diffuses it slightly. The Bott scheme has a lower mean absolute error that the SOL scheme.

5.4. Test 4(a): cyclogenisis on a uniform mesh

This final problem provides an exacting test of any numerical scheme since it involves regions of both high and low spatial gradients which evolve in time and space. The Bott scheme and the SOL

DOI: 10.1002/fld

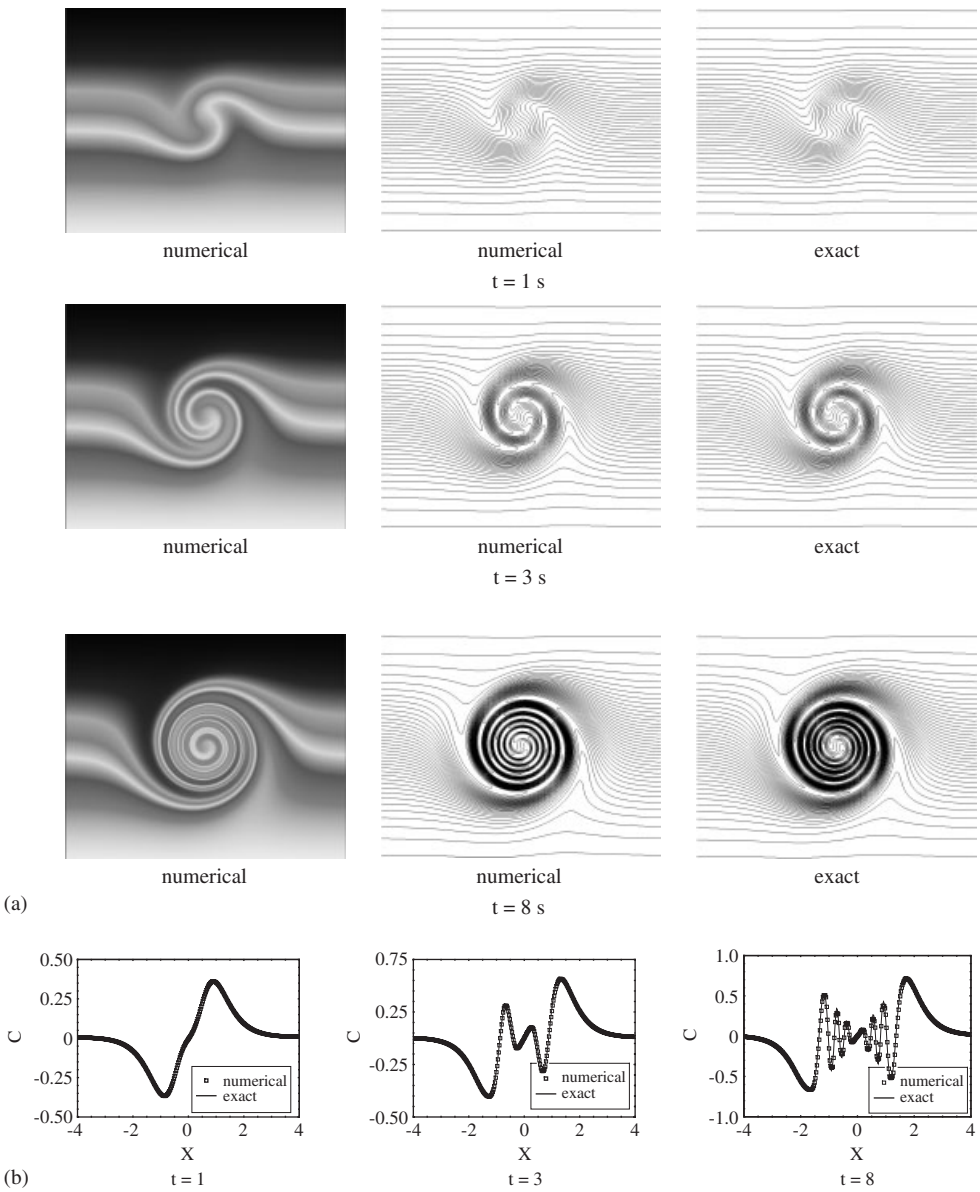

Figure 5. Test 4(a): cyclogenisis problem on a uniform mesh; comparison of SOL scheme and exact solutions: (a) contour plots and (b) horizontal cross sections at $y = 0$.

scheme in its unsplit form are applied to the cyclogenisis problem which describes the rotational mixing of a warm (positive c values) and a cold (negative c values) front under the influence of an applied angular velocity [25]. The computational region was a square of side length 8 centred

DOI: 10.1002/fld

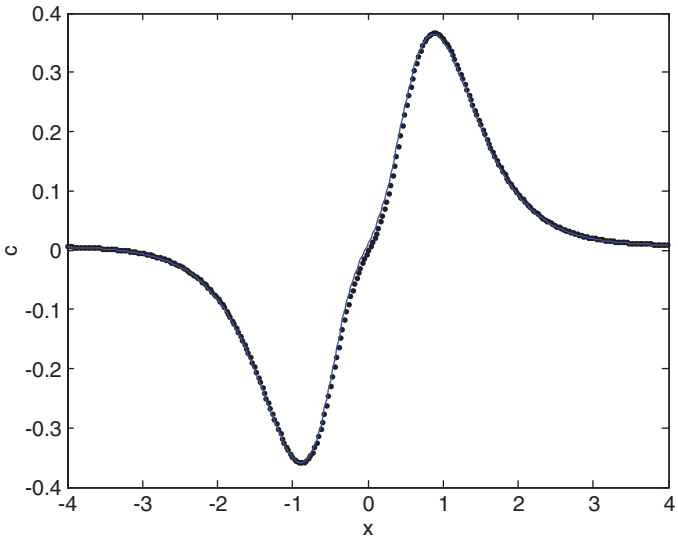

Figure 6. Test 4(a): comparison of SOL (dotted line) and Bott (solid line) results *via* a horizontal cross section at $y = 0$ at $t = 8$.

on the origin. The exact solution for c in non-dimensionalized form is given by

$$c(x, y, t) = -\tanh(0.5(y\cos(wt) - x\sin(wt))$$

where w, d and P denote rotation frequency, distance from the origin and tangential velocity, respectively and are given by

$$w(x, y) = P/(d\max(P)), \quad d = \sqrt{x^2 + y^2}, \quad P = \sec h^2(d)\tanh(d)$$

It is easily shown that the maximum tangential velocity, $\max(P)$, over the computational region is 0.385. The velocity field is given by

$$u(x, y) = -wy, \quad v(x, y) = wx$$

Contour plots of results from the SOL scheme and exact results after $t = 1$, 3 and 8 for a 300×300 uniform mesh displayed in Figure 5(a) and (b) show corresponding cross-sectional distributions of the advected scalar along the x-axis. Corresponding contour plots for the Bott scheme are indistinguishable from those of the SOL scheme so are not displayed. SOL and Bott results are compared via a graph of cross sectional distribution along the x-axis at $t = 8$ in Figure 6. Table I(d) summarizes accuracy. SOL and Bott schemes give the same maximum and minimum c values for the three times but overall the mean absolute error of the Bott scheme is slightly lower than that of the SOL scheme.

As time increases the structure of the computed solutions at the cross section become more complicated and there are rapid changes in gradients. Both the Bott and unsplit SOL schemes

DOI: 10.1002/fld

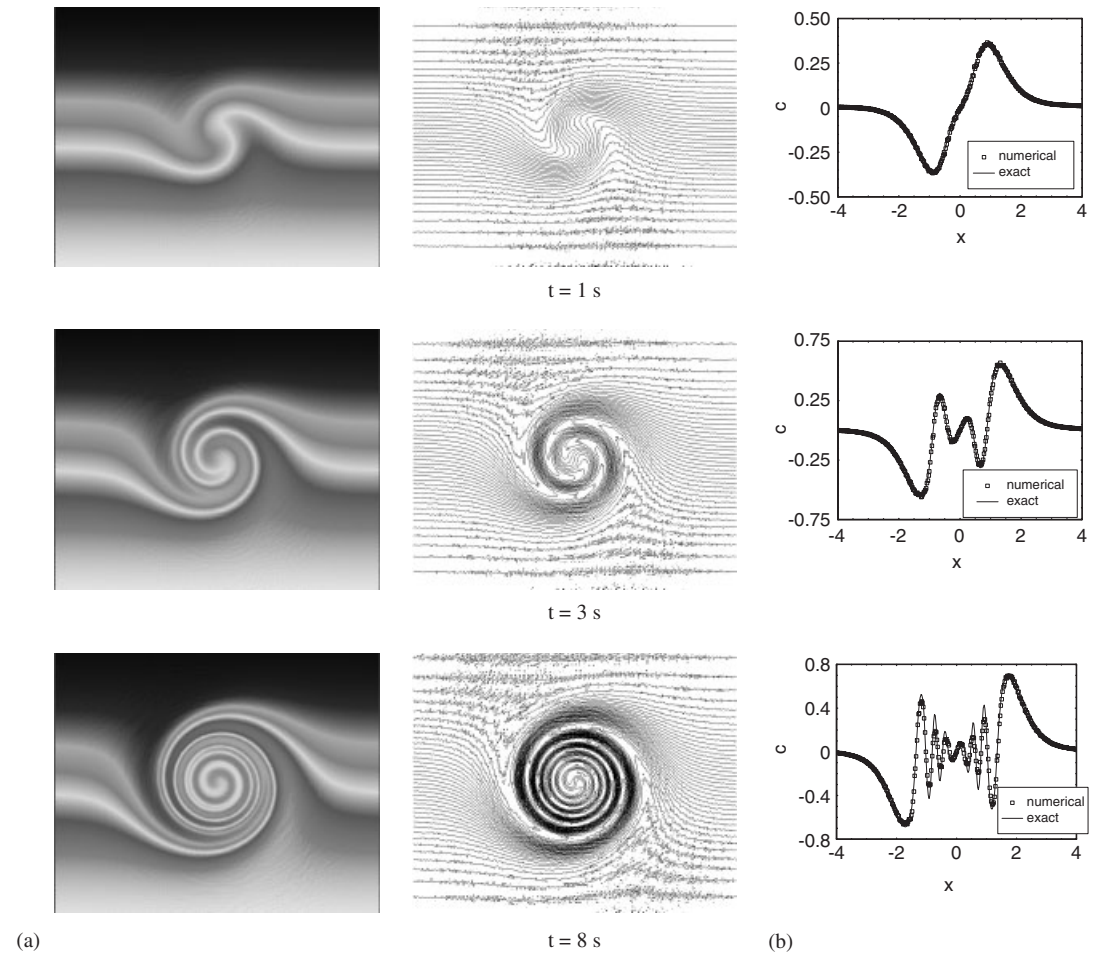

Figure 7. Test 4(b): cyclogenisis problem on a non-uniform mesh; numerical solutions using SOL scheme: (a) contour plots and (b) horizontal cross sections at $y = 0$.

reproduce the structure and position of the exact solution very accurately and numerical and analytical results in each case are virtually indistinguishable as can be seen from Figure 6.

5.5. Test 4(b): cyclogenisis on a non-uniform mesh

The flexibility of the finite volume version of the SOL scheme is illustrated by repeating the previous problem using a non-uniform mesh. Although the mesh remains structured this does not have any bearing on the scheme which simply computes fluxes across cell interfaces regardless of how cells are indexed. It should be noted that the Bott scheme cannot be applied to an unstructured mesh as it requires dimensional splitting to be extended to more than one dimension. The mesh is structured so that it can easily be reproduced by the reader. The mesh was generated by adding $0.0088\cos(9.876543ij)$ to the x coordinate and $0.0088\cos(29.876543ij)$ to the y co-ordinate of

DOI: 10.1002/fld

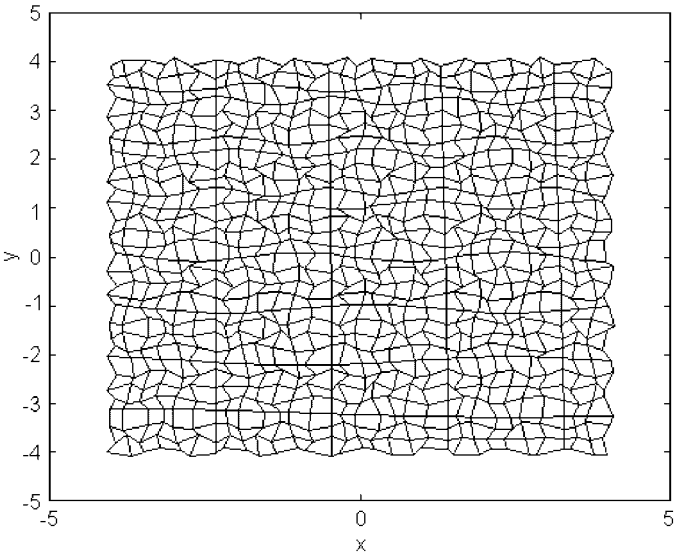

Figure 8. Test 4(b): 30×30 non-uniform coarse mesh for the cyclogenisis problem.

the lower left hand corner of cell i, j in the previous uniform mesh. Corresponding results are displayed in Figure 7(a), (b) and Table I(e). Despite the distorted mesh the SOL scheme once again produces accurate results and resolves the complicated solution structure and rapid changes in gradients. It should be noted that the mesh, a coarse version of which is illustrated in Figure 8, may be much more distorted than would be necessary in a real problem.

6. CONCLUSIONS

The presented numerical scheme, SOL, provides accurate solutions to the pure advection equation in one and two dimensions. It is able to resolve both high and low spatial gradients and capture peak values without exhibiting excessive numerical dissipation or dispersion. The scheme is robust and easy to implement and results show significant gains in accuracy when compared to a range of classical schemes.

The SOL scheme is slightly less accurate than the Bott scheme over the test cases presented. The advantage of the SOL scheme is that it intrinsically multidimensional unlike the Bott scheme which requires dimensional splitting to extend it to multidimensional problems and thus cannot be implemented on an unstructured mesh. The SOL scheme can be implemented in unsplit finite volume form to solve the advection equation on boundary-conforming structured or unstructured meshes and thus can be used for practical problems where the geometry of the computational region is complex.

REFERENCES

1. Chen Y, Falconer RA. Advection–diffusion modelling using the modified QUICK scheme. *International Journal for Numerical Methods in Fluids* 1992; **15**:1171–1196.

DOI: 10.1002/fld

2. Hirt CW, Nichols BD. Volume of fluid (VOF) method for dynamics of free boundaries. *Journal of Computational Physics* 1981; **39**:201–225.
3. Osher S, Sethian J. Fronts propagating with curvature-dependent speed: algorithms based on Hamilton–Jacobi formulations. *Journal of Computational Physics* 1988; **79**:12–49.
4. Falconer RA. Flow and water quality modelling in coastal and inland waters. *Journal of Hydraulic Research* 1992; **30**:437–452.
5. Bott A. A positive definite advection scheme obtained by nonlinear renormalization of the advective fluxes. *Monthly Weather Review* 1989; **117**:1006–1015.
6. Birman A, Har'El NY, Falcovitz J, Ben-Artzi M, Feldman U. Operator-split computation of 3D symmetric flow. *Twenty-second International Symposium on Shock Waves*, Southampton, U.K., 1999, paper 3461.
7. Giraldo FX, Neta B. A comparison of a family of Eulerian and semi-Lagrangian finite element methods for the advection–diffusion equation. *Computer Modelling of Seas and Coastal Regions 3—Coastal 97*. La Coruna: Spain, 1997; ISBN 1-85312-4990.
8. Manson JR, Wallis SG. An accurate numerical algorithm for advective transport. *Communications in Numerical Methods in Engineering* 1995; **11**:1039–1045.
9. Yeh GT. A Lagrangian–Eulerian method with zoomable hidden fine-mesh approach to solving advection–dispersion equations. *Water Resources Research* 1990; **26**:1133–1144.
10. Carmichael GR, Peters LK, Kitada T. A second generation model for region-scale transport/chemistry/deposition. *Atmospheric Environment* 1986; **20**:173–188.
11. Li CW. Least square-characteristics and finite elements for advection–dispersion simulation. *International Journal for Numerical Methods in Engineering* 1990; **29**:1343–1364.
12. Van Leer B. Towards the ultimate conservative difference scheme II: monotonicity and conservation combined in a single scheme. *Journal of Computational Physics* 1974; **14**:361–370.
13. Borthwick AGL, Barber RW. The River and reservoir flow modelling using the transformed shallow water equations. *International Journal for Numerical Methods in Fluids* 1996; **14**:1193–1217.
14. Barber RW, Pearson RV. Modelling tide-induced currents and pollutant transport using a generalised curvilinear co-ordinate system. *Proceedings of the 1st International Conference on Environmental Problems in Coastal Regions,* Rio, Brasil. Wessex Institute of Technology: U.K., 1996; 263–274.
15. Yang G, Causon DM, Ingram DM, Saunders R, Batten P. A Cartesian cut cell method for compressible flows—Part A: static body problems. *Aeronautical Journal* 1997; **101**(1002):47–56.
16. Yang G, Causon DM, Ingram DM, Saunders R, Batten P. A Cartesian cut cell method for compressible flows—Part B: moving body problems. *Aeronautical Journal* 1997; **101**(1002):57–65.
17. Strang G. On the construction and comparison of finite difference schemes. *SIAM Journal on Numerical Analysis* 1968; **5**:506–517.
18. Valocchi AJ, Malmstead M. Accuracy of operator splitting for advection–dispersion reaction problems. *Water Resources Research* 1992; **28**(5):1471–1476.
19. Van Leer B. On the relation between the upwind differencing schemes of Godunov, Enquist–Osher and Roe. *SIAM Journal on Scientific and Statistical Computing* 1984; **5**(1):1–20.
20. Toro E. *Riemann Solvers and Numerical Methods for Fluid Dynamics*. Springer: Berlin, 1997.
21. Leveque RJ. *Numerical Methods for Conservation Laws*. Birkhauser: Basel, 1994; 101. ISBN 3-8176-2723-5.
22. MacCormack RW. The effect of hypervelocity impact cratering. *AIAA Paper No. 69-354*, 1969.
23. Mingham CG, Causon DM, Ingram DM. A TVD MacCormack scheme for transcritical flow. *Proceedings of the ICE*: *Water and Maritime Engineering* 2001; **148**(3):167–175. ISSN 1472-4561.
24. Leonard BP. Simple high-accuracy resolution program for convective modelling of discontinuities. *International Journal for Numerical Methods in Fluids* 1988; **8**(10):1291–1318.
25. Tamamidis P, Assanis DN. Evaluation of various high-order-accuracy schemes with and without flux limiters. *International Journal for Numerical Methods in Fluids* 1993; **16**:931–948.

DOI: 10.1002/fld