RELIABLE FINITE VOLUME METHODS FOR NAVIER STOKES EQUATIONS.

M. Berzins and J.M. Ware.
Centre for the Development of CFD and School of Computer Studies,
The University of Leeds, Leeds LS2 9JT, UK.

SUMMARY.

The use of adaptive mesh spatial discretisation methods, coupled spatial and temporal error control and domain decomposition methods make it possible to construct efficient automatic methods for the numerical solution of time-dependent Navier Stokes problems. This paper describes the unstructured triangular mesh spatial discretisation method being used in a prototype package for compressible flows. The scheme is a cell-centred, second-order finite volume scheme that uses a ten triangle stencil. Previous work has concentrated on algorithms and error estimates for convection dominated problems. In this paper the algorithm is extended to include a new treatment of the diffusion terms. The prototype software uses an adaptive time error control and space remeshing strategy is used to attempt to control the numerical error in the solution.

TRIANGULAR MESH SPATIAL DISCRETISATION METHOD.

Although finite element and finite volume schemes based on unstructured triangular meshes have been used for many years, only recently have a number of high-order cell-centred finite volume schemes been developed, [5, 11, 8] . This paper is concerned with the Ware and Berzins [11, 2, 3] method. Although this method has been developed for systems of equations, for ease of exposition, consider the class of scalar p.d.e.s:

$$\frac{\partial u}{\partial t} + \frac{\partial f}{\partial x} + \frac{\partial g}{\partial y} = 0 \tag{1}$$

where $f = f(x, y, u, \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y})$ and $g = g(x, y, u, \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y})$ are the flux functions in x and y respectively and with appropriate boundary and initial conditions. The cell-centred finite volume scheme described here uses triangular elements as the control volumes over which the divergence theorem is applied. The solution values are deemed to be associated with the centroids of the triangles. In Figure 1, for example, the solution at the centroid of triangle i is U_i , the solutions at the centroids of the triangles surrounding triangle i are U_l, U_j and U_k and the next level of centroid values used by the discretisation method on the ith triangle are: U_m, U_n, U_p, U_q, U_r and U_s. The mesh point at which a solution value, say U_s, is defined is denoted by (x_s, y_s) . Integration of equation (1) on the ith triangle, which has area A_i, and use of the divergence theorem gives:

$$A_i \frac{\partial U_i}{\partial t} = - \oint_{C_i} (f.\underline{n}_x + g.\underline{n}_y) dS,$$

where C_i is the circumference of triangle i. The line integral along each edge is approximated by using the midpoint quadrature rule. The numerical flux is evaluated at the midpoint of the edge:

$$\frac{\partial u}{\partial t} = -\frac{1}{A_i}(f_{ik}\Delta y_{0,1} - g_{ik}\Delta x_{0,1} + f_{ij}\Delta y_{1,2} - g_{ij}\Delta x_{1,2} + f_{il}\Delta y_{2,0} - g_{il}\Delta x_{2,0}), \tag{2}$$

where $\Delta x_{i,j} = x_j - x_i$, $\Delta y_{i,j} = y_j - y_i$. The fluxes f_{ij} and g_{ij} in the x and y directions respectively are evaluated at the midpoint of the triangle edge separating the triangles associated with U_i and U_j. The convective parts of these fluxes are evaluated by using approximate Riemann solvers f_{Rm} and g_{Rm} respectively with the *left* solution value being defined as that internal to triangle i and the *right* solution value being defined as that external to triangle i:

$$\begin{aligned}\frac{\partial u}{\partial t} = \frac{-1}{A_i}(\; & f_{Rm}(U^l_{ik}, U^r_{ik}, (U_{ik})_x, (U_{ik})_y)\Delta y_{0,1} \; - \; g_{Rm}(U^l_{ik}, U^r_{ik}, (U_{ik})_x, (U_{ik})_y)\Delta x_{0,1} \; + \\ & f_{Rm}(U^l_{ij}, U^r_{ij}, (U_{ij})_x, (U_{ij})_y)\Delta y_{1,2} \; - \; g_{Rm}(U^l_{ij}, U^r_{ij}, (U_{ij})_x, (U_{ij})_y)\Delta x_{1,2} \; + \\ & f_{Rm}(U^l_{il}, U^r_{il}, (U_{il})_x, (U_{il})_y)\Delta y_{2,0} \; - \; g_{Rm}(U^l_{il}, U^r_{il}, (U_{il})_x, (U_{il})_y)\Delta x_{2,0} \;),\end{aligned} \tag{3}$$

where U^l_{ij} is the internal solution, with respect to triangle i, at the midpoint of the edge between U_i and U_j and U^r_{ij} is the external solution, with respect to triangle i, on edge j. Note that $U^r_{i,j} = U^l_{j,i}$ as a consequence of this notation. Standard approximate Riemann solvers such as those of Osher and Roe are used to define the convective fluxes. The left and right values for the Riemann solver are created using limited linear upwind values. The internal and external values at cell interface of two triangular elements, U^l_{ij} and U^r_{ij} in equation (3) are replaced with the limited linearly interpolated values defined by

$$U^l_{ij} = U_i + \Phi(r^l_{ij})\left(U^L_{ij} - U_i\right) \quad \text{and} \quad \mathrm{U^r_{ij}} = \mathrm{U_j} + \Phi(\mathrm{r^r_{ij}})\left(\mathrm{U^R_{ij}} - \mathrm{U_j}\right), \tag{4}$$

where U^L_{ij} is the internal linear upwind value, U^R_{ij} is the external linear upwind value, r^l_{ij} is the internal upwind bias ratio of gradients and r^r_{ij} is the external upwind bias ratio of gradients. The internal and external ratio of linear gradients are defined by

$$r^l_{ij} = \frac{U^C_{ij} - U_i}{U^L_{ij} - U_i} \quad \text{and} \quad \mathrm{r^r_{ij}} = \frac{\mathrm{U^C_{ij}} - \mathrm{U_j}}{\mathrm{U^R_{ij}} - \mathrm{U_j}}. \tag{5}$$

U^C_{ij} is the linear centred value at the cell interface. The choice of limiter function $\Phi(.)$ is left open at this point although it should be noted that a zero limiter gives a first-order method. Equations (4) and (5) depend on the as yet undefined, interpolated and extrapolated values: U^L_{ij}, U^R_{ij} and U^C_{ij}.
The value U^L_{ij} is constructed by using linear extrapolation based on the solution value U_i and an intermediate solution value (again calculated by linear interpolation) U_{lk} which lies on the line joining the centroids at which U_l and U_k are defined (see Figure 1) i.e.

$$U^L_{ij} = U_i + d_{ij,i}\,\frac{U_i - U_{lk}}{d_{i,lk}}, \tag{6}$$

where the term $d_{a,b}$ denotes the positive distance between points a and b, so for example $d_{ij,i}$ denotes the positive distance between points ij and i, see Figure 1, as defined by

$$d_{i,ij} = \sqrt{(x_i - x_{ij})^2 + (y_i - y_{ij})^2}\;, \tag{7}$$

where (x_{ij}, y_{ij}) are the co-ordinates of U_{ij} . The value U^R_{ij} is defined in a similar way using linear extrapolation based on the solution value U_j and an intermediate solution value (itself calculated by linear interpolation) U_{rs} which lies on the line joining the centroids at which U_r and U_s are defined, see Figure 1. In the case when the three

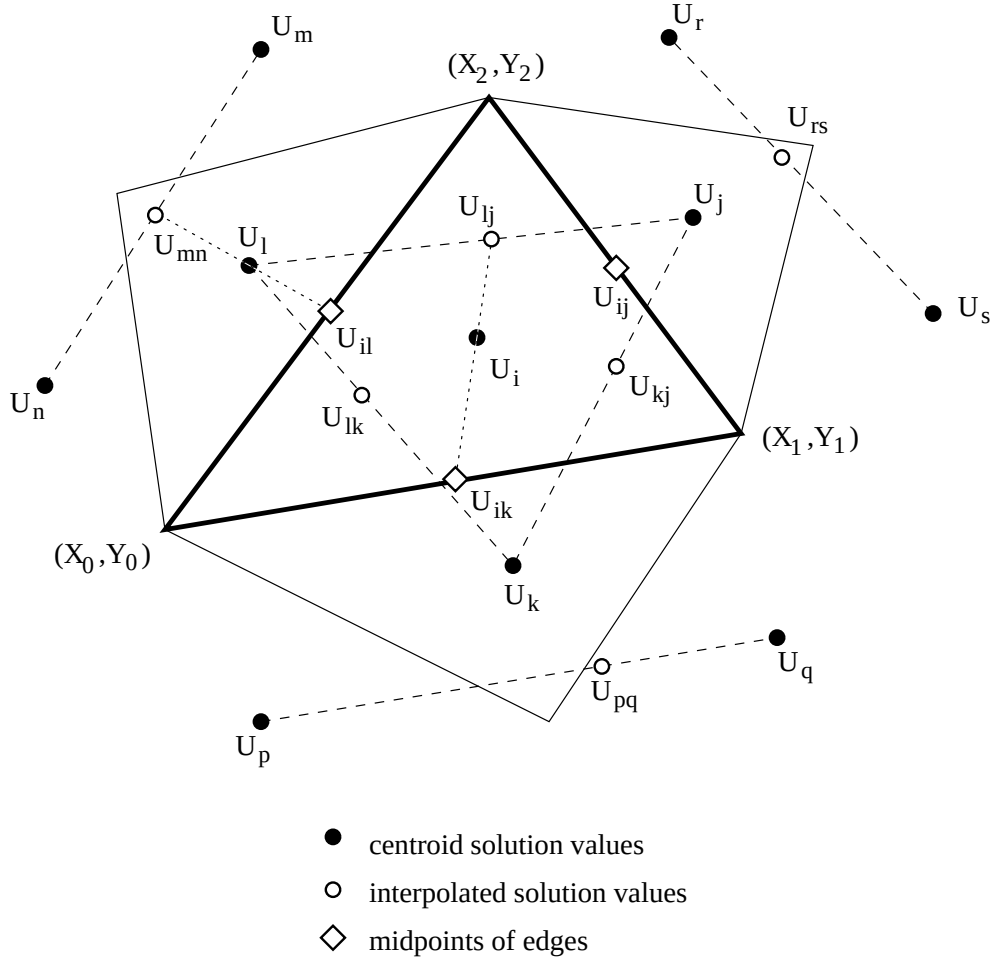

Figure 1: Construction of Interpolants

centroid points are collinear it is not possible to define a linear interpolant and so the immediate upwind centroid value will be used: internally U_i or externally U_j.
Assuming that all the centroid values are exact then the interpolation errors associated with the linear interpolants defined above may be determined by standard Taylor's series analysis. which shows that both interpolation errors are second order in the mesh spacing distances d_{**}, [3].
The centered value, U_{ij}^C, is constructed from the six values: U_i, U_j, U_k, U_l, U_s and U_r by a series of one-dimensional linear interpolations. Three linear interpolations onto the edge being considered are performed using *opposing* pairs of centroid values, see Figure 1. U_{lr}, U_{ij} and U_{ks} are found using the pairs U_l and U_r, U_i and U_j and U_k and U_s respectively. If the midpoint of the edge lies between U_{ks} and U_{ij} then the centred value is found by linear interpolation using these two values. Otherwise the values U_{lr} and U_{ij} are used to compute the centred value at the midpoint by using linear interpolation.

APPROXIMATION OF DIFFUSIVE FLUXES.

In order to compute the diffusive flux contributions at mid-points of edges it is necessary to estimate the derivatives $\partial u/\partial x$ and $\partial u/\partial y$ at these points. Consider the mid-point (x_{il}, y_{il}) which lies inside the triangle formed by the centroids i, l and k . Durlofsky

et. al. [5] construct first-order derivative approximations by differentiating the linear interpolant defined by the solution values at these points. An alternative is to use the six centroid values U_i, U_l, U_k, U_m, U_n and U_j to form a quadratic interpolant and then to differentiate this. Hyman et. al. , [6], show that this is not possible for an arbitrary set of points.
An alternative to this is to use the four points U_i, U_l, U_k, and U_n to form a bilinear interpolant and then to differentiate this. For ease of notation suppose that the edge mid-point il is the origin and assume that all derivatives are evaluated there. Standard Taylor's series expansions then yield:

$$\Delta U_{li} \;= \Delta x_{li}\frac{\partial u}{\partial x} + \Delta y_{li}\frac{\partial u}{\partial y} + \Delta xy_{li}\frac{\partial^2 u}{\partial x \partial y} \;+\; h.o.t. \tag{8}$$

$$\text{where} \;\; \Delta \mathrm{U_{li}} \;= \mathrm{U_l} - \mathrm{U_i} \;, \Delta \mathrm{x_{li}} = \mathrm{x_l} - \mathrm{x_i} \text{ and } \Delta \mathrm{xy_{li}} = \mathrm{x_l x_i} - \mathrm{y_l y_i} \;.$$

Similar equations for ΔU_{lk} and ΔU_{in} may be written using matrix notation as

$$\begin{bmatrix} \Delta U_{lk} \\ \Delta U_{in} \end{bmatrix} \;=\; M_{in}^{lk} \begin{bmatrix} \partial u/\partial x \\ \partial u/\partial y \end{bmatrix} \;+\; \frac{\partial^2 u}{\partial x \partial y} \begin{bmatrix} \Delta xy_{lk} \\ \Delta xy_{in} \end{bmatrix} \;+\; h.o.t. \tag{9}$$

$$\text{where} \;\; \mathrm{M_{in}^{lk}} = \begin{bmatrix} \Delta x_{lk} & , & \Delta y_{lk} \\ \Delta x_{in} & , & \Delta y_{in} \end{bmatrix}$$

which is an invertible matrix. Applying the inverse of this matrix to equation (9) and then substituting for $\partial u/\partial x$ and $\partial u/\partial y$ in equation (8) gives:

$$\alpha \, \frac{\partial^2 u}{\partial x \partial y} \;=\; \Delta U_{li} - \begin{bmatrix} \Delta x_{li} & , & \Delta y_{li} \end{bmatrix} \left[M_{in}^{lk}\right]^{-1} \begin{bmatrix} \Delta U_{lk} \\ \Delta U_{in} \end{bmatrix} \;+\; h.o.t. \tag{10}$$

$$\text{where} \;\; \alpha \;=\; \Delta \mathrm{xy_{li}} - \begin{bmatrix} \Delta x_{li} & , & \Delta y_{li} \end{bmatrix} \left[\mathrm{M_{in}^{lk}}\right]^{-1} \begin{bmatrix} \Delta xy_{lk} \\ \Delta xy_{in} \end{bmatrix} \tag{11}$$

An approximation to the required derivatives is then given by

$$\begin{bmatrix} \partial u/\partial x \\ \partial u/\partial y \end{bmatrix} \;=\; \left[M_{in}^{lk}\right]^{-1} \begin{bmatrix} \Delta U_{lk} & - & \Delta xy_{lk}\frac{\partial^2 u}{\partial x \partial y} \\ \Delta U_{in} & - & \Delta xy_{in}\frac{\partial^2 u}{\partial x \partial y} \end{bmatrix} \;+\; h.o.t. \tag{12}$$

In order to avoid the case $\alpha = 0$ it necessary to take care when choosing the local co-ordinate system in which to calculate the bilinear approximation. The Durlofsky et. al. approach is still used at the boundaries and gives a fallback position should α be zero.

PROPERTIES OF SPATIAL DISCRETIZATION METHOD.

Berzins and Ware [3] have considered whether or not the new scheme has the properties of *linearity preservation* and *positivity*, as proposed by Struijs et. al., [9]. The definition of positivity requires that every new value at a particular time can be written as a convex combination of old values at the previous time step. Berzins and Ware considered different flow paths through the triangle in Figure 1 and showed that three sufficient conditions for positivity are:

1. For every upwind interpolant the centroid value nearest the edge at whose midpoint the upwind value is being calculated is the maximum or minimum of the three values used to form the interpolant.
2. The centred interpolant must be bounded by the centroid values on either side i.e.

$$U_{il}^{C} \;= \alpha U_l + (1-\alpha)U_i \;, \quad \text{for} \;\; 0 \leq \alpha \leq \; 1. \tag{13}$$

3. The limiter $\Phi(.)$ must be positive and $\Phi(S)/S \leq 1$. This last condition is satisfied, for example, by a modified van Leer limiter defined by

$$\Phi(S) = \;\frac{S+|S|}{1+v} \quad \text{where} \;\; \mathrm{v} = \mathrm{Max}(1, |\mathrm{S}|) \;\; . \tag{14}$$

A linearity-preserving spatial discretization method is defined by [9], as one which preserves the exact steady state solution whenever this is a linear function of the space coordinates x and y, for any arbitrary triangulation of the domain. This is equivalent to second order accuracy on regular meshes, see [9]. Berzins and Ware were able to show that the method in its unlimited form is linearity preserving but that in some cases condition 2 above may force linearity preservation to be violated.
The previous results on interpolation errors may be combined with standard results for the effect of quadrature errors, see [7], to show how the errors at the mid-points of edges accumulate in the truncation error. Consider the spatial truncation error in the approximation of the Laplacian $c\left[\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right]$ on the ith triangle, as denoted by TE_i. This is, after ignoring the second order quadrature error, see Jeng and Chen [7], a combination of the derivative errors at the mid-points of the edges i.e.

$$\begin{array}{rcl} TE_i = \dfrac{-c}{A_i}[& (E_{ik})_x \Delta y_{0,1} + (E_{ij})_x \Delta y_{1,2} + (E_{il})_x \Delta y_{2,0} - \\ & (E_{ik})_y \Delta x_{0,1} - (E_{ij})_y \Delta x_{1,2} - (E_{il})_y \Delta x_{2,0} \;]. \end{array} \tag{15}$$

where the individual errors in the derivative approximations are defined such that $(E_{ij})_x$ is the error in $\partial u/\partial x$ at the ikth midpoint for example. Assuming from the definitions of the differentiation approximations that it is possible to extract a constant factor, say d_{min}, depending on the minimum of the distances, d_{ab}, as defined in equation (7), from each of the errors in this equation and assuming still further that the individual errors all have the form

$$(E_{ik})_x = \; d_{min} \; (e_{ik})_x \;\; \text{and} \;\; (\mathrm{E_{ik}})_\mathrm{y} = \; \mathrm{d_{min}} \; (\mathrm{e_{ik}})_\mathrm{y},$$

the expression for the truncation error may be rewritten as:

$$\begin{array}{rcl} TE_i = -\dfrac{c\; d_{min}}{A_i}[& (e_{ik})_x \Delta y_{0,1} + (e_{ij})_x \Delta y_{1,2} + (e_{il})_x \Delta y_{2,0} - \\ & (e_{ik})_y \Delta x_{0,1} - (e_{ij})_y \Delta x_{1,2} - (e_{il})_y \Delta x_{2,0} \;]. \end{array} \tag{16}$$

It is now possible to define two linear functions on the ith triangle $E_f(x,y)$ and $E_g(x,y)$ such that $E_f(x,y)$ has values $(e_{ik})_x, (e_{ij})_x$ and $(e_{il})_x$ at the midpoints ik, ij and il , $E_g(x,y)$ has values $(e_{ik})_y, (e_{ij})_y$ and $(e_{il})_y$ at the midpoints ik, ij and il . ¿From the linearity of these functions and the divergence theorem it follows that

$$\frac{\partial E_f}{\partial x} = \frac{1}{A_i}[(e_{ik})_x \Delta y_{0,1} + (e_{ij})_x \Delta y_{1,2} + (e_{il})_x \Delta y_{2,0}] \tag{17}$$

and

$$\frac{\partial E_g}{\partial y} = -\frac{1}{A_i}[(e_{ik})_y \Delta x_{0,1} + (e_{ij})_y \Delta x_{1,2} + (e_{il})_y \Delta x_{2,0}]. \tag{18}$$

Hence the truncation error (ignoring the quadrature error due to the use of the mid-point rule) may be written as

$$TE_i = -c\ d_{min} \left[\frac{\partial E_f}{\partial x} + \frac{\partial E_g}{\partial y} \right]. \tag{19}$$

The error due to the use of the quadrature rule is derived by Jeng and Chen [7].

TIME INTEGRATION AND ERROR CONTROL.

The above spatial discretization scheme results in a system of differential equations, which can be written as the initial value problem:

$$\dot{\underline{U}} \ = \ \underline{F}_N \ \ (\ t,\ \underline{U}(t)\ ) \ \ , \ \underline{U}(0) \ \ \text{given} \ , \tag{20}$$

where the vector, $\underline{U}(t)$, is defined by $\underline{U}(t) \ = \ [U(x_1, y_1, t), U(x_2, y_2, t), ..., U(x_N, y_N, t)\ ]^T$. The point x_i, y_i is the centre of the i th cell and $U_i(t)$ is a numerical approximation to $u(x_i, y_i, t)$. Numerical integration of equation (20) provides the approximation, $\underline{V}(t)$, to the vector of exact p.d.e. solution values at the mesh points, $\underline{u}(t)$. The global error in the numerical solution can be expressed as the sum of the spatial discretization error, $\underline{e}(t) = \underline{u}(t) - \underline{U}(t)$, and the global time error, $\underline{g}(t) = \underline{U}(t) - \underline{V}(t)$. That is,

$$\begin{aligned} \underline{E}(t) = \underline{u}(t) - \underline{V}(t) \ &= \ (\underline{u}(t) - \underline{U}(t)) + (\underline{U}(t) - \underline{V}(t)) \\ &= \ \underline{e}(t) + \underline{g}(t). \end{aligned} \tag{21}$$

Efficient time integration requires that the spatial and temporal are roughly the same order of magnitude. The need for spatial error estimates unpolluted by temporal error requires the spatial error to be the larger of the two errors.
The Theta method code, see [1] used here defines the numerical solution at $t_{n+1} = t_n + k$, where k is the time step size, as denoted by $\underline{V}(t_{n+1})$, by

$$\underline{V}(t_{n+1}) \ = \ \underline{V}(t_n) \ + \ (1-\theta)k\ \dot{\underline{V}}(t_n) + \theta\ k\ \underline{F}_N(t_{n+1}, \underline{V}(t_{n+1})),\ \theta = 0.55, \tag{22}$$

in which $\underline{V}(t_n)$ and $\dot{\underline{V}}(t_n)$ are the numerical solution and its time derivative at the previous time t_n . Berzins and Ware [3] show that the method will preserve positivity if a CFL-like condition is satisfied. Although such a condition is often used to choose a stable timestep it may be imprecise as an accuracy control. In contrast when a standard local error $\underline{l}_{n+1}(t_{n+1})$ control i.e. $||\ \ \underline{l}_{n+1}(t_{n+1})\ \ || \ < \ TOL$ is used it is difficult to establish a relationship between the accuracy tolerance, TOL , and the global time error.
An alternative approach is described by Berzins [1] who balances the spatial and temporal errors by controling the local time error to be a fraction of the local growth in the spatial discretization error. The local-in-time spatial error, $\underline{\hat{e}}(t_{n+1})$, for the timestep from t_n to t_{n+1} is defined as the spatial error at time t_{n+1} given the assumption that the spatial error, $\underline{e}(t_n)$, at time t_n is zero. A local error balancing approach is then:

$$||\ \ \underline{l}_{n+1}(t_{n+1})\ \ || \ < \ \epsilon\ ||\ \ \underline{\hat{e}}(t_{n+1})\ \ ||, \ \ 0 < \epsilon < 1. \tag{23}$$

The error $\underline{\hat{e}}(t_{n+1})$ is estimated by the difference between the computed solution and the first-order solution which satisfies a modified o.d.e. system denoted by

$$\dot{\underline{v}}_{n+1}\ (t) = \underline{G}_N(t, \underline{v}_{n+1}(t)), \tag{24}$$

where $\underline{v}_{n+1}(t_n) = \underline{V}(t_n)$, $\dot{\underline{v}}_{n+1}(t_n) = \underline{G}_N(t, \underline{V}(t_n))$ and where $\underline{G}_N(.,.)$ is obtained simply by setting the limiter function in the space discretisation to zero and by using the first order space derivative approximations. The local-in-time space error is then given by

$$\underline{\hat{e}}(t_{n+1}) = \underline{V}(t_{n+1}) - \underline{v}_{n+1}(t_{n+1}) \tag{25}$$

and is computed by applying the θ method with one functional iteration to equation (24). Equations (22) and (25) combined with the conditions on $\underline{v}_{n+1}(t_n)$ then give, [1],

$$\begin{aligned} \underline{\hat{e}}(t_{n+1}) = \theta \quad k \quad & [\underline{F}_N(t_{n+1}, \underline{V}(t_{n+1})) - \underline{G}_N(t_{n+1}, \underline{V}(t_{n+1}))] \ + \\ (1-\theta) \quad k \quad & [\underline{F}_N(t_n, \underline{V}(t_n)) - \underline{G}_N(t_n, \underline{V}(t_n))]. \end{aligned} \tag{26}$$

NUMERICAL EXAMPLES.

The properties of the diffusive approximation may be illustrated by two example problems with analytic solutions on $(0,1) \times (0,1)$. Problem A is a simple Poisson equation with an analytic solution $u(x,y) = 3e^{x+y}(x-x^2)(y-y^2)$. Problem B is the system of two p.d.e.s used by de Goede and Boonkamp [4] and is similar to the Navier Stokes equations while still having an exact solution. The equations are modified to be in conservative form, a Reynolds number of 100 is used so that accuracy in the diffusive part is important and integration halted at $t = 2.5$. The L1 norms obtained by discretization on the unit square are given in Table 1. On regular meshes both the Durlofsky and the bilinear approximation are second order accurate . The fixed meshes were varied so as to be irregular and have between 136 and 8704 triangles.

Table 1: L1 Error Norms.

		No. of Triangles			
Method	Prob.	136	544	2176	8704
Durlo.	A	6.9617e-3	2.0171e-3	7.2810e-4	2.1478e-4
Bilin.	A	6.5321e-3	1.7258e-3	5.5671e-4	1.5385e-4
Durlo.	B	9.6889e-3	3.0749e-3	7.5211e-4	2.1395e-4
Bilin.	B	9.5294e-3	2.7675e-3	6.8262e-4	1.8787e-4

CONCLUSIONS.

The improved accuracy of the new bilinear derivative on fixed meshes is demonstrated by Table 1. Preliminary results suggest that this improvement carries across to adaptive unstructured meshes.

The prototype adaptive software based on this discretisation method is being used to solve a variety of problems using fully automatic mesh generation and mesh adaptation software. The adaptivity tracks features in the solution automatically whilst using large elements away from these features to increase the efficiency. The spatial error estimate is used successfully in these examples to control the error through mesh adaptivity. Time integration is performed in such a way that the spatial error dominates, see [1]. The selection of appropriate times to adapt the spatial mesh is made by using a combination of estimated errors and predicted future errors, [2] . The prototype package also can be used on both shared and distributed memory computers as the flux calculation used in the residual is designed to operate in parallel. The mapping of unstructured meshes onto distributed memory processors is achieved by using graph-theoretic techniques, [10]; this ensures good speed-ups on both shared and distributed memory parallel computers.
Acknowledgement. The authors would like to thank Shell Research Ltd. for funding.

REFERENCES.

[1] M. Berzins. *Temporal error control for convection-dominated equations in two space dimensions.*, SIAM Journal of Scientific Computing (to appear), 199x.

[2] M. Berzins, J.M. Ware and J. Lawson. *Spatial and temporal error control in the adaptive solution of systems of conservation laws.* Advances in Computer Methods for Partial Differential Equations: IMACS PDE VII. IMACS, 1992.

[3] M. Berzins and J.M. Ware. *Positive discretization methods for hyperbolic equations on irregular meshes.* Submitted to Applied Numerical Mathematics, 1994.

[5] L. J. Durlofsky, B. Enquist and S. Osher. *Triangle based adaptive stencils for the solution of hyperbolic conservation laws.* Jour.Of Comp. Phys., 98:64–73, 1992.

[4] E. de Goede and M. Boonkamp .*Vectorisation of the Odd/Even Hopscotch Scheme and the ADI scheme for the Two Dimensional Burgers' Equation.*, SIAM Jour. of Sci. Comp., 11:354–367, 1990.

[6] J. M. Hyman, R. J. Knapp, and J. E. Scovel. *High order finite volume approximations of differential operators on nonuniform grids.* Physica D. , 60:112–138, 1992.

[7] Y.N. Jeng and J.L. Chen. *Truncation error analysis of the finite volume method for a model steady convective equation.* Jour. of Comp. Phys., 100:64–76, 1992.

[8] S.Y. Lin, T.M. Wu and Y.S. Chin. *Upwind finite-volume method with a triangular mesh for conservation laws.* Jour. of Comp. Phys., 107:324–337, 1993.

[9] R. Struijs, H. Deconinck and P. L. Roe. *Fluctuation splitting schemes for the 2D Euler equations.* Technical report, von Karman Institute for Fluid Dynamics, Chaussee de Waterloo, 72, B-1640 Rhode Saint Genese - Belgium, 1991.

[10] C.H. Walshaw and M. Berzins. *Enhanced dynamic load-balancing of adaptive unstructured meshes.* In R.F. Sincovec et. al., editor, Parallel Processing for Scientific Computing, pages 971–978. SIAM, 1993.

[11] J.M. Ware and M. Berzins. *Finite volume techniques for time-dependent fluid-flow problems.* In Advances in Computer Methods for Partial Differential Equations: IMACS PDE VII. IMACS, 1992.