

Chapter 2

Finite Volume Differencing Schemes

This chapter discusses the basic techniques for the numerical solution of Partial Differential Equations (PDEs) using Finite Volume approximations. With analytic methods the solution to a PDE is found for all locations within the domain of interest. However, with finite volume or finite difference methods the solution is found only for a set of discrete points within the space of the domain. For finite difference methods the domain has a number of points placed within it, and the equations are solved at each point. In contrast, for finite volume methods the domain is divided up into a number of control volumes, with the value at the centre of the control volume being held to be representative for the value over the entire control volume. By integrating the original PDE over the control volume the equation is cast into a form that ensures conservation. The derivatives at the faces of the volume are approximated by finite difference equations, and a system of sparse linear equations is generated that can be solved using a standard linear method. The process of transforming a PDE into a set of linear equations is termed discretisation.

In this chapter we will only consider the discretisation of a generic transport equation upon a Cartesian mesh (ie: one aligned with the x , y and z axis of the domains coordinate system). The notation used for the discretisation is described in Section 2.1, with the steady transport equation being discretised in Section 2.2, and the transient transport equation being discretised in Section 2.4. The accuracy of the different schemes is tested in Section 2.5 for both transient and steady flow problems.

The methods used to solve the linear systems generated by the finite volume discretisations are given in Chapter 3, whilst the methods used to solve the Navier–Stokes equations will be described in Chapter 4. Further finite volume discretisations that are not restricted to Cartesian meshes and can be used on a general curvilinear mesh are derived in Chapter 5.

2.1 Equations and Notation

The transient transport equation for the conservation of a specific¹ scalar ϕ in a fluid undergoing advection and diffusion can be written as

$$\frac{\partial}{\partial t}(\rho\phi) + \nabla \cdot (\rho\mathbf{u}\phi) = \nabla \cdot (\Gamma\nabla\phi) + S, \quad (2.1)$$

where Γ is the diffusion coefficient, S any source term for the scalar ϕ per unit volume, ρ the density of the fluid and \mathbf{u} the fluid's velocity field. The first term in Equation (2.1) is the time derivative measuring the rate of change of ϕ with respect to time, and the second term is the advective component representing the transport of ϕ by the ambient velocity field. On the right hand side of the equation

¹ie: one defined *per unit mass*

the first term is the transport due to diffusion, whilst the final term is the contribution due to sources of ϕ within the field.

For a one dimensional domain Equation (2.1) can be written

$$\frac{\partial}{\partial t}(\rho\phi) + \frac{\partial}{\partial x}(\rho u\phi) = \frac{\partial}{\partial x}\left(\Gamma\frac{\partial\phi}{\partial x}\right) + S, \quad (2.2)$$

whilst in two and three dimensions it may be written as

$$\frac{\partial}{\partial t}(\rho\phi) + \frac{\partial}{\partial x}(\rho u\phi) + \frac{\partial}{\partial y}(\rho v\phi) = \frac{\partial}{\partial x}\left(\Gamma\frac{\partial\phi}{\partial x}\right) + \frac{\partial}{\partial y}\left(\Gamma\frac{\partial\phi}{\partial y}\right) + S, \quad (2.3)$$

and

$$\begin{aligned} \frac{\partial}{\partial t}(\rho\phi) + \frac{\partial}{\partial x}(\rho u\phi) + \frac{\partial}{\partial y}(\rho v\phi) + \frac{\partial}{\partial z}(\rho w\phi) = \\ \frac{\partial}{\partial x}\left(\Gamma\frac{\partial\phi}{\partial x}\right) + \frac{\partial}{\partial y}\left(\Gamma\frac{\partial\phi}{\partial y}\right) + \frac{\partial}{\partial z}\left(\Gamma\frac{\partial\phi}{\partial z}\right) + S, \end{aligned} \quad (2.4)$$

respectively, with u , v and w being the x , y and z components of velocity.

For the steady transport equation the time derivative is set to zero, and so the leading term may be dropped from Equations (2.1) to (2.4).

For a finite volume discretisation the domain over which the transport equation is to be solved is divided up into a number of discrete volumes, as in Figure 2.1, with each volume having a representative value located at its centre. For a one dimensional domain the cells are numbered from 1 to N , with the cell at point i having neighbours $i - 1$ and $i + 1$, the interfaces with those neighbouring volumes being $i - \frac{1}{2}$ and $i + \frac{1}{2}$. For a two dimensional domain the cells are numbered (i, j) , and for a three dimensional domain they are numbered (i, j, k) .

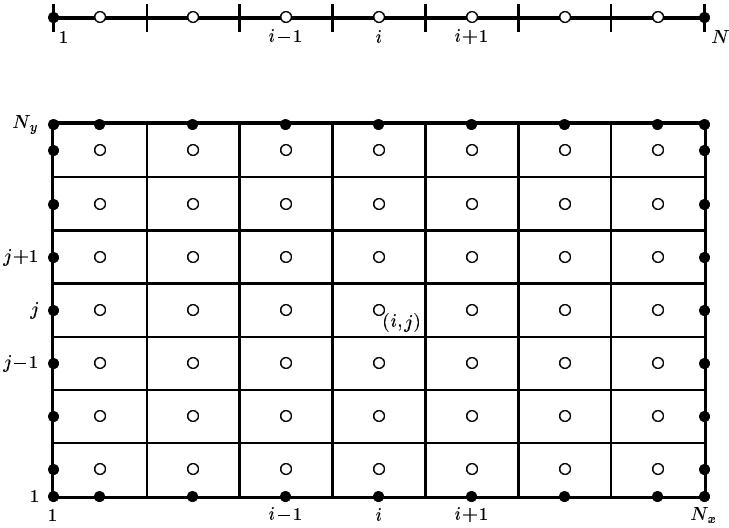


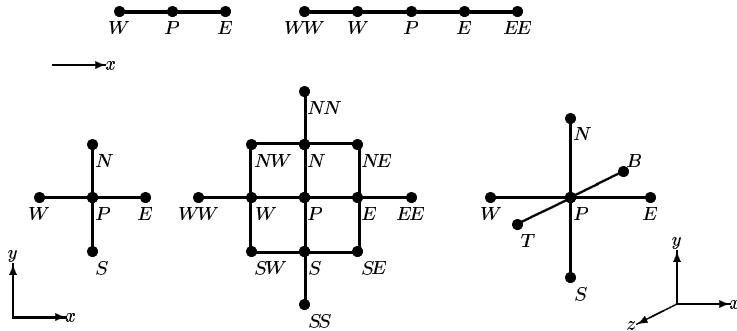
Figure 2.1: The discretisation of a 1D (top) and 2D (bottom) domain into Cartesian finite volumes.

For convenience, the commonly used compass notation originating from Imperial College[43, 122] will be used when referring to the neighbours of a cell. The subscript P signifies the cell upon which the equation is being discretised, whilst its immediate neighbours in the x axis are labelled E and W (for East and West), in the y axis N and S (for North and South), and in the z axis T and B (for Top and Bottom). A capital letter signifies a value at the centre of the neighbouring cell, whilst a

Grid location	Compass notation
i, j, k	P
$i-1, j, k$	W
$i+1, j, k$	E
$i, j-1, k$	S
$i, j+1, k$	N
$i, j, k-1$	B
$i, j, k+1$	T
$i-\frac{1}{2}, j, k$	w
$i+\frac{1}{2}, j, k$	e
$i, j-\frac{1}{2}, k$	s
$i, j+\frac{1}{2}, k$	n
$i, j, k-\frac{1}{2}$	b
$i, j, k+\frac{1}{2}$	t

Table 2.1: Conversion between the mesh index and the compass notation.

lower case letter signifies a value at the interface between the two cells. Cells further away from the P cell are named using the chain of locations that link them back to the central point P . For instance the cell that is to the North of the East neighbour is called NE , whilst the cell to the West of the West neighbour is WW . A diagram of these neighbouring nodes which make up the computational molecule at a point is given in Figure 2.2, whilst a table converting between this compass notation and the cell indexing previously referred to is given in Table 2.1.

Figure 2.2: The computational molecule at point P for one-, two- and three-dimensional meshes, illustrating the compass notation used.

This chapter will only describe discretisations on Cartesian meshes, with the complications of non-orthogonal meshes being left to Chapter 5. The geometry for a cell in a one-dimensional mesh is given in Figure 2.3 with the geometry for a cell in a two-dimensional Cartesian mesh being given in Figure 2.4. The cell has width Δx and height Δy , whilst the distance between the centre of the cell and that of its immediate neighbour on the right (or east) side is δx_e , and that with its left (or west) neighbour is δx_w .

For a one-dimensional mesh the area of the cell faces are taken to be unitary, ie: $A_e = A_w = 1$, whilst the volume of the cell is $\Delta\Omega = \Delta x$. With the two-dimensional Cartesian mesh the cell face areas are $A_e = A_w = \Delta y$ and $A_n = A_s = \Delta x$, whilst the volume is $\Delta\Omega = \Delta x \Delta y$. For a three-dimensional Cartesian mesh the cell face areas are $A_e = A_w = \Delta y \Delta z$, $A_n = A_s = \Delta z \Delta x$ and $A_t = A_b = \Delta x \Delta y$ and the cell volume is $\Delta\Omega = \Delta x \Delta y \Delta z$.

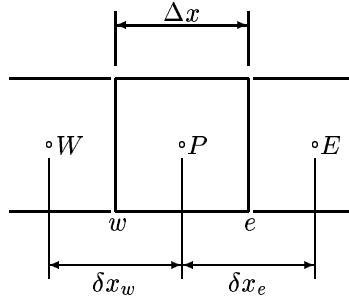


Figure 2.3: The geometry for a single volume in a one-dimensional mesh.

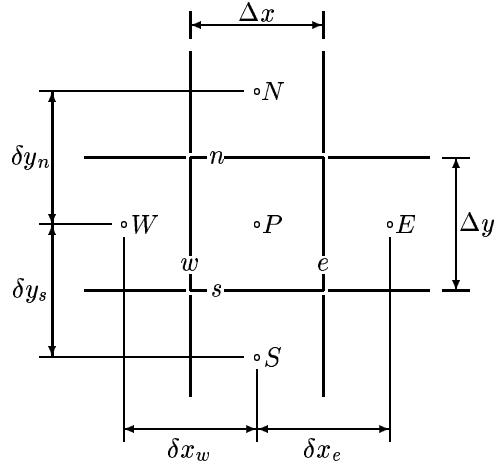


Figure 2.4: The geometry for a single volume in a two-dimensional Cartesian mesh.

For transient cases, discussed in Section 2.4, the values over the domain are prescribed for an initial time, and then the transport equation is solved for a series of time steps, at each time step the solution for the previous time step being known.

At time step n , the time is t and the solution for the n^{th} field is known. The PDE is solved for a time Δt in the future at time step $n + 1$.

2.2 Discretisation of the Steady Transport Equation

The steady transport equation is obtained by dropping the transient term from Equation (2.1) to give,

$$\nabla \cdot (\rho \mathbf{u} \phi) = \nabla \cdot (\Gamma \nabla \phi) + S, \quad (2.5)$$

with the term on the left hand side of the equation being the transport of ϕ due to advection, whilst the terms on the right hand side quantify the diffusion of ϕ and any sources of ϕ within the domain.

By explicitly writing out the terms of the ∇ operator, Equation (2.5) can be written in one, two and three dimensions as

$$\frac{\partial}{\partial x} (\rho u \phi) = \frac{\partial}{\partial x} \left(\Gamma \frac{\partial \phi}{\partial x} \right) + S, \quad (2.6)$$

$$\frac{\partial}{\partial x}(\rho u \phi) + \frac{\partial}{\partial y}(\rho v \phi) = \frac{\partial}{\partial x}\left(\Gamma \frac{\partial \phi}{\partial x}\right) + \frac{\partial}{\partial y}\left(\Gamma \frac{\partial \phi}{\partial y}\right) + S, \quad (2.7)$$

$$\frac{\partial}{\partial x}(\rho u \phi) + \frac{\partial}{\partial y}(\rho v \phi) + \frac{\partial}{\partial z}(\rho w \phi) = \frac{\partial}{\partial x}\left(\Gamma \frac{\partial \phi}{\partial x}\right) + \frac{\partial}{\partial y}\left(\Gamma \frac{\partial \phi}{\partial y}\right) + \frac{\partial}{\partial z}\left(\Gamma \frac{\partial \phi}{\partial z}\right) + S. \quad (2.8)$$

We will initially discretise the equations on a one-dimensional mesh, by taking Equation (2.6) and integrating in the x axis over the volume of the cell shown in Figure 2.3

$$\int_w^e \frac{\partial}{\partial x}(\rho u \phi) dx = \int_w^e \frac{\partial}{\partial x}\left(\Gamma \frac{\partial \phi}{\partial x}\right) dx + \int_w^e S dx. \quad (2.9)$$

By using S_P as a representative value for the source distribution within the volume, then the integral of the source term can be approximated by

$$\int_w^e S dx \approx S_P \Delta \Omega. \quad (2.10)$$

The second derivative term in the diffusion term can be approximated by two nested central difference approximations

$$\begin{aligned} \int_w^e \frac{\partial}{\partial x}\left(\Gamma \frac{\partial \phi}{\partial x}\right) dx &= \left(\Gamma \frac{\partial \phi}{\partial x}\right)_e - \left(\Gamma \frac{\partial \phi}{\partial x}\right)_w \\ &\approx \Gamma_e \frac{\phi_E - \phi_P}{\delta x_e} - \Gamma_w \frac{\phi_P - \phi_W}{\delta x_w}. \end{aligned} \quad (2.11)$$

Finally, assuming that velocities across the faces of the volume are known, then the advection term can be approximated by the use of a linear interpolation for the scalar value at the face,

$$\int_w^e \frac{\partial}{\partial x}(\rho u \phi) dx \approx (\rho u)_e \frac{\phi_E + \phi_P}{2} - (\rho u)_w \frac{\phi_P + \phi_W}{2}. \quad (2.12)$$

When describing different interpolations and finite difference approximations, such as have been made in Equations (2.10), (2.11) and (2.12), a commonly used description of the expected accuracy of the expression is its *order*, which is the leading truncation error of an equivalent Taylor series expansion, ie: the order of the next highest polynomial fit. Thus the interpolations used in Equations (2.11) and (2.12) are second order since the next highest polynomial above the linear fit used would be a quadratic or second order polynomial. Assuming that the first dropped term dominates the error of the approximation, then the error $\varepsilon \propto \Delta x^2$, and so as the mesh is refined $\Delta x \rightarrow 0$ then $\varepsilon \rightarrow 0$ quadratically.

Other commonly used interpolations are the quadratic, which gives a third order truncation error with $\varepsilon \propto \Delta x^3$, and the first order approximation such as is used in the integration of the source term in Equation (2.10), where the interpolating function is constant and has $\varepsilon \propto \Delta x$. Since as $\Delta x \rightarrow 0$ the higher order error terms tend to zero at a faster rate than the lower order terms, and so the higher order interpolations are formally more accurate than their low order counterparts. However, on coarse meshes where the interpolated function isn't sufficiently fine to resolve a discontinuity, the high order schemes can cause spurious oscillations in the interpolations which can lead to large errors and problems with numerical stability.

Returning to the discretisation, using the finite approximations in Equations (2.10), (2.11) and (2.12) the integral of the steady transport equation over the volume Ω can be approximated by

$$(\rho u)_e \frac{\phi_E + \phi_P}{2} - (\rho u)_w \frac{\phi_P + \phi_W}{2} = \Gamma_e \frac{\phi_E - \phi_P}{\delta x_e} - \Gamma_w \frac{\phi_P - \phi_W}{\delta x_w} + S_P \Delta \Omega, \quad (2.13)$$

which can be factorised to give

$$\begin{aligned} & \left(\frac{\Gamma_e}{\delta x_e} + \frac{\Gamma_w}{\delta x_w} + \frac{(\rho u)_e}{2} - \frac{(\rho u)_w}{2} \right) \phi_P \\ & - \left(\frac{\Gamma_e}{\delta x_e} - \frac{(\rho u)_e}{2} \right) \phi_E - \left(\frac{\Gamma_w}{\delta x_w} + \frac{(\rho u)_w}{2} \right) \phi_W = S_P \Delta \Omega. \end{aligned} \quad (2.14)$$

Using the compass notation introduced in Section 2.1, this can be rewritten as

$$a_P \phi_P + a_E \phi_E + a_W \phi_W = c, \quad (2.15)$$

with

$$\begin{aligned} a_E &= - \left(\frac{\Gamma_e}{\delta x_e} - \frac{(\rho u)_e}{2} \right), \\ a_W &= - \left(\frac{\Gamma_w}{\delta x_w} + \frac{(\rho u)_w}{2} \right), \\ a_P &= \frac{\Gamma_e}{\delta x_e} + \frac{(\rho u)_e}{2} + \frac{\Gamma_w}{\delta x_w} - \frac{(\rho u)_w}{2} \\ &= -(a_E + a_W) + (\rho u)_e - (\rho u)_w, \\ c &= S \Delta \Omega. \end{aligned} \quad (2.16)$$

The diffusive and mass fluxes at a face_f are defined as,

$$\begin{aligned} d_f &= A_f \frac{\Gamma_f}{\delta x_f}, \\ m_f &= A_f (\rho u)_f, \end{aligned} \quad (2.17)$$

with d_f being the diffusion flux across face_f, m_f the mass flux through the face, and A_f the area of the face, with _f being one of _e, _w, _n, _s, _t and _b, representing the East, West, North, South, Top and Bottom faces respectively. Note that the diffusion flux will always be positive (or zero) whilst the mass flux can take on positive and negative values. By using these mass and diffusion fluxes, the Equations (2.16) can be rewritten as

$$\begin{aligned} a_E &= -d_e + \frac{1}{2}m_e, \\ a_W &= -d_w - \frac{1}{2}m_w, \\ a_P &= d_e + \frac{1}{2}m_e + d_w - \frac{1}{2}m_w \\ &= -(a_E + a_W) + (m_e - m_w), \\ c &= S_P \Delta \Omega. \end{aligned} \quad (2.18)$$

The a_P term is the sum of the neighbouring equation coefficients with the addition of the $(m_e - m_w)$ term, which is the net loss of mass from the cell (and which should be zero for an incompressible flow).

By discretising all the points in the domain one obtains a system of linear equations,

$$\mathbf{A} \Phi = \mathbf{c}, \quad (2.19)$$

where each element of the vector Φ is the scalar ϕ_i of the i^{th} volume of the domain, each element of the vector \mathbf{c} is the source term c_i for that volume, and the matrix \mathbf{A} contains the factors on the left hand side of Equation (2.15). The matrix \mathbf{A} is sparse, and for this one-dimensional discretisation has only three non-zero diagonals, which are the diagonal and the immediate sub- and super-diagonals. This tridiagonal matrix system can be easily solved using the Thomas Tridiagonal linear solver which is described in Section 3.1.2.

For the two dimensional transport equation, given in Equation (2.7), the discretisation involves an integration over the volume in both the x and the y axis,

$$\begin{aligned} & \int_{s}^{n} \int_{w}^{e} \frac{\partial}{\partial x} (\rho u \phi) dx dy + \int_{s}^{n} \int_{w}^{e} \frac{\partial}{\partial y} (\rho v \phi) dx dy \\ &= \int_{s}^{n} \int_{w}^{e} \frac{\partial}{\partial x} \left(\Gamma \frac{\partial \phi}{\partial x} \right) dx dy + \int_{s}^{n} \int_{w}^{e} \frac{\partial}{\partial y} \left(\Gamma \frac{\partial \phi}{\partial y} \right) dx dy + \int_{s}^{n} \int_{w}^{e} S dx dy. \end{aligned} \quad (2.20)$$

Using the same integral approximations as for the one-dimensional transport equation (but noting the extra integration in the y axis) gives

$$\begin{aligned} & (\rho u)_e \Delta y \frac{\phi_E + \phi_P}{2} - (\rho u)_w \Delta y \frac{\phi_P + \phi_W}{2} \\ &+ (\rho v)_n \Delta x \frac{\phi_N + \phi_P}{2} - (\rho v)_s \Delta x \frac{\phi_P + \phi_S}{2} \\ &= \Gamma_e \Delta y \frac{\phi_E - \phi_P}{\delta x_e} - \Gamma_w \Delta y \frac{\phi_P - \phi_W}{\delta x_w} \\ &+ \Gamma_n \Delta x \frac{\phi_N - \phi_P}{\delta x_n} - \Gamma_s \Delta x \frac{\phi_P - \phi_S}{\delta x_s} + S_P \Delta \Omega. \end{aligned} \quad (2.21)$$

This can be factorised to give

$$a_P \phi_P + a_E \phi_E + a_W \phi_W + a_N \phi_N + a_S \phi_S = c, \quad (2.22)$$

where

$$\begin{aligned} a_E &= -d_e + \frac{1}{2}m_e, \\ a_W &= -d_w - \frac{1}{2}m_w, \\ a_N &= -d_n + \frac{1}{2}m_n, \\ a_S &= -d_s - \frac{1}{2}m_s, \\ a_P &= d_e + \frac{1}{2}m_e + d_w - \frac{1}{2}m_w + d_n + \frac{1}{2}m_n + d_s - \frac{1}{2}m_s \\ &= -(a_E + a_W + a_N + a_S) + (m_e - m_w + m_n - m_s), \\ c &= S_P \Delta \Omega. \end{aligned} \quad (2.23)$$

Finally, the finite volume discretisation of the three-dimensional steady transport equation, given in Equation (2.8), is found by integrating the equation in the x , y and z axis,

$$\begin{aligned} & \int_b^t \int_s^n \int_w^e \frac{\partial}{\partial x} (\rho u \phi) dx dy dz + \int_b^t \int_s^n \int_w^e \frac{\partial}{\partial y} (\rho v \phi) dx dy dz + \int_b^t \int_s^n \int_w^e \frac{\partial}{\partial z} (\rho w \phi) dx dy dz \\ &= \int_b^t \int_s^n \int_w^e \frac{\partial}{\partial x} \left(\Gamma \frac{\partial \phi}{\partial x} \right) dx dy dz + \int_b^t \int_s^n \int_w^e \frac{\partial}{\partial y} \left(\Gamma \frac{\partial \phi}{\partial y} \right) dx dy dz \\ &+ \int_b^t \int_s^n \int_w^e \frac{\partial}{\partial z} \left(\Gamma \frac{\partial \phi}{\partial z} \right) dx dy dz + \int_b^t \int_s^n \int_w^e S dx dy dz. \end{aligned} \quad (2.24)$$

As before the derivatives and face values can be approximated with central differences and linear interpolation, and the resulting equations factorised to give,

$$a_P \phi_P + a_E \phi_E + a_W \phi_W + a_N \phi_N + a_S \phi_S + a_T \phi_T + a_B \phi_B = c, \quad (2.25)$$

where

$$\begin{aligned}
 a_E &= -d_e + \frac{1}{2}m_e, \\
 a_W &= -d_w - \frac{1}{2}m_w, \\
 a_N &= -d_n + \frac{1}{2}m_n, \\
 a_S &= -d_s - \frac{1}{2}m_s, \\
 a_T &= -d_t + \frac{1}{2}m_t, \\
 a_B &= -d_b - \frac{1}{2}m_b, \\
 a_P &= d_e + \frac{1}{2}m_e + d_w - \frac{1}{2}m_w + d_n + \frac{1}{2}m_n \\
 &\quad + d_s - \frac{1}{2}m_s + d_t + \frac{1}{2}m_t + d_b - \frac{1}{2}m_b \\
 &= -(a_E + a_W + a_N + a_S + a_T + a_B) \\
 &\quad + (m_e - m_w + m_n - m_s + m_t - m_b), \\
 c &= S_P \Delta\Omega.
 \end{aligned} \tag{2.26}$$

The two- and three-dimensional discretisations discussed are just a pair (or trio) of orthogonal one-dimensional discretisations carried out along each axis of the space being covered. This is borne out by an examination of the one-dimensional system in Equation (2.18) and comparing it with the two- and three-dimensional systems in Equations (2.23) and (2.26). This is true for the other discretisations covered in this chapter, and so for reasons of economy only one-dimensional discretisations will be provided for the rest of the chapter, with the understanding that two- and three-dimensional versions are easily derived by adding together the orthogonal one-dimensional discretisations that are given.

The solution of the linear systems resulting from the finite volume discretisation of the two- and three-dimensional transport equations is far more problematical than is the case with the system resulting from the discretisation of the one-dimensional transport equation. There we saw that the linear system $\mathbf{A}\Phi = \mathbf{c}$ had a matrix \mathbf{A} with a compact tridiagonal structure that allowed for an efficient solution to the system. For the two- and three-dimensional systems the matrix has five and seven non-zero bands respectively, with some of the off-diagonal bands being located far from the diagonal of the system. Direct solvers become inefficient for such systems and so iterative methods are usually employed. In either case the multidimensional systems require a much greater computational effort per mesh point than does the one-dimensional system.

2.2.1 Numerical Stability of the Equations Resulting from a Finite Volume Discretisation of the Steady Transport Equation

The finite volume discretisation described in the previous section assumed that the mass flux across the face, $m_f = (\rho u)_f$, was known, and that a linear interpolation could be used to approximate the value of the scalar ϕ at the face. From this the advective flux of ϕ across the eastern face of the cell is given as

$$A_e (\rho u \phi)_e \approx A_e (\rho u)_e \frac{\phi_E + \phi_P}{2}. \tag{2.27}$$

This linear interpolation is commonly referred to as the Central Differencing scheme due to its similarity to the central differencing scheme used in finite difference approximations of the transport equation. Whilst this approximation is formally second order accurate, it has some numerical limitations. For transient problems (discussed below in Section 2.4) the system may become unstable with time, whilst for the steady transport equation it may cause unphysical short wavelength oscillations in the solution which in turn can cause problems in nonlinear numerical schemes.

In addition the equations resulting from the finite volume discretisations are normally solved using iterative methods. For many iterative linear solvers, a requirement for convergence is that the system

of equations must exhibit diagonal dominance, that is

$$|a_{i,i}| \geq \sum_{\substack{j=1 \\ j \neq i}}^{j=N} |a_{i,j}|, \quad (2.28)$$

which for a system of equations resulting from the finite volume discretisations given above is

$$|a_P| \geq \sum_{nb} |a_{nb}|, \quad (2.29)$$

where a_{nb} are the neighbouring nodes, a_E, a_W, a_N, a_S, a_T and a_B . Defining a cell based Péclet number as

$$\text{Pe} = \frac{U \Delta x}{\Gamma}, \quad (2.30)$$

then it can be demonstrated that the central difference scheme is diagonal dominant only when the cell Péclet number is less than or equal to 2. For cases where the cell Péclet number is greater than 2 (which for the momentum equations is when the cell Reynolds number is greater than 2) equations resulting from a central difference approximation loose their diagonal dominance and may not be solvable with an iterative linear solver.

To prevent these problems arising a number of alternative interpolation schemes have been proposed over the years. The earliest scheme was proposed by Courant, Isaacson and Rees[27] who observed the instability in a finite difference approximation to the transient transport equation, and who proposed the first order upwind scheme to solve the problem. This scheme has commonly been used, and for one-dimensional problems it can be shown to be reasonably accurate when the flow is aligned with the mesh[161, 134, 122]. However for multidimensional problems the first order schemes give an approximation that is over-diffusive, the excess or numerical diffusivity being approximated for a two dimensional flow on a Cartesian mesh by de Vahl Davis and Mallinson[33, 35] as

$$\Gamma_{\text{num}} = \frac{\rho U \Delta x \Delta y \sin 2\theta}{4(\Delta y \sin^3 \theta + \Delta x \cos^3 \theta)}. \quad (2.31)$$

This numerical diffusion can modify solutions to be not only quantitatively but qualitatively different from the correct solution. Since its discovery there has been a shift to using higher order schemes, with some journals having a policy of not publishing work that is calculated using first order methods[94]². To overcome the stability problems with central differencing, high order schemes that use some form of upwinding have been used, whilst the problem of oscillatory solutions has more recently been addressed with the use of flux limiter methods that aim to provide monotonic solutions in regions of high gradients whilst preserving the accuracy of the higher order methods.

One method to force diagonal dominance in otherwise non-dominant systems is by use of the method of deferred correction by Kholsa and Rubin[78]. If the system is repeatedly solved by some linear solver, with the solution after the n^{th} call to the solver being Φ^n , then the system $\mathbf{A}\Phi = \mathbf{c}$ can be recast as

$$\mathbf{A}_\ell \Phi^n = \mathbf{c} + (\mathbf{A}_\ell \Phi^{n-1} - \mathbf{A}_h \Phi^{n-1}), \quad (2.32)$$

where the system \mathbf{A}_ℓ is the system of equations resulting from some stable low-order scheme, and \mathbf{A}_h is the unstable high-order scheme that is being solved. The system on the left hand side of Equation (2.32) is diagonally dominant and so amenable to solution with an iterative linear solver, whilst the difference between the low- and high-order approximations is added as a source term (and so would be absorbed into the \mathbf{c} term in Equation (2.25)). As the system (2.32) is repeatedly solved the solutions Φ^n and Φ^{n-1} hopefully converge to a solution that is consistent with the higher order discretisation. The method has been used by a number of authors, a good description of the stability requirements of such a scheme being given by Hayase, Humphrey and Greif[58].

In the following sections a selection of first, second and third order schemes will be derived, with their comparative accuracy being tested later in Section 2.5.

²A related paper by Gresho titled “Don’t Suppress the Wiggles—They’re Telling You Something!”[53] is not about state censorship of kiddies rock bands, but is instead an analysis of the dangers of using first order schemes in finite element solvers.

2.2.2 The First Order Differencing Schemes

The first order differencing schemes are the oldest differencing schemes specifically designed to fix the stability problems of the discrete advection equation. They are simple, stable, easily implemented, and provide a smooth solution. Unfortunately this is due to their being excessively diffusive for problems with moderate to high Reynolds or Péclet numbers, resulting in solutions that are both quantitatively and qualitatively different from the correct solution.

Whilst these schemes are no longer commonly used they are included for historical interest. In addition they form the basis of some deferred correction forms of higher order methods. The methods discussed in this section are the First Order Upwind scheme, and the Hybrid, Exponential and Power schemes, a summary of which are to be found in Patankar[122]. A more modern and critical discussion of the methods and their limitations is given by Leonard and Drummond[94].

The First Order Upwind Scheme (FOU)

The simplest first order method is the First Order Upwind scheme, first proposed by Courant, Isaacson and Rees[27] and independently rediscovered by Lelevier[141], Gentry, Martin and Daly[48], Torrance[172] and Runchal and Wolfshtein[144].

The method was originally developed from a comparison with boundary layer flow, where the transport equations are parabolic, with scalars being transported downstream by the flow, and so the value of a scalar at a face can be approximated by it's value at the first node upstream of the face. Thus for the east face of a cell

$$\phi_e \approx \begin{cases} \phi_P & : m_e > 0 \\ \phi_E & : m_e < 0, \end{cases} \quad (2.33)$$

where m_e is the mass flux through the east face of the cell, which is given in Equation (2.17).

Using the $\llbracket \cdot \rrbracket$ operator of Patankar[122], where $\llbracket x, y \rrbracket$ is the maximum of x or y (ie: it is equivalent to the Fortran MAX intrinsic function), then the advection term in Equation (2.9) can be approximated by

$$\int_w^e \frac{\partial}{\partial x} (\rho u \phi) dx \approx \llbracket m_e, 0 \rrbracket \phi_P - \llbracket -m_e, 0 \rrbracket \phi_E - \llbracket m_w, 0 \rrbracket \phi_W + \llbracket -m_w, 0 \rrbracket \phi_P. \quad (2.34)$$

Using this approximation, together with the approximations given in Equations (2.11) and (2.10) for the diffusive and source terms, then the one-dimensional transport equation is discretised as,

$$\begin{aligned} \llbracket m_e, 0 \rrbracket \phi_P - \llbracket -m_e, 0 \rrbracket \phi_E - \llbracket m_w, 0 \rrbracket \phi_W + \llbracket -m_w, 0 \rrbracket \phi_P \\ = \Gamma_e \frac{\phi_E - \phi_P}{\delta x_e} - \Gamma_w \frac{\phi_P - \phi_W}{\delta x_w} + S_P \Delta \Omega, \end{aligned} \quad (2.35)$$

which can be factorised to give

$$\begin{aligned} \left(\frac{\Gamma_e}{\delta x_e} + \frac{\Gamma_w}{\delta x_w} + \llbracket m_e, 0 \rrbracket + \llbracket -m_w, 0 \rrbracket \right) \phi_P \\ - \left(\frac{\Gamma_e}{\delta x_e} + \llbracket -m_e, 0 \rrbracket \right) \phi_E - \left(\frac{\Gamma_w}{\delta x_w} + \llbracket m_w, 0 \rrbracket \right) \phi_W = S_P \Delta \Omega. \end{aligned} \quad (2.36)$$

This may be written as

$$a_P \phi_P + a_E \phi_E + a_W \phi_W = c, \quad (2.37)$$

where

$$\begin{aligned} a_E &= -d_e - \llbracket -m_e, 0 \rrbracket, \\ a_W &= -d_w - \llbracket m_w, 0 \rrbracket, \\ a_P &= d_e + \llbracket m_e, 0 \rrbracket + d_w + \llbracket -m_w, 0 \rrbracket \\ &= -(a_E + a_W) + (m_e - m_w), \\ c &= S_P \Delta \Omega. \end{aligned} \quad (2.38)$$

Similar expressions may be developed for discretisations in two and three dimensions.

The Hybrid Scheme

In order to combine the accuracy of Central Differencing with the stability of the First Order Upwind scheme, a hybrid of the two schemes was suggested by Spalding[161]. As was mentioned in Section 2.2.1 the central difference scheme is only unstable for cases where the cell based Péclet number is greater than 2. Thus Spalding used a scheme where the central difference scheme was used for faces with a cell Péclet number less than 2, with a switch being made to upwind differencing for faces with a cell Péclet number greater than 2.

Using this combination of first and second order differencing the hybrid scheme approximates the value of the scalar at a cells east face as

$$\phi_e \approx \begin{cases} \phi_P & : \quad 2 < \text{Pe}_e \\ \frac{\phi_P + \phi_e}{2} & : \quad -2 < \text{Pe}_e < 2 \\ \phi_E & : \quad \text{Pe}_e < -2, \end{cases} \quad (2.39)$$

which when substituted into a finite volume approximation to the one-dimensional transport equations and factorised leads to the system Equation (2.37), the equations coefficients being,

$$\begin{aligned} a_E &= -\llbracket -m_e, d_e - m_e/2, 0 \rrbracket, \\ a_W &= -\llbracket m_w, d_w + m_w/2, 0 \rrbracket, \\ a_P &= \llbracket m_e, d_e - m_e/2, 0 \rrbracket + \llbracket -m_w, d_w + m_w/2, 0 \rrbracket \\ &= -(a_E + a_W) + (m_e - m_w), \\ c &= S_P \Delta \Omega. \end{aligned} \quad (2.40)$$

This scheme improves the accuracy of the First Order Upwind scheme when being applied to one-dimensional flows. For multidimensional flows however it still exhibits the numerical diffusion of the FOU scheme unless the flow has such a low Péclet number that the Central differencing approximation is being used, in which case the Central differencing scheme could have been used anyhow. Nevertheless the scheme is still sometimes used, and is the default option in many commercial CFD codes, Flow3D being an example[72].

The Exponential and Power Schemes

Analytic solutions are available for the one-dimensional transport equation. By using these methods a discretisation scheme was developed by Allen and Southwell[1] and subsequently redeveloped by Spalding[161] and Raithby and Torrence[136]. For one-dimensional flows the scheme can be shown to give exact solutions to the flow, but as for all the first order schemes it exhibits numerical diffusion in multidimensional flows. The scheme uses an exponential approximation to the scalar field and is called the exponential scheme, although it is sometimes referred to as the exact scheme in some early papers.

For the east face of a cell the scalar is interpolated as

$$\phi_e \approx \phi_E \frac{e^{Pe_e/2} - 1}{e^{Pe_e} - 1}, \quad (2.41)$$

giving the discretised equations as

$$\begin{aligned} a_E &= m_e / (1 - e^{Pe_e}), \\ a_W &= m_w (e^{Pe_w}) / (1 - e^{Pe_w}), \\ a_P &= -(a_E + a_W) + (m_e - m_w), \\ c &= S_P \Delta \Omega. \end{aligned} \quad (2.42)$$

Since the exponential scheme requires the evaluation of many exponential functions, which is a computationally expensive task, more efficient versions of the scheme were developed by Patankar[122] and Raithby and Schneider[135] using a polynomial approximation to the exponential expressions. Patankar's scheme gives a discretisation of

$$\begin{aligned} a_E &= -d_e [(1 - 0.1|Pe_e|)^5, 0] - [-m_e, 0], \\ a_E &= -d_w [(1 - 0.1|Pe_w|)^5, 0] - [-m_w, 0], \\ a_P &= -(a_E + a_W) + (m_e - m_w), \\ c &= S_P \Delta \Omega. \end{aligned} \quad (2.43)$$

As with all the first order scheme, these schemes are only accurate for one-dimensional flow, and exhibit numerical diffusion in multidimensional flow.

2.2.3 Second Order Differencing Schemes

The second order schemes are less commonly used than their first and third order counterparts. The simplest scheme is the previously discussed central difference method, which is not commonly used due to its problems with numerical stability. To overcome these limitations other second order schemes have been developed that use upwinding, with the value of a scalar at the cell face being extrapolated from two upstream points rather than using just one point as is the case with first order upwinding. These methods do not suffer the stability problems inherent in central differencing, but still exhibit the problem of oscillatory solutions, with the solved field often being wiggly. To prevent this oscillatory behaviour flux limiter schemes have been developed which preserve the higher order accuracy of such schemes whilst retaining monotonic solutions in regions of high gradients. The second order flux limiter method discussed below is one developed by Roe[142] and Sweby[165].

Central Difference Scheme (CDS)

The central differencing scheme was derived above in Section 2.2 but for the sake of completeness it is repeated here alongside the other second order methods. For the one-dimensional cell given in Figure 2.3, the value of the scalar ϕ at the right or eastern face of the cell is estimated using a linear interpolation between the two neighbouring values, so

$$\phi_e \approx \frac{\phi_P + \phi_E}{2}, \quad (2.44)$$

assuming a regular mesh (ie: Δx is constant).

Substituting into the transport equation, factorising and rewriting as a linear equation, the central difference approximation for the one-dimensional transport equation becomes

$$a_P \phi_P + a_E \phi_E + a_W \phi_W = c, \quad (2.45)$$

where

$$\begin{aligned} a_E &= -d_e + \frac{1}{2}m_e, \\ a_W &= -d_w - \frac{1}{2}m_w, \\ a_P &= d_e + \frac{1}{2}m_e + d_w - \frac{1}{2}m_w \\ &= -(a_E + a_W) + (m_e - m_w), \\ c &= S_P \Delta\Omega. \end{aligned} \quad (2.46)$$

This discretisation scheme can cause oscillatory solutions, and for cell Péclet numbers greater than 2 the equations loose their diagonal dominance which can cause the failure of the iterative linear solvers used to solve the systems. However, by using the method of deferred correction given in Equation (2.32) central differencing can be made diagonally dominant. For such a system Equation (2.45) is modified to

$$a_P \phi_P^n + a_E \phi_E^n + a_W \phi_W^n = c + b_P \phi_P^{n-1} + b_E \phi_E^{n-1} + b_W \phi_W^{n-1}, \quad (2.47)$$

where

$$\begin{aligned} a_E &= -d_e - [-m_e, 0], \\ a_W &= -d_w - [m_w, 0], \\ a_P &= -(a_E + a_W) + (m_e - m_w), \\ c &= S_P \Delta\Omega, \\ b_E &= -[-m_e, 0] - \frac{1}{2}m_e, \\ b_W &= -[m_w, 0] + \frac{1}{2}m_w, \\ b_P &= [-m_e, 0] + [m_w, 0] + \frac{1}{2}m_e - \frac{1}{2}m_w. \end{aligned} \quad (2.48)$$

The system is solved using the first order upwind scheme, with the source term being modified to contain the difference between the first order scheme and central differencing, evaluated using the solution from the previous iteration (ie: at iteration $n - 1$). Such a scheme has been described by Hayase et al[58].

Second Order Upwind Scheme (SOU)

The second order upwind schemes were described for finite difference discretisations by Warming and Beam[176] and Hodge, Stone and Miller[65]. The first finite volume implementations were by Tamamidis and Assanis[168], as an explicit transient scheme, and by Thompson and Wilkes[170] who implemented a steady state implicit version.

For the second order upwind scheme the value of a scalar at a face is approximated by a second order upwind extrapolation. For the east face of a finite volume mesh the scalar is estimated as

$$\phi_e \approx \begin{cases} \frac{3}{2}\phi_P - \frac{1}{2}\phi_W & : m_e > 0 \\ \frac{3}{2}\phi_E - \frac{1}{2}\phi_{EE} & : m_e < 0, \end{cases} \quad (2.49)$$

assuming a regular mesh with constant Δx . Discretising yields a system of equations

$$a_P \phi_P + a_E \phi_E + a_W \phi_W + a_{EE} \phi_{EE} + a_{WW} \phi_{WW} = c, \quad (2.50)$$

where

$$\begin{aligned} a_E &= -d_e - \frac{3}{2}[-m_e, 0] - \frac{1}{2}[-m_w, 0], \\ a_W &= -d_w - \frac{3}{2}[m_w, 0] - \frac{1}{2}[m_e, 0], \\ a_{EE} &= \frac{1}{2}[-m_e, 0], \\ a_{WW} &= \frac{1}{2}[m_w, 0], \\ a_P &= -(a_E + a_W + a_{EE} + a_{WW}) + (m_e - m_w), \\ c &= S_P \Delta\Omega. \end{aligned} \quad (2.51)$$

The use of more than one upwind point expands the computational molecule from the compact three point scheme (in one dimension) to a wider five point molecule. In two dimensions the molecule is expanded from five to nine points, and in three dimensions from seven to thirteen points (illustrated in Figure 2.2). To reduced the number of non-zeros in linear system the system of equations is commonly reduced using the deferred correction method, with the linear system having the same number of non-zero diagonals as the central difference scheme, and with the other terms being absorbed into the source term. This allows linear solvers developed for the central and first-order upwind schemes to be used on equations resulting from a larger computational molecule.

Such a scheme has been implemented by using a first order upwind scheme with a second order upwind correction. Using the deferred correction method the second order upwind discretisation given in Equations (2.50) and (2.51) is transformed into

$$\begin{aligned} a_P \phi_P^n + a_E \phi_E^n + a_W \phi_W^n = \\ c + b_P \phi_P^{n-1} + b_{EE} \phi_{EE}^{n-1} + b_E \phi_E^{n-1} + b_W \phi_W^{n-1} + b_{WW} \phi_{WW}^{n-1}, \end{aligned} \quad (2.52)$$

where

$$\begin{aligned} a_E &= -d_e - \llbracket -m_e, 0 \rrbracket, \\ a_W &= -d_w - \llbracket m_w, 0 \rrbracket, \\ a_P &= -(a_E + a_W) + (m_e - m_w), \\ c &= S_P \Delta \Omega, \\ b_E &= \frac{1}{2} (\llbracket -m_e, 0 \rrbracket + \llbracket -m_w, 0 \rrbracket), \\ b_W &= \frac{1}{2} (\llbracket m_e, 0 \rrbracket + \llbracket m_w, 0 \rrbracket), \\ b_{EE} &= -\frac{1}{2} (\llbracket -m_e, 0 \rrbracket), \\ b_{WW} &= -\frac{1}{2} (\llbracket m_w, 0 \rrbracket), \\ b_P &= -\frac{1}{2} (\llbracket m_e, 0 \rrbracket + \llbracket -m_w, 0 \rrbracket). \end{aligned} \quad (2.53)$$

Monotonic Second Order Upwind Differencing Scheme (MSOU)

The Monotonic Second Order Upwind scheme was developed by Sweby[165] from the work of Roe[142]. A flux limiter is added to the SOU differencing scheme to prevent the formation of oscillations in the scalar field. The method is implemented as a second order correction to first order upwind differencing (FOU) and so is numerically stable.

The approximation of the value of a scalar at the east face of a volume is given by

$$\phi_e \approx \begin{cases} \phi_P + \frac{1}{2} \varphi_P^- (\phi_P - \phi_W) & : m_e > 0 \\ \phi_E - \frac{1}{2} \varphi_E^+ (\phi_{EE} - \phi_E) & : m_e < 0, \end{cases} \quad (2.54)$$

where the functions φ^+ and φ^- are defined at the point P by

$$\begin{aligned} \varphi_P^- &= \llbracket 0, \langle\langle 2r_P^-, 1 \rangle\rangle, \langle\langle r_P^-, 2 \rangle\rangle \rrbracket, \\ \varphi_P^+ &= \llbracket 0, \langle\langle 2r_P^+, 1 \rangle\rangle, \langle\langle r_P^+, 2 \rangle\rangle \rrbracket, \\ r_P^- &= \frac{\phi_E - \phi_P}{\phi_P - \phi_W}, \\ r_P^+ &= \frac{\phi_P - \phi_E}{\phi_E - \phi_{EE}} \\ &= \frac{1}{r_E}, \end{aligned} \quad (2.55)$$

and the $\langle\langle\rangle\rangle$ operator is defined such that $\langle\langle x, y \rangle\rangle$ is the minimum of x or y (ie: it is equivalent to the Fortran MIN intrinsic function). The r^- and r^+ functions are the ratio of consecutive gradients, and for cases where the denominator equals zero (and they become infinite) they should be set to an arbitrary scalar value that is greater than 2 to prevent floating point overflow. Also note that r^+ at point P is equal to $1/r^-$ at point E . The φ function as defined by Sweby is equivalent to the “superbee” flux limiter of Roe.

By substituting the interpolations in Equation (2.55) into the discretised advection equation and factorising the following system is generated,

$$\begin{aligned} a_P \phi_P^n + a_E \phi_E^n + a_W \phi_W^n = \\ c + b_P \phi_P^{n-1} + b_{EE} \phi_{EE}^{n-1} + b_E \phi_E^{n-1} + b_W \phi_W^{n-1} + b_{WW} \phi_{WW}^{n-1}, \end{aligned} \quad (2.56)$$

where

$$\begin{aligned} a_E &= -d_e - [-m_e, 0], \\ a_W &= -d_w - [m_w, 0], \\ a_P &= -(a_E + a_W) + (m_e - m_w), \\ c &= S_P \Delta \Omega, \\ b_E &= \frac{1}{2} (\llbracket -m_e, 0 \rrbracket \varphi_E^+ + \llbracket -m_w, 0 \rrbracket \varphi_P^+), \\ b_W &= \frac{1}{2} (\llbracket m_e, 0 \rrbracket \varphi_P^- + \llbracket m_w, 0 \rrbracket \varphi_W^-), \\ b_{EE} &= -\frac{1}{2} (\llbracket -m_e, 0 \rrbracket \varphi_E^+), \\ b_{WW} &= -\frac{1}{2} (\llbracket m_w, 0 \rrbracket \varphi_W^-), \\ b_P &= -\frac{1}{2} (\llbracket m_e, 0 \rrbracket \varphi_P^- + \llbracket -m_w, 0 \rrbracket \varphi_P^+). \end{aligned} \quad (2.57)$$

Since this is a deferred correction scheme, the r^+ and r^- gradient ratios are evaluated using the values of ϕ from the $n - 1^{th}$ iteration. A comparison with Equation (2.53) reveals that the MSOU scheme is just the SOU scheme modified with the φ flux limiters.

The resulting differencing scheme is monotonic and does not exhibit high-frequency oscillations as often happens with higher order schemes.

2.2.4 Third Order Differencing Schemes

Third order upwinding schemes were first described by Leonard[88], who since his first paper has written many papers developing and promoting the use of higher order differencing schemes. Two third order scheme are discussed here, both by Leonard. The first is the regular third order upwind scheme that he named QUICK, and the second is a flux-limited third order upwind scheme.

Third Order Upwind differencing (QUICK)

The third order upwind scheme was first described by Leonard in 1979[88, 90]. The two schemes in his first paper, QUICK and QUICKEST, were both explicit finite volume methods, but other authors quickly applied them to implicit finite volume discretisations. In what has become typical Leonard style, he named the schemes with a tortuously derived but catchy acronym, QUICK standing for “Quadratic Upwind Interpolation for Convective Kinematics” whilst the QUICKEST scheme had “Estimated Streaming Terms” tagged on the end. Recent research has shown that the use of catchy acronyms ensures citations.

For a third order upwind scheme the value of a scalar at the east face of a volume is approximated by

$$\phi_e \approx \begin{cases} \frac{3}{8}\phi_E + \frac{3}{4}\phi_P - \frac{1}{8}\phi_W & : m_e > 0 \\ \frac{3}{8}\phi_P + \frac{3}{4}\phi_E - \frac{1}{8}\phi_{EE} & : m_e < 0, \end{cases} \quad (2.58)$$

assuming a regular mesh with constant Δx . Leonard cast this in the form of a central difference scheme with a third order correction,

$$\phi_e \approx \begin{cases} \frac{1}{2}(\phi_P + \phi_E) - \frac{1}{8}(\phi_W - 2\phi_P + \phi_E) & : m_e > 0 \\ \frac{1}{2}(\phi_P + \phi_E) - \frac{1}{8}(\phi_P - 2\phi_E + \phi_{EE}) & : m_e < 0. \end{cases} \quad (2.59)$$

As with the second order upwind schemes, third order upwinding increases the number of points in the computational molecule over that of the first order and central differencing schemes. In addition it can result in systems that are not diagonally dominant, and so it is normally recast into a deferred correction form. A number of forms of the deferred correction QUICK scheme have been published and Hayase, Humphrey and Greif[58] provide a unification of previous work, generalising the process of generating a deferred correction scheme and providing their own version of the third order upwind scheme which they show to be faster to converge than other schemes, whilst being unconditionally stable.

For a deferred correction form of interpolation the interpolated value of a scalar at a face is found from an interpolation of the surrounding values plus the addition of a source term from the previous estimate of the scalar's values. For the generic QUICK interpolation of a scalar ϕ at the east and west faces of a cell, Hayase et al cast the interpolating functions as

$$\phi_e \approx \begin{cases} a_1\phi_W + a_2\phi_P + a_3\phi_E + S_e^+ & : m_e > 0 \\ b_1\phi_W + b_2\phi_P + b_3\phi_E + S_e^- & : m_e < 0, \end{cases} \quad (2.60)$$

$$\phi_w \approx \begin{cases} b_1\phi_W + b_2\phi_P + b_3\phi_E + S_w^+ & : m_e > 0 \\ a_1\phi_W + a_2\phi_P + a_3\phi_E + S_w^- & : m_e < 0, \end{cases} \quad (2.61)$$

where the source terms S are defined as

$$\begin{aligned} S_e^+ &= (-\frac{1}{8} - a_1)\phi_W + (\frac{3}{4} - a_2)\phi_P + (\frac{3}{8} - a_3)\phi_E, \\ S_e^- &= -\frac{1}{8}\phi_{EE} + (\frac{3}{4} - b_1)\phi_E + (\frac{3}{8} - b_2)\phi_P + -b_3)\phi_W, \\ S_w^+ &= -\frac{1}{8}\phi_{WW} + (\frac{3}{4} - b_1)\phi_W + (\frac{3}{8} - b_2)\phi_P + -b_3)\phi_E, \\ S_w^- &= (-\frac{1}{8} - a_1)\phi_E + (\frac{3}{4} - a_2)\phi_P + (\frac{3}{8} - a_3)\phi_W. \end{aligned} \quad (2.62)$$

Early authors to convert the QUICK scheme into a deferred correction finite volume scheme were Leschziner[101], Han, Humphrey and Launder[56], Pollard and Siu[131] and Freitas[45]. Hayase et al apply these authors schemes to the generic interpolation method given above, and then generate their own scheme using a first order upwind scheme with a third order deferred correction, which is not only simple to implement, but which they show to have superior stability and convergence characteristics than the earlier schemes. A summary of the different schemes is given in Table 2.2.

Two versions of the QUICK differencing scheme are given here – the first is the original method of Leonard cast into an implicit finite volume form, using a central difference approximation modified with a third order correction. The second scheme is that of Hayase et al and consists of a first order upwind scheme with a third order correction.

Using the interpolation of Leonard given in Equation (2.58) the deferred correction form of the one-dimensional transport equation can be written as

$$\begin{aligned} a_P \phi_P^n + a_E \phi_E^n + a_W \phi_W^n &= \\ c + b_P \phi_P^{n-1} + b_{EE} \phi_{EE}^{n-1} + b_E \phi_E^{n-1} + b_W \phi_W^{n-1} + b_{WW} \phi_{WW}^{n-1}, \end{aligned} \quad (2.63)$$

Author	a_1	a_2	a_3	b_1	b_2	b_3
Leschziner[101]	$-\frac{1}{8}$	$\frac{6}{8}$	$\frac{3}{8}$	$\frac{6}{8}$	$\frac{3}{8}$	0
Han et al[56]	0	$\frac{6}{8}$	$\frac{4}{8}$	$\frac{4}{8}$	$\frac{3}{8}$	0
Pollard and Siu[131]	$-\frac{1}{8}$	$\frac{7}{8}$	$-\frac{6}{8}$	$\frac{6}{8}$	$-\frac{6}{8}$	0
Freitas[45]	$-\frac{1}{8}$	$\frac{6}{8}$	$-\frac{1}{8}$	$\frac{6}{8}$	$\frac{3}{8}$	0
Hayase[58]	0	1	0	0	1	0

Table 2.2: Deferred approximation schemes for the generic QUICK scheme given in Equations (2.60) to (2.62). After Hayase et al[58].

where

$$\begin{aligned}
 a_E &= -d_e + \frac{1}{2}m_e, \\
 a_W &= -d_w - \frac{1}{2}m_w, \\
 a_P &= -(a_E + a_W) + (m_e - m_w), \\
 c &= S_P \Delta\Omega, \\
 b_E &= \frac{1}{8}[-m_e, 0] + \frac{2}{8}[-m_e, 0] + \frac{1}{8}[-m_w, 0], \\
 b_W &= \frac{1}{8}[-m_w, 0] + \frac{2}{8}[m_w, 0] + \frac{1}{8}[m_e, 0], \\
 b_{EE} &= -\frac{1}{8}[-m_e, 0], \\
 b_{WW} &= -\frac{1}{8}[m_w, 0], \\
 b_P &= -(b_E + b_W + b_{EE} + b_{WW}).
 \end{aligned} \tag{2.64}$$

The deferred correction scheme of Hayase et al uses a first order upwind scheme with a third order correction. The equations take on the form of Equation (2.63), but with the equation coefficients being given by

$$\begin{aligned}
 a_E &= -d_e - [-m_e, 0], \\
 a_W &= -d_w - [m_w, 0], \\
 a_P &= -(a_E + a_W) + (m_e - m_w), \\
 c &= S_P \Delta\Omega, \\
 b_E &= -\frac{3}{8}[m_e, 0] - \frac{2}{8}[-m_e, 0] + \frac{1}{8}[-m_w, 0], \\
 b_W &= -\frac{3}{8}[-m_w, 0] - \frac{2}{8}[m_w, 0] + \frac{1}{8}[m_e, 0], \\
 b_{EE} &= -\frac{1}{8}[-m_e, 0], \\
 b_{WW} &= -\frac{1}{8}[m_w, 0], \\
 b_P &= -(b_E + b_W + b_{EE} + b_{WW}).
 \end{aligned} \tag{2.65}$$

Flux Limited Third Order Schemes

Whilst the deferred correction schemes improve the numerical stability of the linear systems generated by the QUICK differencing, one of the fundamental problems of high order differencing schemes remains, with the solution field often exhibiting spurious (ie: non-physical) oscillations. To overcome this problem Leonard has applied a number of flux-limiter schemes to the basic third order upwind biased method. The initial scheme, SHARP[91], was later developed into the ULTRA-SHARP scheme[100, 99] which is discussed here. Later efforts (both numerically and acronymically) lead to the ULTIMATE[92], ULTIMATE QUICKEST[93], UTOPIA[98], NIRVANA[96], ENIGMATIC[95], MACHO and COSMIC[97] schemes. Whilst only the ULTRA-SHARP scheme is given in this section, a summary of the (tortured) acronyms of the other schemes is given in Table 2.3³.

³One can only be thankful that he hasn't developed a scheme for "Explicit Non-oscillatory Extrapolation for Multidimensional Advection" although his writing suggests that he might want to apply such a name to the first order upwind scheme.

SHARP	Simple High-Accuracy Resolution Program	Leonard[91]
ULTIMATE	Universal Limiter for Transport Interpolation Modelling of the Advective Transport Equation	Leonard[92]
ULTIMATE-QUICKEST	Universal Limiter for Transport Interpolation Modelling of the Advective Transport Equation applied to Quadratic Upstream Interpolation for Convective Kinematics with Estimated Streaming Terms	Leonard[93]
ULTRA-SHARP	Universal Limiter for Tight Resolution and Accuracy in combination with the Simple High-Accuracy Resolution Program	Leonard and Mokhtari[100, 99]
UTOPIA	Uniformly Third Order Polynomial Interpolation Algorithm	Leonard, MacVean and Lock[98]
NIRVANA	Non-oscillatory Integrally Reconstructed Volume-Averaged Numerical Advection scheme	Leonard, Lock and MacVean[96]
ENIGMATIC	Extended Numerical Integration for Genuinely Multi-dimensional Advective Transport Insuring Conservation	Leonard, Lock and MacVean[95]
MACHO	Multidimensional Advective-Conservative Hybrid Operator	Leonard, Lock and MacVean[97]
COSMIC	Conservative Operator Splitting for Multidimensions with Internal Constancy	Leonard, Lock and MacVean[97]
QUICK	Quadratic Upstream Interpolation for Convective Kinematics	Leonard [88, 90]
QUICKEST	Quadratic Upstream Interpolation for Convective Kinematics with Estimated Streaming Terms	Leonard [88, 90]
AQUATIC	Adjusted Quadratic Upstream Algorithm for Transient Incompressible Convection	Leonard [89]
EXQUISITE	Exponential or Quadratic Upstream Interpolation for Solution of the Incompressible Transport Equation	Leonard [89]

Table 2.3: A summary of the names of the flux-limited and unlimited third order schemes of Leonard.

The ULTRA flux limiter is a bounds checking algorithm which can be applied to any higher order differencing scheme, placing upper and lower limit on the values that the interpolating polynomial can take. It is similar to the flux limiters of Roe that are the basis of the MSOU scheme described in Section 2.2.3.

For any high order interpolation on the east face of a cell, the ULTRA limited interpolation for that face is given by

$$\phi_e \approx \begin{cases} \langle\!\langle \phi_P, \phi'_e, \langle\!\langle \phi_P, \phi_E, \phi_W + 10(\phi_P - \phi_W) \rangle\!\rangle \rangle\!\rangle & : m_e > 0 \\ \langle\!\langle \phi_E, \phi'_e, \langle\!\langle \phi_E, \phi_P, \phi_{EE} + 10(\phi_E - \phi_{EE}) \rangle\!\rangle \rangle\!\rangle & : m_e < 0, \end{cases} \quad (2.66)$$

where ϕ'_e is the original higher order interpolation, and $\langle\!\langle \cdot \rangle\!\rangle$ is the median operator, defined so $\langle\!\langle x, y, z \rangle\!\rangle$ is the median value of x, y and z . This can be constructed from the standard Fortran MAX and MIN operators with

$$\langle\!\langle x, y, z \rangle\!\rangle = \text{MAX}(\text{MIN}(x, y), \text{MIN}(y, z), \text{MIN}(z, x)). \quad (2.67)$$

Another (more complicated) version of the operator is given by Huynh[67] who uses the Fortran MIN and SIGN functions.

$$\langle\!\langle x, y, z \rangle\!\rangle = x + (\text{SIGN}(0.5, y - x) + \text{SIGN}(0.5, z - x))\text{MIN}(\text{ABS}(y - x), \text{ABS}(z - x)). \quad (2.68)$$

The ULTRA-SHARP scheme (sometimes referred to by Leonard as ULTRA-QUICK) applies the ULTRA limiter to the QUICK interpolation scheme, where

$$\phi'_f \approx \begin{cases} \frac{1}{8}(3\phi_E + 6\phi_P - \phi_W) & : m_e > 0 \\ \frac{1}{8}(3\phi_P + 6\phi_E - \phi_{EE}) & : m_e < 0. \end{cases} \quad (2.69)$$

As with Hayase et al's version of the QUICK scheme and the MSOU flux limited second order upwind scheme, the ULTRA-SHARP scheme is implemented as a first order upwind scheme with a higher order correction. The system is generated as

$$a_P \phi_P^n + a_E \phi_E^n + a_W \phi_W^n = c, \quad (2.70)$$

where

$$\begin{aligned} a_E &= -d_e - [-m_e, 0], \\ a_W &= -d_w - [m_w, 0], \\ a_P &= -(a_E + a_W) + (m_e - m_w), \\ c &= S_P \Delta \Omega + (m_e \phi_e - m_w \phi_w)^{\text{ULTRA}} - (m_e \phi_e - m_w \phi_w)^{\text{FOU}} \\ &= S_P \Delta \Omega + ([m_e, 0] \phi_e^+ - [-m_e] \phi_e^- - [m_w, 0] \phi_w^+ + [-m_w] \phi_w^-) \\ &\quad + ([m_e, 0] \phi_E - [-m_e] \phi_E - [m_w, 0] \phi_W + [-m_w] \phi_W), \end{aligned} \quad (2.71)$$

with the ϕ_e^\pm functions being defined by

$$\begin{aligned} \phi_e^+ &= \langle\!\langle \phi_P, \frac{1}{8}(3\phi_E + 6\phi_P - \phi_W), \langle\!\langle \phi_P, \phi_E, \phi_W + 10(\phi_P - \phi_W) \rangle\!\rangle \rangle\!\rangle, \\ \phi_e^- &= \langle\!\langle \phi_E, \frac{1}{8}(3\phi_P + 6\phi_E - \phi_{EE}), \langle\!\langle \phi_E, \phi_P, \phi_{EE} + 10(\phi_E - \phi_{EE}) \rangle\!\rangle \rangle\!\rangle, \\ \phi_w^+ &= \langle\!\langle \phi_W, \frac{1}{8}(3\phi_P + 6\phi_W - \phi_{WW}), \langle\!\langle \phi_W, \phi_P, \phi_{WW} + 10(\phi_W - \phi_{WW}) \rangle\!\rangle \rangle\!\rangle, \\ \phi_w^- &= \langle\!\langle \phi_P, \frac{1}{8}(3\phi_W + 6\phi_P - \phi_E), \langle\!\langle \phi_P, \phi_W, \phi_E + 10(\phi_P - \phi_E) \rangle\!\rangle \rangle\!\rangle. \end{aligned} \quad (2.72)$$

In addition to being applied to the QUICK differencing scheme, in [99] Leonard also applies the ULTRA limiter to second, fifth and seventh order upwind schemes.

2.2.5 Summary of the Advective Discretisation Schemes

Whilst the discretisation of the diffusive terms in the transport equation provide no complications, the straightforward discretisation of the advective terms using a central difference approximation can cause numerical instability and oscillatory solutions. Innumerable alternative discretisations have been proposed over the years, most using some form of upwind biasing.

The oldest of the upwind methods are the first order schemes, first proposed by Courant. These methods are numerically stable and give smooth solutions, but unfortunately they are excessively diffusive for all but one-dimensional flow. The second and third order upwind biased schemes are more accurate than their first order counterparts, and when implemented as deferred correction methods they are numerically stable. However, they tend to generate oscillatory solutions. Recently the use of flux-limiters has been promoted as a means to remove the oscillatory behaviour of the higher order schemes, whilst retaining their greater accuracy compared with the first order methods.

Of the methods described the third order upwind scheme (or QUICK) is probably the most commonly used differencing scheme. However, the flux limited second and third order schemes are of interest in their ability to generate accurate non-oscillatory solutions. A comparative test of the various methods is undertaken in Section 2.5 at the end of this chapter.

2.3 Boundary Conditions for the Transport Equation

So far we have ignored the boundaries of the domain over which we are solving the PDE, which is an important omission since the conditions imposed at the edge of the domain (ie: the boundary conditions) largely define the the solution that is obtained. For the scalar field being considered there are two main types of boundary condition, the Dirichlet boundary condition where the value of the scalar is defined at the boundary

$$\phi = \phi_b, \quad (2.73)$$

and the Neumann boundary condition where the gradient of the scalar normal to the boundary is specified

$$\frac{\partial \phi}{\partial n} = \phi'_b. \quad (2.74)$$

Another less commonly encountered boundary condition is the periodic boundary, where the domain at one boundary maps onto the boundary at another part of the domain. For example, if the right and left faces of the one dimensional domain in Figure 2.1 were periodic, then the value of the scalar at the faces, at point 1 and point N , would be the same, and the $N + 1^{\text{th}}$ cell would correspond to cell number 2, whilst the cell 0 would correspond to cell $N - 1$. Fluid flowing out the face at the eastern face at N re-enters the domain at the western face at 1 and vis-versa. For a regular mesh CFD code, Periodic boundary conditions can be trivially implemented using the Fortran 90 CSHIFT intrinsic function, which wraps points from side of a domain to the other. However, their implementation for the Direct and Incomplete LU solvers is rather more problematic and so they will be ignored for now and we will concentrate on the Dirichlet and Neumann boundaries.

For Cartesian meshes the implementation of the Dirichlet and Neumann boundary conditions is quite simple. A boundary cell is placed outside the domain, with the boundary for the domain lying at the face between this external boundary cell and the first cell inside the domain. This addition of false boundary cells results in a mesh slightly modified from that shown in Figure 2.1, with it instead having the form shown in Figure 2.5. Note that the cells at the edge of the computational domain lie outside the domain being solved for. A detail of two cells at a boundary is shown in Figure 2.6

For the cells in Figure 2.6 which lie at the Eastern edge of the domain, the Eastern face of the cell P , labelled e , lies at the boundary, whilst the cell E lies outside the solution domain. Using the

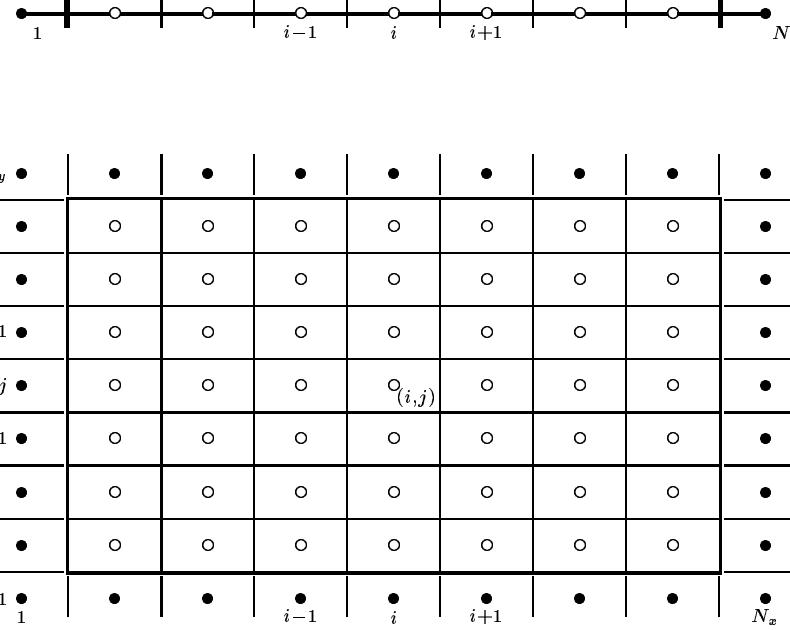


Figure 2.5: The discretisation of a 1D (top) and 2D (bottom) domain into Cartesian finite volumes, with external boundary volumes at the edges of the domain.

discretisation techniques described above a finite volume approximation for the transport at the cell P results in an equation of the form

$$a_E \phi_E + a_P \phi_P + a_W \phi_W = c. \quad (2.75)$$

For Dirichlet boundary conditions the value of ϕ at the domain boundary is provided and has the value ϕ_b . This boundary lies at the eastern face of cell P , and so $\phi_e = \phi_b$. Using a central difference approximation

$$\phi_e = \phi_b \approx \frac{\phi_P + \phi_E}{2}, \quad (2.76)$$

we get an expression for ϕ_E as

$$\phi_E = 2\phi_b - \phi_P. \quad (2.77)$$

This can be substituted into Equation (2.75) and factorised to yield

$$a_E^* \phi_E + a_P^* \phi_P + a_W^* \phi_W = c^*, \quad (2.78)$$

with

$$\begin{aligned} a_E^* &= 0, \\ a_W^* &= a_W, \\ a_P^* &= a_P - a_E, \\ c^* &= c - 2a_E\phi_b. \end{aligned} \quad (2.79)$$

By performing this substitution the equation for the external cell (cell E) has been dropped from the system, reducing the number of equations in the system. After this reduced system has been solved the values of the boundary cells can be found using Equation (2.77).

A similar process can be used for Neumann boundaries where the gradient of the ϕ field normal to the boundary is specified as ϕ'_b . Using a central difference approximation for the gradient at the boundary

$$\phi'_b = \frac{\partial \phi}{\partial x_e} \approx \frac{\phi_E - \phi_P}{\delta x_e}, \quad (2.80)$$

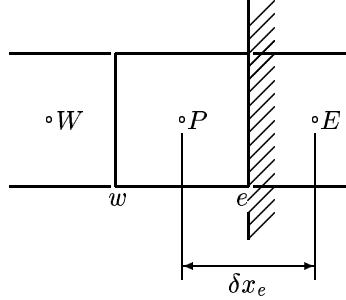


Figure 2.6: The geometry for the cells at the right boundary of the domain. The point E lies outside, whilst P and W are within the domain. The boundary coincides with face e .

then the value of ϕ at the external cell is given by

$$\phi_E = \phi_P + \delta x_e \phi_b^l. \quad (2.81)$$

Substituting into Equation (2.75) and factorising gives

$$a_E^* \phi_E + a_P^* \phi_P + a_W^* \phi_W = c^*, \quad (2.82)$$

with

$$\begin{aligned} a_E^* &= 0, \\ a_W^* &= a_W, \\ a_P^* &= a_P + a_E, \\ c^* &= c - a_E \phi_b^l \delta x_e. \end{aligned} \quad (2.83)$$

As before the equation for cell E has been dropped from the system, and after the system is solved the boundary value can be found using Equation (2.81).

When solving the linear systems arising from PDE's on a SIMD parallel computer like the CM-5, the additional boundary condition solution step can be computationally expensive, since it is performed as an operation on the complete solution array, with the non-boundary elements being masked out. A small saving in computation time can be effected by including the boundary element equations (2.77) and (2.81) in the system that is solved by the linear solver. Then when the system is solved it includes the solution for the boundary, with no additional step to solve for the boundary elements.

2.4 Discretisation of the Transient Transport Equation

The discretisations discussed so far have been for steady flow and they require extension to deal with the case of transient flow. Thankfully the temporal dimension is slightly easier to deal with than the spatial dimensions, since the transient transport equation is hyperbolic/parabolic in time and not elliptic as it is in space. Thus the solution of the equation at time t depends upon its previous state (or history), but not on its future state. Transient systems are usually modelled using a time stepping procedure, with an initial condition being provided, and with the solution algorithm marching forward in time, solving for the domain at each time step.

The transient transport equation was originally given in Equation (2.1) as

$$\frac{\partial}{\partial t}(\rho\phi) + \nabla(\rho\mathbf{u}\phi) = \nabla(\Gamma\nabla\phi) + S, \quad (2.84)$$

with a one-dimensional version of the equation being given in Equation (2.5)

$$\frac{\partial}{\partial t}(\rho\phi) + \frac{\partial}{\partial x}(\rho u\phi) = \frac{\partial}{\partial x}\left(\Gamma\frac{\partial\phi}{\partial x}\right) + S. \quad (2.85)$$

To generate a finite volume discretisation to this equation it is integrated over the one-dimensional cell given in Figure 2.3 in a similar manner to the procedure followed in Equation (2.9), except with an additional integration forward in time from the current time step n at time t (for which we know the field for ϕ), to the next time step $n + 1$ at time $t + \Delta t$ (for which the field ϕ is as yet unknown), with a division by the time interval Δt . The resulting integral equation is

$$\begin{aligned} \frac{1}{\Delta t} \int_t^{t+\Delta t} \int_w^e \frac{\partial}{\partial t}(\rho\phi) dx dt + \frac{1}{\Delta t} \int_t^{t+\Delta t} \int_w^e \frac{\partial}{\partial x}(\rho u\phi) dx dt = \\ \frac{1}{\Delta t} \int_t^{t+\Delta t} \int_w^e \frac{\partial}{\partial x}\left(\Gamma\frac{\partial\phi}{\partial x}\right) dx dt + \frac{1}{\Delta t} \int_t^{t+\Delta t} \int_w^e S dx dt. \end{aligned} \quad (2.86)$$

The integrations in x can be made using the same approximations as were used in the discretisation of the steady transport equation in Section 2.2, with the integration of the transient term,

$$\frac{\partial}{\partial t}(\rho\phi), \quad (2.87)$$

being made using the same approximation as the source term integration in Equation (2.10). This leaves

$$\begin{aligned} \frac{1}{\Delta t} \int_t^{t+\Delta t} \frac{\partial}{\partial t}(\rho\phi) \Delta x dt + \frac{1}{\Delta t} \int_t^{t+\Delta t} \{(\rho u\phi)_e - (\rho u\phi)_w\} dt = \\ \frac{1}{\Delta t} \int_t^{t+\Delta t} \left\{ \left(\Gamma_e \frac{\phi_E - \phi_P}{\delta x_e} \right) - \left(\Gamma_w \frac{\phi_P - \phi_W}{\delta x_w} \right) \right\} dt + \frac{1}{\Delta t} \int_t^{t+\Delta t} S \Delta x dt, \end{aligned} \quad (2.88)$$

where the $\rho u\phi$ parameters in the integral of the advective terms are to be interpolated using one of the schemes described in Sections 2.2.2 to 2.2.4.

The transient term is integrated in time as

$$\frac{1}{\Delta t} \int_t^{t+\Delta t} \frac{\partial}{\partial t}(\rho\phi) \Delta x dt = \frac{\rho \Delta x}{\Delta t} (\phi_P^{n+1} - \phi_P^n), \quad (2.89)$$

assuming a constant density, and with the superscript n signifying a variable is at the current time step, which is known, whilst the $n + 1$ superscript signifies a variable at the next time step, the unknown for which the equation is being solved.

For the other terms in the equation some choice must be made on how to approximate them over the interval $t \rightarrow t + \Delta t$. Four time stepping schemes are discussed below. The first is an explicit first order scheme, Forward Euler differencing, where the value of the derivatives are estimated over the interval Δt by their initial values at the n^{th} time step. Explicit schemes have stability limitations, but they have the great advantage that they don't result in a set of linear equations, but rather the value of the scalar ϕ at the new time step $n + 1$ is explicitly given by its value at the old time step n . In contrast the other three methods are implicit, and so whilst they have no stability restrictions their solution requires the solution of a linear system to calculate the scalar field for the new time step. The three implicit schemes covered are the Backwards Euler, where the derivatives are estimated over the time step by their values at the new $n + 1^{\text{th}}$ time step, the Crank–Nicolson scheme where the derivatives are estimated by a linear interpolation of their values at the n and $n + 1^{\text{th}}$ time steps, and an Adams–Bashforth scheme where the advective terms are estimated by a linear extrapolation from the $n - 1$ and n time steps.

2.4.1 The Forward Euler Differencing Scheme

With Forward Euler time differencing of the one-dimensional transport equation, the diffusion and advection terms in Equation (2.88) are represented over the interval $t \rightarrow t + \Delta t$ by their values at time t . The integral for the diffusive terms in the equation then becomes

$$\begin{aligned} \frac{1}{\Delta t} \int_t^{t+\Delta t} \left\{ \left(\Gamma_e \frac{\phi_E - \phi_P}{\delta x_e} \right) - \left(\Gamma_w \frac{\phi_P - \phi_W}{\delta x_w} \right) \right\} dt \\ \approx \left(\Gamma_e \frac{\phi_E^n - \phi_P^n}{\delta x_e} \right) - \left(\Gamma_w \frac{\phi_P^n - \phi_W^n}{\delta x_w} \right), \end{aligned} \quad (2.90)$$

where the superscript n signifies that a variable is evaluated at time t at the start of the time step. Similarly the integral of the advective components is

$$\begin{aligned} \frac{1}{\Delta t} \int_t^{t+\Delta t} \left\{ (\rho u \phi)_e - (\rho u \phi)_w \right\} dt &\approx \rho u_e^n \phi_e^n - \rho u_w^n \phi_w^n \\ &\approx \rho u_e^n \frac{\phi_P^n + \phi_E^n}{2} - \rho u_w^n \frac{\phi_P^n + \phi_W^n}{2}. \end{aligned} \quad (2.91)$$

The face cell face values in the advective terms can be interpolated using any of the schemes given in Sections 2.2.2 to 2.2.4, and here we have used the central difference approximation described in Section 2.2.3.

Substituting these approximations into Equation (2.88) gives

$$\begin{aligned} \frac{\rho \Delta x}{\Delta t} (\phi_P^{n+1} - \phi_P^n) + \left(\frac{m_e^n}{2} (\phi_P^n + \phi_E^n) - \frac{m_w^n}{2} (\phi_P^n + \phi_W^n) \right) = \\ (d_e^n (\phi_E^n - \phi_P^n) - d_w^n (\phi_P^n - \phi_W^n)) + S \Delta \Omega, \end{aligned} \quad (2.92)$$

where we have used the d_f and m_f expressions for the face mass and diffusion fluxes, their definitions being given in Equation (2.17). Factorising Equation (2.92) gives

$$a_P \phi_P^{n+1} + b_E \phi_E^n + b_P \phi_P^n + b_W \phi_W^n = c, \quad (2.93)$$

where

$$\begin{aligned} a_P &= \frac{\rho \Delta x}{\Delta t}, \\ b_E &= -d_e^n + \frac{1}{2} m_e^n, \\ b_W &= -d_w^n - \frac{1}{2} m_w^n, \\ b_P &= d_e^n + \frac{1}{2} m_e^n + d_w^n - \frac{1}{2} m_w^n - \frac{\rho \Delta x}{\Delta t} \\ &= -(b_E + b_W) + (m_e^n - m_w^n) - \frac{\rho \Delta x}{\Delta t}, \\ c &= S \Delta \Omega. \end{aligned} \quad (2.94)$$

Performing such a differencing upon all the points within a domain generates a system of equations

$$\mathbf{A} \Phi^{n+1} + \mathbf{B} \Phi^n = \mathbf{c}, \quad (2.95)$$

which is expressed as

$$\mathbf{A} \Phi^{n+1} = \mathbf{c} - \mathbf{B} \Phi^n, \quad (2.96)$$

where Φ^n is known and the system is solved for Φ^{n+1} .

Two things should be noted about the above system. Firstly, the equation terms in the \mathbf{B} array are identical to those for the steady central difference scheme given in Equation (2.46), baring the addition of the $\rho\Delta x/\Delta t$ transient term to b_P . Thus the differencing schemes previously derived for the steady transport equation can be readily used in the transient transport equation. Secondly the only non-zero elements of the \mathbf{A} array are on the main diagonal, so it is trivial to solve for Φ^{n+1} given that we know Φ^n , and so the scheme is explicit.

Unfortunately there are stability restrictions on the system, the original stability analysis of the system being performed by Courant, Friedrichs and Lewy[25, 26]. The Diffusion and Courant numbers for a cell are

$$\begin{aligned} D &= \frac{\Gamma \Delta t}{\Delta x^2}, \\ Cr &= \frac{u \Delta t}{\Delta x}, \end{aligned} \quad (2.97)$$

where D is the ratio of the time step Δt to the characteristic diffusion time $\Gamma/\Delta x^2$ and the Courant number Cr is the ratio of the time step to the characteristic convection time $\Delta x/u$. For a system discretised using Central differencing in space and Forward Euler differencing in time Courant found the requirements for stability to be

$$\begin{aligned} D &< \frac{1}{2}, \\ Cr &< 2D, \end{aligned} \quad (2.98)$$

which effectively place a limit on the time step and mesh size of

$$\begin{aligned} \Delta t &< \frac{\Delta x^2}{2\Gamma}, \\ \Delta x &< \frac{2u}{\Gamma}. \end{aligned} \quad (2.99)$$

The second restriction in Equation 2.98 may be rearranged as

$$\begin{aligned} Cr/D &< 2, \\ \frac{u \Delta t}{\Delta x} \frac{\Delta x^2}{\Gamma \Delta t} &= \frac{u \Delta x}{\Gamma} < 2, \end{aligned} \quad (2.100)$$

which is the cell Péclet number restriction that was previously encountered with the Central Difference scheme in Section 2.2.1.

By changing to a First Order Upwind differencing scheme for the advective fluxes, Courant relaxed the conditions in Equation (2.98) to

$$Cr < 1, \quad (2.101)$$

which gives a restriction on the time step of

$$\Delta x < \frac{\Delta x}{u}. \quad (2.102)$$

This may be interpreted to say that the time step must be less than the residence time for the cell, that is the time taken for a fluid particle to cross the cell.

This is less restrictive than Equation (2.98), but still places a restriction on the time step that can be used in a transient calculation. As a mesh is refined so the time step must be similarly reduced to preserve the Courant number restriction.

2.4.2 The Backward Euler Differencing Scheme

One way to overcome the stability limitations of Equation (2.98) is by integrating the diffusion and advection terms in Equation (2.88) by assuming that the values at the time $t + \Delta t$ are representative

for the interval $t \rightarrow t + \Delta t$. The integral for the diffusive term then becomes

$$\begin{aligned} \frac{1}{\Delta t} \int_t^{t+\Delta t} \left\{ \left(\Gamma_e \frac{\phi_E - \phi_P}{\delta x_e} \right) - \left(\Gamma_w \frac{\phi_P - \phi_P}{\delta x_w} \right) \right\} dt \\ \approx \left(\Gamma_e \frac{\phi_E^{n+1} - \phi_P^{n+1}}{\delta x_e} \right) - \left(\Gamma_w \frac{\phi_P^{n+1} - \phi_W^{n+1}}{\delta x_w} \right), \end{aligned} \quad (2.103)$$

where the superscript $n+1$ signifies that a variable is evaluated at the end of the time step at time $t + \Delta t$. Similarly the integral of the advective components is

$$\begin{aligned} \frac{1}{\Delta t} \int_t^{t+\Delta t} \left\{ (\rho u \phi)_w - (\rho u \phi)_e \right\} dt &\approx \rho u_w^{n+1} \phi_w^{n+1} - \rho u_e^{n+1} \phi_e^{n+1} \\ &\approx \rho u_e^{n+1} \frac{\phi_P^{n+1} + \phi_E^{n+1}}{2} - \rho u_w^{n+1} \frac{\phi_P^{n+1} + \phi_W^{n+1}}{2}. \end{aligned} \quad (2.104)$$

As with the explicit scheme, the face cell face values in the advective terms can be interpolated using any of the schemes given in Sections 2.2.2 to 2.2.4, and again we have used the central difference approximation described in Section 2.2.3.

Substituting these approximations into Equation (2.88) gives

$$\begin{aligned} \frac{\rho \Delta x}{\Delta t} (\phi_P^{n+1} - \phi_P^n) + \left(\frac{m_e^{n+1}}{2} (\phi_P^{n+1} + \phi_E^{n+1}) - \frac{m_w^{n+1}}{2} (\phi_P^{n+1} + \phi_W^{n+1}) \right) = \\ (d_e^{n+1} (\phi_E^{n+1} - \phi_P^{n+1}) - d_w^{n+1} (\phi_P^{n+1} - \phi_W^{n+1})) + S \Delta \Omega, \end{aligned} \quad (2.105)$$

where we have used the d_f and m_f expressions for the face mass and diffusion fluxes, their definitions being given in Equation (2.17). Factorising Equation (2.105) gives

$$a_E \phi_E^{n+1} + a_P \phi_P^{n+1} + a_W \phi_W^{n+1} + b_P \phi_P^n = c, \quad (2.106)$$

where

$$\begin{aligned} a_E &= -d_e^{n+1} + \frac{1}{2} m_e^{n+1}, \\ a_W &= -d_w^{n+1} - \frac{1}{2} m_w^{n+1}, \\ a_P &= d_e^{n+1} + \frac{1}{2} m_e^{n+1} + d_w^{n+1} - \frac{1}{2} m_w^{n+1} + \frac{\rho \Delta x}{\Delta t} \\ &= -(a_E + a_W) + (m_e^{n+1} - m_w^{n+1}) + \frac{\rho \Delta x}{\Delta t}, \\ b_P &= -\frac{\rho \Delta x}{\Delta t}, \\ c &= S \Delta \Omega. \end{aligned} \quad (2.107)$$

As with the explicit scheme this generates the linear system

$$\mathbf{A} \Phi^{n+1} = \mathbf{c} - \mathbf{B} \Phi^n, \quad (2.108)$$

where Φ^n is known and Φ^{n+1} is to be obtained.

Unlike the explicit scheme this method has no limits on stability with respect to the time step. However it is still first order in time and so is relatively inaccurate unless small time steps are taken. Also, unlike the explicit scheme, the matrix \mathbf{A} has three diagonals with non-zero elements. This means that it is no longer trivial to solve for Φ^{n+1} and some sort of linear inversion method must be used to solve the system. For the system arising from the one-dimensional equation this is not problematic since the extremely efficient Thomas Tridiagonal solver can be used (see Section 3.1.2). However, for the equations arising from discretisations of the two and three dimensional diffusion equation the equations are much harder to solve, some methods for doing so being discussed in Chapter 3.

2.4.3 Crank–Nicolson or Centred Differencing

In the Crank–Nicolson discretisation, the advection and diffusion terms in Equation (2.88) are integrated over the time interval $t \rightarrow t + \Delta t$ using the average of the values at t and $t + \Delta t$. This scheme is second order in time and so more accurate than the first order explicit and implicit schemes, and like the implicit scheme it is stable. However, for large time steps the scheme can produce non-physical oscillations in the calculated scalar field.

Using the Crank–Nicolson discretisation the integral for the diffusive term becomes

$$\begin{aligned} \frac{1}{\Delta t} \int_t^{t+\Delta t} & \left\{ \left(\Gamma_e \frac{\phi_E - \phi_P}{\delta x_e} \right) - \left(\Gamma_w \frac{\phi_P - \phi_W}{\delta x_w} \right) \right\} dt \\ & \approx \left(\Gamma_e \frac{\phi_E^{n+1} - \phi_P^{n+1}}{2\delta x_e} \right) - \left(\Gamma_w \frac{\phi_P^{n+1} - \phi_W^{n+1}}{2\delta x_w} \right) \\ & \quad + \left(\Gamma_e \frac{\phi_E^n - \phi_P^n}{2\delta x_e} \right) - \left(\Gamma_w \frac{\phi_P^n - \phi_W^n}{2\delta x_w} \right), \end{aligned} \tag{2.109}$$

with n and $n + 1$ signifying values at the start and end of the time step. Similarly the integral of the advective components is

$$\begin{aligned} \frac{1}{\Delta t} \int_t^{t+\Delta t} & \left\{ (\rho u \phi)_w - (\rho u \phi)_e \right\} dt \\ & \approx \frac{1}{2} (\rho u_w^{n+1} \phi_w^{n+1} + \rho u_w^n \phi_w^n) - \frac{1}{2} (\rho u_e^{n+1} \phi_e^{n+1} + \rho u_e^n \phi_e^n) \\ & \approx \frac{1}{2} \left(\rho u_e^{n+1} \frac{\phi_P^{n+1} + \phi_E^{n+1}}{2} + \rho u_e^n \frac{\phi_P^n + \phi_E^n}{2} \right) \\ & \quad - \frac{1}{2} \left(\rho u_w^{n+1} \frac{\phi_P^{n+1} + \phi_W^{n+1}}{2} + \rho u_w^n \frac{\phi_P^n + \phi_W^n}{2} \right). \end{aligned} \tag{2.110}$$

As with the first order schemes previously described, the cell face values in the advective terms can be interpolated using any of the schemes given in Sections 2.2.2 to 2.2.4. And as before, the central difference scheme has been used to approximate the advective fluxes.

Substituting these approximations into Equation (2.88) gives

$$\begin{aligned} \frac{\rho \Delta x}{\Delta t} (\phi_P^{n+1} - \phi_P^n) & + \frac{1}{4} (m_e^{n+1} (\phi_P^{n+1} + \phi_E^{n+1}) + m_e^n (\phi_P^n + \phi_E^n)) - \\ & \frac{1}{4} (m_w^{n+1} (\phi_P^{n+1} + \phi_W^{n+1}) + m_w^n (\phi_P^n + \phi_W^n)) \\ & = \frac{1}{2} (d_e^{n+1} (\phi_E^{n+1} - \phi_P^{n+1}) + d_e^n (\phi_E^n - \phi_P^n)) - \\ & \frac{1}{2} (d_w^{n+1} (\phi_P^{n+1} - \phi_W^{n+1}) + d_w^n (\phi_P^n - \phi_W^n)) + S \Delta \Omega, \end{aligned} \tag{2.111}$$

and factorising yields the system

$$a_E \phi_E^{n+1} + a_P \phi_P^{n+1} + a_W \phi_W^{n+1} + b_E \phi_E^n + b_P \phi_P^n + b_W \phi_W^n = c, \tag{2.112}$$

where

$$\begin{aligned}
a_E &= \frac{1}{2} (-d_e^{n+1} + \frac{1}{2} m_e^{n+1}), \\
a_W &= \frac{1}{2} (-d_w^{n+1} - \frac{1}{2} m_w^{n+1}), \\
a_P &= \frac{1}{2} (d_e^{n+1} + \frac{1}{2} m_e^{n+1} + d_w^{n+1} - \frac{1}{2} m_w^{n+1}) + \frac{\rho \Delta x}{\Delta t} \\
&= -(a_E + a_W) + \frac{1}{2} (m_e^{n+1} - m_w^{n+1}) + \frac{\rho \Delta x}{\Delta t}, \\
b_E &= \frac{1}{2} (-d_e^n + \frac{1}{2} m_e^n), \\
b_W &= \frac{1}{2} (-d_w^n - \frac{1}{2} m_w^n), \\
b_P &= \frac{1}{2} (d_e^n + \frac{1}{2} m_e^n + d_w^n - \frac{1}{2} m_w^n) - \frac{\rho \Delta x}{\Delta t} \\
&= -(b_E + b_W) + \frac{1}{2} (m_e^n - m_w^n) - \frac{\rho \Delta x}{\Delta t}, \\
c &= S_P \Delta \Omega.
\end{aligned} \tag{2.113}$$

The terms in (2.113) are the same as those in the steady state equation divided by two, with the a_P and b_P terms having the addition of the $\pm \rho \Delta x / \Delta t$ term. The system is solved as

$$\mathbf{A} \Phi^{n+1} = \mathbf{c} - \mathbf{B} \Phi^n, \tag{2.114}$$

where Φ^{n+1} is the vector of unknowns. As with the first order implicit scheme the \mathbf{A} array has off-diagonal entries and must be inverted with a linear solver.

2.4.4 Adams–Bashforth Differencing

The two implicit schemes described above require the mass flux fields, m^{n+1} , at the new time step for the approximation of the advective fluxes over the time interval. Unless there is a prescribed velocity field these mass fluxes are normally found from a transport equation for momentum. Thus the transport equations are normally coupled together and must be solved in some iterative matter, updating the values for the mass fluxes at the new time step (this is discussed further in Section 4.2.3).

One method for overcoming this problem is a scheme where the advective terms are extrapolated from previous time steps in an explicit manner using Adams–Bashforth differencing, whilst the remaining diffusion terms are handled implicitly with a Crank–Nicolson second order scheme.

For the advective terms, the integration of the scalar ϕ of the mass flux m forward in time from time t to $t + \Delta t$ (ie: from time step n to $n + 1$) is approximated by

$$\begin{aligned}
\frac{1}{\Delta t} \int_n^{n+1} \phi dt &\approx \frac{3}{2} \phi^n - \frac{1}{2} \phi^{n-1}, \\
\frac{1}{\Delta t} \int_n^{n+1} m dt &\approx \frac{3}{2} m^n - \frac{1}{2} m^{n-1},
\end{aligned} \tag{2.115}$$

and so using a central difference scheme to approximate the value of the scalar ϕ at the face of the cell, the integral of the advective terms is approximated by

$$\begin{aligned}
\frac{1}{\Delta t} \int_t^{t+\Delta t} \left\{ (\rho u \phi)_e - (\rho u \phi)_w \right\} dt \\
&\approx \left(\frac{3}{2} \rho u_e^n \phi_e^n - \frac{1}{2} \rho u_e^{n-1} \phi_e^{n-1} \right) - \left(\frac{3}{2} \rho u_w^n \phi_w^n - \frac{1}{2} \rho u_w^{n-1} \phi_w^{n-1} \right) \\
&\approx \left(\frac{3}{4} \rho u_e^n (\phi_P^n + \phi_E^n) - \frac{1}{4} \rho u_e^{n-1} (\phi_P^{n-1} + \phi_E^{n-1}) \right) \\
&\quad - \left(\frac{3}{4} \rho u_w^n (\phi_P^n + \phi_W^n) - \frac{1}{4} \rho u_w^{n-1} (\phi_P^{n-1} + \phi_W^{n-1}) \right).
\end{aligned} \tag{2.116}$$

The approximation of the diffusive terms is that which was used for the Crank–Nicolson scheme given above in Equation (2.109). Substituting these expressions into Equation (2.88) gives

$$\begin{aligned} \frac{\rho\Delta x}{\Delta t} (\phi_P^{n+1} - \phi_P^n) + \left(\frac{3}{4}m_e^n(\phi_P^n + \phi_E^n) - \frac{1}{4}m_e^{n-1}(\phi_P^{n-1} + \phi_E^{n-1}) \right) - \\ \left(\frac{3}{4}m_w^n(\phi_P^n + \phi_W^n) - \frac{1}{4}m_w^{n-1}(\phi_P^{n-1} + \phi_W^{n-1}) \right) \\ = \left(\frac{1}{2}d_e^{n+1}(\phi_E^{n+1} - \phi_P^{n+1}) + \frac{1}{2}d_e^n(\phi_E^n - \phi_P^n) \right) - \\ \left(\frac{1}{2}d_w^{n+1}(\phi_P^{n+1} - \phi_W^{n+1}) + \frac{1}{2}d_w^n(\phi_P^n - \phi_W^n) \right) + S\Delta\Omega, \end{aligned} \quad (2.117)$$

which can be factorised to give

$$\begin{aligned} a_E \phi_E^{n+1} + a_P \phi_P^{n+1} + a_W \phi_W^{n+1} + b_E \phi_E^n + b_P \phi_P^n + b_W \phi_W^n \\ + c_E \phi_E^{n-1} + c_P \phi_P^{n-1} + c_W \phi_W^{n-1} = q, \end{aligned} \quad (2.118)$$

where

$$\begin{aligned} a_E &= -\frac{1}{2}d_e^{n+1}, \\ a_W &= -\frac{1}{2}d_w^{n+1}, \\ a_P &= \frac{1}{2}(d_e^{n+1} + d_w^{n+1}) + \frac{\rho\Delta x}{\Delta t} \\ &= -(a_E + a_W) + \frac{\rho\Delta x}{\Delta t}, \\ b_E &= -\frac{1}{2}d_e^n + \frac{3}{4}m_e^n, \\ b_W &= -\frac{1}{2}d_w^n - \frac{3}{4}m_w^n, \\ b_P &= \frac{1}{2}(d_e^n + d_w^n) + \frac{3}{4}(m_e^n - m_w^n) - \frac{\rho\Delta x}{\Delta t} \\ &= -(b_E + b_W) + \frac{3}{2}(m_e^n - m_w^n) - \frac{\rho\Delta x}{\Delta t}, \\ c &= S_P\Delta\Omega. \end{aligned} \quad (2.119)$$

The Adams–Bashforth scheme was so developed by Lilly[105], the original form of the difference in Equation (2.115) being given by Bashforth and Adams[8]. An alternative form of the method can be made by only extrapolating the mass flux using Equation (2.115) and otherwise using a Crank–Nicolson scheme for the scalar ϕ in the advection terms. For interpolating the advective fluxes other spatial differencing schemes can be used apart from the central difference scheme used above.

Other multi-step schemes that can be used for advancing the transport equation forward in time include Runge–Kutta integration[86], Richardson extrapolation[140] and the Bulirsch–Stoer method[162]. For reasons of simplicity these schemes weren't evaluated in this study.

For the initial time step there is no $n - 1^{\text{th}}$ time step to use for extrapolation, and so an explicit Forward Euler extrapolation is made for the advective terms.

2.4.5 Summary of the Temporal Discretisation Schemes

Of the four temporal discretisation schemes discussed, the easiest to solve is the explicit Forwards Euler scheme, since it doesn't require the solution of a linear system. However this scheme has stability restrictions on the time step. In comparison the two implicit schemes described, the backwards Euler and Crank–Nicolson schemes, have no such stability limitations, but in turn are more difficult to solve, requiring the solution of a system of linear equations. As regards accuracy, the Forward and Backward Euler methods are first order schemes in time, whilst the Crank–Nicolson method is second order in time. Thus the Crank–Nicolson method would be expected to converge to the correct solution at a faster rate as the time step is reduced.

The fourth scheme discussed used a second order explicit Adams–Bashforth scheme for the advective terms, with a second order Crank–Nicolson implicit scheme being used for the diffusive terms. This scheme is more efficient than the Crank–Nicolson scheme when modelling transient flows, since the iterative process that is commonly needed to estimate the velocity field is dispensed with, and so the time necessary for solution at any time step is greatly reduced, whilst it retains the second order accuracy of the Crank–Nicolson scheme. Unfortunately it has a Courant number stability restriction similar to the Forwards Euler scheme. However, provided the extra computational effort for the reduced time step is less than the effort need for the iterative coupling of the velocity fields used with Crank–Nicolson differencing there is an overall efficiency gain.

2.5 A Comparison of the Discretisation Methods

To test the implementation of the differencing schemes described above, and to compare their accuracy, the methods were applied to two benchmark test problems. The first, the “advecting witches hat” problem, models the transient advection of a scalar field in a two-dimensional steady velocity field with no diffusion. For the velocity fields chosen the initial conical distribution of the scalar should retain its shape, and so any variation in the distribution is attributable to numerical error. The second benchmark, the Smith–Hutton problem[157], is for advection and diffusion in a steady flow. A sharp discontinuity in the scalar field is imposed at the flows inlet, and the calculated outlet profile is examined to compare the accuracy of the various schemes.

Further tests of the differencing schemes have been undertaken at the end of Chapter 4, where they have been implemented within a full Navier–Stokes solver and used to model two two-dimensional steady state internal flows for which published benchmarks are available—a driven cavity flow, and a natural convection problem.

2.5.1 The Advecting Witches Hat Problem

The witches hat test models the advection of a blob of a scalar through a steady shear-free velocity field. The diffusivity is set to zero, and so the initial disturbance of the scalar field is advected through the flow, with its shape remaining unchanged. Since the exact solution is known, any discrepancy between the exact and calculated solution is attributable to numerical error, and so it is an effective test of the discretisation of the advective and transient terms in the transport equation. Typically the blob is given a conical distribution, which when projected in 3D gives the test its name.

The test case modelled follows that used by Tamamidis and Assanis[168], who in turn based their test on those of Crowley, Molenkamp and Orszag[28, 120, 121]. The two-dimensional transport equation is solved for a scalar ϕ over a unit square centred on the origin, so the domain has boundaries at $x = \pm 0.5$ and $y \in [-0.5, 0.5]$. The flow in the domain undergoes an anticlockwise solid body rotation centred on the origin, with an angular velocity of $\omega = 1$. The streamfunction for such a flow is

$$\psi = \frac{1}{2} (x^2 + y^2), \quad (2.120)$$

which gives a velocity field of

$$\begin{aligned} u &= y, \\ v &= -x. \end{aligned} \quad (2.121)$$

To ensure that the face mass fluxes satisfy the continuity equation, the streamfunction is calculated for each corner of a cell, and then the volume flux across each face is calculated from the difference in the streamfunction at each end of the face.

An initial scalar distribution is prescribed that is zero everywhere except for a conical blob located at $(0, 0.25)$. Defining a radius r centred at this location

$$r = \sqrt{x^2 + (y - 0.25)^2}, \quad (2.122)$$

then the initial distribution is

$$\phi = \begin{cases} \sin^2\left(\frac{\pi}{2}\left(1 - \frac{r}{0.1}\right)\right) & : r \leq 0.1 \\ 0 & : r \geq 0.1, \end{cases} \quad (2.123)$$

which gives the distribution plotted in Figure 2.7.

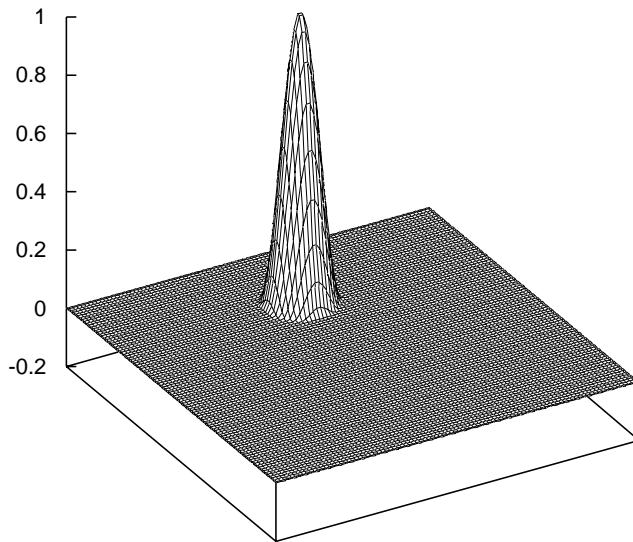


Figure 2.7: The initial field for the rotating flow test. Field shown for a 98×98 mesh.

The transient transport equation is solved for this domain, being stepped forward for a total time of 2π , with zero derivative Neumann boundary conditions being imposed at the edges of the domain

$$\begin{aligned} \frac{d\phi}{dx} &= 0 & : x = \pm 0.5, \\ \frac{d\phi}{dy} &= 0 & : y = \pm 0.5. \end{aligned} \quad (2.124)$$

For an angular velocity of $\omega = 1$, the time 2π is exactly that for a single rotation of the flow field. Since the diffusivity is zero (ie: $\Gamma = 0$, $\text{Pe} = \infty$) and the flow undergoes no shear, then the shape of the scalar field will be undistorted and will undergo solid body rotation about the origin, and at time 2π the field should be identical to the initial field at time 0.

For such a flow the velocity goes through a range of angles with the mesh. To examine the effect of the angle between the velocity and the mesh axes, two further series of test runs were made similar to those of Wolfshtein[182], with a scalar field being translated in a steady uniform flow that runs

parallel and at 45° to the mesh axes. For both cases the same domain was solved for, with the same boundary conditions. For the parallel runs the streamfunction was given by

$$\psi = x, \quad (2.125)$$

which gives a velocity field of

$$\begin{aligned} u &= 0, \\ v &= 1, \end{aligned} \quad (2.126)$$

and the run was made for a total $\Delta t = 0.3125$, with the initial scalar cone being centred at $(0, -0.15625)$. For the 45° flow the streamfunction was

$$\psi = x - y, \quad (2.127)$$

with a velocity field of

$$\begin{aligned} u &= 1, \\ v &= 1, \end{aligned} \quad (2.128)$$

with a total $\Delta t = 0.3125$, and an initial scalar cone centred at $(-0.15625, -0.15625)$.

For each of the three test cases the flow was solved using the FOU, Exponential, Central, SOU, MSOU, QUICK and ULTRA differencing schemes, using the Forward and Backwards Euler, and the Crank–Nicolson differencing schemes. The problem was also modelled using the Adams–Bashforth time stepping scheme, but this method is unstable in the case of zero diffusivity, and to calculate a solution the boundary conditions had to be modified to a Dirichlet condition of $\phi = 0$ at the edges of the domain. Since the Adams–Bashforth time stepping method was implemented in a separate code, only the QUICK differencing scheme was tested with this time stepping method. Also, for the zero diffusivity problem modelled all the first order schemes revert to first order upwinding and give identical solutions, and so for reasons of space not all first order results are shown.

In the absence of diffusion, the combination of the Crank–Nicolson time discretisation with the second and third order discretisations of the advective term (the Central, SOU and QUICK schemes) is neutrally stable, and so any errors are expected to accumulate in time. Thus as time tends to infinity the solutions calculated with a combination of these schemes are expected to grow progressively worse.

For each combination of differencing and time stepping scheme the flow was calculated on meshes of 18^2 , 26^2 , 34^2 , 50^2 , 66^2 , 98^2 and 130^2 cells (including boundary cells). The flows were calculated for Courant numbers of $5, 1, 0.5, 0.1, 0.05, 0.01$ and 0.005 , except for the explicit Forward Euler and Adams–Bashforth schemes which were unstable for Courant numbers ≥ 0.5 .

The RMS error of the final solution of the rotation test case is plotted in Figure 2.8 as a function of Courant number and mesh spacing. Data is presented for the calculations made using the QUICK differencing scheme with Crank–Nicolson time stepping. As can be seen, for Courant numbers < 1 the solution is converged with respect to Courant number (or time step), whilst the error due to the coarseness of the mesh seems to dominate for all meshes tested, with there being no sign of mesh convergence.

The same data is plotted in Figure 2.9 as a function of Courant number only, along with the errors from calculations made using the QUICK differencing scheme and the Forward and Backward Euler, and Adams–Bashforth time stepping schemes. The Forward and Backward Euler schemes are seen to be slower to converge with respect to the Courant number (or time step) than the Crank–Nicolson and Adams–Bashforth schemes. The Adams–Bashforth scheme seems to have similar convergence properties to the Crank–Nicolson time stepping method. Note that for Courant numbers ≤ 0.01 all four schemes converge to the same solution suggesting that the remaining error is not dependent on the time stepping scheme used, but instead is due only to the coarseness of the mesh.

The RMS errors of all the final solutions are plotted in Figure 2.10 as functions of Courant number and mesh spacing. As with the results calculated using the QUICK differencing scheme, the solutions

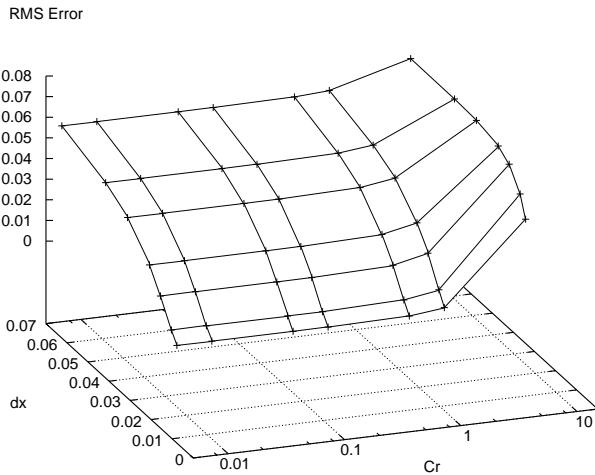


Figure 2.8: The RMS error for the rotating test problem, plotted as a function of Courant number (Cr) and mesh spacing (dx), for calculations made using the QUICK differencing scheme.

converge to be independent of the time stepping scheme used for Courant numbers ≤ 0.01 , and the Crank–Nicolson scheme converges for Courant numbers < 1 . Note that the Adams–Bashforth time stepping scheme was only tested with QUICK differencing. Also for the explicit Forward Euler time stepping method, the SOU and Central difference schemes would only converge for Courant numbers < 0.1 and 0.05 respectively, and were otherwise unstable and so could not be calculated.

There are typically two types of numerical error encountered when modelling transient advective flow, dispersion and dissipation. With dispersion the different wavelengths of the solution propagate at different velocities in the flow, and so waves tend to be drawn out and become wiggly, whilst dissipation describes the false or numerical diffusion that occurs, spreading the width of waves and reducing their amplitude.

Plots of the final solutions for the three test cases are shown plotted in Figures 2.11, 2.12 and 2.13, for flows calculated using the various differencing schemes and the Crank–Nicolson time stepping scheme, at a Courant number of 0.1 on a 98×98 mesh. For comparison the exact solution is also shown.

The most striking feature of these solutions is the extreme dissipation exhibited by the first order schemes. For the 45° and parallel test runs the total Δt was less than for the rotating test case, and the amplitude of the scalar cone was reduced to 33% and 58% of its initial value respectively. For the rotating test case the amplitude was reduced to 13% of its initial value. The methods did show good conservative properties, with the integral of ϕ over the domain remaining constant, and so the loss of amplitude was accompanied by a spreading of the base of the cone. For the rotational and 45° test cases the scalar cone remained roughly circular, but for the parallel flow test case the dissipation occurred only in the streamwise direction, with no cross-stream dissipation taking place. This accounts for the reduced loss in amplitude of the parallel test case, but the resulting solution is spread out in the streamwise direction giving a long narrow scalar blob.

The other notable feature is the dispersion in the solutions calculated with the Central, SOU and QUICK differencing schemes. The Central and QUICK schemes both have an oscillatory wake behind the advecting cone, whilst the SOU scheme has an oscillatory disturbance moving ahead of the cone. For the parallel test case the oscillations are restricted to the streamlines of the scalar cone, whilst for the 45° test case the oscillations align themselves with the mesh axis. Of the three schemes the QUICK differencing seems to give the least oscillatory solution, whilst the Central difference scheme generates short wavelength disturbances that are visibly perturb the whole of the domain in

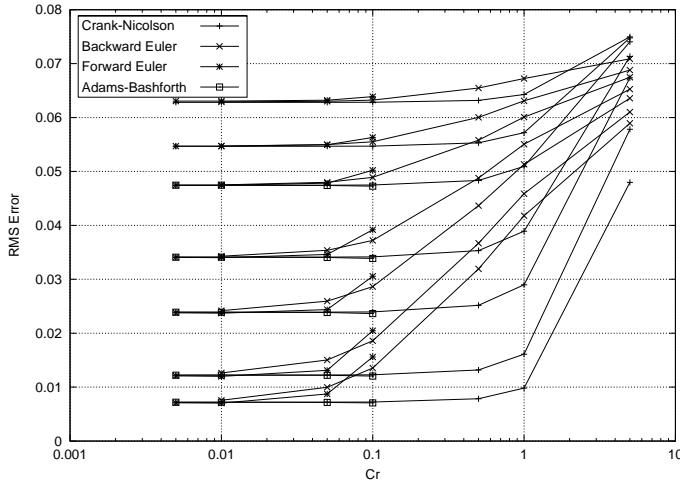


Figure 2.9: The RMS error for the rotating test problem, plotted as a function of Courant number (Cr), for calculations made using the QUICK differencing scheme. For each time stepping scheme the plots (from the top down) are for meshes of 18^2 , 26^2 , 34^2 , 50^2 , 66^2 , 98^2 and 130^2 cells.

the rotational test.

The flux limiters applied to the MSOU and ULTRA differencing schemes effectively suppress the oscillations typically generated by the higher order methods, and the solutions appear free of dispersion. However the shape of the cone is distorted, most visibly with the MSOU scheme where the cone has been somewhat flattened. In addition the amplitude of the MSOU and ULTRA solutions is less than that of their SOU and QUICK counterparts. This suggests that the limiters act in a dissipative manner.

The amplitudes of the cones are plotted as a function of time step in Figure 2.14, the data being presented for a calculation at a Courant number of 0.1 on a 130^2 mesh using Crank–Nicolson time stepping. The loss of amplitude of the first order schemes is notable, with the transient for these methods dropping off the bottom of the plot. The central and QUICK solutions exhibit the least decrease in amplitude, their solutions having a high frequency wave of the order of $Cr\Delta x/u$ imposed on them. The MSOU and ULTRA flux limited solutions exhibit a greater loss in amplitude, and are much smoother than their unlimited high order counterparts.

A detail of the amplitude plot, in Figure 2.15, reveals the oscillatory nature of the Central, SOU and QUICK amplitudes. The reason for these oscillations is the discrete nature of the data being used. As the peak of the scalar cone moves from the edge to the centre of a cell, the value at the cell centre rises accordingly, with it dropping once the peak passes and travels out of the cell. Since the amplitude was measured as the maximum of any of the cells in the domain, then this maximum will rise and fall as the peak passes through each cell, with a time period of $Cr\Delta x/u$ which is the time taken to pass through the cell.

In contrast the MSOU and ULTRA schemes do not exhibit this amplitude oscillation, but instead experience “staircasing”, with the maximum value dropping as the peak passes from cell to cell. For the ULTRA scheme, since this method does not allow overshoots in the interpolated function, as the peak of the scalar cone passes through a cell the value at the centre is restrained to the maximum of the interpolated boundary values. No restraint is made as the peak passes out of the cell and so a corresponding loss in amplitude occurs. With the MSOU scheme the loss occurs abruptly whilst the peak is at the centre of the cell, and it is uncertain what mechanism is causing the drop. Of the two methods the MSOU experiences less amplitude loss than the ULTRA scheme.

The growth of the rms error for the rotational test case is shown in Figure 2.16. The large error in the first order schemes is apparent, as is the large errors of the solutions calculated with the Central and

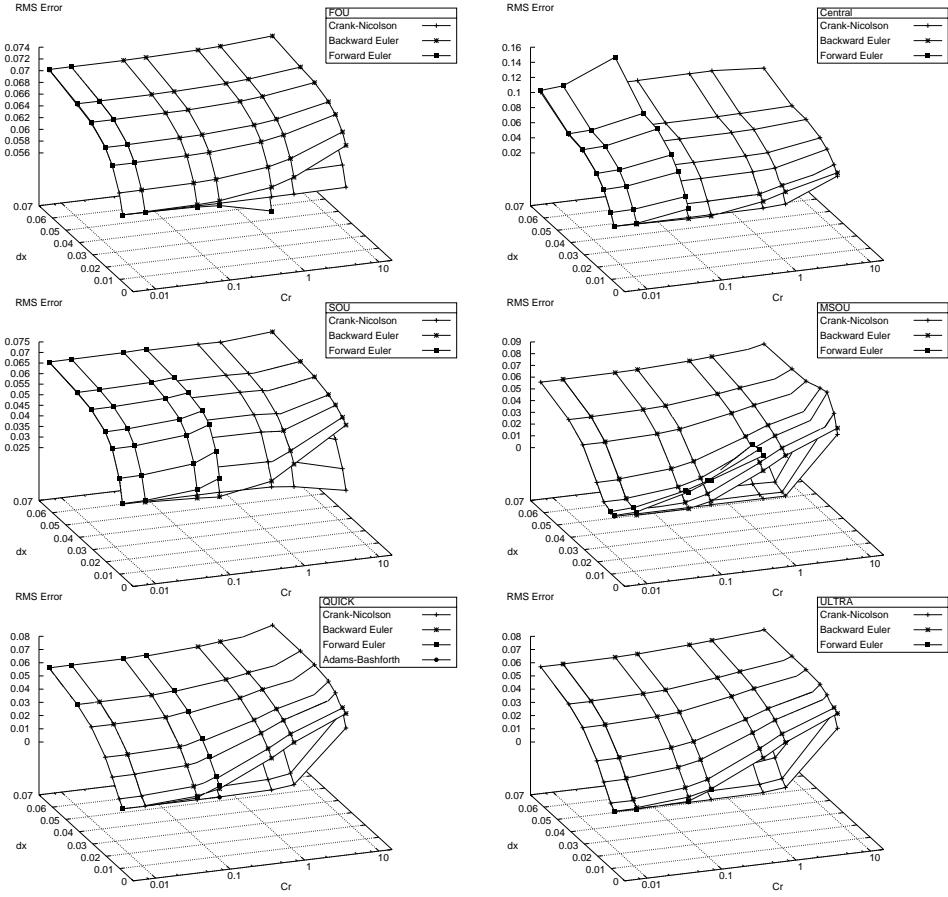


Figure 2.10: The RMS error for the rotational test problem, plotted as a function of Courant number (Cr) and mesh spacing (dx). Results are plotted for (left to right, top to bottom), FOU, Central, SOU, MSOU, QUICK and ULTRA differencing.

SOU schemes. The QUICK, MSOU and ULTRA methods all show a reduced error, with the ULTRA scheme having the lowest rms error despite it having a greater loss in amplitude than the MSOU and QUICK schemes.

Finally the phase error for the rotational test case is shown in Figure 2.17, the error being measured as the angle in radians between the centre of the exact solution and the peak of the calculated distribution. This gives a noisy measurement with much variation as the peak of the scalar cone passes from one cell to the next, and perhaps a centre of mass calculation would give a better measure of the centre of the scalar cone. The solutions are all reasonably accurate, with the SOU and Central differencing schemes giving the outlying solutions. The first order solution has an oscillatory nature, with the phase error increasing when the angle between the flow and the mesh increases from $0^\circ \rightarrow 45^\circ$, with a decrease as the angle continues from $45^\circ \rightarrow 90^\circ$.

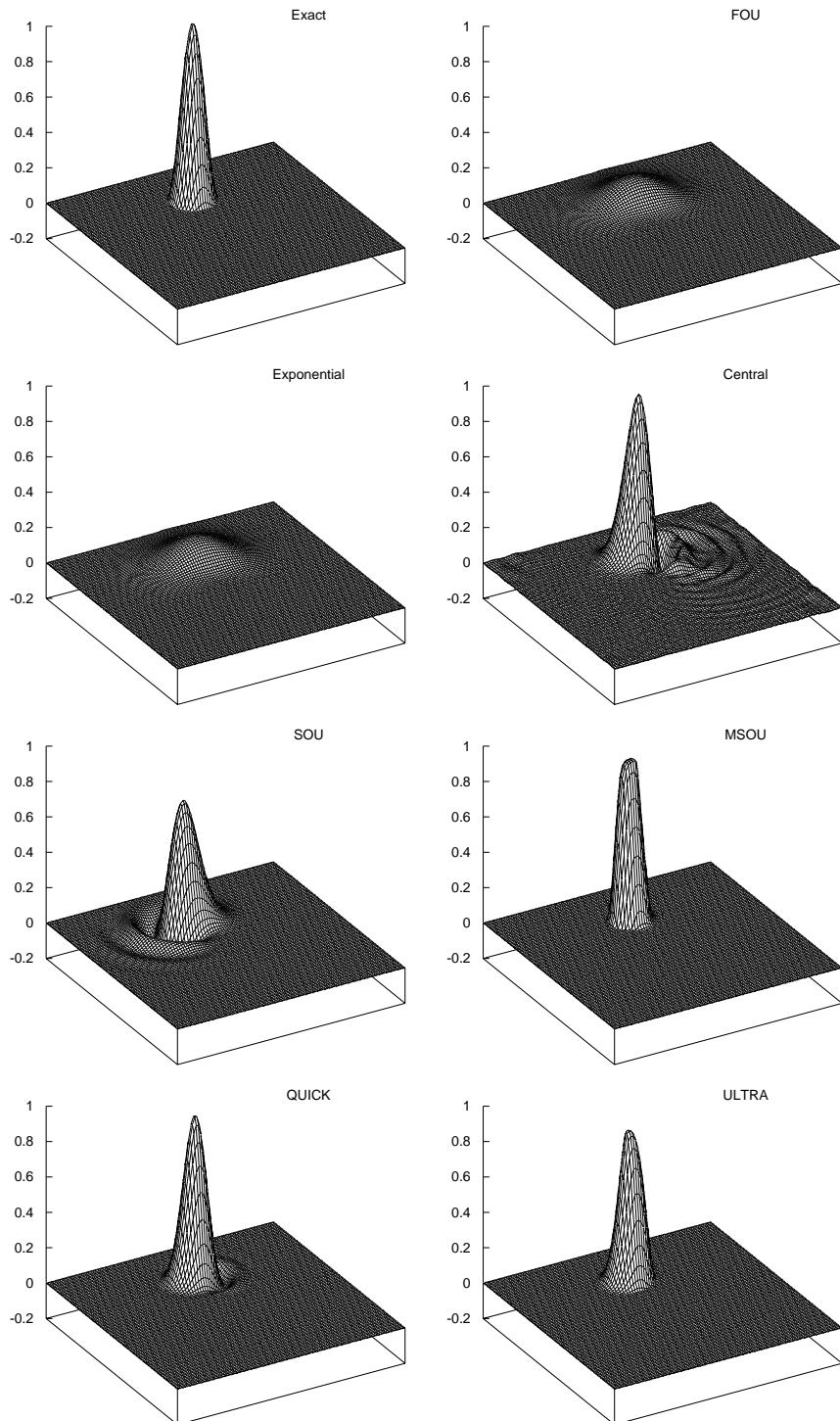


Figure 2.11: The scalar field after one rotation of the rotating flow test, calculated using Crank–Nicolson time stepping on a 98×98 mesh with a Courant number of 0.1. The plots are (left to right, top to bottom), The exact solution, and solutions calculated using the FOU, Exponential, Central, SOU, MSOU, QUICK and ULTRA differencing schemes.

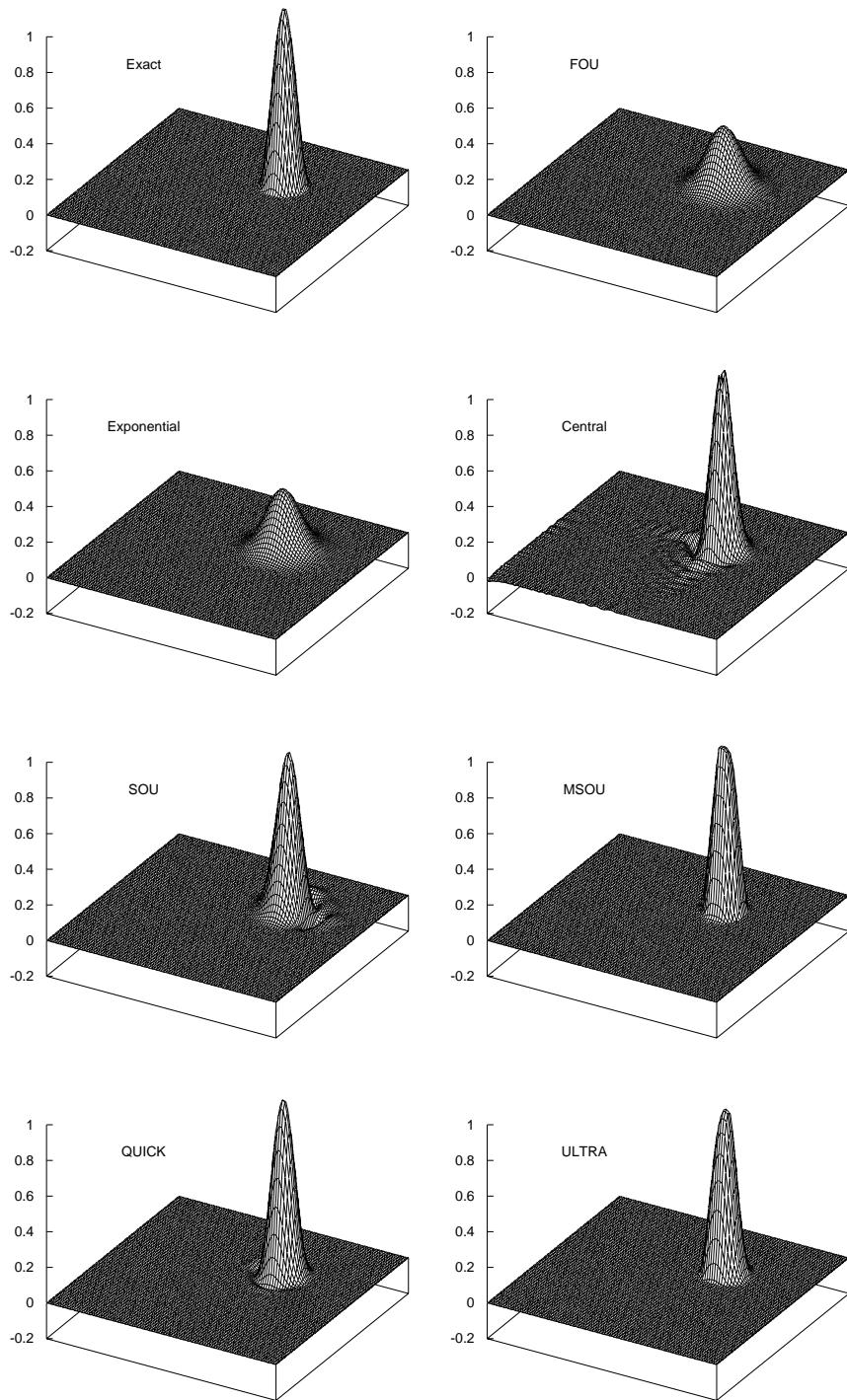


Figure 2.12: The final scalar field after the 45° flow test. The plots are (left to right, top to bottom), The exact solution, and solutions calculated using the FOU, Exponential, Central, SOU, MSOU, QUICK and ULTRA differencing schemes.

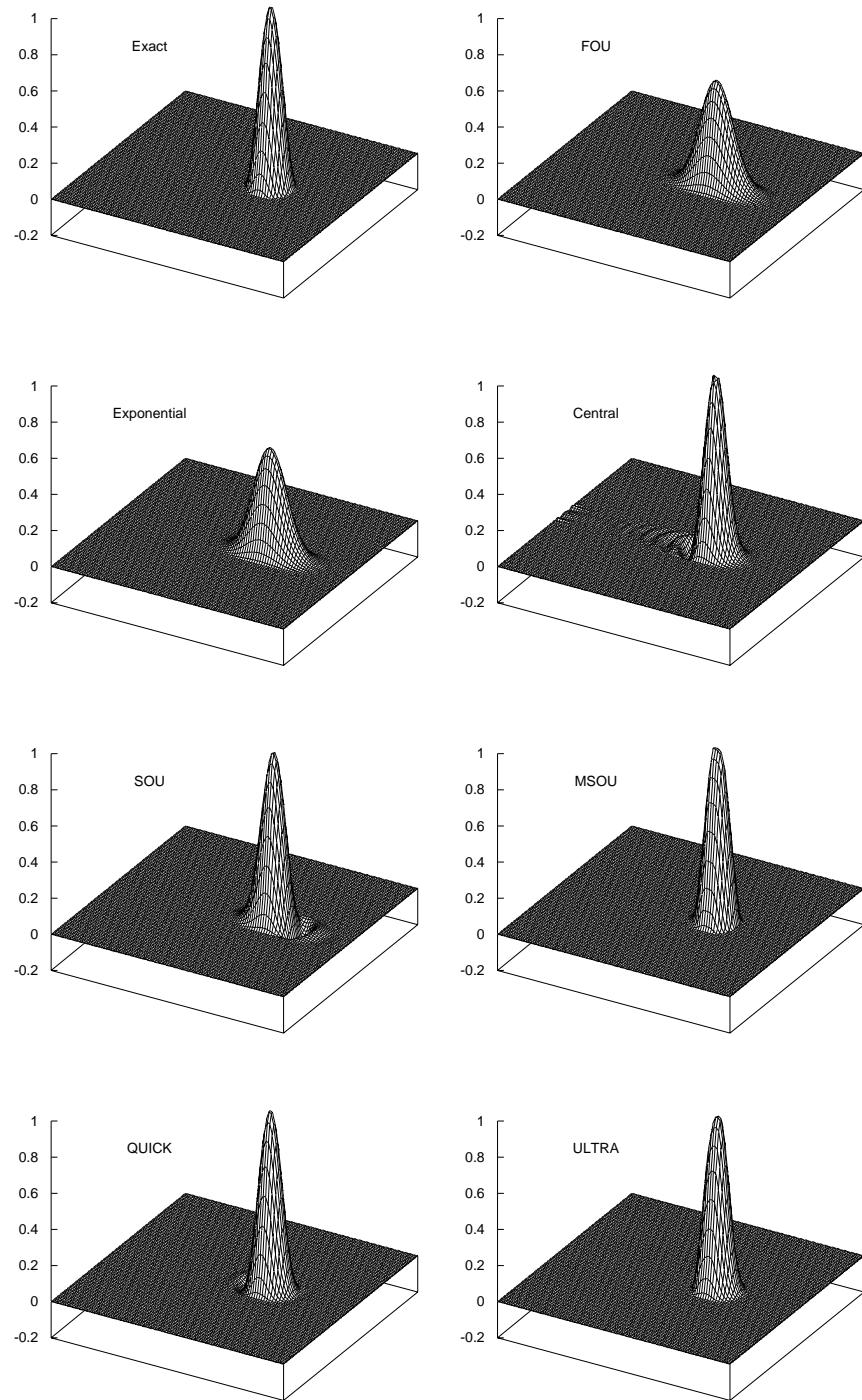


Figure 2.13: The final scalar field after the parallel flow test. The plots are (left to right, top to bottom), The exact solution, and solutions calculated using the FOU, Exponential, Central, SOU, MSOU, QUICK and ULTRA differencing schemes.

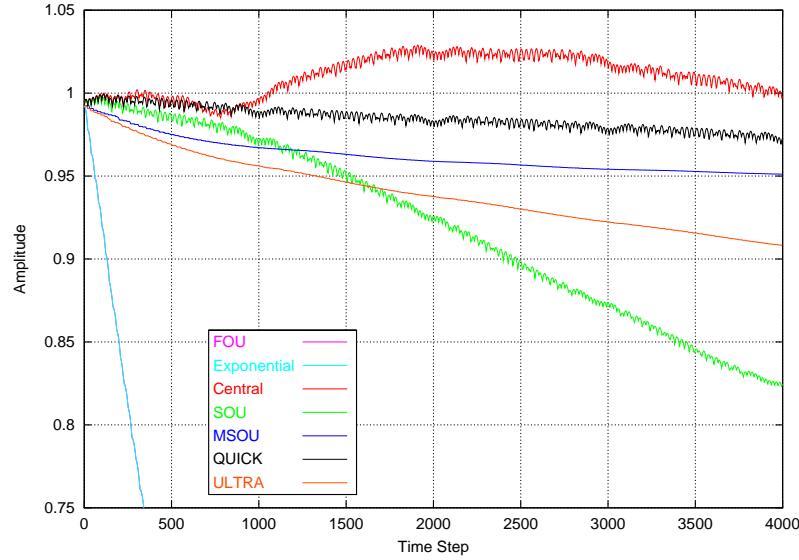


Figure 2.14: The amplitude of the rotating scalar cone plotted as a function of time step. The problem is calculated on a 130×130 with a Courant number of 0.1 using Crank–Nicolson time stepping. Note that the Exponential scheme transient overlays that calculated using the FOU scheme.

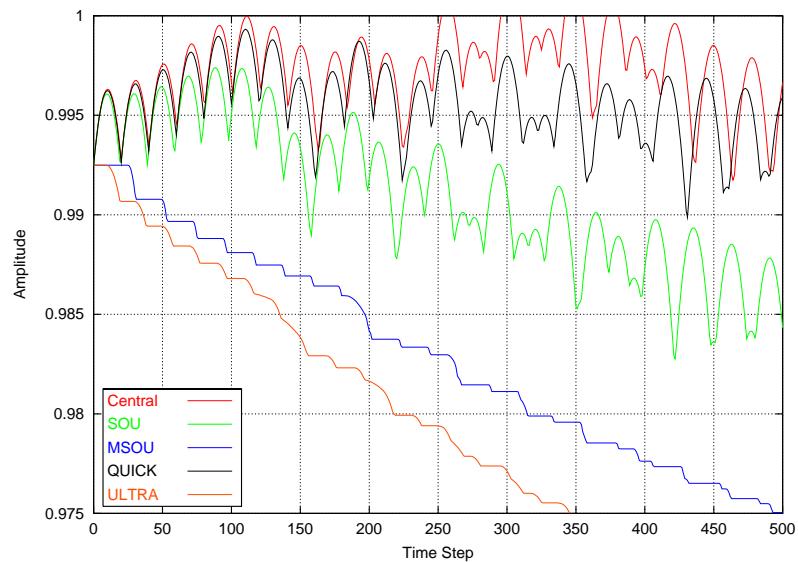


Figure 2.15: Detail of the amplitude transient shown in Figure 2.14. Note the staircase appearance of the MSOU and ULTRA flux limited solutions.

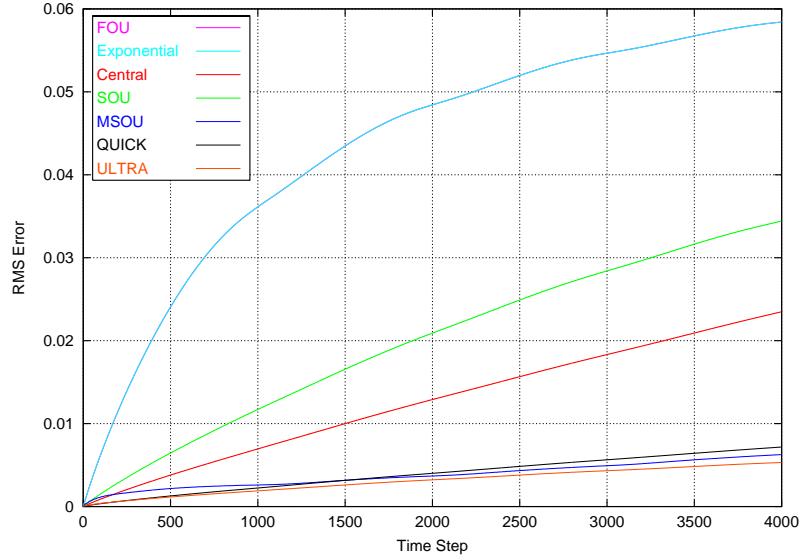


Figure 2.16: The rms error of the rotating scalar cone plotted as a function of time. The problem is calculated on a 130×130 with a Courant number of 0.1 using Crank–Nicolson time stepping.

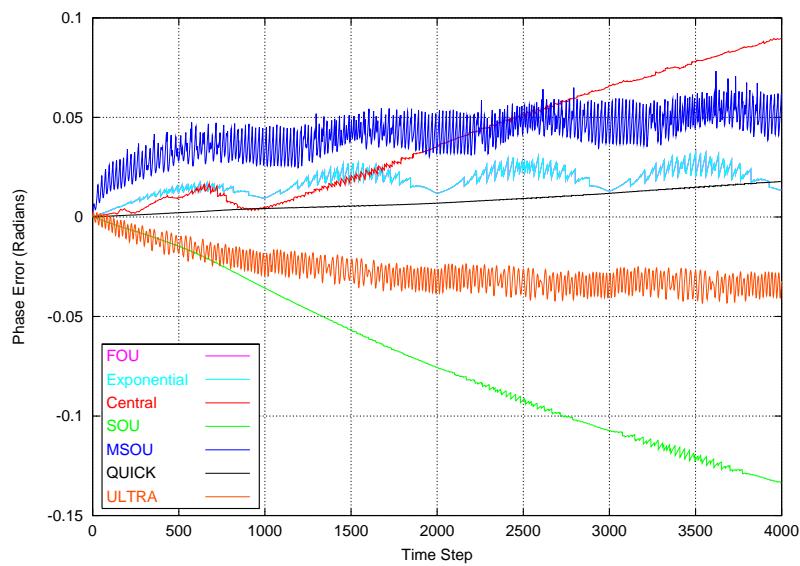


Figure 2.17: The phase error (in radians) of the rotating scalar cone plotted as a function of time. The problem is calculated on a 130×130 mesh with a Courant number of 0.1 using Crank–Nicolson time stepping.

2.5.2 The Smith–Hutton Problem

The Smith–Hutton problem[157] was designed to test advection–diffusion discretisation schemes, and provides a simple problem with a strong discontinuity in a scalar profile and flow that is not parallel to the boundaries of the domain being tested. As such it should reveal the poor convergence of the first order schemes, which exhibit false diffusion on flow that is not parallel to the grid, whilst the sharp gradient should generate oscillations in the solutions generated using the second and third order schemes.

The steady transport equation is solved in the region $-1 \leq x \leq 1, 0 \leq y \leq 1$, with the streamfunction being specified as

$$\psi = -(1 - x^2)(1 - y^2), \quad (2.129)$$

which is shown in Figure 2.18. This streamfunction gives a velocity field of

$$\begin{aligned} u &= 2y(1 - x^2), \\ v &= -2x(1 - y^2). \end{aligned} \quad (2.130)$$

The velocity field has an inlet flow over the region $-1 \leq x \leq 0$ on $y = 0$, with the flow exiting the domain over $0 \leq x \leq 1$ on $y = 0$.

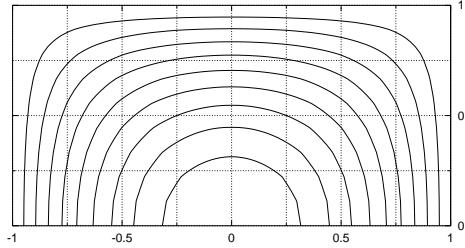


Figure 2.18: The streamfunction for the Smith-Hutton problem. Flow is inward over $-1 \leq x \leq 0 : y = 0$ whilst $0 \leq x \leq 1 : y = 0$ is an outlet.

The scalar ϕ is solved over the domain, with the value of ϕ being prescribed at the inlet and on the left, right and top boundaries, whilst on the outlet the derivative of ϕ normal to the boundary is set to zero. The inlet profile is given as

$$\phi = 1 + \tanh[\alpha(2x + 1)] \quad : \quad y = 0 \quad -1 \leq x \leq 0, \quad (2.131)$$

where α is a parameter that defines the sharpness of the inlet profile. The other boundaries are prescribed as

$$\phi = 1 - \tanh \alpha \quad : \quad \begin{cases} x = -1 & 0 \leq y \leq 1 \\ y = 1 & -1 \leq x \leq 1 \\ x = 1 & 0 \leq y \leq 1. \end{cases} \quad (2.132)$$

Thus ϕ is 0 on $x = \pm 1$ and $y = 1$, and is 2 at the origin. At the outlet a zero normal derivative is prescribed

$$\frac{d\phi}{dy} = 0 \quad : \quad y = 0 \quad 0 \leq x \leq 1. \quad (2.133)$$

The two parameters which define the scalar field are the Péclet number $\text{Pe} = \frac{UL}{\Gamma}$, which specifies the diffusivity of the problem, and α which is a parameter that defines the sharpness of the inlet profile. For the tests described here $\alpha = 100$ and $\text{Pe} = 500$ which were the parameters used in the tests by

Leonard[94]. The problem was modelled using the differencing schemes described in this chapter, with the number of mesh volumes varying between 16×8 to 258×128 (including boundary cells). A solution obtained using the ULTRA differencing scheme on a mesh of 514×258 was used as the benchmark against which the other solutions were compared. The scalar distribution along the x axis from this fine mesh solution is shown in Figure 2.19, whilst the scalar field calculated using a coarser 64×32 mesh is shown in Figure 2.20.

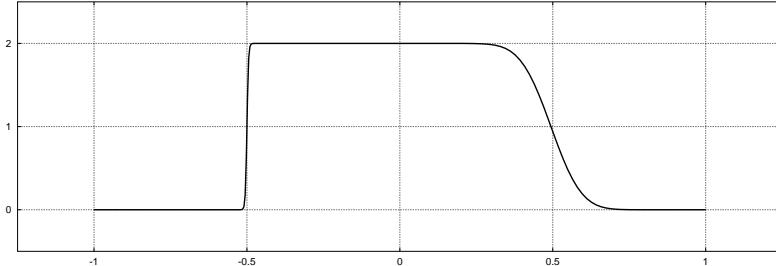


Figure 2.19: The profile of ϕ along the x axis for $\text{Pe} = 500$ and $\alpha = 100$. For $-1 \leq x \leq 0$ the fluid flows into the domain and ϕ is prescribed. The $0 \leq x \leq 1$ region has the fluid flowing out of the domain, and a derivative boundary condition of $\frac{\partial \phi}{\partial y} = 0$ is applied.

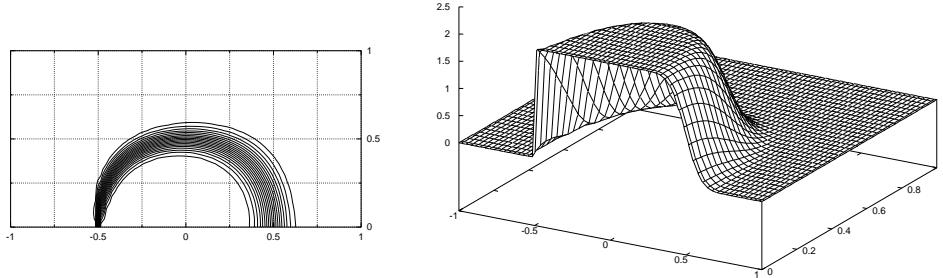


Figure 2.20: The distribution of ϕ through the domain for $\text{Pe} = 500, \alpha = 100$. The solution is calculated on a 66×34 mesh using the ULTRA differencing scheme.

The calculated outlet profiles for the different differencing schemes are given in Figure 2.21. For each differencing scheme the flow was calculated on a range of meshes, and these plots show the reduction in error as the mesh is refined, with the fine mesh solution being given as a reference in each graph. For reasons of space no solution is given for the Power differencing scheme, since it's solutions were identical to those given by the Exponential method.

The first order schemes are grossly in error for the coarse mesh solutions, with the overly diffusive nature of the methods smearing out the discontinuity in the profile of ϕ , this error slowly reducing as the mesh is refined. The FOU scheme is the worst of the methods and still has appreciable error on the finest mesh that a solution was calculated upon. The fine mesh Exponential and Hybrid solutions are better than the FOU solutions, but they are still slow to converge. The first order methods do however give a smooth solution with no overshoots, with their solutions remaining bounded by $0 \leq \phi \leq 2$.

Of the second order methods the Central difference scheme is the fastest to converge to the fine mesh solution. On the coarse meshes however it does exhibit a minor overshoot in the values of ϕ on either side of the discontinuity, with the values of ϕ becoming greater than 2 and less than 0. However, as

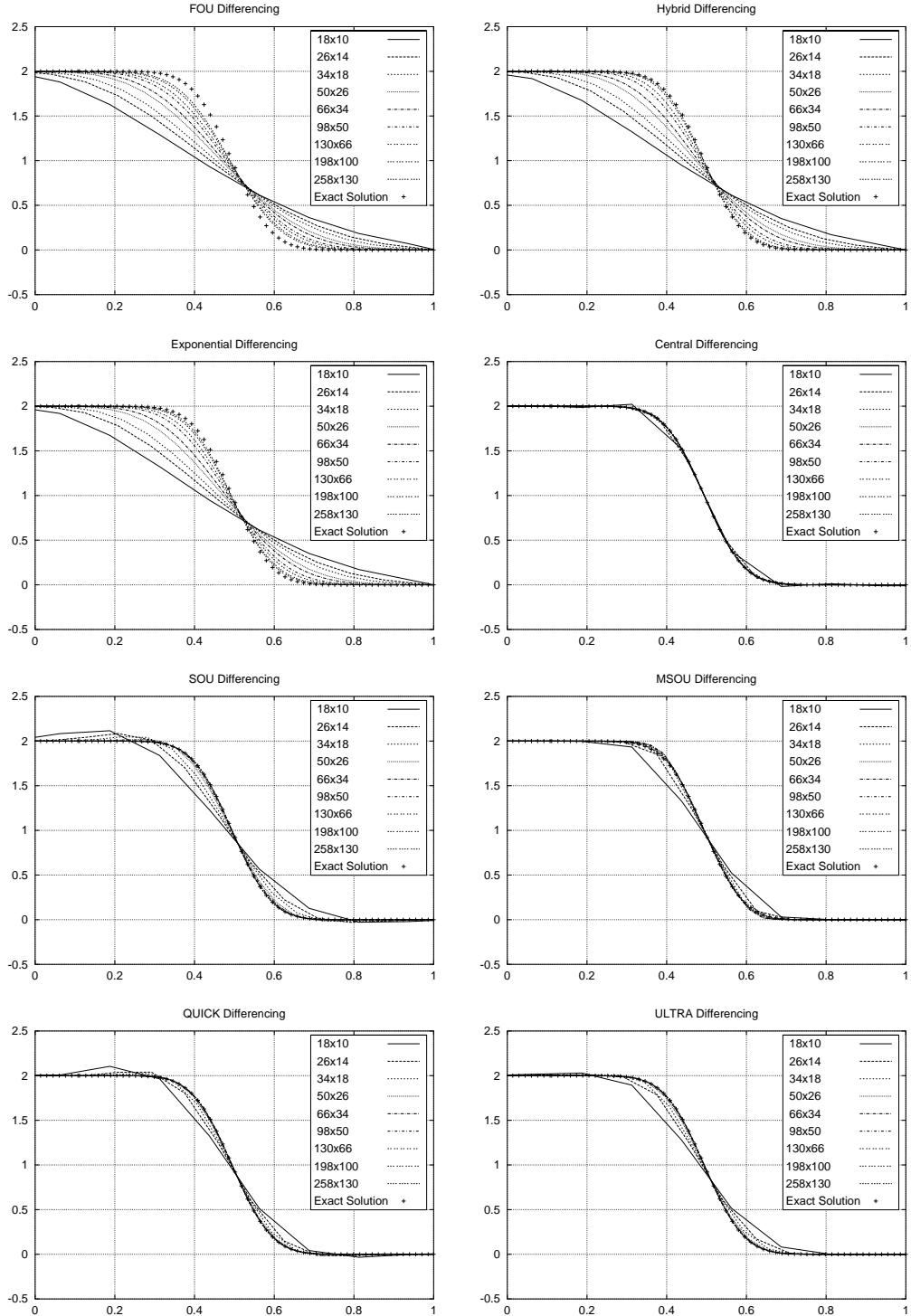


Figure 2.21: Profiles of ϕ at the outlet ($0 \pm x \pm 1, y = 0$) calculated using difference differencing schemes for a range of mesh sizes, and compared with a fine mesh solution. The schemes used are (clockwise from top right), Hybrid, Central, MSOU, ULTRA, QUICK, SOU, Exponential and FOU differencing. For reasons of space the Power differencing scheme has been omitted.

the mesh is refined these oscillations disappear from the solutions. The SOU scheme displays much larger overshoots and is slower to converge than the central difference scheme. The MSOU scheme features no such overshoots, and is faster to converge than the SOU scheme. It is however slower to converge than the central differencing scheme, and it seems to over-sharpen the discontinuity in the ϕ profile, with the values of ϕ in $0.3 < x < 0.4$ having too great a value, whilst the values in $0.6 < x < 0.7$ are underpredicted.

The third order QUICK scheme also gives an oscillatory solution on coarse meshes, with the values of ϕ falling outside the range $0 \leq \phi \leq 2$. However, it does converge to the fine mesh solution at a faster rate than the SOU scheme. The ULTRA flux limiter scheme features a minor overshoot on the coarsest meshes, but converges at a similar rate to the QUICK scheme, and it does not exhibit the over-sharpening behaviour of the MSOU flux limited scheme.

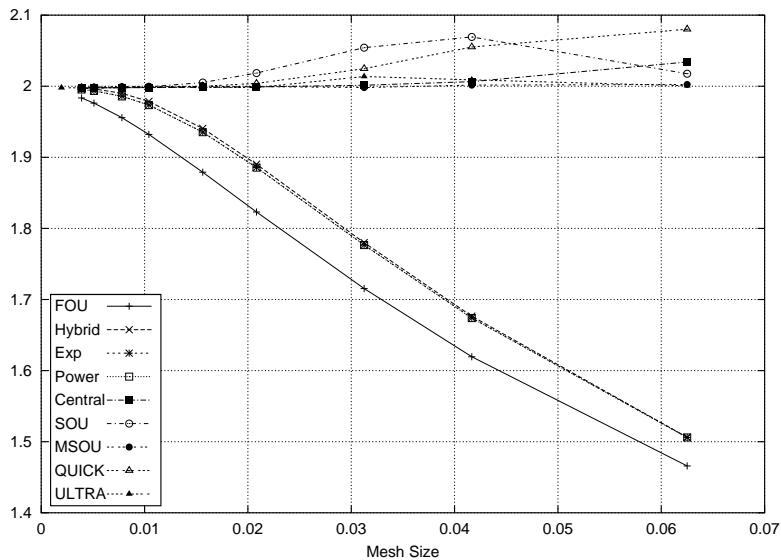


Figure 2.22: The value of the scalar ϕ at the location $(0.25, 0)$ on the flow outlet. The calculated value is plotted as a function of mesh size to show the convergence of the solution as the mesh is refined.

To show the rate of convergence, the value of the scalar ϕ as calculated at $(0.25, 0)$ on the outlet is plotted in Figure 2.22 as a function of the mesh size Δx . Plots are made for each of the differencing schemes described in this Chapter.

The slow convergence of the first order schemes is readily apparent, with the excessive diffusivities of the methods causing an underprediction of the scalar value at this location. The Power and Exponential schemes give almost identical solutions, which in turn are similar to those calculated using the Hybrid scheme.

In contrast the oscillatory nature of the higher order schemes causes an overprediction of the scalar value, with the coarse mesh solutions calculated with the Central, SOU and QUICK differencing schemes giving solutions greater than 2, the SOU and QUICK schemes being the worst offenders in this regard. However, the magnitude of the error from these schemes is still less than those of the first order methods calculated on the same mesh.

Finally the flux limited higher order methods, the ULTRA and MSOU schemes, show little overshoot, and converge rapidly to the exact solution, although the ULTRA scheme does show some overshoot on the 66×34 mesh (with a mesh size of $1/32 = 0.03125$).

2.6 Conclusions

The transient and steady state transport equations have been discretised using finite volume techniques, with a number of different approximations being employed in the discretisation of the advective and transient terms of the equations.

For the transient term four methods were described; the first order Forward and Backward Euler schemes, and the second order Crank–Nicolson and Adams–Bashforth schemes.

As for the discretisation of the advective components, a total of nine schemes were derived and described. Four were first order methods; the First Order Upwind (FOU), Hybrid, Exponential and Power Law schemes. Three of the methods were second order schemes; Central, Second Order Upwind (SOU), and Monotonic Second Order Upwind (MSOU) differencing, the last named scheme being a flux limited method. The final two methods were third order schemes; the QUICK and ULTRA methods of Leonard, the latter being a flux limited scheme.

To test the relative accuracy of the methods they were employed to solve two test problems, one involving the transient advection of a scalar field in the absence of diffusion (the Witches Hat problem), the second modelling steady-state advection-diffusion (the Smith–Hutton problem).

For the transient tests, the second order temporal differencing schemes, the Crank–Nicolson and Adams–Bashforth methods, showed a rapid reduction in error as the time step was decreased. In comparison the first order methods exhibited a much slower convergence. For Courant numbers less than 0.1 (with the second order schemes) or 0.01 (for the first order schemes) the error showed negligible variation with respect to the Courant number, and the error seemed to be mainly a function of mesh coarseness.

With regards to the advection discretisations, for both the transient and steady state tests the plain second and third order advective schemes (Central, SOU and QUICK differencing) displayed oscillatory solutions, with under- and over-shoots to the solution in the Smith–Hutton problem, and wavy wakes in the transient tests. The Central differencing scheme was particularly bad, with the waves filling the entire domain for one transient test case. However the solutions displayed little loss of amplitude in the transient problem, gave a good approximation to the gradients in the Smith–Hutton steady-state problem, and converged rapidly as the mesh was refined.

In comparison, the first order techniques were non-oscillatory but excessively diffusive, showing very slow rates of convergence. The excessive diffusivity resulted in a large loss of amplitude for the transient problems, with the amplitude being reduced to 13% of its correct value in one test.

The flux limited higher order techniques, the MSOU and ULTRA schemes, effectively suppressed the oscillatory nature of the other second and third order techniques, and showed a similar fast convergence with mesh refinement. However, for the transient tests they showed a greater loss in amplitude compared to the schemes without flux limiters, with the amplitude of the scalar distribution exhibiting a curious “staircasing” effect as the scalar was advected from cell to cell. In addition, the scalar distribution showed some distortion, with the MSOU scheme in particular flattening the sharp peak of the scalar distribution.

Overall, the flux limited second and third order schemes (MSOU and ULTRA) gave the best performance, along with the third order upwind (QUICK) and the second order Central differencing schemes. In terms of amplitude the Central differencing and QUICK schemes suffered from the least error, whilst in terms of RMS error the MSOU and QUICK schemes were the best.