A PARALLEL, IMPLICIT, MULTI-DIMENSIONAL UPWIND, RESIDUAL DISTRIBUTION METHOD FOR THE NAVIER-STOKES EQUATIONS ON UNSTRUCTURED GRIDS

E. van der Weide, H. Deconinck, E. Issman & G. Degrez
von Karman Institute for Fluid Dynamics
Waterloose Steenweg 72
Sint-Genesius-Rode, Belgium

Abstract

A multi-dimensional cell-vertex upwind discretization technique for the Navier-Stokes equations on unstructured grids is presented. The grids are composed of linear triangles in two and linear tetrahedra in three space dimensions. The nonlinear upwind schemes for the inviscid part can be viewed as a multi-dimensional generalization of the Roe-scheme, but also as a special class of Petrov-Galerkin schemes. They share with these schemes a compact Galerkin stencil, and are in addition monotonic by construction. The Petrov-Galerkin interpretation of the discretization technique allows a straightforward extension to the Navier-Stokes equations. For linear elements this boils down to a Galerkin discretization for the viscous terms. Compared to standard finite-volume methods on these grids, the method shows an increased accuracy, which becomes comparable with structured grid algorithms.

The spatially discretized set of equations is integrated in time using the Backward Euler time integration method. The full Jacobian matrix is computed, either numerically by finite differences or approximated analytically, and stored. The resulting set of linear equations is solved by a Block MILU(0) preconditioned Krylov subspace method. For this purpose the Aztec library of SANDIA National Laboratories is used, which also takes care of the parallelization process and completely hides the details for the user.

Results are presented for a two-dimensional turbulent shock wave boundary layer interaction in a nozzle and the turbulent flow over an ogive cylinder. All computations have been performed on the Cray T3E of the Technical University of Delft.

INTRODUCTION

The nowadays standard finite-volume method, based on dimensional splitting and approximate Riemann solvers, performs quite well on smooth, quasi-orthogonal structured grids composed of quadrilaterals (in 2D) and hexahedra (in 3D). However, the generation of such grids for complex three dimensional geometries requires a large human input. Consequently it takes time to generate them, usually several weeks to a few months for complex shapes, which is the major obstacle for fast turnaround times, needed in design cycles. Alternatively, unstructured grid generation can be highly automated and therefore is potentially able to meet the requirement of fast turnaround times. However, the current finite-volume schemes are only second-order accurate on highly regular grids, a requirement which is not fulfilled by unstructured grids. Attempts for quadratic reconstruction, second-order accuracy on arbitrary grids, have been made, see Barth and Frederickson [3], Barth [2] and Delanaye [13], but the algorithm becomes complex and expensive in three dimensions. Another possibility is represented by stabilized Galerkin finite-element discretizations, e.g. Brooks and Hughes [5], Johnson [17] and Tezduyar [32], but most of them fail the monotonicity property and a nonlinear artificial dissipation term must be added for monotonic discontinuity capturing. As can be expected this dissipation term has a negative influence on the accuracy, see Paillère [21].

Belonging to the same category of Petrov-Galerkin finite-element methods, the multi-dimensional upwind discretization technique for *scalar* advection was developed by Roe, Deconinck and coworkers [25, 26, 31, 12] with the aim of incorporating multi-dimensional physics into the upwind discretizations. Some of these schemes are monotonic and linearity preserving (the ability to exactly reproduce linear steady-state solutions) by construction and therefore no shock capturing artificial dissipation term is needed. Following the work of Sidilkover, e.g. Sidilkover [28, 27, 29], the extension of these schemes to non-commuting hyperbolic systems has been made, see van der Weide and Deconinck [33] and Catalano *et al.*[7, 8]. The system schemes do not rely on wave-modeling [24], or approximate diagonalization [9, 10], the earlier attempts of generalizing the scalar schemes to systems.

One of the advantages of this discretization technique over the standard finite-volume method, is the compactness of the stencil, at most the Galerkin finite-element stencil, which makes the algorithm easy to parallelize, see Issman [16]. First, the spatial discretization technique is discussed, followed by the parallel, implicit time integration method and the turbulence modeling. Finally, numerical examples are presented for transonic and supersonic turbulent flows in two and three space dimensions.

SPATIAL DISCRETIZATION

The spatial discretization is discussed in three space dimensions only. From this, the two-dimensional case can easily be recovered and the interested reader is referred to Deconinck *et al.*[10].

Consider first the unsteady hyperbolic system of (Euler) equations (e.g. in conservative variables U), given by:

$$\frac{\partial U}{\partial t} + A_U \frac{\partial U}{\partial x} + B_U \frac{\partial U}{\partial y} + C_U \frac{\partial U}{\partial z} = 0. \qquad (1)$$

Assume further that this system has been linearized over a tetrahedral computational cell T (see next section), such that the Jacobian matrices A_U, B_U and C_U are constant. The geometry of the cell is given in figure 1. Using Gauss' theorem the residual in integral form of equation (1) over cell T is

$$\begin{aligned} \Phi^T &= \iint_T \left(A_U \frac{\partial U}{\partial x} + B_U \frac{\partial U}{\partial y} + C_U \frac{\partial U}{\partial z} \right) d\Omega \qquad (2) \\ &= -\oint_\Gamma \left(A_U n_{x_{int}} + B_U n_{y_{int}} + C_U n_{z_{int}} \right) U \, d\Gamma, (3) \end{aligned}$$

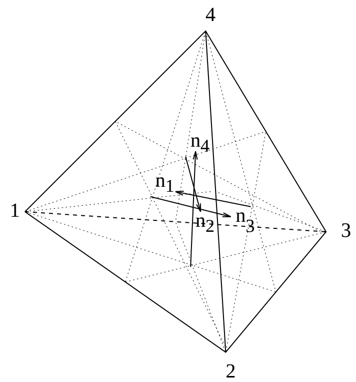

Fig. 1: A generic tetrahedron with its inner scaled normals

where Φ^T is the vector containing the cell residuals, Γ is the boundary of the control volume T and $n_{x_{int}}$, $n_{y_{int}}$ and $n_{z_{int}}$ are the x, y and z component of the inward normal respectively. In the residual distribution approach, the contour integral is evaluated using the unknowns stored at the corners, and fractions of Φ^T are sent to the cell vertices. Assembling the contributions from all cells the nodal update is obtained. The semi-discretization in point i is then

$$\frac{dU_i}{dt} = -\frac{1}{V_i} \sum_T \beta_i^T \Phi^T. \qquad (4)$$

Here V_i is the volume of the median dual cell of node i and β_i^T is the distribution matrix of cell T to node i. For consistency one has $\sum_j \beta_j^T = I$, where the summation extends over all 4 vertices of a tetrahedron.

Assuming a linear variation of U, or equivalently using the trapezium integration rule, the cell residual Φ^T, equation (3), can be written as

$$\Phi^T = \sum_{i=1}^{4} K_i U_i \ , \qquad (5)$$

where U_i is the solution at vertex i and K_i the Jacobian based on the face normal i:

$$K_i = \frac{1}{3} \left(A_U n_{x,i} + B_U n_{y,i} + C_U n_{z,i} \right), \qquad (6)$$

with $n_{x,i}$, $n_{y,i}$ and $n_{z,i}$ the x, y and z component of the inward normal $\vec{n}_i$, scaled with the area of the face opposed to vertex i, see figure 1. The Jacobians K_i can be written as

$$K_i = R_i \Lambda_i L_i, \qquad (7)$$

where the columns of R_i contain the right eigenvectors, Λ_i is a diagonal matrix of the eigenvalues and $L_i = R_i^{-1}$. As the system (1) is hyperbolic, the matrices R_i, Λ_i and L_i will be real. The matrices K_i^+ and K_i^- are given by

$$K_i^+ = R_i \Lambda_i^+ L_i, \qquad K_i^- = R_i \Lambda_i^- L_i. \qquad (8)$$

Here Λ_i^+ contains the positive and Λ_i^- the negative eigenvalues, $\Lambda_i^\pm = \frac{\Lambda_i \pm |\Lambda_i|}{2}$.

The properties of the different schemes are determined by the way β_i^T is defined. Important properties are:

- **Positivity**
  A monotonic scheme can be obtained by demanding positivity. Suppose that the numerical solution at mesh point i is U_i. Then the positivity property requires that in the semi-discrete form of equation (1),

  $$\frac{dU_i}{dt} V_i = -\sum_{k \neq i} C_{ki} (U_i - U_k), \qquad (9)$$

  the matrices C_{ki} are all positive semi-definite, which is a generalization of the scalar positivity also called

Local Extremum Diminishing (LED) property. Condition (9) is called global positivity and is difficult to impose. Therefore a more restricted property, local positivity, see Roe [25] or Deconinck *et al.* [12], is introduced. This means that condition (9) is imposed for the contribution of every cell, which is very easy to check.

- **Linearity Preservation or Residual Property**
  The ability to reproduce linear steady-state solutions exactly is obtained by demanding that no updates are sent to the vertices if the cell residual is zero. This is satisfied when the distribution matrices β_i^T are bounded, such that

$$\beta_i^T \Phi^T \to 0 \qquad \text{when} \qquad \Phi^T \to 0. \tag{10}$$

  It can be proven that only nonlinear schemes can satisfy positivity and linearity preservation at the same time (Godunov's theorem). Therefore, linear Petrov-Galerkin finite-element schemes like the SUPG-scheme, cannot be monotonic, as they are linearity preserving, by construction.

- **Invariance for similarity transformations**
  An important design property of the system distribution schemes is that the residual sent to the nodes in conservative variables is independent of the choice of variables used for the actual distribution. To illustrate this, consider the similarity transformation $\partial W = \frac{\partial W}{\partial U}\partial U$. The hyperbolic system (1) transforms into

$$\frac{\partial W}{\partial t} + A_W \frac{\partial W}{\partial x} + B_W \frac{\partial W}{\partial y} + C_W \frac{\partial W}{\partial z} = 0, \tag{11}$$

  where

$$A_W = \frac{\partial W}{\partial U} A_U \frac{\partial U}{\partial W} \tag{12}$$

  and similar expressions hold for B_U and C_U. Consequently we impose

$$\beta_i^W = \frac{\partial W}{\partial U} \beta_i^U \frac{\partial U}{\partial W}. \tag{13}$$

  As a result of this, the earlier mentioned characteristic decompositions based on similarity transformations, such as the pseudo-Mach angle model [22] or Deconinck-Hirsch decomposition [9] all lead to the same result.

In the following sections the discussion is restricted to two monotonic schemes: the system N-scheme and the limited system N-scheme (PSI-scheme). For a more extensive discussion, see van der Weide and Deconinck [34].

The system first-order N-scheme

This scheme is a generalization of the optimal first-order scalar scheme for advection problems on triangular grids, see Struijs *et al.*[31]. Using the Jacobians based on the edge-normals, equations (6) and (8), the distribution to node i is defined as

$$\Phi_i^N = \beta_i^N \Phi^T = K_i^+ (U_i - U_{in}), \tag{14}$$

where the state U_{in} is given by

$$U_{in} = \left(\sum_j K_j^- \right)^{-1} \sum_j K_j^- U_j. \tag{15}$$

For a symmetrizable system it is possible to verify that this scheme satisfies the positivity property (9). The condition of symmetrizability is fulfilled by most physical conservation laws as far as they posses a convex entropy function. Energy stability, the only functional which possibly can be bounded for a multi-dimensional hyperbolic system, see Liu and Lax [19], was shown for linear systems by Barth [4], again with the assumption that the system can be transformed into a symmetric form.

The system N-scheme is invariant for similarity transformations. To show this, consider the set of variables $\partial W = \frac{\partial W}{\partial U}\partial U$. The transformed hyperbolic system is given in equation (11). The Jacobians K_i also obey the transformation (12) and consequently

$$K_i^{W+} = \frac{\partial W}{\partial U} K_i^{U+} \frac{\partial U}{\partial W}, \quad K_i^{W-} = \frac{\partial W}{\partial U} K_i^{U-} \frac{\partial U}{\partial W}. \tag{16}$$

Hence, the update for the N-scheme in conservative U variables when the distribution is calculated in the variables W is given by

$$\begin{aligned} \Phi_i^N &= \frac{\partial U}{\partial W} K_i^{W+} (W_i - W_{in}) \\ &= \frac{\partial U}{\partial W} K_i^{W+} \frac{\partial W}{\partial U} \frac{\partial U}{\partial W} (W_i - W_{in}) \\ &= K_i^{U+} (U_i - U_{in}), \end{aligned} \tag{17}$$

which is identical to (14) obtained by applying the system N-scheme directly to the variables U.

The limited system N-scheme (PSI-scheme)

The N-scheme is monotonic, but only first-order accurate in space, as it does not satisfy equation (10). A second-order scheme is obtained for the scalar case by limiting the distribution coefficients of the N-scheme as follows, see Deconinck *et al.*[10]

$$\beta_i^{PSI} = \frac{\max(0, \beta_i^N)}{\sum_j \max(0, \beta_i^N)} \tag{18}$$

where

$$\beta_i^N = \frac{\Phi_i^N}{\Phi^T}. \tag{19}$$

The extension of equation (18) to systems is

$$\beta_i^{PSI} = \left(\sum_j \beta_j^{N+} \right)^{-1} \beta_i^{N+}. \tag{20}$$

Here $\beta_i^{N^+}$ is the matrix based on the positive eigenvalues of the first-order distribution matrix β_i^N, implicitly defined by equation (14). In combination with the property of invariance for similarity transformations, this is enough to define β_i^N uniquely for a commuting system:

$$\beta_i^N = RD_iL, \tag{21}$$

where D_i is a diagonal matrix with the k^{th} element d_i^k given by

$$d_i^k = \frac{l^k \cdot \Phi_i^N}{l^k \cdot \Phi^T}, \tag{22}$$

while L (with rows l^k) and R are the left and right eigenvector matrices of the K_i, which are independent of the normal. It is easy to check that this definition of the limited system N-scheme is equivalent to applying the scalar limited N-scheme to each of the decoupled characteristic equations and transforming the result back to the variable U. For a general non-commuting system these expressions can be generalized by introducing the following form for β_i^N:

$$\beta_i^N = R_iD_iL_i + \beta_i^{\mathrm{corr}}, \tag{23}$$

where R_i and L_i (with rows l_i^k) contain the right respectively left eigenvectors of K_i and D_i is a diagonal matrix whose k^{th} element, d_i^k, is given by

$$d_i^k = \frac{l_i^k \cdot \Phi_i^N}{l_i^k \cdot \Phi^T}. \tag{24}$$

The β_i^{corr} must satisfy the following requirement:

$$\sum_j \beta_j^{\mathrm{corr}} = I - \sum_j R_jD_jL_j \tag{25}$$

to have a consistent distribution ($\sum_j \beta_j^N = I$). Again this is not sufficient to determine β_i^{corr} completely, but it is used as a constraint in a minimization problem for the $L2$ norm of the elements of β_i^{corr}. Because of symmetry reasons the solution of this problem is very simple. The result is:

$$\beta_i^N = R_iD_iL_i + \frac{1}{4}\left(I - \sum_j R_jD_jL_j\right) \tag{26}$$

It can be proven that if the system is symmetrizable (as is the case for the Euler equations), the eigenvalues of β_i^N are real. The form (26) is invariant for similarity transformations and obeys the one-dimensional limit; it returns to the scalar scheme if the system is (partially) decoupled, it gives the correct expression if a system is diagonalizable and it is continuous. However, it is not satisfactory in terms of robustness (convergence problems) and further improvements are under investigation.

As the limiting procedure (20) is expensive (determination of the eigenvectors and eigenvalues of β_i^N), an alternative is to apply scalar limiting in characteristic residuals, where the characteristic variables must be chosen by the user. Of course this approach is not invariant for similarity transformations, but the results are (at the moment) comparable with the matrix limiting approach. Also the scalar limiting suffers from convergence problems.

Extension to advective-diffusive systems

The diffusive terms are discretized in a central manner. As the residual distribution schemes are vertex centered, the Galerkin discretization of node i of cell T for the diffusive part of the scalar advection-diffusion equation,

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} + b\frac{\partial u}{\partial y} + c\frac{\partial u}{\partial z} - \nu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}\right) = 0, \tag{27}$$

becomes:

$$\Phi_i^{dif} = \frac{\vec{n}_i}{3} \cdot \nu \sum_{j=1}^{4} u_j \frac{\vec{n}_j}{3V_T}, \tag{28}$$

where $\vec{n}_i$ are the inward scaled normals, see figure 1, and V_T the volume of cell T. The discretization of the viscous fluxes of e.g. the Navier-Stokes equations is similar.

CONSERVATIVE LINEARIZATION

From the definition of the matrix schemes in the previous sections it is clear that they use the quasi-linear form of the equations with constant Jacobians per cell and especially the positivity property heavily relies on this fact. For solutions with discontinuities the conservative form must be solved to obtain the correct jump relations. The link between both formulations is given by the conservative linearization, briefly discussed here. For more details, see Deconinck *et al.*[11].

Consider the conservation law

$$\frac{\partial U}{\partial t} + \vec{\nabla} \cdot \vec{F} = 0, \tag{29}$$

and its quasi-linear form

$$\frac{\partial U}{\partial t} + \frac{\partial \vec{F}}{\partial U} \cdot \vec{\nabla} U = 0. \tag{30}$$

Here the vector notation is used for simplicity. The quasi-linear form (30) is conservative if for any cell T the relation

$$\oint_{\partial T} \vec{F} \cdot \vec{n}\, d\partial T \equiv \frac{\partial \vec{F}}{\partial Z}\left(\bar{Z}\right) \cdot \oint_{\partial T} Z\, \vec{n}\, d\partial T \tag{31}$$

is identically satisfied. Here Z is an arbitrary set of independent variables, $\bar{Z}$ a cell averaged state of these variables, ∂T the edge of T and $\vec{n}$ the normal of ∂T. On triangles in 2D and tetrahedra in 3D it can be shown, see Deconinck *et al.*[11], that the arithmetic average of the nodal values of the Roe parameter vector $Z = \sqrt{\rho}(1, u, v, w, H)^T$ satisfies (31), thus extending 1D Roe-averaging to two and three space dimensions.

PARALLEL TEMPORAL DISCRETIZATION

In this section a parallel implicit algorithm is explained for the integration in time of the ordinary set of differential equations

$$\frac{dU}{dt} = RES\left(U\right), \tag{32}$$

which is the result of the spatial discretization, see equation (4). First the sequential algorithm is explained, followed by the parallelization technique.

As only steady-state problems are considered, time-accuracy is not important and the backward Euler method is used to integrate equation (32). This results in the non-linear system for the state on time $t = t^{n+1}$

$$\frac{U^{n+1} - U^n}{\Delta t} = RES\left(U^{n+1}\right), \tag{33}$$

where Δt indicates the local time step. The right hand side of equation (33) is approximated using a Newton linearization, which leads to the following linear system for the update $\Delta U = U^{n+1} - U^n$:

$$\left[\frac{I}{\Delta t} - \frac{\partial RES\left(U^n\right)}{\partial U^n}\right] \Delta U = RES\left(U^n\right). \tag{34}$$

This loop is repeated in time until convergence. Analytical computation of the exact Jacobian matrix $\frac{\partial RES}{\partial U}$ becomes increasingly intractable and a numerical evaluation becomes attractive. Truncating the Taylor expansion of $RES_i(U_j + \varepsilon_m 1_m)$ (the nodal residual at node i with the m^{th} component of U at node j perturbed by a small quantity ε_m) to the first-order terms, one has:

$$\left[\frac{\partial RES_i(U)}{\partial U_j}\right]_m \simeq \frac{RES_i(U_j + \varepsilon_m 1_m) - RES_i(U_j)}{\varepsilon_m}. \tag{35}$$

Because of the compact stencil of the discretization, this numerical computation of the Jacobian requires a rather limited number of additional cell residual evaluations (12 and 20 for laminar flows in two and three dimensions respectively). The expression used for ε_m is

$$\varepsilon_m = \epsilon \ \text{sign}(U_m) \ \max\left(|U_m|, 10^{-3}\right), \tag{36}$$

where sign(U_m) is 1 for positive and -1 for negative U_m. The constant ϵ is a user specified value and is of the order 10^{-7} to 10^{-9}. Because of these small values of the disturbances it is important that all computations are done in double precision and the dimensionless Navier-Stokes equations are used to minimize errors due to the finite arithmetic.

Alternatively, approximate analytical Jacobians can be used, especially for turbulent problems and the nonlinear second-order discretization. These approximate Jacobians, when carefully designed, enhance stability in the early stages of the iteration process. The price paid is that no Newton convergence can be obtained, something which could not be achieved anyway in the above mentioned cases. Only the Picard Jacobian of the system N-scheme has been used. As the viscous terms are discretized with the Galerkin method, their Jacobians can be computed analytically without too many difficulties. For turbulent computations the eddy-viscosity is frozen and only the destruction term is linearized.

In combination with the conservative linearization, the distribution of the system N-scheme (14) for a certain cell T can be written as

$$\Phi_i^N = K_i^+ \left(\sum_k K_k^-\right)^{-1} \sum_j K_j^- \frac{\partial U}{\partial Z}\left(Z_i - Z_j\right), \tag{37}$$

where Z_j are the nodal values of the parameter vector and all the matrices are evaluated at the cell average state $\bar{Z}$. Assuming that the nodal differences are small, *i.e.* the derivatives of matrices in front of $(Z_i - Z_j)$ in equation (37) can be neglected, the Jacobians are approximately:

$$\frac{\partial \Phi_i^N}{\partial U_i} = \underbrace{K_i^+ \left[\left(\sum_k K_k^-\right)^{-1} K_i^- - I\right]}_{\text{evaluated at } \bar{Z}} \frac{\partial U}{\partial Z}\frac{\partial Z_i}{\partial U_i} \tag{38}$$

$$\frac{\partial \Phi_i^N}{\partial U_j} = \underbrace{-K_i^+ \left(\sum_k K_k^-\right)^{-1}}_{\text{evaluated at } \bar{Z}} K_j^- \frac{\partial U}{\partial Z}\frac{\partial Z_j}{\partial U_j} \quad i \neq j. \tag{39}$$

A further approximation would be to assume $\frac{\partial U}{\partial Z}\left(\bar{Z}\right)\frac{\partial Z_j}{\partial U_j} \approx I$, which saves two matrix multiplications, but this has not been done. To improve stability the turbulence equations are solved in a decoupled manner, *i.e.* ignoring the Jacobian entries of the flow variables with respect to the turbulent quantities and vice versa. An extra advantage of this approach is a factor two saving in the memory requirements.

The linear system (34) is iteratively solved in parallel using the AZTEC library of SANDIA National Laboratories, see Hutchinson *et al.*[15]. This library completely hides the details of the parallelization for the user, whose only responsibility is to provide the matrix and right hand side in the suitable format. The domain decomposition method used by AZTEC is the vertex oriented decomposition (VOD)

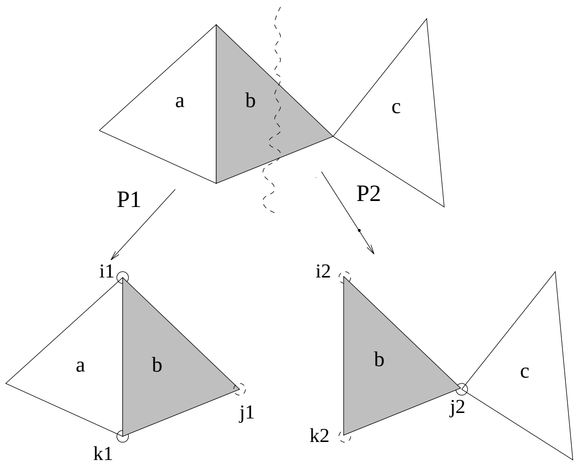

Fig. 2: VOD, Close-up of the overlapping strip and respective mapping onto processors, internal (solid circles) and external (dashed circles) nodes.

technique, see figure 2. As a result, the separation between neighboring partitions occurs at the edge-level. If overlapping of neighboring partitions is performed (effectively half a layer of ghost cells, as cell b is stored on two processors), any computational task can be carried out in the same way as in the sequential code, provided that nodes on the outer boundaries of each domain have been properly updated.

A closer look at the three cells a, b and c of figure 2 shows two different types of vertices in the overlapped strip of cells: Internal nodes, nodes i_1 and k_1 in partition 1 and node j_2 in partition 2, and external nodes, node j_1 in partition 1 and nodes i_2 and k_2 in partition 2. Internal nodes are surrounded by their complete set of triangles and consequently the residual and all Jacobian entries can be computed. For external nodes this is not the case and neither the residual nor the Jacobians are stored. Their values must be obtained by communication after every iteration. In AZTEC, computation and communication overlap as much as possible, resulting in a smaller parallel overhead.

The iterative solvers inside AZTEC are Krylov subspace methods, like GMRES, BiCGSTAB, etc., which in general must be preconditioned to achieve an acceptable performance. The only preconditioner inside the current AZTEC release, which performed good enough for our applications, is the Block Incomplete LU factorization without additional fill in, BILU(0). Unfortunately, this requires an additional storage of a full matrix, which is unacceptable in 3D. Therefore a modified version of this preconditioner, BMILU(0), which only requires the storage of an extra block diagonal, like Block Jacobi, has been added to the AZTEC library. In the future releases, this preconditioner will become standard available.

For the partitioning into the required number of subdomains (usually the number of processors), Metis has been used, see Karypis and Kumar [18].

TURBULENCE MODELING

To account for the effects of turbulence, three turbulence models have been used.

- **$k - \omega$ model**
  The $k - \omega$ model, see Wilcox [35], is nowadays a standard model. Unlike the $k - \varepsilon$ model, it does not need damping functions in the viscous sublayer and is relatively easy to handle numerically. Its biggest disadvantage is its sensitivity to the rather arbitrary free-stream values of k and ω.

- **SST-model**
  The two-equation SST model of Menter [20], is a mixture of the $k - \omega$ model, used in the inner region of the boundary layer, and the $k - \varepsilon$ model, used in rest of the flow field. In this way the free-stream sensitivity of the $k - \omega$ model is avoided, while it still does not need damping functions in the viscous sublayer. Furthermore, like the Johnson-King model, it takes the transport of the principal turbulent shear stress into account, which leads to much better predictions of flows with adverse pressure gradients.

- **Spalart-Allmaras model**
  This model, see Spalart and Allmaras [30], is an empirical one equation model for the eddy-viscosity and is mainly meant for aerodynamical flow fields. Numerically seen it is relatively easy to handle.

The generic form of the additional transport equation(s) in two space dimensions is

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} + b\frac{\partial u}{\partial y} = \frac{\partial}{\partial x}\left(\nu_u \frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial x}\left(\nu_u \frac{\partial u}{\partial y}\right) + \Sigma^+ + \Sigma^-. \tag{40}$$

Here u stands for any of the turbulent quantities, ν_u a viscosity or diffusion coefficient typically for u, Σ^+ the production source term and Σ^- the destruction source term. The discretization of the advective and diffusive part of equation (40) has been described in the previous section and will therefore not be repeated here. For stability reasons the source terms Σ^+ and Σ^- are not upwinded, but treated in a pointwise manner. To enhance stability even further, only the destruction source term Σ^- is treated implicitly, which can be interpreted as an automatic under-relaxation.

RESULTS

Results are presented for two different test cases: a 2D turbulent shock-wave/boundary layer interaction in a transonic channel and a 3D supersonic turbulent flow over an ogive cylinder. The computations were run under MPI on the Cray T3E of the University of Delft, where each node has 128 Mbytes of RAM.

Transonic turbulent channel, Délery's Experiment C

This well-known test case, whose geometry is given in figure 3, was experimentally investigated in the early eighties by Délery [14]. The subsonic inlet boundary conditions are $T_t = 300K$, $P_t = 96kPa$ and $\alpha = 0°$. Furthermore a zero

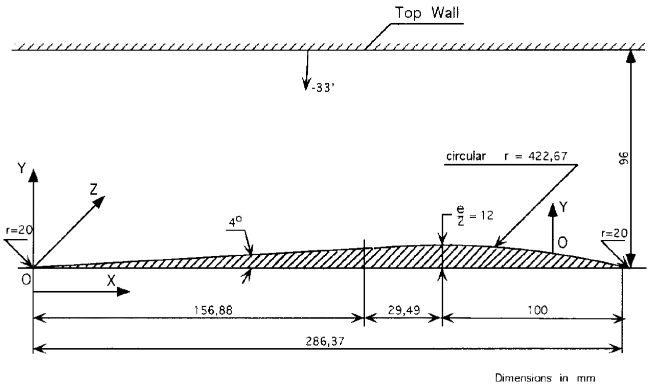

Fig. 3: Geometry description for Délery bump (case C).

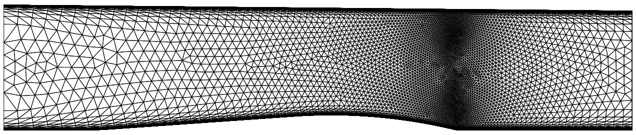

Fig. 4: Grid for Délery's test case C: 10420 nodes, 20466 elements

thickness inlet boundary layer is assumed. The static pressure at the outlet boundary should be adjusted such that the position of the normal shock wave coincides with the experiment. This very much depends on the turbulence model used; for the SST model a back pressure of 61.5 kPa must be prescribed, for the Spalart-Allmaras model 62.5 kPa and for the $k-\omega$ model even 63.0 kPa.

The hybrid grid used for this computation is obtained by combining hyperbolic structured grids in the boundary layer with frontal Delaunay generated isotropic grids in the mean flow, see Carette [6]. The grid is shown in figure 4. It consists of 10420 nodes and 20466 triangular elements and is refined in the region of interest. The initial normal spacing is variable and taken according to a Blasius boundary layer growth, which is always on the safe side. At the inlet it is $1.0 \times 10^{-6} m$ and at the outlet $5.7 \times 10^{-6} m$. Mach number isolines in the region of interest of the second-order scheme in combination with the SST turbulence model are given in figure 5. The lambda-shock, which linear $k-\varepsilon$ and $k-\omega$ models fail to predict, is clearly visible. Wall pressure distributions, made dimensionless by the total pressure of 96.0 kPa, are compared with experiments in the figures 6 and 7. It can be seen that upstream of the shock the computed pressure distributions perfectly match the experimental data, which is logical, as these distributions are determined by inviscid phenomena. However, after the shock all computations are quite far off from the experimental data, although the SST model performs significantly better than Spalart-Allmaras and the $k-\omega$ model. A possible explanation for the deviation between computation and experiment is the treatment of the subsonic outlet boundary. As stated earlier, the back pressure, with a constant value over the entire outlet, has been adjusted such that the shock on the lower wall matches the experimental position. In the experiment

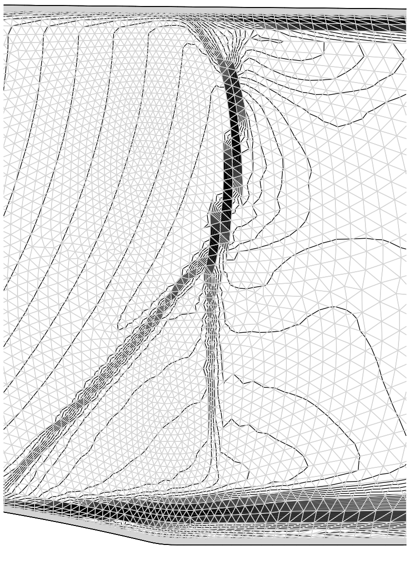

Fig. 5: Mach number isolines superposed on the mesh, in the interaction region for the Délery test case C: min = 0.0, max = 1.40, step = 0.025

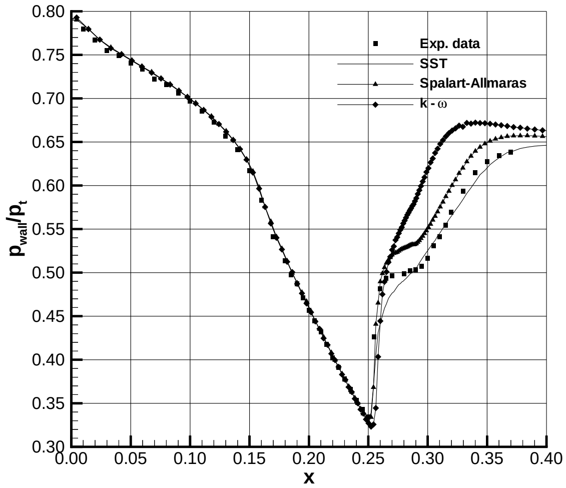

Fig. 6: Experimental and computed pressure distribution on the lower wall

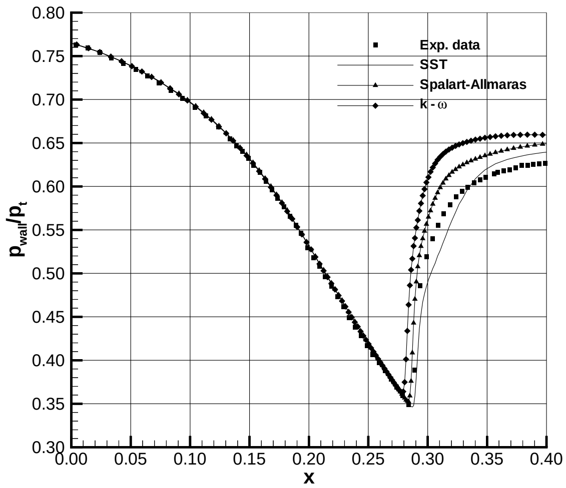

Fig. 7: Experimental and computed pressure distribution on the upper wall

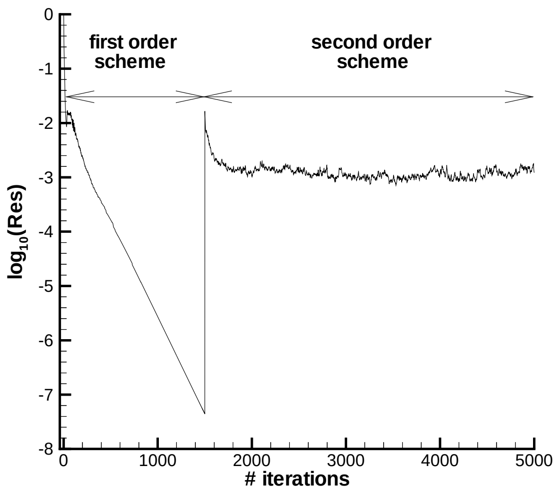

Fig. 8: Convergence history for the Délery test case C, SST turbulence model.

the shock position was controlled by an adjustable second throat, which of course allows a varying pressure distribution over the computational outlet boundary. However, a computation with this outlet boundary located much further downstream gave virtually identical results, and therefore it can be concluded that the deficiencies in the turbulence models cause the differences between experiment and computation.

The convergence history for the SST-model, based on the residuals of all but the turbulence equations, is depicted in figure 8. This shows that the first-order scheme, which has been computed with a constant CFL number of 100, had converged about 7 orders of magnitude before the second-order scheme was started, also with a constant CFL number of 100. As can be seen, this nonlinear scheme has convergence problems and stalls after one order of magnitude. However the maximum value of the eddy-viscosity did not change more than 0.2% between iteration 4000 and 5000 and the solution has been assumed converged. The convergence histories for the $k-\omega$ and Spalart-Allmaras model show a similar behavior. The total amount of CPU-time on 16 processors was less than 2 hours.

This test case has been used to determine the speed-ups on different number of processors. For comparison, 1500 time steps are taken for the first-order scheme, starting from a uniform flow field. The results are given in table 1, which compares the amount of CPU time needed to perform 1500 time steps. Per time step, the linear system (34) is solved by preconditioned GMRES. As interprocessor links are neglected for the preconditioner, its performance decreases as the number of processors increases. Consequently more GMRES iterations must be taken to reach the same accuracy. For example, on two processors 25% more GMRES steps had to be taken than on one processor, which results in an

# procs	CPU-time (s)	Speed up w.r.t. 1 procs	Speed up w.r.t. N/2 procs
1	$2.3910 \cdot 10^4$	-	-
2	$1.3414 \cdot 10^4$	1.78	1.78
4	$7.2687 \cdot 10^3$	3.29	1.85
8	$3.8505 \cdot 10^3$	6.21	1.89
16	$2.1943 \cdot 10^3$	10.90	1.75
32	$1.2863 \cdot 10^3$	18.44	1.71

Table 1: Speed-ups for the Délery test case C.

extra computational effort of 10 t0 15%. If this would be taken into account, almost perfect speed-ups are obtained for 2, 4 and 8 processors. This is not true anymore for 16 and 32 processors, mainly because the size of the test problem is very small. Indeed, on 32 processors, only about 300 internal nodes are stored per processor.

Supersonic flow over an ogive cylinder

The details of this 3D test case can be found in [1]. The flow conditions are given in table 2. In the experiments the

M_∞	2.0
Re_∞	$5.33 * 10^6/m$
T^0_∞	330 K
p^0_∞	50000 Pa
α	10 deg
D	0.03 m

Table 2: Flow conditions for the ogive cylinder

flow was tripped near the apex and a fully turbulent flow has been assumed in the computation. Only the SST turbulence model has been used.

A fully unstructured grid has been generated using the FLITE3D grid generator of the University of Swansea [23]. Due to limitations of this grid generator a symmetry plane cannot be used and the full problem must be computed. The grid consist of 850,196 nodes (5,048,481 tetrahedra), the initial normal spacing is $10^{-6}m$ and 38 viscous layers with a stretching factor of 1.2 have been used to guarantee a smooth transition from the viscous layers to the isotropic part of the grid. The surface grid of the ogive-cylinder,

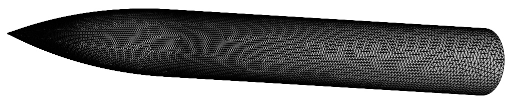

Fig. 9: Unstructured surface grid for the ogive-cylinder, 23187 nodes

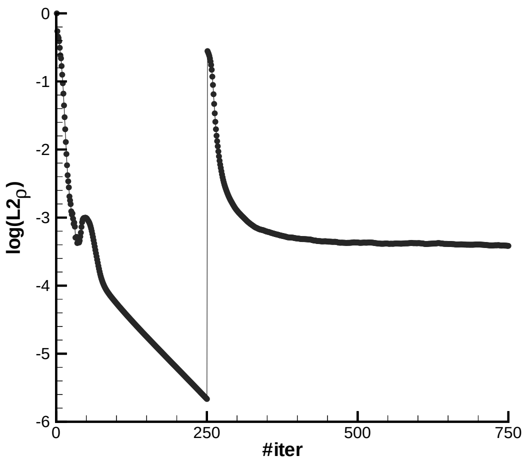

Fig. 10: Convergence history for the turbulent ogive cylinder, SST-model.

shown in figure 9, has 23187 nodes. The computation could only be done on 64 processors, because of the memory requirement of the implicit solver. Approximate analytical Jacobians are used, in combination with a CFL number of 100, for both the first and second-order computation. As shown in the convergence history, figure 10, the first-order N-scheme is converging quite well and probably more orders of magnitude can be reached than the approximately 6 orders where it has been stopped. The PSI-scheme converges about 3 orders relative from the restart and then it stalls. As the maximum value of the eddy-viscosity did not change more than 0.1% between iteration 700 and 750, it has been assumed that the solution was converged.

To illustrate the flow field, Mach isolines are shown in figure 11. In the outflow plane the main vortices are clearly visible.

Comparison with the experimental surface pressure distributions at several cross sections, figures 12 to 14, shows

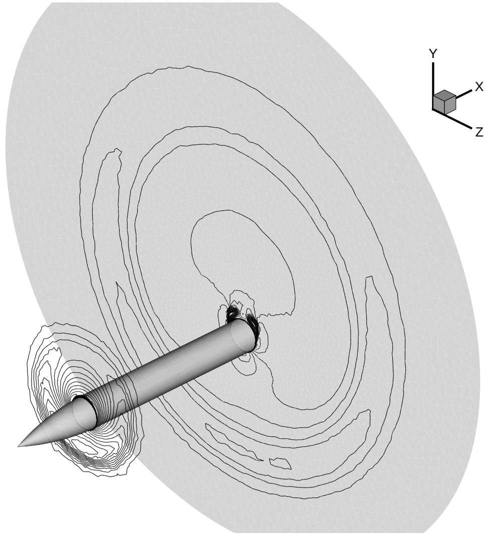

Fig. 11: Mach number isolines for the ogive cylinder

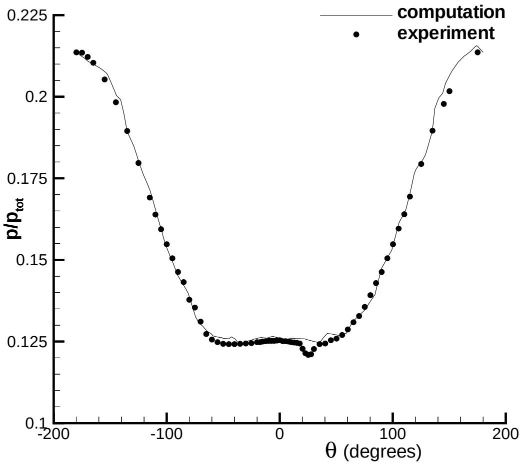

Fig. 12: Surface pressure distribution, x/D = 1.5

very good agreement between computation and measurements. From turbulence modeling point of view, this flow is much simpler than Délery's Experiment C and therefore the deviations from the experimental values are less.

CONCLUSIONS

An attempt has been made to extend previously developed multi-dimensional upwind scalar schemes to noncommuting hyperbolic systems in general and the Euler equations in particular. This extension is straightforward for the linear N-scheme, but for the nonlinear PSI-scheme, which combines second-order accuracy with monotonicity, it is more difficult. A formulation is given, which gives acceptable results, but its convergence behavior is not optimal and further improvements are under investigation. The Galerkin

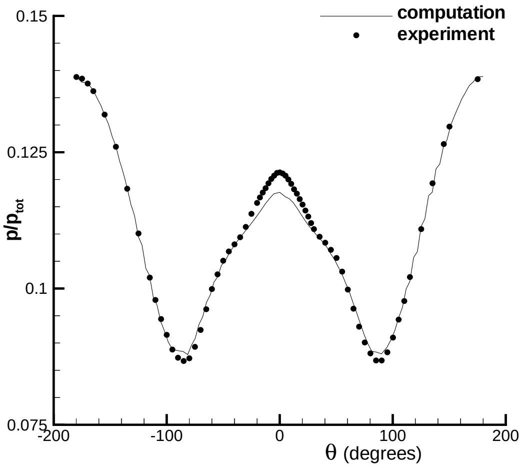

Fig. 13: Surface pressure distribution, x/D = 4.5

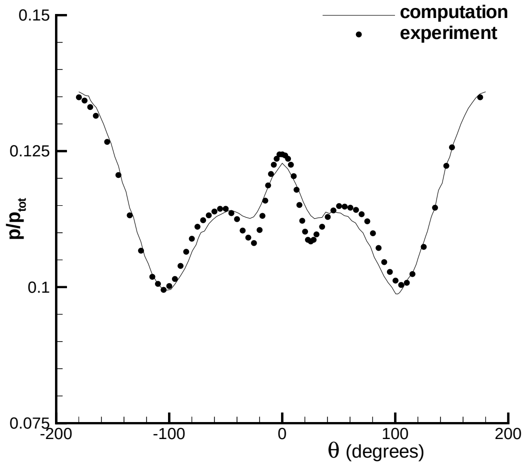

Fig. 14: Surface pressure distribution, x/D = 7.5

method has been used to discretize the viscous fluxes of the Navier-Stokes equations.

Due to the compact stencil, the parallelization of the fully implicit time-integration algorithm is relatively easy. Also the use of the AZTEC iterative solver library simplifies this task considerably. Speed-up measurement for a quite coarse grid on a Cray T3E showed satisfactory results up to 32 processors.

The implementation of turbulence models with the current discretization technique is very natural. Only the strong production and destruction source terms should be handled with care. To enhance stability, these terms are treated in a pointwise manner and only the destruction term is linearized. In combination with approximate Picard Jacobians, this gives a very stable discretization technique. For the problems considered, a uniform flow field has been used as initial solution, also for the turbulent quantities, without any problem.

The results of the turbulent shock-wave/boundary layer interaction, Délery's Experiment C, showed that none of the turbulence models used, is able to reproduce the experimental values exactly, although the SST-model is significantly better than the Spalart-Allmaras and $k-\omega$ model for the wall pressure distribution. The flow over an ogive cylinder showed that the discretization technique is capable to compute 3D turbulent flows without any problem. The better agreement with the experimental wall pressure distribution for this test case is caused by the fact that no shock-wave/boundary layer interaction occurs and probably any turbulence model will give reasonable results. The CPU-time of 36 hours on 64 processors of the Cray T3E is rather high and multigrid techniques might be interesting to reduce this.

ACKNOWLEDGMENT

The authors would like to thank Bert Houtman of the Aerospace Department of the Delft University of Technology for his advice to get the code running on the Cray T3E.

REFERENCES

1. D. Barberis. Supersonic Vortex Flow around a Missile Body. In *A Selection of Experimental Test Cases for the Validation of CFD Codes*, August 1994. AGARD-AR-303, Vol. II.

2. T.J. Barth. Recent Developments in High Order k-Exact Reconstruction on Unstructured Meshes. 1993. AIAA-93-0668.

3. T.J. Barth and P.O. Frederickson. Higher Order Solution of the Euler Equations on Unstructured Grids using Quadratic Reconstruction. 1990. AIAA-90-0013.

4. T.J. Barth, E. van der Weide, H. Deconinck, and R. Abgrall. Matrix Distribution Schemes and Energy-Stability for Hyperbolic Systems, with Application to Compressible Flows. *in Proc.* 6$^{\text{th}}$ *International Conference on Hyperbolic Problems, Hongkong*, 1996.

5. A. Brooks and T. Hughes. Streamline Upwind/Petrov Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 32:199–259, 1982.

6. J.-C Carette. *Adaptive Unstructured Mesh Algorithms and SUPG Finite Element Method for High Reynolds Number Compressible Flows.* PhD thesis, Université Libre de Bruxelles, Belgium, Dec. 1997.

7. L.A. Catalano, P. De Palma, G. Pascazio, and M. Napolitano. Matrix Fluctuation Splitting Schemes for accurate Solutions to Transonic flows. 1996. 15th ICN-MFD, Monterey.

8. L.A. Catalano, P. De Palma, G. Pascazio, and M. Napolitano. A Contribution to Multidimensional Upwinding and Fluctuation Splitting Schemes: Nonlinear Matrix Schemes. 1997. 13th AIAA CFD Conference, Snow Mass, Paper 97-2031.

9. H. Deconinck, Ch. Hirsch, and J. Peuteman. Characteristic decomposition methods for the multidimensional Euler equations. In *Lecture Notes in Physics*, volume 264. Springer-Verlag, 1986.

10. H. Deconinck, H. Paillère, R. Struijs, and P.L. Roe. Multidimensional upwind schemes based on fluctuation-splitting for systems of conservation laws. *Journal of Computational Mechanics*, 11(5/6):323–340, 1993.

11. H. Deconinck, P.L. Roe, and R. Struijs. A Multidimensional Generalization of Roe's Flux Difference

Splitter for the Euler Equations. *Journal of Computers and Fluids*, 22:215–222, 1993.

12. H. Deconinck, R. Struijs, G. Bourgois, and P.L. Roe. Compact advection schemes on unstructured grids. In *VKI LS 1993-04, Computational Fluid Dynamics*, 1993.

13. M. Delanaye. *Polynomial reconstruction finite volume schemes for the compressible Euler and Navier-Stokes equations on unstructured adaptive grids.* PhD thesis, Université de Liège, Belgium, 1996.

14. J.M. Délery. Experminental Investigation of Turbulence Properties in Transonic Shock/Boundary-Layer Interactions. *AIAA-Journal*, 21(2):180–185, 193.

15. S.A. Hutchinson, J.N. Shadid, and R.S. Tuminaro. Aztec User's Guide: Version 1.1. Technical Report SAND95-1559, Sandia National Laboratories, 1995.

16. E. Issman. *Implicit Solution Strategies for Compressible Flow Equations on Unstructured Meshes.* PhD thesis, Université Libre de Bruxelles, Belgium, 1997.

17. C. Johnson. Finite element methods for flow problems. In *Proceedings of the AGARD-VKI Lecture Series on Unstructured Grid Methods for Advection Dominated Flows*, 1992. AGARD Report 787.

18. G. Karypis and V. Kumar. Metis, Unstructured Graph Partitioning and Sparse Matrix Ordering System, v.2.0. Technical report, University of Minnesota, Department of Computer Science, 1995.

19. X. Liu and P.D. Lax. Positive Schemes for Solving Multi-dimensional Hyperbolic Systems of Conservation Laws. *CFD Journal*, 5(2):133–156, 1996.

20. F.R. Menter. Two-Equation Eddy-Viscosity Turbulence Models for Engineering Applications. *AIAA-Journal*, 32(8):1598–1605, 1994.

21. H. Paillère. *Multidimensional Upwind Residual Distribution Schemes for the Euler and Navier-Stokes Equations on Unstructured Grids.* PhD thesis, Université Libre de Bruxelles, Belgium, 1995.

22. H. Paillère, J.-C. Carette, and H. Deconinck. Multidimensional upwind and SUPG methods for the solution of the compressible flow equations on unstructured grids. In *VKI LS 1994-05, Computational Fluid Dynamics*, 1994.

23. O. Hassan; K. Morgan; E.J. Probert; J. Peraire. Unstructured tetrahedral mesh generation for three-dimensional viscous flows. *International journal for numerical methods in engineering*, 39:549–567, 1996.

24. P.L. Roe. Discrete Models for the numerical analysis of time-dependent multidimensional gas dynamics. *Journal of Computational Physics*, 63, 1986.

25. P.L. Roe. Linear advection schemes on triangular meshes. Technical report, Cranfield Institute of Technology, November 1987. CoA 8720.

26. P.L. Roe. "Optimum" upwind advection on a triangular mesh. Technical Report 90-75, ICASE, 1990.

27. D. Sidilkover. A genuinely multidimensional upwind scheme and efficient multigrid solver for the compressible Euler equations. Technical Report 94-84, ICASE, 1994.

28. D. Sidilkover. A genuinely multidimensional upwind scheme for the compressible Euler equations. In J. Glimm, M.J. Graham, J.W. Grove, and B.J. Plohr, editors, *Proceedings of the Fifth International Conference on Hyperbolic Problems: Theory, Numerics, Applications.* World Scientific, June 1994.

29. D. Sidilkover. Multidimensional upwinding and multigrid. 1995. 12th AIAA CFD Conference, San Diego, Paper 95-1759.

30. P. R. Spalart and S. R. Allmaras. A one-equation turbulence model for aerodynamical flows. *La Recherche Aérospatiale*, (1):5–21, 1994.

31. R. Struijs, H. Deconinck, and P.L. Roe. Fluctuation Splitting Schemes for the 2D Euler Equations. In *VKI LS 1991-01, Computational Fluid Dynamics*, 1991.

32. T.E. Tezduyar and Y.J. Park. Discontinuity-capturing finite element formulations for nonlinear convection-diffusion-reaction equations. *Computer Methods in Applied Mechanics and Engineering*, 59:307–325, 1986.

33. E. van der Weide and H. Deconinck. Positive Matrix Distribution Schemes for Hyperbolic Systems, with application to the Euler Equations. pages 747–753. Wiley, 1996. Proceedings of the 3rd European CFD Conference.

34. E. van der Weide and H. Deconinck. Upwind Residual Distribution methods for compressible flow: An alternative to Finite Volume and Finite Element methods; Part II: System Schemes and Applications. In *VKI LS 1997-02, Computational Fluid Dynamics*, 1997.

35. D.C. Wilcox. Reassessment of the Scale-Determining Equation for Advanced Turbulence Models. *AIAA-Journal*, 26(11):1299–1310, 1988.