INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 18, 313–331 (1994)

TREATMENT OF NUMERICAL DIFFUSION IN STRONG CONVECTIVE FLOWS

G. ARAMPATZIS AND D. ASSIMACOPOULOS

Section II, Department of Chemical Engineering, National Technical University of Athens, Athens 157-73, Greece

E. MITSOULIS*

Department of Chemical Engineering, University of Ottawa, Ottawa, Ont. K1N 6N5, Canada

SUMMARY

A three-dimensional extension of the QUICK scheme adapted for the finite volume method and non-uniform grids is presented to handle convection–diffusion problems for high Peclet numbers and steep gradients. The algorithm is based on three-dimensional quadratic interpolation functions in which the transverse curvature terms are maintained and the diagonal dominance of the coefficient matrix is preserved. All formulae are explicitly given in an appendix. Results obtained with the classical upwind (UDS), the simplified QUICK (transverse terms neglected) and the present full QUICK schemes are given for two benchmark problems, one two-dimensional, steady state and the other three-dimensional, unsteady state. Both QUICK schemes are shown to give superior solutions compared with the UDS in terms of accuracy and efficiency. The full QUICK scheme performs better than the simplified QUICK, giving even for coarse grids acceptable results closer to the analytical solutions, while the computational time is not affected much.

KEY WORDS Convective transport Upwinding QUICK scheme Benchmark problems Finite volume method

1. INTRODUCTION

Computational techniques for the solution of fluid problems have reached a level of maturity which permits safe and accurate predictions. A major challenge in the development of a numerical model lies in the treatment of the convection terms. The application of the classical central difference scheme leads to instabilities wherever the convection mechanism dominates over diffusion (phenomena at large Paclet numbers).[1,2] The use of the alternative upwind scheme,[1] although it ensures stability for any value of the Peclet number, is burdened by its lack of accuracy, introducing an additional diffusion term commonly referred to as 'numerical diffusion'.[2,3] This problem may become extremely serious for unsteady, three-dimensional (3D) flows.

With the aim to reduce numerical diffusion and at the same time to secure stability, Leonard[2] developed a discretization scheme which uses quadratic interpolation functions (QUICK). Use of this scheme in a number of one- and two-dimensional problems, in conjunction with an explicit time intergration method, showed an improved behaviour.

Several attempts followed in order to apply the QUICK scheme with an implicit time integration method.[4–11] The implicit form of the QUICK scheme presents stability problems,

* Author to whom correspondence should be addressed.

0271–2091/94/030313–19$14.50

Received 26 January 1993
Revised 28 July 1993

because under conditions of highly convective flows the coefficient matrix resulting from the finite difference equations may lose its diagonal dominance. Han *et al.*,[4] in an effort to express exactly the convected face values of the dependent variable in the way suggested by Leonard,[2] discovered that they could obtain converged solutions only upon the introduction of a pseudosource term in the equations. The type of pseudosources is entirely problem-dependent and no general formula can be derived for other cases. Han *et al.*[4] tried a recasting of the relationships given by Leonard,[2] which was effective only in the case of low Reynolds numbers.

On the basis of these observations, Pollard and Siu[5] realized that the stability problem was caused by the coefficients of the linear system of equations, which under certain conditions may assume negative values. They proceeded to a recasting of the finite difference equations in order to secure positive definite coefficients; this version of the QUICK scheme was named QUICKER. A different approach was used by Shyy,[6] who introduced an appropriately normalized relaxation factor and obtained satisfactory results for a wide Peclet range.

It should be noted that all the above efforts apply to one- and two-dimensional problems with uniform grids. They are also based on the simplified QUICK version, in which all transverse curvature terms in the expressions of the face values are eliminated.

Freitas *et al.*[7] developed a three-dimensional QUICK version for non-uniform grids, in which the diagonal dominance of the coefficient matrix is preserved in the way suggested by Pollard and Siu.[5] Their analysis, however, is based on one-dimensional interpolation functions, which corresponds to the elimination of the transverse curvature terms according to the simplified QUICK version.

In the present work a new variant of the 3D QUICK scheme is developed for the general case of non-uniform grids. It uses 3D quadratic interpolation functions and can be easily incorporated in an explicit or implicit time intergration algorithm. Since implicit integration leads to the solution of a 'linearized' system of equations, a special procedure is adopted so that a convergent solution can be ensured. All formulae are explicitly given in the Appendix for completeness.

Two benchmark problems are studied for the evaluation of different algorithms applied to highly convective flows. These benchmark problems combine characteristics that favour numerical diffusion, i.e. three-dimensionality and steep gradients. Their simple formulation permits comparison of numerical results and analytical solutions. A computer programme was developed and results for the upwind, simplified QUICK and full QUICK schemes are presented and compared.

2. MATHEMATICAL FORMULATION

2.1. Convection–diffusion equation

The general, unsteady, convection–diffusion equation describing the conservation of the scalar quantity φ written in Cartesian tensorial form can be stated as

$$\frac{\partial}{\partial t}(\rho\varphi) + \frac{\partial}{\partial x_j}\left(\rho u_j \varphi - \Gamma \frac{\partial \varphi}{\partial x_j}\right) = S, \tag{1}$$

where ρ is the density, u_j is the velocity vector having components u, v and w in the directions $x_1 = x$, $x_2 = y$ and $x_3 = z$ respectively, Γ is the exchange coefficient and S is a source term. It can be shown[1] that the momentum and continuity equations may be expressed in the above form, where for different φs the exchange coefficient Γ and the source term S take on different values.

2.2. Finite volume formulation

Equation (1) is solved with the appropriate boundary conditions by first integrating it over a finite number of control volumes (Figure 1) for the time interval from $n\Delta t$ to $(n+1)\Delta t$. Using the divergence theorem, the volume integral of the terms under the differential operator may be converted into surface integrals representing fluxes over the faces of each control volume according to

$$\frac{(\overline{\rho\varphi_P^{n+1}} - \overline{\rho\varphi_P^n})V_P}{\Delta t} = \left(\rho u\varphi - \Gamma\frac{\partial\varphi}{\partial x}\right)_w A_w - \left(\rho u\varphi - \Gamma\frac{\partial\varphi}{\partial x}\right)_e A_e + \left(\rho v\varphi - \Gamma\frac{\partial\varphi}{\partial y}\right)_s A_s - \left(\rho v\varphi - \Gamma\frac{\partial\varphi}{\partial y}\right)_n A_n + \left(\rho w\varphi - \Gamma\frac{\partial\varphi}{\partial z}\right)_l A_l - \left(\rho w\varphi - \Gamma\frac{\partial\varphi}{\partial z}\right)_h A_h + S^*, \quad (2)$$

where the A_is represent the areas of the cell faces and V_P is the cell volume. The overbars in the terms on the left-hand side denote average values in the control volume, S^* is a space–time-averaged source term and the other terms on the right-hand side represent time- and area-averaged values.

Equation (2) represents the balance of fluxes over the typical control volume for node P and is an exact equation. The numerical problem becomes one of estimating the local space–time behaviour of φ in the vicinity of each control volume in order to evaluate the indicated average values.

The time-averaged quantities are approximated by their values at time level $(n+1)\Delta t$ (fully implicit scheme). The standard practice is to approximate the diffusion terms using the central difference scheme, where a piecewise linear profile for the variation of φ is assumed. Hence the term representing the diffusion flux through the east face takes the form

$$\left(\Gamma\frac{\partial\varphi}{\partial x}\right)_e A_e = D_e(\varphi_E - \varphi_P), \quad (3)$$

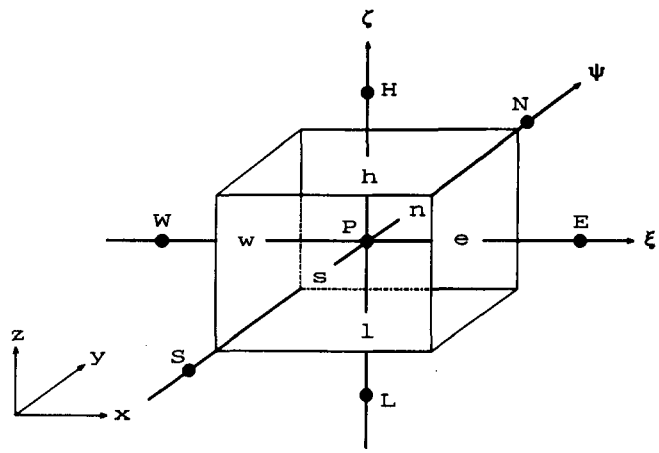

Figure 1. Typical control volume for node P. Capital letters denote neighbouring nodes while small letters denote volume faces.

where

$$D_e = \frac{\Gamma_e A_e}{x_E - x_P}. \tag{4}$$

The source term is assumed to be represented by the linear form

$$S^* = S_U + S_P \varphi_P \tag{5}$$

for the general case where the source may depend on the actual value of φ.

In this work attention is focused on the convection terms and the schemes that are used to approximate the convected face values of the dependent variable (i.e. φ_e for the east cell face).

If the computational mesh were refined infinitely, one would expect to get the exact solution of the differential equation irrespective of the differencing scheme used to evaluate the convective fluxes. In practice one can only employ a finite number of cells and it is then important to ensure that the resulting discretized equations and their solution satisfy certain desirable properties such as conservativeness,[12] convective stability,[2] transportiveness[12] and boundedness.[13,14]

The last property is generally associated with the condition of diagonal dominance of the coefficient matrix. If diagonal dominance holds, then φ_P is a weighted average of its neighbours and any non-physical overshoots or undershoots are avoided. However, this is only a sufficient condition for guaranteeing boundedness and in some cases perfectly acceptable results are obtained from schemes not satisfying this condition.[13]

2.3. Upwind differencing scheme (UDS)

The upwind differencing scheme (also known as the donor cell scheme) is based on the assumption that the convected cell face value is equal to that at the upstream cell along the same co-ordinate direction. This assumption leads to the following computational formula for φ_e:

$$\varphi_e = \begin{cases} \varphi_P & \text{if } u_e \geqslant 0, \\ \varphi_E & \text{if } u_e < 0. \end{cases} \tag{6}$$

This scheme is broadly used mainly because of its simplicity and excellent stability properties. It can be shown[1] that the resulting coefficient matrix satisfies the diagonal dominance criterion. It is also obvious that the transportive property is fully accounted for (no reference is made to downstream cells) and therefore convective stability is always ensured. However, stability is actually being achieved through the introduction (injection) of a stabilizing second-order diffusion term (false or numerical diffusion).[2] This term, which dominates physical diffusion under high-convection conditions, is responsible for many numerical inaccuracies.

2.4. Quadratic upwind differencing scheme (QUICK)

QUICK is a third-order, upwind-biased scheme which possesses the convective stability of the first-order UDS but is free of its second-order numerical diffusion errors. The development proceeds by first defining a general three-dimensional quadratic interpolating polynomial function locally in every control volume of the following form:

$$\varphi(\xi, \psi, \zeta)_P = k_1 + k_2\xi + k_3\xi^2 + k_4\psi + k_5\psi^2 + k_6\zeta + k_7\zeta^2, \tag{7}$$

where (ξ, ψ, ζ) is a local co-ordinate system attached to node P (Figure 1).

The seven coefficients k_i are determined by the values of φ at the current control volume (P) and its six neighbours (W, E, S, N, L and H in Figure 1):

$$k_1 = \varphi_P, \tag{8}$$

$$k_2 = \frac{1}{\xi_E - \xi_W}\left(-\frac{\xi_E}{\xi_W}(\varphi_P - \varphi_W) - \frac{\xi_W}{\xi_E}(\varphi_E - \varphi_P)\right), \tag{9}$$

$$k_3 = \frac{1}{\xi_E - \xi_W}\left(\frac{1}{\xi_W}(\varphi_P - \varphi_W) + \frac{1}{\xi_E}(\varphi_E - \varphi_P)\right), \tag{10}$$

$$k_4 = \frac{1}{\psi_N - \psi_S}\left(-\frac{\psi_N}{\psi_S}(\varphi_P - \varphi_S) - \frac{\psi_S}{\psi_N}(\varphi_N - \varphi_P)\right), \tag{11}$$

$$k_5 = \frac{1}{\psi_N - \psi_S}\left(\frac{1}{\psi_S}(\varphi_P - \varphi_S) + \frac{1}{\psi_N}(\varphi_N - \varphi_P)\right), \tag{12}$$

$$k_6 = \frac{1}{\zeta_H - \zeta_L}\left(-\frac{\zeta_H}{\zeta_L}(\varphi_P - \varphi_L) - \frac{\zeta_L}{\zeta_H}(\varphi_H - \varphi_P)\right), \tag{13}$$

$$k_7 = \frac{1}{\zeta_H - \zeta_L}\left(\frac{1}{\zeta_L}(\varphi_P - \varphi_L) + \frac{1}{\zeta_H}(\varphi_H - \varphi_P)\right). \tag{14}$$

The convected value of φ on the common surface between two consecutive control volumes is evaluated by integrating the interpolating function of the upstream control volume over this surface:

$$\varphi_e = \begin{cases} \dfrac{1}{\Delta y_P \Delta z_P}\displaystyle\int_{-\Delta y_P/2}^{\Delta y_P/2}\int_{-\Delta z_P/2}^{\Delta z_P/2} \varphi(\xi_e, \psi, \zeta)_P \, d\psi \, d\zeta & \text{if } u_e \geqslant 0, \\ \dfrac{1}{\Delta y_E \Delta z_E}\displaystyle\int_{-\Delta y_E/2}^{\Delta y_E/2}\int_{-\Delta z_E/2}^{\Delta z_E/2} \varphi(\xi_e, \psi, \zeta)_E \, d\psi \, d\zeta & \text{if } u_e < 0. \end{cases} \tag{15}$$

Substituting (7)–(14) into equation (15) results in the following formula for φ_e:

$$\varphi_e = \begin{cases} \varphi_P + QAE_P(\varphi_P - \varphi_W) + QBE_P(\varphi_E - \varphi_P) + QC_P(\varphi_S - \varphi_P) & \text{if } u_e \geqslant 0, \\ \quad + QD_P(\varphi_N - \varphi_P) + QE_P(\varphi_L - \varphi_P) + QF_P(\varphi_H - \varphi_P) & \\ \varphi_E + QAW_E(\varphi_P - \varphi_E) + QBW_E(\varphi_E - \varphi_{EE}) + QC_E(\varphi_{SE} - \varphi_E) & \text{if } u_e < 0, \\ \quad + QD_E(\varphi_{NE} - \varphi_E) + QE_E(\varphi_{LE} - \varphi_E) + QF_E(\varphi_{HE} - \varphi_E) & \end{cases} \tag{16}$$

where QAE_P, QBE_P, etc. are appropriate geometric coefficients arising from the integration. These coefficients depend only on the geometry and are evaluated only once at the start of the calculations. Furthermore, because they are one-dimensional, they do not require a lot of computational space. Finally, they are always positive, which simplifies the sorting-out process in writing the final set of algebraic equations. The full set of equations is given in the Appendix.

It should be noted that equation (7) does not contain 'cross'-terms in $\xi\psi$, $\psi\zeta$ and $\zeta\xi$, which represent 'twist' in the quadratic interpolating function. However, as pointed out by Leonard,[15] the 'twist' terms cancel out upon integration over the control volume and thus their contribution is zero.

In contrast with the simplified 2D and 3D versions presented in the literature so far, the procedure described above follows exactly the philosophy of the QUICK development. The simplified version of the QUICK scheme is based on one-dimensional interpolation functions. For example, for the x-direction,

$$\varphi(\xi, \psi, \zeta)_P = k_1 + k_2\xi + k_3\xi^2. \tag{17}$$

This practice leads to the following expression for φ_e:

$$\varphi_e = \begin{cases} \varphi_P + QAE_P(\varphi_P - \varphi_W) + QBE_P(\varphi_E - \varphi_P) & \text{if } u_e \geqslant 0, \\ \varphi_E + QAW_E(\varphi_P - \varphi_E) + QBW_E(\varphi_E - \varphi_{EE}) & \text{if } u_e < 0. \end{cases} \tag{18}$$

It can be seen that in equation (18) all the transverse curvature terms $QC_P(\varphi_S - \varphi_P)$, $QD_P(\varphi_N - \varphi_P)$, etc. were eliminated, which drastically simplifies both the handling of the final algebraic equations and their programming. This fact explains why all previous researchers have adopted the simplified QUICK version. In Figure 2 the grid nodes involved in the calculation of the face values are shown for all three schemes. The number of grid nodes involved is seven for the UDS, 13 for the simplified QUICK and 25 for the full QUICK scheme.

Upon substitution of equations (3), (5) and (16) in the finite volume equation (2), a system of algebraic equations is derived, the equation for node P being

$$\alpha_P\varphi_P = \alpha_W\varphi_W + \alpha_E\varphi_E + \alpha_S\varphi_S + \alpha_N\varphi_N + \alpha_L\varphi_L + \alpha_H\varphi_H + b. \tag{19}$$

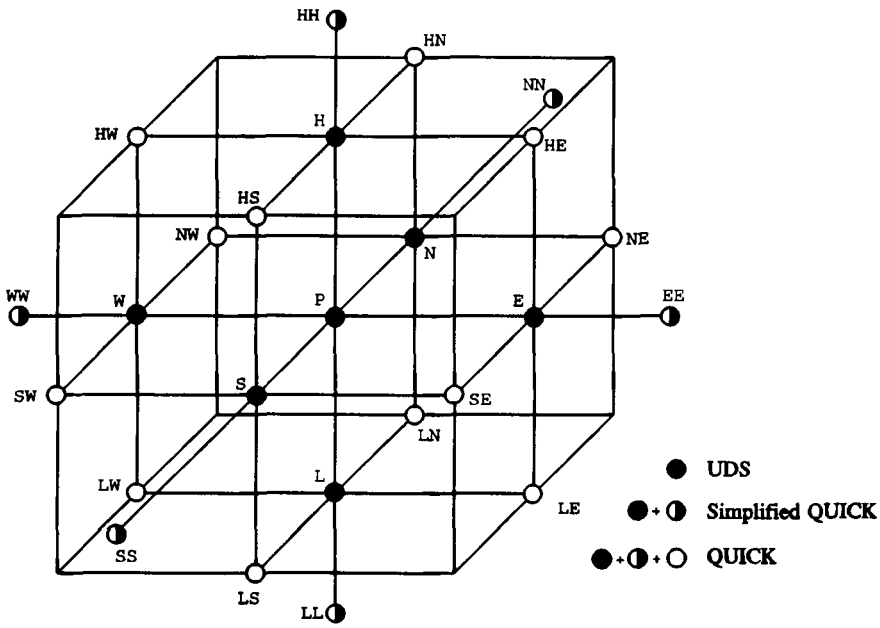

Figure 2. Computational molecules for the three schemes

The form of the equations is suitable for the application of Stone's[16] strongly implicit procedure (SIP). This algorithm was used in this work. The coefficients α_i (i = W, E, S, N, L, H, P) and b are given in the Appendix.

The QUICK scheme partially satisfies the transportive property because the downstream cells appear in equations (16). However, as Leonard[2] proved, it fully satisfies the convective stability criterion. The form of the coefficients α_i shows that in strongly convective flows they can take negative values and the system of algebraic equations loses its diagonal dominance. This leads to 'overshoots' and 'undershoots' in high-Peclet-number problems (unbounded solutions).[2,13,14] It also presents numerical stability problems whenever iterative algorithms (such as SIP) are used. This is why the full QUICK scheme has not been used at high Peclet numbers.[8]

In the present work and in order to overcome the last-mentioned restriction, we keep only those terms which guarantee positive coefficients and relocate all others into the constant term b. With this modification the coefficient α_E, for example, becomes

$$\alpha_E = D_e - C_e^- - C_w^- QBW_P \tag{20}$$

and the contribution of this coefficient to the constant term b is

$$[C_e^- QAW_E - C_e^+ QBE_P + QB_P(C_s^- - C_n^+ + C_l^- - C_h^+)](\varphi_E - \varphi_P). \tag{21}$$

The new pattern of the algebraic equations has a less implicit form, which results in a slower rate of convergence. However, this formulation aids in maintaining the iterative stability of the scheme.

3. TEST CASES

In this section two test cases are presented and the results of the UDS, simplified QUICK and full QUICK are compared. Both test cases concern the transport of a scalar quantity by convection and diffusion. The first one is a two-dimensional, steady state problem and the second is a three-dimensional, transient problem.

3.1. Two-dimensional, steady state problem

The physical problem under investigation is shown schematically in Figure 3. The governing equation is the 2D, steady state analogue of equation (1). The velocities u and v and the exchange coefficient Γ have constant values. The scalar variable φ is zero on three side boundaries whereas it has an exponential distribution on the fourth, i.e.

$$\varphi = 0 \quad \begin{cases} \text{at } x = 0, 0 \le y \le L, \\ \text{at } x = L, 0 \le y \le L, \\ \text{at } y = 0, 0 \le x \le L, \end{cases} \tag{22}$$

$$\varphi = \varphi_0 e^{Pe_x/2} \sin(\pi x) \quad \text{at } y = L,\ 0 \le x \le L.$$

It can be shown that the analytical solution has the form[17]

$$\varphi = \frac{\varphi_0}{e^{r_1} - e^{r_2}} e^{Pe_x/2} \sin(\pi x)(e^{r_1 y} - e^{r_2 y}), \tag{23}$$

where

$$r_{1,2} = \tfrac{1}{2}Pe_y \pm \tfrac{1}{2}\sqrt{(Pe_y^2 + 4\beta)}, \qquad \beta = (4\pi^2 + Pe_x^2)/4 \tag{24}$$

and $Pe_x = \rho uL/\Gamma$ and $Pe_y = \rho vL/\Gamma$ are the Peclet numbers in the x- and y-directions respectively.

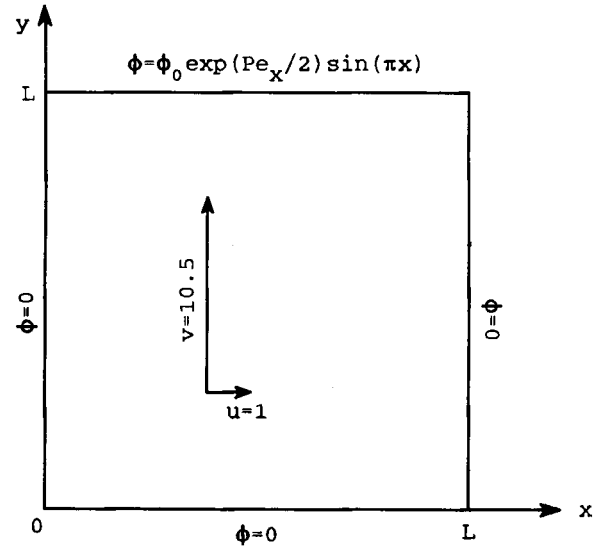

Figure 3. Definition of the two-dimensional, steady state benchmark problem and its boundary conditions

For the simulation presented here the following values have been used: $L = 1$, $\varphi_0 = 1$, $\rho = 1$, $\Gamma = 1$, $u = 1$ and $v = 10{\cdot}5$ (angle of 5·5°), giving rise to Peclet numbers $Pe_x = 1$ and $Pe_y = 10{\cdot}5$. The computational domain consists of 15 cells in each direction ($NX = NY = 15$). All solutions satisfy the convergence criterion

$$\sum_{\text{all nodes}} |\varphi - \varphi_a| \leq 10^{-4}, \tag{25}$$

where φ and φ_a are the numerical and analytical solutions respectively.

In Figure 4 the value of the relative error $|\varphi - \varphi_a|/\varphi_0$ along the y-direction for the location $x = 0{\cdot}5$ is presented using the UDS and full QUICK scheme. The computational grid is uniform. The superiority of QUICK with respect to the UDS is evident as well as the large errors of numerical diffusion shown by the latter. The maximum values of the relative error are 12·5% and 5·6% for the UDS and QUICK respectively.

Both schemes show large errors in the region where steep gradients of the φ-variable exist (large values of y). This fact favours the use of a non-uniform grid (locally denser in the critical region). The following transformation[18] was used for the grid generation:

$$y_i = \alpha \left[\left(\frac{\alpha + 1}{\alpha - 1} \right)^{i/NY} - 1 \right] \Big/ \left[1 + \left(\frac{\alpha + 1}{\alpha - 1} \right)^{i/NY} \right], \tag{26}$$

with the parameter $\alpha = 1{\cdot}1$. Figure 5 depicts the new relative errors. An appreciable error reduction is noticeable, mainly with the QUICK scheme. The UDS errors, although reduced, are still large. The maximum values of the relative error are now 6·8% and 1% for the UDS and QUICK respectively. Thus the full QUICK scheme performs better for both uniform and non-uniform grids. The solution accuracy is remarkable with the QUICK scheme when we consider that it is achieved with relatively coarse grids.

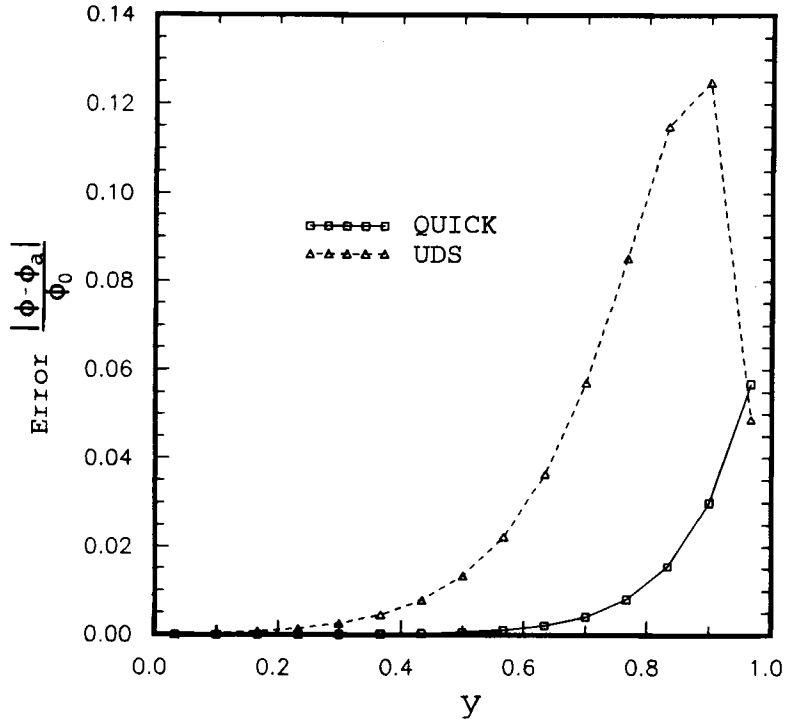

Figure 4. Relative error along y at $x = 0{\cdot}5$ using a uniform grid ($NX = NY = 15$) for the two-dimensional benchmark problem of Figure 3

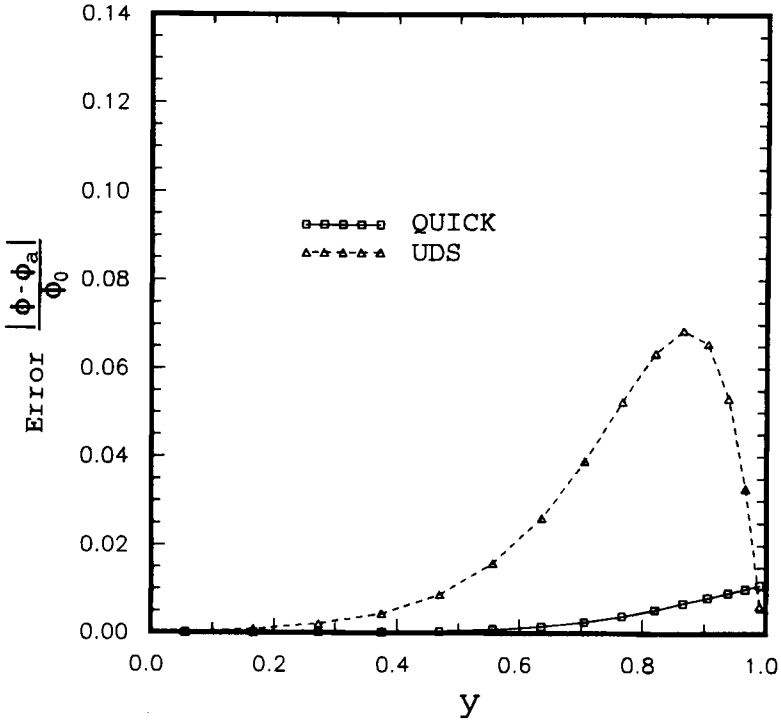

Figure 5. Relative error along y at $x = 0{\cdot}5$ using a non-uniform grid according to equation (26) for the two-dimensional benchmark problem of Figure 3

The differences between the full and simplified QUICK schemes are presented in Figure 6. The ratio of the relative errors of the two schemes shows the superiority of the full QUICK scheme, which reduces errors by as much as 9%.

A decisive characteristic of every arithmetic model is the total central processing unit (CPU) time needed for the completion of the calculations. In order to have a better understanding about the time required with respect to the accuracy achieved, the total percentage error is used here defined by

$$\varepsilon = \left(\sum_{\text{all nodes}} \left| \frac{\varphi - \varphi_a}{\varphi_0} \right| \times 100 \right) \Big/ N, \tag{27}$$

where N is the total number of cells ($NX \times NY$). CPU times and corresponding total errors ε for various grid sizes are presented in Figures 7 and 8 respectively. Time is given on an 80286 personal computer. Although the UDS appears to be faster for the same uniform grid size, its large errors make it finally uneconomical. The error with the UDS for a uniform grid of size 19 × 19 is 1·53, whereas QUICK presents the same error for a grid size of only 7 × 7. The CPU times of the two schemes are 64·21 and 19·28 s respectively and therefore the UDS is 3·33 times more expensive than QUICK in order to produce the same accuracy. The slow rate of convergence of the UDS is evident (Figure 8), which in certain cases may give the false impression that a grid-independent solution has been reached.

It is seen in Figure 7 that as the number of cells becomes larger, the difference in CPU time

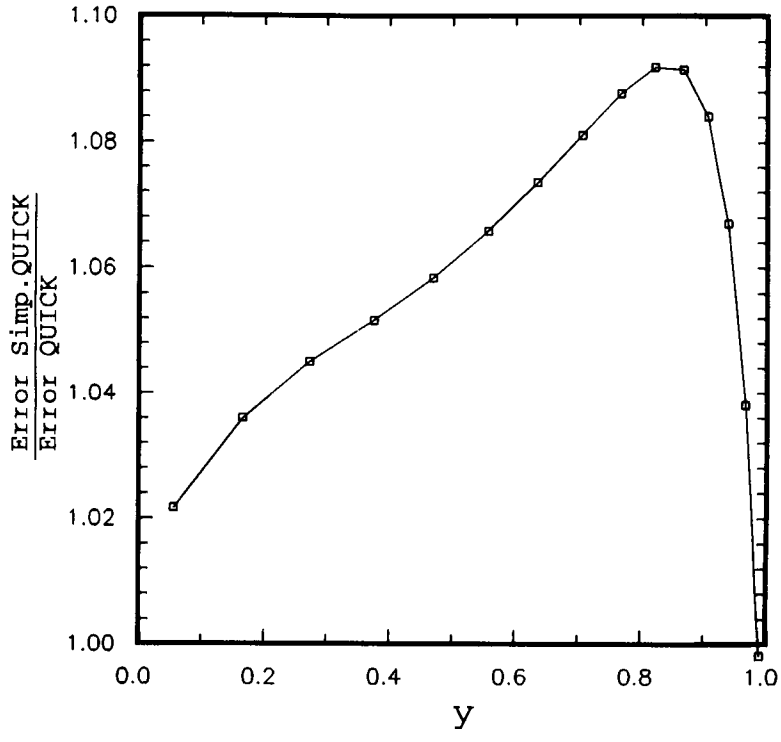

Figure 6. Ratio of relative errors for the full and simplified QUICK schemes along y at $x = 0{\cdot}5$ using a non-uniform grid for the two-dimensional benchmark problem of Figure 3. The discrepancies reach up to 9%

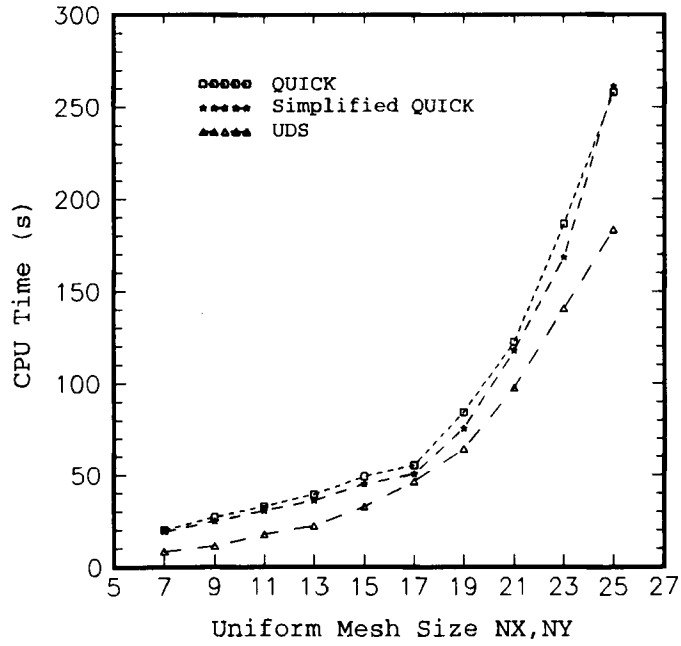

Figure 7. Computation time versus mesh size for the three different schemes used in the two-dimensional benchmark problem of Figure 3 (CPU time given on a PC with an 80286 processor)

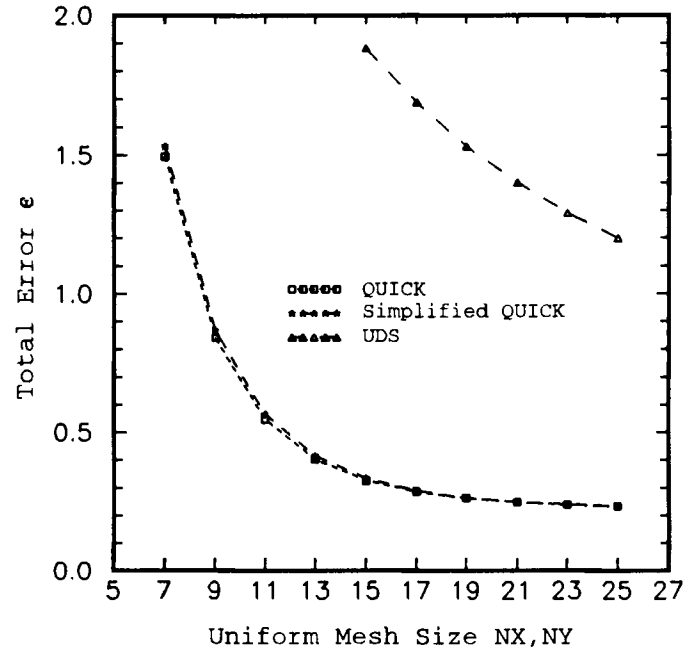

Figure 8. Total error (equation (27)) versus mesh size for the three different schemes used in the two-dimensional benchmark problem of Figure 3

between QUICK and the UDS becomes even larger. This is due to the non-linear (second-order) dependence of CPU time on grid size (NX or NY) and because the UDS requires less computation for the same number of equations. The fact that not much extra CPU time is required between the simplified and full QUICK schemes is not suprising. The extra equations with the geometric coefficients arising from the integration are evaluated only once at the beginning of the calculations, as mentioned earlier. Most of the CPU time is spent on the iterative solution of the system of equations.

Finally, to check the performance of the two QUICK schemes, two other flow direction angles of 25° and 37° have been used ($u = 4{\cdot}5$, $v = 9{\cdot}6$ and $u = 6{\cdot}3$, $v = 8{\cdot}4$). The full QUICK scheme again gave better results, reaching about 10% less in error relative to the simplified scheme. This shows that the performance of the full QUICK scheme presented here is not much influenced by the angle of the flow direction.

3.2. Three-dimensional, unsteady state problem

The physical phenomenon under investigation is the transport of a scalar quantity φ (pullutant concentration) by convection and diffusion in a 3D flow field. Initially φ is zero throughout the flow field. At $t = 0$ a pollutant mass is released instantly in a small volume centred at (x_0, y_0, z_0) with dimensions L_x, L_y and L_z as shown in Figure 9. In this volume φ is initially uniform and equal to φ_0. The air entering the computational domain does not convey any pollutant mass. All the other boundary surfaces have zero-gradient pollutant concentration. The governing

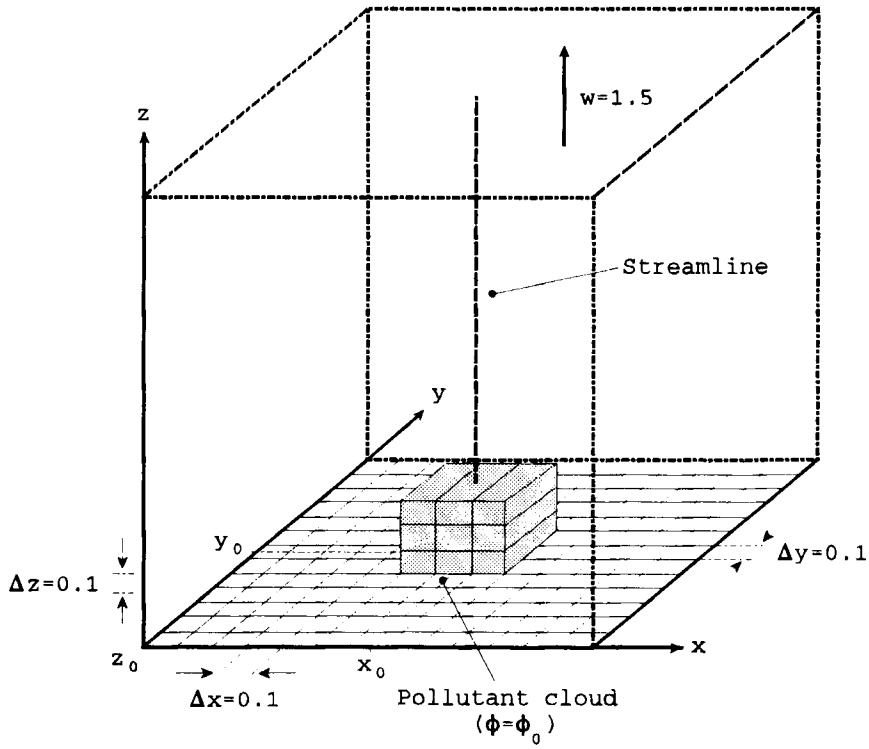

Figure 9. Definition of the three-dimensional, unsteady state benchmark problem and its initial conditions ($NX = NY = NZ = 13$)

equation has the form of (1) with constant u, v, w, ρ and Γ. The analytical solution of the corresponding one-dimensional problem is given by Crank.[19] The solution of the 3D problem is simply the product of the three one-dimensional solutions, i.e.

$$\varphi = \frac{\varphi_0}{8}\left[\operatorname{erf}\left(\frac{L_x/2 - X}{2\sqrt{(\Gamma t/\rho)}}\right) + \operatorname{erf}\left(\frac{L_x/2 + X}{2\sqrt{(\Gamma t/\rho)}}\right)\right]\left[\operatorname{erf}\left(\frac{L_y/2 - Y}{2\sqrt{(\Gamma t/\rho)}}\right) + \operatorname{erf}\left(\frac{L_y/2 + Y}{2\sqrt{(\Gamma t/\rho)}}\right)\right]$$
$$\times\left[\operatorname{erf}\left(\frac{L_z/2 - Z}{2\sqrt{(\Gamma t/\rho)}}\right) + \operatorname{erf}\left(\frac{L_z/2 + Z}{2\sqrt{(\Gamma t/\rho)}}\right)\right], \tag{28}$$

where X, Y and Z are the distances from the centre of the cloud defined by $X = x - x_0 - ut$, $Y = y - y_0 - vt$ and $Z = z - z_0 - wt$. The following non-dimensional quantities are used:

$$\text{concentration } \Phi = \varphi/\varphi_0, \tag{29}$$

$$\text{distance } L_x = (l - l_0 - Vt)/\Delta x, \tag{30}$$

$$\text{time } T = Vt/\Delta x, \tag{31}$$

where $l_0 = \sqrt{(x_0^2 + y_0^2 + z_0^2)}$, $l = \sqrt{(x^2 + y^2 + z^2)}$ and $V = \sqrt{(u^2 + v^2 + w^2)}$.

Results have been obtained for a uniform grid of 13 cells in each direction with $\Delta x = \Delta y = \Delta z = 0{\cdot}1$ and an integration time step of $\Delta t = 0{\cdot}01$. The initial pollutant mass covers $3 \times 3 \times 3$ cells in the computational domain ($L_x = 3\Delta x$, $L_y = 3\Delta y$, $L_z = 3\Delta z$) with $x_0 = 7\Delta x$, $y_0 = 7\Delta y$ and $z_0 = 7\Delta z$. The grids have been chosen intentionally coarse so as to better show discrepancies between the three schemes. The other parameters have values $\varphi_0 = 1$, $\rho = 1$ and $V = 1{\cdot}5$ and the exchange coefficient Γ has values according to the grid Peclet number $Pe_\Delta = \rho V \Delta x/\Gamma$.

Three cases have been studied, one with the flow velocity aligned with the computational grid ($u = 1{\cdot}5$, $v = w = 0$), another where it forms an angle of 45° with the grid lines of the xy-plane ($u = v = \sqrt{1{\cdot}5}$, $w = 0$) and another with an angle of 22·5° ($u = 1{\cdot}38$, $v = 0{\cdot}57$, $w = 0$). The results are presented in the form of concentration distribution with distance along the streamline passing through the centre of the cloud. Figures 10 and 11 show results for grid Peclet numbers of 100 and 150 respectively, while Figures 12 and 13 shows results for $Pe_\Delta = 100$. In all cases the time is set at $T = 6$.

Errors of numerical diffusion are clearly obvious in all solutions with the three schemes. This is not surprising, however, taking into account that only $3 \times 3 \times 3$ cells have been used to cover the pollutant mass. It is seen that the UDS performs worse than the QUICK schemes. Its poor performance is twofold: first, it provides an underestimation of the peak value of concentration (up to 55%–70%); second, it disperses concentration values outside the boundaries of the analytical solution. The latter results in non-zero values in positions where the concentration should have been zero. An increase in numerical diffusion exists in the case of a non-aligned flow (Figures 12 and 13). The combination of the two mechanisms generating numerical diffusion (truncation error and cross-flow diffusion)[3] leads to unrealistic results.

The QUICK scheme shows a better behaviour and the underestimation of the peak concentration is of the order of 20%. The numerical diffusion in the case of a non-aligned flow is even smaller than for the corresponding UDS in parallel flow. This observation reveals that the main cause of numerical diffusion is the truncation error, which is less for QUICK than the UDS. In the QUICK scheme, however, phase errors are present which do not exist in the UDS, since in the latter the low-wavelength harmonics are reduced because of numerical diffusion.[20]

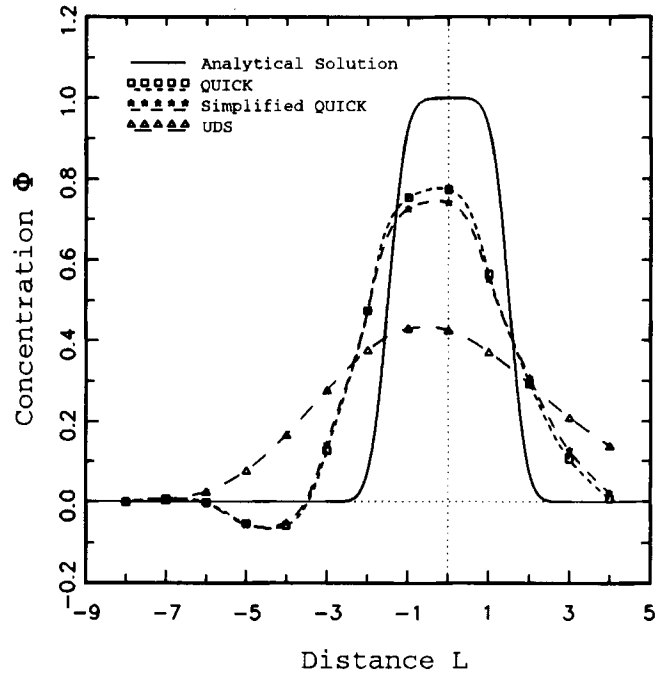

Figure 10. Comparison of results by different methods for the three-dimensional, unsteady state benchmark problem. Flow aligned with the grid; $Pe_{\Delta} = 100$, $T = 6$

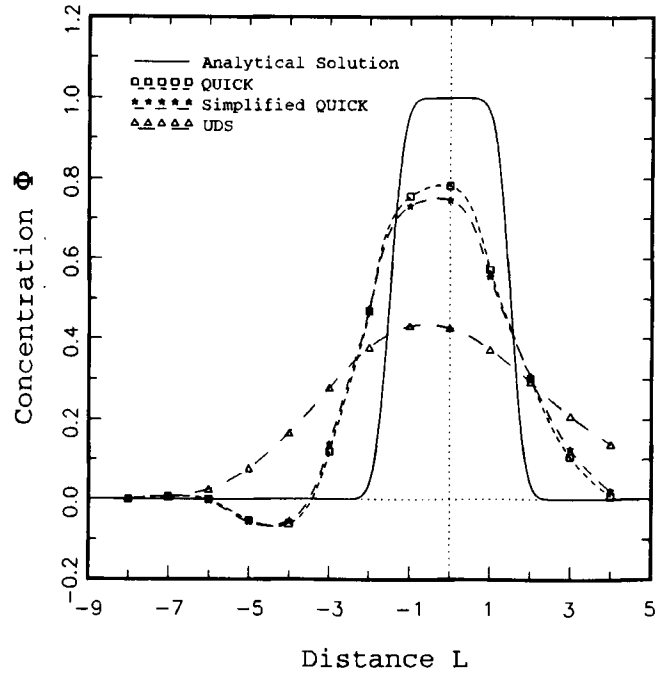

Figure 11. Comparison of results by different methods for the three-dimensional, unsteady state benchmark problem. Flow aligned with the grid; $Pe_{\Delta} = 150$, $T = 6$

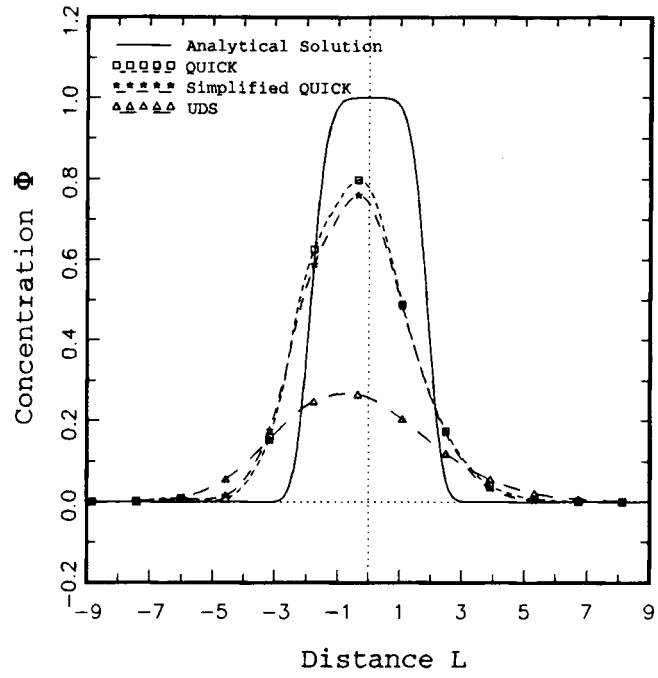

Figure 12. Comparison of results by different methods for the three-dimensional, unsteady state benchmark problem. Flow at angle of 45° to the grid; $Pe_\Delta = 100$, $T = 6$

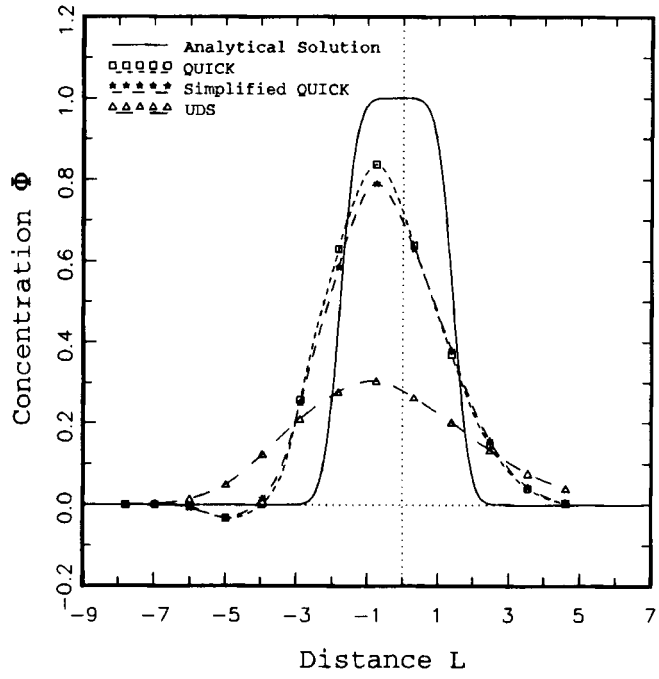

Figure 13. Comparison of results by different methods for the three-dimensional, unsteady state benchmark problem. Flow at angle of 22·5° to the grid; $Pe_\Delta = 100$, $T = 6$

The differences between the full QUICK and simplified QUICK schemes, although not large, are evident for all flow directions studied. The underestimation of the peak concentration value in the simplified QUICK scheme is around 25%. It also gives larger errors at the points where the analytical solution has zero values.

4. CONCLUSIONS

Two different benchmark problems have been studied for the application of three different schemes in highly convective flows. The first test case concerns a two-dimensional, steady state problem, while the second concerns a three-dimensional, unsteady state problem. The general convection–diffusion equation has been solved numerically using the control volume approach. The three different schemes used to handle convection are the upwind, simplified QUICK and full QUICK discretization methods. For the two-dimensional case it is shown that for uniform grids the simplified QUICK scheme reduces the error by almost a half compared with the upwind scheme. Non-uniform grids further reduce the error, which is negligible for the QUICK scheme. The simplified QUICK scheme gives errors of up to 9% compared with the full QUICK scheme in approximating the analytical solution. The CPU time required using the full QUICK scheme is not much different than for the simplified QUICK scheme, while the accuracy is improved.

The three-dimensional, unsteady state benchmark problem again shows the markedly superior performance of the QUICK scheme compared with the UDS. The full QUICK scheme gives a better approximation than its simplified version, especially for coarse grids, while the CPU time is not much affected. It is believed that the full QUICK scheme, equally applied to uniform and non-uniform grids in its 3D version, is more appropriate and should be used in further computations of highly convective flows.

ACKNOWLEDGEMENT

Financial support for one of the authors (E.M.) by the Natural Sciences and Engineering Research Council (NSERC) of Canada is gratefully acknowledged.

APPENDIX: QUICK FORMULAE FOR CONVECTED FACE VALUES

The equations used to approximate the convected face values of the dependent variable φ are

$$\varphi_e = \begin{cases} \varphi_P + QAE_P(\varphi_P - \varphi_W) + QBE_P(\varphi_E - \varphi_P) + QC_P(\varphi_S - \varphi_P) & \text{if } u_e \geqslant 0, \\ \quad + QD_P(\varphi_N - \varphi_P) + QE_P(\varphi_L - \varphi_P) + QF_P(\varphi_H - \varphi_P) & \\ \varphi_E + QAW_E(\varphi_P - \varphi_E) + QBW_E(\varphi_E - \varphi_{EE}) + QC_E(\varphi_{SE} - \varphi_E) & \text{if } u_e < 0, \\ \quad + QD_E(\varphi_{NE} - \varphi_E) + QE_E(\varphi_{LE} - \varphi_E) + QF_E(\varphi_{HE} - \varphi_E) & \end{cases} \tag{32}$$

$$\varphi_w = \begin{cases} \varphi_P + QAW_P(\varphi_W - \varphi_P) + QBW_P(\varphi_P - \varphi_E) + QC_P(\varphi_S - \varphi_P) & \text{if } u_w < 0, \\ \quad + QD_P(\varphi_N - \varphi_P) + QE_P(\varphi_L - \varphi_P) + QF_P(\varphi_H - \varphi_P) & \\ \varphi_W + QAE_W(\varphi_W - \varphi_{WW}) + QBE_W(\varphi_P - \varphi_W) + QC_W(\varphi_{SW} - \varphi_W) & \text{if } u_w \geqslant 0, \\ \quad + QD_W(\varphi_{NW} - \varphi_W) + QE_W(\varphi_{LW} - \varphi_W) + QF_W(\varphi_{HW} - \varphi_W) & \end{cases} \tag{33}$$

$$\varphi_n = \begin{cases} \varphi_P + QA_P(\varphi_W - \varphi_P) + QB_P(\varphi_E - \varphi_P) + QCN_P(\varphi_P - \varphi_S) & \text{if } v_n \geqslant 0, \\ \quad + QDN_P(\varphi_N - \varphi_P) + QE_P(\varphi_L - \varphi_P) + QF_P(\varphi_H - \varphi_P) & \\ \varphi_N + QA_N(\varphi_{NW} - \varphi_N) + QB_N(\varphi_{NE} - \varphi_N) + QCS_N(\varphi_P - \varphi_N) & \text{if } v_n < 0, \\ \quad + QDS_N(\varphi_N - \varphi_{NN}) + QE_N(\varphi_{LN} - \varphi_N) + QF_N(\varphi_{HN} - \varphi_N) & \end{cases} \tag{34}$$

$$\varphi_s = \begin{cases} \varphi_P + QA_P(\varphi_W - \varphi_P) + QB_P(\varphi_E - \varphi_P) + QCS_P(\varphi_S - \varphi_P) & \text{if } v_s < 0, \\ \quad + QDS_P(\varphi_P - \varphi_N) + QE_P(\varphi_L - \varphi_P) + QF_P(\varphi_H - \varphi_P) & \\ \varphi_S + QA_S(\varphi_{SW} - \varphi_S) + QB_S(\varphi_{SE} - \varphi_S) + QCN_S(\varphi_S - \varphi_{SS}) & \text{if } v_s \geqslant 0, \\ \quad + QDN_S(\varphi_P - \varphi_S) + QE_S(\varphi_{LS} - \varphi_S) + QF_S(\varphi_{HS} - \varphi_S) & \end{cases} \tag{35}$$

$$\varphi_h = \begin{cases} \varphi_P + QA_P(\varphi_W - \varphi_P) + QB_P(\varphi_E - \varphi_P) + QC_P(\varphi_S - \varphi_P) & \text{if } w_h \geqslant 0, \\ \quad + QD_P(\varphi_N - \varphi_P) + QEH_P(\varphi_P - \varphi_L) + QFH_P(\varphi_H - \varphi_P) & \\ \varphi_H + QA_H(\varphi_{HW} - \varphi_H) + QB_H(\varphi_{HE} - \varphi_H) + QC_H(\varphi_{HS} - \varphi_H) & \text{if } w_h < 0, \\ \quad + QD_H(\varphi_{HN} - \varphi_H) + QEL_H(\varphi_P - \varphi_H) + QFL_H(\varphi_H - \varphi_{HH}) & \end{cases} \tag{36}$$

$$\varphi_l = \begin{cases} \varphi_P + QA_P(\varphi_W - \varphi_P) + QB_P(\varphi_E - \varphi_P) + QC_P(\varphi_S - \varphi_P) & \text{if } w_l < 0, \\ \quad + QD_P(\varphi_N - \varphi_P) + QEL_P(\varphi_L - \varphi_P) + QFL_P(\varphi_P - \varphi_H) & \\ \varphi_L + QA_L(\varphi_{LW} - \varphi_L) + QB_L(\varphi_{LE} - \varphi_L) + QC_L(\varphi_{LS} - \varphi_L) & \text{if } w_l \geqslant 0. \\ \quad + QD_L(\varphi_{LN} - \varphi_L) + QEH_L(\varphi_L - \varphi_{LL}) + QFH_L(\varphi_P - \varphi_L) & \end{cases} \tag{37}$$

The geometric coefficients appearing in the equations above are defined by

$$QA_P = \frac{\Delta x_P^2}{3(\Delta x_P + \Delta x_W)(\Delta x_W + 2\Delta x_P + \Delta x_E)}, \tag{38}$$

$$QB_P = \frac{\Delta x_P^2}{3(\Delta x_P + \Delta x_E)(\Delta x_W + 2\Delta x_P + \Delta x_E)}, \tag{39}$$

$$QC_P = \frac{\Delta y_P^2}{3(\Delta y_P + \Delta y_S)(\Delta y_S + 2\Delta y_P + \Delta y_N)}, \tag{40}$$

$$QD_P = \frac{\Delta y_P^2}{3(\Delta y_P + \Delta y_N)(\Delta y_S + 2\Delta y_P + \Delta y_N)}, \tag{41}$$

$$QE_P = \frac{\Delta z_P^2}{3(\Delta z_P + \Delta z_L)(\Delta z_L + 2\Delta z_P + \Delta z_H)}, \tag{42}$$

$$QF_P = \frac{\Delta z_P^2}{3(\Delta z_P + \Delta z_H)(\Delta z_L + 2\Delta z_P + \Delta z_H)}, \tag{43}$$

$$QAE_P = \frac{\Delta x_P \Delta x_E}{(\Delta x_P + \Delta x_W)(\Delta x_W + 2\Delta x_P + \Delta x_E)}, \tag{44}$$

$$QBE_P = \frac{2\Delta x_P^2 + \Delta x_P \Delta x_W}{(\Delta x_P + \Delta x_E)(\Delta x_W + 2\Delta x_P + \Delta x_E)}, \tag{45}$$

$$QAW_P = \frac{2\Delta x_P^2 + \Delta x_P \Delta x_E}{(\Delta x_P + \Delta x_W)(\Delta x_W + 2\Delta x_P + \Delta x_E)}, \tag{46}$$

$$QBW_P = \frac{\Delta x_P \Delta x_W}{(\Delta x_P + \Delta x_E)(\Delta x_W + 2\Delta x_P + \Delta x_E)}, \tag{47}$$

$$QCN_P = \frac{\Delta y_P \Delta y_N}{(\Delta y_P + \Delta y_S)(\Delta y_S + 2\Delta y_P + \Delta y_N)}, \tag{48}$$

$$QDN_P = \frac{2\Delta y_P^2 + \Delta y_P \Delta y_S}{(\Delta y_P + \Delta y_N)(\Delta y_S + 2\Delta y_P + \Delta y_N)}, \tag{49}$$

$$QCS_P = \frac{2\Delta y_P^2 + \Delta y_P \Delta y_N}{(\Delta y_P + \Delta y_S)(\Delta y_S + 2\Delta y_P + \Delta y_N)}, \tag{50}$$

$$QDS_P = \frac{\Delta y_P \Delta y_S}{(\Delta y_P + \Delta y_N)(\Delta y_S + 2\Delta y_P + \Delta y_N)}, \tag{51}$$

$$QEH_P = \frac{\Delta z_P \Delta z_H}{(\Delta z_P + \Delta z_L)(\Delta z_L + 2\Delta z_P + \Delta z_H)}, \tag{52}$$

$$QFH_P = \frac{2\Delta z_P^2 + \Delta z_P \Delta z_L}{(\Delta z_P + \Delta z_H)(\Delta z_L + 2\Delta z_P + \Delta z_H)}, \tag{53}$$

$$QEL_P = \frac{2\Delta z_P^2 + \Delta z_P \Delta z_H}{(\Delta z_P + \Delta z_L)(\Delta z_L + 2\Delta z_P + \Delta z_H)}, \tag{54}$$

$$QFL_P = \frac{\Delta z_P \Delta z_L}{(\Delta z_P + \Delta z_H)(\Delta z_L + 2\Delta z_P + \Delta z_H)}. \tag{55}$$

The coefficients α_i and b in equation (19) have the form

$$\alpha_W = D_w + C_w^+ + C_e^+ QAE_P + C_w^- QAW_P - C_w^+ QBE_W + QA_P(C_s^- - C_n^+ + C_l^- - C_h^+), \tag{56}$$

$$\alpha_E = D_e - C_e^- - C_w^- QBW_P + C_e^- QAW_E - C_e^+ QBE_P + QB_P(C_s^- - C_n^+ + C_l^- - C_h^+), \tag{57}$$

$$\alpha_S = D_s + C_s^+ + C_n^+ QCN_P + C_s^- QCS_P - C_s^+ QDN_S + QC_P(C_w^- - C_e^+ + C_l^- - C_h^+), \tag{58}$$

$$\alpha_N = D_n - C_n^- - C_s^- QDS_P + C_n^- QCS_N - C_n^+ QDN_P + QD_P(C_w^- - C_e^+ + C_l^- - C_h^+), \tag{59}$$

$$\alpha_L = D_l + C_l^+ + C_h^+ QEH_P + C_l^- QEL_P - C_l^+ QFH_L + QE_P(C_w^- - C_e^+ + C_s^- - C_n^+), \tag{60}$$

$$\alpha_H = D_h - C_h^- - C_l^- QFL_P + C_h^- QEL_H - C_h^+ QFH_P + QF_P(C_w^- - C_e^+ + C_s^- - C_n^+), \tag{61}$$

$$\alpha_P = \alpha_W + \alpha_E + \alpha_S + \alpha_N + \alpha_L + \alpha_H + C_T^0 - S_P, \tag{62}$$

$$\begin{aligned} b = {} & C_T^0 \varphi_P^0 + S_U \\ & + C_w^+ [QAE_W(\varphi_W - \varphi_{WW}) + QC_W(\varphi_{SW} - \varphi_W) + QD_W(\varphi_{NW} - \varphi_W) \\ & + QE_W(\varphi_{LW} - \varphi_W) + QF_W(\varphi_{HW} - \varphi_W)] \\ & - C_e^- [QBW_E(\varphi_E - \varphi_{EE}) + QC_E(\varphi_{SE} - \varphi_E) + QD_E(\varphi_{NE} - \varphi_E) \\ & + QE_E(\varphi_{LE} - \varphi_E) + QF_E(\varphi_{HE} - \varphi_E)] \\ & + C_s^+ [QA_S(\varphi_{SW} - \varphi_S) + QB_S(\varphi_{SE} - \varphi_S) + QCN_S(\varphi_S - \varphi_{SS}) \\ & + QE_S(\varphi_{LS} - \varphi_S) + QF_S(\varphi_{HS} - \varphi_S)] \end{aligned}$$

$$
\begin{aligned}
&- C_n^-[QA_N(\varphi_{NW} - \varphi_N) + QB_N(\varphi_{NE} - \varphi_N) + QDS_N(\varphi_N - \varphi_{NN}) \\
&+ QE_N(\varphi_{LN} - \varphi_N) + QF_N(\varphi_{HN} - \varphi_N)] \\
&+ C_l^+[QA_L(\varphi_{LW} - \varphi_L) + QB_L(\varphi_{LE} - \varphi_L) + QC_L(\varphi_{LS} - \varphi_L) \\
&+ QD_L(\varphi_{LN} - \varphi_L) + QEH_L(\varphi_L - \varphi_{LL})] \\
&- C_h^-[QA_H(\varphi_{HW} - \varphi_H) + QB_H(\varphi_{HE} - \varphi_H) + QC_H(\varphi_{HS} - \varphi_H) \\
&+ QD_H(\varphi_{HN} - \varphi_H) + QFL_H(\varphi_H - \varphi_{HH})]. \qquad (63)
\end{aligned}
$$

where $C_T^0 = \rho_P^0 V_P \Delta t$, $C_e^+ = \max(0, C_e)$, $C_e^- = \max(0, -C_e)$, $C_e = \rho_e u_e A_e$, etc.

REFERENCES

1. S. V. Patankar, *Numerical Heat Transfer and Fluid Flow*, Hemisphere, New York, 1980.
2. B. P. Leonard, 'A stable accurate convective modelling procedure based on quadratic upstream interpolation', *Comput. Methods Appl. Mech. Eng.*, **19**, 59–98 (1979).
3. K. Y. Huh, M. W. Golay and V. P. Manno, 'A method for reduction of numerical diffusion in the donor cell treatment of convection', *J. Comput. Phys.*, **63**, 201–221 (1986).
4. T. Han, J. A. C. Humphrey and B. E. Launder, 'A comparison of hybrid and quadratic-upstream differencing in high Reynolds number elliptic flows', *Comput. Methods Appl. Mech. Eng.*, **29**, 81–95 (1981).
5. A. Pollard and A. L. W. Siu, 'The calculation of some laminar flows using various discretisation schemes', *Comput. Methods Appl. Mech. Eng.*, **35**, 293–313 (1982).
6. W. Shyy, 'Determination of relaxation factors for high cell Peclet number flow simulation', *Comput. Methods Appl. Mech. Eng.*, **43**, 221–230 (1984).
7. C. J. Freitas, R. L. Street, A. N. Findikakis and J. R. Koseff, 'Numerical simulation of three-dimensional flow in a cavity', *Int. j. numer. methods fluids*, **5**, 561–575 (1985).
8. M. Patel, M. Cross and N. C. Markatos, 'An evaluation of eleven discretisation schemes for predicting elliptic flow and heat transfer in supersonic jets', *Int. J. Heat Mass Transfer*, **30**, 1907–1925 (1987).
9. S. K. Aggarwal, 'Numerical study of convection–diffusion–reaction equations for large Damkohler and cell Reynolds numbers', *Numer. Heat Transfer*, **11**, 143–164 (1987).
10. A. J. Stamou, 'Numerical modeling of water quality in rivers using the QUICK scheme', *Proc. Eur. Conf. for Advances in Water Resources Technology*, Athens, 1991, pp. 397–405.
11. L. Ekebjarg and P. Justesen, 'An explicit scheme for advection–diffusion modelling in two dimensions', *Comput. Methods Appl. Mech. Eng.*, **88**, 287–297 (1991).
12. P. J. Roache, *Computational Fluid Dynamics*, Hermosa, Albuquerque, NM, 1982.
13. P. H. Gaskell and A. K. Lau, 'Curvature-compensated convective transport: SMART, a new boundedness-preserving transport algorithm', *Int. j. numer. methods fluids*, **8**, 617–641 (1988).
14. M. Peric, 'A finite volume method for the prediction of three-dimensional fluid flow in complex ducts', *Ph.D. Thesis*, University of London, 1985.
15. B. P. Leonard, 'Elliptic systems: finite-difference method IV', in *Handbook of Numerical Heat Transfer*, Wiley, New York, 1988, Chap. 9, pp. 347–378.
16. H. L. Stone, 'Iterative solution of implicit approximation of multidimensional partial differential equations', *SIAM J. Numer. Anal.*, **5**, 530–558 (1968).
17. D. Bradley, M. Missaghi and S. B. Chin, 'A Taylor-series approach to numerical accuracy and a third-order scheme for strong convective flows', *Comput. Methods Appl. Mech. Eng.*, **69**, 133–151 (1988).
18. D. A. Anderson, J. C. Tanehill and R. H. Pletcher, *Computational Fluid Mechanics and Heat Transfer*, Hemisphere, New York, 1984.
19. J. Crank, *The Mathematics of Diffusion*, Clarendon, Oxford, 1975.
20. M. B. Abbott and D. R. Basco, *Computational Fluid Dynamics: An Introduction for Engineers*, Longman/Wiley, New York, 1989.