Ventura, D. M.
dvsmart@hotmail.com

Lyra, P. R. M.
prmlyra@padmec.org

Willmersdorf, R. B.
ramiro@willmersdorf.net
Universidade Federal de Pernambuco
Mechanical department
Av. Acadêmico Hélio Ramos, s/n
50.740-530 Recife-PE, Brazil

Silva, R. S.
rogerio.soaress@ufpe.br

Antunes, A. R. E.
areantunes@yahoo.com.br
Núcleo de Tecnologia-CAA-UFPE
Rodovia BR-104 Km 59
55.002-970 Caruaru-PE, Brazil

An Edge-Based Unstructured Mesh Formulation for High Speed Tridimensional Compressible Flow Simulation

Numerical simulation of realistic compressible flows is very important and requires accurate and flexible tridimensional formulations, which should furthermore be robust and efficient. In this work we describe the development of a computational tool for numerical simulation of inviscid compressible 3-D fluid flow problems. This tool uses as the main building block an edge-based Galerkin FEM (Finite Element Method) together with a MUSCL (Monotonic Upstream-centered Schemes for Conservations Laws) approach to get a higher-order scheme with LED (Local Extremum Diminishing) property. The code is particularly developed for the simulation of supersonic and hypersonic flow regimes and several important (sometimes unavoidable) numerical procedures incorporated to increase its robustness are described. Some aspects related to the adoption of an edge-based data structure and other implementation issues are also described. Finally, some numerical model problems are analyzed and compared with results found in the literature demonstrating the effectiveness of the developed tool.

***Keywords**: 3-D compressible flows, Euler equations, FEM-MUSCL/LED, unstructured tetrahedral meshes, edge-based formulation*

Introduction

Several industrial applications involve high speed compressible fluid flows, and the development of computational tools to deal with such class of problems, although quite well established nowadays, is still an important research area. The main issues remain obtaining accurate and robust computational models to deal with severe regimes and developing flexible and efficient codes to cope with complex geometries and large scale problems.

In this paper we describe the extension to tridimensional models of the FEM-MUSCL/LED formulation developed by Lyra (1994), Lyra and Morgan (2002). The extension follows the work of Peraire et al. (1993) and includes several numerical strategies proposed or used by Lyra (1994) and Lyra and Morgan (2000, 2002) to increase the robustness and convergence behavior of the final scheme when simulating high speed inviscid compressible flows. Here, these strategies are extended and tested for 3-D model problems. The core of the code consists on an edge-based Galerkin FEM (Finite Element Method), using conservative variables, together with a fully upwind limited MUSCL (Monotonic Upstream-centered Schemes for Conservations Laws) approach to get a higher-order scheme with LED (Local Extremum Diminishing) property. The main concern is the steady-state solution of supersonic (and hypersonic) fluid flows. The solution is integrated in time using a simple explicit Euler forward approach, with a "lumped mass" matrix and local time stepping. A number of specific strategies were adopted to render the final scheme robust, including: adequate choices during limiting stage (limiter functions, set of variables to be limited, etc.); warm start for higher-order simulation; relaxation to guarantee positivity of thermodynamic variables; possible solid wall boundary condition for impulsive start from the free stream condition; and a discussion on the necessity of using a background numerical dissipation and a strategy for freezing the non-linear limiting term.

In what follows, first we briefly describe the physical-mathematical model considered. Then we present the main ingredients necessary to get the computational model, including: the edge-based FEM formulation; the incorporation of a MUSCL-LED high-order "monotonic preserving" reconstruction; the time discretization adopted; how to implement the standard boundary and initial conditions, and several important numerical and computational issues. The following section discusses some numerical benchmark applications and, finally, some conclusions are drawn.

Mathematical Model

The equations which govern the unsteady laminar tridimensional flow of a compressible inviscid fluid, in the absence of external source terms, may be written in the conservation form

$$\partial \boldsymbol{U} / \partial t + \partial \boldsymbol{F}^{j} / \partial x_{j} = 0 \quad \text{in} \quad \Omega \times I \quad \text{for} \quad j = 1,2,3 \tag{1}$$

where the summation convention is employed, x_j is the Cartesian coordinate, Ω is the spatial domain, t is the time independent variable and $I = (t^0, T)$ is the time interval. The solution of this set of equations is sought over the closed spatial domain Ω, with boundary surface Γ, and the time interval I. This initial/boundary value problem requires additionally boundary and initial conditions, which are taken here in the form

$$\begin{aligned} &\boldsymbol{F}^{n} = n_j \boldsymbol{F}^{j} = \overline{\boldsymbol{F}}^{n} && \text{at} \quad \Gamma \times I \\ &\boldsymbol{U}\left(\boldsymbol{x}_j, t^0\right) = \overline{\boldsymbol{U}}\left(\boldsymbol{x}_j\right) && \text{on} \quad \Omega \times t^0 \end{aligned} \tag{2}$$

in which n_j represents the component, in x_j direction, of the unit normal vector to Γ and $\overline{\boldsymbol{U}}\left(\boldsymbol{x}_j\right)$ is a known function at initial time t^0. In Eq. (1), $\boldsymbol{U}$ is the vector of the conservative variables, while $\boldsymbol{F}^{j}$ denotes the inviscid flux vector in the direction x_j. These vectors can be written as

$$\begin{aligned} \boldsymbol{U}^T &= \left[\rho, \rho u_1, \rho u_2, \rho u_3, \rho\varepsilon\right] \\ \boldsymbol{F}^{j} &= \left[\rho u_j, \rho u_1 u_j + p\delta_{1j}, \rho u_2 u_j + \right. \\ &\qquad \left. p\delta_{2j}, \rho u_3 u_j + p\delta_{3j}, \left(\rho\varepsilon + p\right) u_j\right] \end{aligned} \tag{3}$$

Paper received 11 November 2010. Paper accepted 6 December 2011.
Technical Editor: Fernando Rochinha

Here ρ, p, ε represent the fluid density, pressure and total specific energy respectively, and u_j is the component of the velocity vector in x_j direction. In addition, δ_{ij} denotes the Kronecker Delta. The equation set is closed by the addition of the perfect gas equation of state given as

$$p = (\gamma - 1)\rho\left(\varepsilon - (u_i u_i)/2\right) \tag{4}$$

where $\gamma = c_p / c_v$, with c_p and c_v being the specific heats of the fluid at constant pressure and volume, respectively. A non-dimensional counterpart of this mathematical model is then discretized as described in what follows.

Computational Model

The Galerkin finite element formulation, implemented in an edge-based form, together with a MUSCL approach, are used as the basic building block for the construction of a tridimensional unstructured grid scheme using a higher order upwind biased procedure for the solution of compressible Euler system of equations. An explicit Euler forward time discretization is used to advance the solution in time. All that will be described in what follows, together with several important numerical and implementation issues necessary to get a robust computational code.

Variational formulation

Assuming that the spatial domain Ω is discretized into an unstructured assembly of linear tetrahedral elements, with the nodes numbered from 1 to p, the subsets $\boldsymbol{V}^{(p)}$ and $\boldsymbol{W}^{(p)}$ of the trial and weighting function sets $\boldsymbol{V}$ and $\boldsymbol{W}$, respectively, a weak variational approximation to the problem given by Eqs. (1) to (4) using the Galerkin FEM (Zienkiewicz at al., 2006) can be stated as

$$\text{Find } \boldsymbol{U}^{(p)} \in \boldsymbol{V}^{(p)} \mid \forall \boldsymbol{w}^{(p)} \in \boldsymbol{W}^{(p)}, t > t^0$$
$$\sum_{E \supset I} \int_{\Omega_e} \frac{\partial \boldsymbol{U}^{(p)}}{\partial t} N_I d\Omega = \sum_{E \supset I} \int_{\Omega_e} \boldsymbol{F}^j\left(\boldsymbol{U}^{(p)}\right) \frac{\partial N_I}{\partial x_j} d\Omega - \sum_{f \supset I} \int_{\Gamma_f} \overline{\boldsymbol{F}}^n N_I d\Gamma \tag{5}$$

for each, $I = 1, 2, \ldots, p$ where the summations just extend over those elements E and boundary faces f which contain node I, N_I denotes the nodal standard Galerkin finite element shape/weighting functions and the compact support of them were used in order to evaluate the integrals by summing individual element contributions.

To avoid the necessity of numerical integration, both the unknown $\boldsymbol{U}^{(p)}$ and the inviscid flux $F^j(\boldsymbol{U}^{(p)})$ are approximated in a piecewise linear fashion in terms of their nodal values. By inserting the assumed forms for $\boldsymbol{U}^{(p)}$ and $F^j(\boldsymbol{U}^{(p)})$ into Eq. (5), all integrals can be evaluated in closed form and the approximate weak variational statement leads to

$$\begin{aligned} \sum_{E \supset I}\left[\int_{\Omega_e} N_I N_J d\Omega\right] \frac{d\boldsymbol{U}_I}{\partial t} = & \sum_{E \supset I} \frac{\Omega_E}{4} \frac{\partial N_I^E}{\partial x_j}\left(\boldsymbol{F}_I^j + \boldsymbol{F}_J^j + \boldsymbol{F}_K^j + \boldsymbol{F}_L^j\right) - \\ & \sum_{f \supset I} \frac{\Gamma_f}{12}\left(2\overline{\boldsymbol{F}}^n_I + \overline{\boldsymbol{F}}^n_J + \overline{\boldsymbol{F}}^n_K\right) \end{aligned} \tag{6}$$

where Ω_E denotes the volume of the tetrahedral element E, with nodes I, J, K and L, and Γ_f denotes the area of the boundary triangular face with nodes I, J and K. In Eq. (6), the first and second summations extend for all E elements that contain node I and the third extends for all boundary faces f that contain node I.

The standard finite element data structure consists of the physical coordinates listed by node number, a list of the nodal connectivity for each element and a list of the boundary faces connectivities. With this geometrical and topological data, the integrals discussed above can be performed by a loop over the elements and a loop over the boundary faces, with the element and faces contributions to the nodes being accumulated during the process. However, in order to get a more efficient computational implementation, both in terms of memory requirements and CPU time (Peraire et al., 1993; Catabriga and Coutinho, 2002; Kusmin et al., 2005; Smolarkiewicz and Szmelter, 2005; Löhner, 2008), an alternative data structure can be adopted which considers a list of edges and boundary faces, as will be further discussed in the following section.

Edge/Face Based Formulation

All integrals given in Eq. (6) can be performed in an alternative way using an edge/face data structure. By considering that a node I is directly connected with μ_I edges and I_v boundary faces, then an equivalent discrete form of Eq. (6) can be written as (Peraire et al., 1993; Ventura, 2008):

$$\begin{aligned} \left[\boldsymbol{M}\frac{d\boldsymbol{U}}{dt}\right]_I = & -\left\{\sum_{S=1}^{\mu_I} L_{IIs}\boldsymbol{F}_{\mathrm{IIs}}\right\}/2 + \\ & \left\langle \sum_{f=I_1}^{I_v} D_f\left(4\overline{\boldsymbol{F}}^n_I + 2\overline{\boldsymbol{F}}^n_{I1f} + 2\overline{\boldsymbol{F}}^n_{I2f} + 2\boldsymbol{F}_I^n - \boldsymbol{F}_{I1f}^n - \boldsymbol{F}_{I2f}^n\right)\right\rangle_I \end{aligned} \tag{7}$$

where $\boldsymbol{M}$ is the consistent finite element “mass” matrix, the first summation is over all S edges (or sides) which contains node I and $\boldsymbol{F}_{IIs} = \boldsymbol{F}_I^j S_{IIs}^j + \boldsymbol{F}_{Is}^j S_{IIs}^j$. The term $\langle\ \rangle_I$ will exist only when node I is a boundary node. When this happens, node I will be part of I_v boundary faces B_{I1}, B_{I2}, B_{I3},…, B_{Iv}, where the boundary face B_{If} has nodes I, I_{1f}, I_{2f}. For notational convenience, we defined $L_{IIs} = |\boldsymbol{C}_{IIs}|$ and $S_{IIs}^j = C_{IIs}^j / L_{IIs}$ with $\boldsymbol{C}_{IIs}$ being the weighting coefficient vector. The coefficient C_{IIs}^j denotes the weight, in x_j direction, which must be applied to the average flux of the edge S, which joins nodes I and I_s to obtain the contribution made by that edge to node I. The coefficient C_{IsI}^j denotes a similar weight, but now to obtain flux contribution to node I_s. The coefficient D_f denotes the boundary face weight coefficient to complete the approximate computation of all integrals given on the right side of Eq. (5) for nodes I that lie on the boundary. Subscripts I_{1f} and I_{2f} denote the two other nodes of the boundary face which contains node I. Finally, $\overline{\boldsymbol{F}}^n$ and $\boldsymbol{F}^n$ represent the prescribed and calculated fluxes in the boundary faces normal direction, respectively.

The coefficients C_{IIs}^j and D_f are computed by

$$\begin{aligned} C_{IIs}^j &= -\sum_{E \supset IIs} \frac{\Omega_E}{2} \frac{\partial N_I^E}{\partial x_j} + \left\langle \sum_{f \supset IIs} \frac{\Gamma_f}{12} n_j \right\rangle_{IIs} \\ D_f &= -\frac{\Gamma_f}{24} \end{aligned} \tag{8}$$

where first summation extends for all E elements which contains *IIs* edge. The term $\langle \; \rangle_{IIs}$ will be included only if *IIs* edge belongs to the boundary surface. These coefficients satisfy the relations

$$
\begin{aligned}
&\sum_{S=1}^{\mu_I} C_{IIs}^{j} - \left\langle \sum_{f=I_1}^{I_v} 8 D_f n_i \right\rangle_I = 0 \\
&C_{IIs}^{j} + C_{IsI}^{j} = 0 \quad \text{for} \quad S = 1, 2, \ldots, \mu_I
\end{aligned}
\tag{9}
$$

for $j = 1,2,3$. From the anti-symmetry of the edge weights expressed in second relation of Eq. (9), the numerical discretization can be immediately noticed to possess a local conservation property.

For a detailed deduction of the edge based formulation see Lyra (1994) and Ventura (2008) for the 3-D extension. It can be demonstrated, as a result of relations given in Eq. (9), that the scheme given in Eq. (7) is a central difference type discretization, in non-structured meshes, which is not a suitable approximation to the hyperbolic Euler system of equations, Hirsch (2010). However, a combination of a central difference ("Galerkin" FEM) discretization and some type of stabilization and shock-capturing terms can be used to construct an effective scheme. This can be done, using some ideas taken from finite difference method (FDM) and finite volume method (FVM), substituting the edge flux $\boldsymbol{F}_{IIs} = \boldsymbol{F}_I^j S_{IIs}^j + \boldsymbol{F}_{Is}^j S_{IIs}^j$, which is a true Galerkin flux in Eq. (7), by a consistent numerical flux $\boldsymbol{F}_{\mathrm{IIs}}$ defined on generic unstructured tetrahedra meshes.

Stabilization and Shock-Capturing

First order approach – Roe's approximate Riemann solver

A large number of engineering and science problems are governed by conservation laws expressed in terms of hyperbolic partial differential equations, such as the Euler equations. Techniques inherit from the theory of hyperbolic partial differential equations have been incorporated in some numerical methods resulting in successful solution algorithms. In particular, the perturbations of physical propagation along the characteristic lines, which are typical in hyperbolic equations, play an important role in a class of numerical methods known as Upwind methods. The robustness of the Upwind discretization, the possibility of physical interpretation and the high order approximation away from discontinuities emphasize the popularity of Upwind methods among computational fluid dynamics algorithm developers, Lyra (1994).

In the present context, the first order (and also higher order) Upwind methods implementation involves the solution of Riemann problems, Godunov (1959) and Hirsch (2010). In this work, Roe´s approximate Riemann solver (Roe, 1981) is adopted. The main advantage of Roe's approach is its relatively low computational cost and good accuracy for representation of single discontinuities. Here, the Roe scheme is implemented by defining the consistent numerical flux $\boldsymbol{F}_{IIs}$ as

$$
\boldsymbol{F}_{IIs} = \left\{ F_I^j S_{IIs}^j + F_{Is}^j S_{IIs}^j - \left| A_{IIs} \right| \left(U_{Is} - U_I \right) \right\} / 2
\tag{10}
$$

where the first term of the right hand side of Eq (10) represents the Galerkin (central difference-type) term and the second represents an implicit numerical diffusion term that stabilizes the scheme. The Roe Jacobian matrix $\boldsymbol{A}_{IIs}$, is calculated in the edge coefficient direction $\boldsymbol{C}_{IIs}$ using Roe's average (Roe, 1981). The final scheme is very robust and gives monotonic solutions. However, it leads to some drawbacks such as the non-recognition of stationary expansion waves, violating the entropy condition, and low (first) order accuracy.

A remedy to the entropy violation problem of Roe´s scheme was suggested by Harten and Hyman (see Hirsch, 2010) by replacing $\left|\bar{\lambda}_k\right|$ to function $\psi\left(\bar{\lambda}_k\right)$ in the computation of $\left|\boldsymbol{A}_{IIs}\right|$, with

$$
\begin{aligned}
&\psi\left(\bar{\lambda}_k\right) = \left|\bar{\lambda}_k\right| \quad \text{if} \quad \left|\bar{\lambda}_k\right| \geq \delta_k \\
&\psi\left(\bar{\lambda}_k\right) = \left(\bar{\lambda}_k^{\,2} + \delta_k^{\,2}\right) / \delta_k^{\,2} \quad \text{if} \quad \left|\bar{\lambda}_k\right| < \delta_k
\end{aligned}
\tag{11}
$$

where δ_k denote small positive numbers, either constant or the same for all fields "k" or more elaborately defined as a function of the local flow condition (Yee, 1989; Hirsch, 2010), and $\bar{\lambda}_k$ refers to the eigenvalues of Roe´s average Jacobian matrix.

Higher order approach – MUSCL/LED scheme

A flexible approach for building high-order schemes consists in the adoption of the so called MUSCL (Monotonic Upstream-centered Schemes for Conservation Laws) methodology (Van Leer, 1979; Hirsch, 2010). This is achieved by adopting a piecewise linear reconstruction using neighboring information. If, in addition to the reconstruction stage, a proper monotonic constraint is employed, normally in the form of non-linear slope limiters, the final scheme guarantees sharp results, "free" (or almost) from numerical oscillations.

MUSCL reconstruction: To build a high-order scheme an extended stencil must be used. This can be done in different ways.

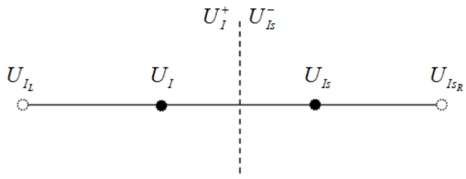


Figure 1. Extended fictitious stencil and interface values for the MUSCL/LED scheme.

Here, we consider a "structured" fictitious extended stencil of four points formed by the edge nodes *I* and *Is* and two "ghost" nodes I_L and Is_R located equidistantly along the line that contains nodes *I* and I_S (see Fig. 1).

In order to obtain the values at "ghost" nodes, one can either use some form of interpolation or alternatively use the gradients (Lyra and Morgan, 2002). In the present work, a gradient reconstruction was adopted. As linear shape functions were adopted, this leads to constant gradients over each element, with multiple values at node, so a global variational recovery (Zienkiewicz at al., 2006) was used to compute a continuous gradient field $\boldsymbol{\nabla U}_I$. Using an edge-based data structure, this can be performed with an expression similar to that of Eq. (7), i.e.

$$
\begin{aligned}
\left[\boldsymbol{M}_L \frac{d\boldsymbol{U}}{dx_j} \right]_I = {} & \sum_{S=1}^{\mu_I} \boldsymbol{C}_{IIs}^{j} \left(\boldsymbol{U}_I + \boldsymbol{U}_{Is} \right) / 2 - \\
& \left\langle \sum_{f=I_1}^{I_v} \boldsymbol{D}_f \left(6\boldsymbol{U}_I^n + \boldsymbol{U}_{I1f}^n + \boldsymbol{U}_{I2f}^n \right) \right\rangle_I
\end{aligned}
\tag{12}
$$

where $\boldsymbol{M_L}$ is the "lumped" (diagonal) "mass" matrix, which replaces the consistent FEM "mass" matrix $\boldsymbol{M}$ to avoid the need to solve an algebraic system of equations.

Using a truncated Taylor series expansion (Hirsch, 2010), it can be shown that the extrapolated values, for the stencil of Fig. 1, can be rewritten as follows:

$$\begin{aligned} \boldsymbol{U}_{I_L} &= \boldsymbol{U}_I - 2\nabla \boldsymbol{U}_I \boldsymbol{IIs} - \left(\boldsymbol{U}_{Is} - \boldsymbol{U}_I\right) \\ \boldsymbol{U}_{Is_R} &= \boldsymbol{U}_{Is} + 2\nabla \boldsymbol{U}_{Is} \boldsymbol{IIs} - \left(\boldsymbol{U}_{Is} - \boldsymbol{U}_I\right) \end{aligned} \qquad (13)$$

where $\boldsymbol{IIs}$ is the vector connecting nodes I and Is, and $\boldsymbol{\nabla U}_I$ represents the gradient of $\boldsymbol{U}$ at node I calculated as given in Eq. (12). The four nodes stencil allows the extension from first order to higher order scheme.

Using the extended stencil, found as described previously, the values at the cell interfaces, for example, U_I^+ and U_{Is}^- (see Fig. 1), can be computed from a totally "Upwind" extrapolation (see Hirsch, 2010; Lyra, 1994) as

$$\begin{aligned} \boldsymbol{U}_I^+ &= \boldsymbol{U}_I + \left(\boldsymbol{U}_I - \boldsymbol{U}_{I_L}\right)/2 \\ \boldsymbol{U}_{Is}^- &= \boldsymbol{U}_{Is} - \left(\boldsymbol{U}_{Is_R} - \boldsymbol{U}_{Is}\right)/\ 2 \end{aligned} \qquad (14)$$

By inserting Eq. (13) into (14), the interface values can be rewritten as

$$\begin{aligned} \boldsymbol{U}_I^+ &= \boldsymbol{U}_I + \left(2\boldsymbol{\nabla U}_I \boldsymbol{IIs} - \boldsymbol{\Delta U}\right)/2 \\ \boldsymbol{U}_{Is}^- &= \boldsymbol{U}_{Is} - \left(2\boldsymbol{\nabla U}_{Is} \boldsymbol{IIs} - \boldsymbol{\Delta U}\right)/\ 2 \end{aligned} \qquad (15)$$

where $\boldsymbol{\Delta U} = \boldsymbol{(U}_{Is} - \boldsymbol{U}_I\boldsymbol{)}$ and the computational implementation does not really requires "explicitly" extended stencils, making directly use of nodal gradients.

The reconstructed interface values allow the construction of stabilized, high order fully "Upwind" scheme, which might present oscillations near discontinuities. Thus, it is necessary to introduce some strategy to guarantee some monotonicity properties for the solution.

Limiting procedure: The main mechanism to guarantee higher order LED (Local Extremum Diminishing) schemes (Jameson, 1993; Lyra, 1994), which is a concept closely connected with TVD (Total Variation Diminishing) property, is the use of non-linear limiter functions. These functions impose restrictions over the dependent variables variations or over the flux functions leading to the possibility of designing monotonicity preserving high-resolution schemes.

To assure monotonicity preservation, limiter functions are introduced at interface extrapolated values, and the expressions given in Eq. (15) are now calculated as follows:

$$\begin{aligned} \boldsymbol{U}_I^+ &= \boldsymbol{U}_I + \varphi_I\left(r_I^+\right)\left(2\boldsymbol{\nabla U}_I \boldsymbol{IIs} - \boldsymbol{\Delta U}\right)/2 \\ \boldsymbol{U}_{Is}^- &= \boldsymbol{U}_{Is} - \varphi_{Is}\left(r_{Is}^-\right)\left(2\boldsymbol{\nabla U}_{Is} \boldsymbol{IIs} - \boldsymbol{\Delta U}\right)/2 \end{aligned} \qquad (16)$$

where $\boldsymbol{r}_I^+ = \boldsymbol{\Delta U}\ /\ \boldsymbol{\Delta U}_I^+$, $\boldsymbol{r}_{Is}^- = \boldsymbol{\Delta U}\ /\ \boldsymbol{\Delta U}_{Is}^-$ with $\boldsymbol{\Delta U}_I^+ = \boldsymbol{(U}_I - \boldsymbol{U}_{I_L}\boldsymbol{)}$, $\boldsymbol{\Delta U}_{Is}^- = \boldsymbol{(U}_{Is_R} - \boldsymbol{U}_{Is}\boldsymbol{)}$ and $\varphi_I\left(r_I^+\right)$, $\varphi_{Is}\left(r_{Is}^-\right)$ are limiter functions.

An alternative notation can be introduced where $\varphi_{(1)}\left(r\right) = L_{(2)}\left(1,r\right) \equiv L_{(2)}\left(a,b\right)$, with $r = b/a$, and the possibility of zero in the denominator and extra conditional sentence to avoid that in the computational implementation are eliminated. Chakravarthy-Osher (see Hirsch, 2010) limiter function is an example of a limiter function and has the following general expression:

$$L\left(a,b\right) = sign\left(a\right) max\left[0, min\left(\left|a\right|, \beta b sign\left(a\right)\right)\right] \qquad (17)$$

where for a fully explicit "upwind" scheme $1 \le \beta \le 2$. The parameter β was introduced to allow the original *minmod* limiter function $\boldsymbol{(}\beta = 1\boldsymbol{)}$ to become more compressible. There are several alternative limiter functions, but, in this work, only the *minmod* limiter function is adopted. More details about limiter functions can be found in Hirsch (2010) and Lyra (1994).

With the interface values in hands, given by Eq. (16), the first-order numerical flux function $\boldsymbol{F}_{IIs}$ given in Eq. (10) can now be replaced by

$$\begin{aligned} \boldsymbol{F}_{\boldsymbol{IIs}} = \Big\{ & F^j\left(U_I^+\right)S_{IIs}^j + F^j\left(U_{Is}^-\right)S_{IIs}^j - \\ & \left|A^S\left(U_I^+, U_{Is}^-\right)\right|\left(U_{Is}^- - U_I^+\right)\Big\}/\ 2 \end{aligned} \qquad (18)$$

and the final formulation given by Eq. (7), with the flux defined in Eq. (18), represents an unstructured higher-order slope-limited (MUSCL/LED) scheme.

Time Integration

Equation (7) represents a system of ordinary differential equations and must be further discretized in time to produce a practical solution algorithm. The time evolution of the unknown vector $\boldsymbol{U}_I\boldsymbol{(t)}$ at node I of the mesh, using a simple Euler forward finite difference approximation to the time derivative, can be written as

$$\boldsymbol{U}_I^{n+1} = \boldsymbol{U}_I^n + \Delta \boldsymbol{t}_I \left[\boldsymbol{M}_L\right]_I^{-1} \boldsymbol{RHS}_I^n \qquad (19)$$

In Eq. (19), $\boldsymbol{RHS_I^n}$ represents the right hand side of Eq. (7) computed at time level t^n, $\boldsymbol{U}_I^n$ is the unknown vector at t^n, $\left[\boldsymbol{M}_L\right]_I^{-1}$ is the inverse of the diagonal (lumped) mass matrix, and $\boldsymbol{\Delta t}_I$ is the nodal time-step interval vector. The consistent finite element mass matrix $\boldsymbol{M}$ was replaced by the standard lumped (diagonal) mass matrix to enable truly explicit time integration and does not alter the final steady state solution, which is of primary concern here. Despite some possible loss of temporal accuracy, this approximation was also adopted for the transient computations performed in the shock tube application presented in this paper.

Performing a stability analysis, based on the energy method (Giles, 1987), provides the following criteria to compute $\boldsymbol{\Delta t}$ for unstructured mesh "cell-vertex" algorithms

$$\begin{aligned} \Delta \boldsymbol{t}_I &= 2CFL\left[\boldsymbol{M}_L\right]_I \left[\frac{\mu_I}{\sum\limits_{S=1} L_{IIs}\left|\left(\lambda_{\boldsymbol{max}}\right)_{IIs}\right|}\right]^{-1} \\ \text{with}\quad \left(\lambda_{\boldsymbol{max}}\right)_{IIs} &= \left|\boldsymbol{u}_{IIs} \cdot \boldsymbol{S}_{IIs}\right| + c_{IIs} \end{aligned} \qquad (20)$$

which was written conveniently for the edge-based notation adopted here, and with $\boldsymbol{u}_{IIs}$ and C_{IIs} denoting the edge values of the fluid velocity vector and the speed of sound, respectively, and λ_{max} representing the largest eigenvalue (spectral radius) of the Jacobian matrix $\boldsymbol{A}_{IIs}$. These edge values are obtained by averaging the appropriate nodal values.

It must be remarked that the time step $\Delta \boldsymbol{t}_I$ must satisfy a CFL-like stability condition, which is more restrictive than the linear stability limit, to guarantee that the final scheme is TVD (or LED) stable (Hirsch, 2010; Jameson, 1993; Lyra, 1994). Also, when a steady-state analysis is studied, a local time stepping is employed to accelerate the convergence rate towards steady-state. This is implemented by specifying a constant CFL, in Eq. (20), throughout the mesh. For true transient simulation, the minimum local time-step, i.e. $min(\Delta \boldsymbol{t}_I) \ \forall I$, is used for the whole discretization.

Boundary and Initial Conditions

To have a fully discretized model we must describe how we deal with the different boundary and initial conditions normally present on inviscid laminar flow regimes.

Far field boundary condition: For a node I located at the far field boundary, the "prescribed" flux $\overline{F}_I^n$ of the boundary face loop presented in Eq. (7) is determined by solving a Riemann problem (Lyra and Morgan, 2002), employing, once again, Roe's Riemann solver (Roe, 1981), to resolve the interface between the computed value $\boldsymbol{U}_I$ and the free stream value $\boldsymbol{U}_\infty$. This means that

$$\overline{\boldsymbol{F}}_I^n = \left\{ \boldsymbol{F}^n(\boldsymbol{U}_I) + \boldsymbol{F}^n(\boldsymbol{U}_\infty) - \left|\boldsymbol{A}^n(\boldsymbol{U}_I, \boldsymbol{U}_\infty)\right| (\boldsymbol{U}_\infty - \boldsymbol{U}_I) \right\} / 2 \qquad (21)$$

in which the Roe matrix $\left|\boldsymbol{A}^n(\boldsymbol{U}_I, \boldsymbol{U}_\infty)\right|$ is evaluated in the direction normal to the boundary.

Solid wall and symmetric boundary condition: At a solid wall (Γ_w) the "prescribed" flux $\overline{F}_I^n$ is set equal to the computed flux $\overline{F}_I^n$ and the normal component of the velocity $\boldsymbol{u}^n$ is set to zero at each time step, i.e. $\boldsymbol{u}^n \cdot \boldsymbol{n} = 0$ at Γ_w.

Initial condition: The initial value of the vector $\boldsymbol{U}_I$ is set equal to a known value $\overline{\boldsymbol{U}}_I$, Eq. (2), which is a function of a prescribed Mach number and angle of attack.

Important Numerical and Computational Aspects

Several remarks concerning some numerical peculiarities of the formulation previously described are opportune. The importance of the strategies to be discussed in what follows is such that they can be responsible for the success or failure of the analysis. This is particularly true for hypersonic flow simulation. Some brief comments on general aspects of the computational implementation and the use of some available enabling technologies are also presented.

Numerical Issues

Variable choice during limiting stage: The limiting procedure can be imposed on the primitive, conservative or characteristic variables. Lyra (1994) reports that the use of the primitive variables led to better behavior than the use of the conservative variables or mixing primitive and conservative variables. Also, the use of the characteristic variables would require some extra computation, being computationally more expensive. These facts and several numerical results (robust and oscillation-free solutions for the analyzed problems) support the choice of primitive variables adopted in this work. However, it should be mentioned that Yee (1989) reports that the choice of characteristic variables plays an important role in the stability and convergence rate as the Mach number increases, and also that when using characteristic variables one can use different limiter functions for each variable, exploiting some characteristics of specific limiters which are better designed for linear or non-linear fields (Lyra, 1994; Lyra and Morgan, 2000).

To increase the robustness of the final scheme, Lyra (1994) reports that, when using primitive variables it was very important to apply the limiting procedure for the velocity field in the weighting coefficient direction and the normal direction to this coefficient. Thus, the velocities are initially projected onto these directions and after applying the limiters they are projected back onto the Cartesian direction. This idea was inspired in FVM (Finite Volume Method) which works with normal and tangential directions to the control volume surface and was adopted here.

Enhancement of stability and convergence rate on high speed flow simulation

Warm start for higher-order simulation: One very efficient and robust strategy adopted to find the steady state solution of challenging applications consists on advancing the solution a few steps using a first order scheme and then advancing the solution towards steady-state using a higher order scheme.

Positivity of thermodynamic variables: Due to the presence of high gradients, rarefaction zones and impulsive initial conditions, negative values of density and pressure can occur during the time integration process, mainly at the initial stages. To avoid negative values of thermodynamic variables during the iterative process, density and pressure are updated, whenever necessary, using some kind of relaxation. This must assure that they are always positive. For instance, pressure update is modified according to equation

$$p^{n+1} = p^n + \Delta p\left[1 + \eta\left(\alpha + \left|\Delta p \setminus p\right|\right)\right]^{-1} \quad \text{if} \quad (\Delta p \setminus p) \leq \alpha \qquad (22)$$

and the values adopted to η and α are 2.0 and -0.2, respectively. For more details see Lyra (1994).

Initial condition x solid wall boundary condition**:** The boundary condition $\boldsymbol{u}^n \cdot \boldsymbol{n} = 0$ at Γ_w is not consistent with the initial data (free stream condition) and an additional difficulty might appear when attempting to simulate severe flow regimes which contain obstacles. In fact, truly impulsive start of any mechanical system is not physically possible, but rapid start or impulsive acceleration is quite legitimate and can be implemented by considering that the free stream condition is achieved after a certain small time interval. Alternatively, one can adopt the free stream condition directly, but together with a relaxation on the solid wall boundary condition (Lyra, 1994). This is accomplished here by taking

$$\boldsymbol{u}^n \cdot \boldsymbol{n} = \boldsymbol{u}^{n-1} \cdot \boldsymbol{n}(1-e) \quad \text{at} \quad \Gamma_w \qquad (23)$$

where e $(0 < e \leq 1)$ is a parameter such that when its value is different from one, the solution will slip and penetrate the wall at the start of the transient, but as time evolves the normal velocity goes to zero at the wall. This procedure has been found to be very important for the simulation of high speed flow past blunt bodies, where the value e = 0.8 has been typically used.

Other Issues: The lack of background numerical dissipation and the non-linear nature of the limiting procedure were found to be the main causes of a bad convergence rate observed for some 2-D applications (Lyra, 1994). So, the existence of some form of background diffusion might be important for helping to damp high frequency modes on challenging high speed applications. This can be obtained if instead of using a central difference (or Galerkin), second-order method as the building block for the final scheme, an alternative method which has implicitly some background diffusion is adopted (Lyra, 1994; Lyra and Morgan 2002). Alternatively, a freezing strategy, by stopping the update of the limited corrective flux (non-linear) term, proposed in Lyra (1994), can be used when the solution approaches steady-state, helping to drop the residual towards machine zero, with no measurable error on the final solution.

The aspects described previously were not required for the 3-D applications analyzed here, but they could be necessary for more complex flow regimes. This probably happens due to the extra numerical diffusion associated with the extension of a 1-D like Upwind discretization into multidimensional models, being even bigger for 3-D models than for 2-D. Further investigation is required to confirm such conjecture.

Some Computational and Implementation Issues

The CFD simulator, which implements the scheme described in this work, was written in Fortran90 and was developed as an academic computer program to solve Euler equations. During the complete computational modeling and simulation process, some open source libraries and programs were adopted to enable fast and secure implementation and or to perform the pre- and post-processing steps:

The Gmsh system was adopted as a finite element grid generator which has a build-in CAD (Computer Aided Design) engine and a post-processor. It provides a fast, light and user-friendly meshing tool with parametric input and with some visualization capabilities. Of course, any other available CAD and 3-D tetrahedral mesh generator could have been used.

The adoption of an edge/face data structure and tetrahedral meshes has demanded a pre-processor program, which must extract this alternative data structure from the standard element data structure and pre-compute several data, such as: the edge and boundary face weighting coefficients, C_{IIs} and D_f respectively, the volume associated with each node (i.e. the lumped mass matrix) and the average nodal external normal vector. The pre-processor program was written in C++, making use of the FMDB (Flexible Distributed Mesh Data Base) library, which was adopted as an object manager to easily create the required data structure from a tetrahedral mesh without an explicit implementation of hash tables or similar algorithms.

The operations performed by edge/face based algorithm are, basically, loops over the edges, boundary faces and nodes of the mesh. For instance, a loop over the edges consists on: gather information from the nodes of each edge; operate on this information; scatter it back to the nodes of the edges and add it to the nodal quantities. These are the basic operations present in the developed simulator, easily implemented using Fortran90.

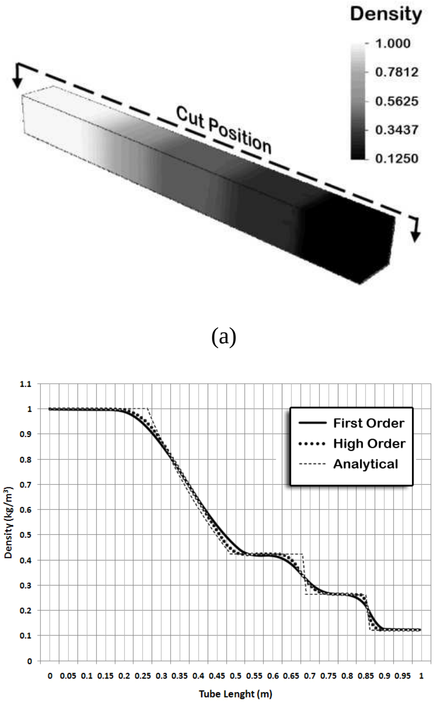
Paraview and Visit, which are free scientific data visualization tools, were used for post-processing simulations. They allow manipulating and post-processing data in a variety of ways to visualize scalar, vector and tensor fields on multidimensional structured and unstructured meshes.

Numerical Results

Shock tube problem

The first model analyzed was the classical transient shock tube problem. The numerical model simulates a long tube, initially divided by a diaphragm into two regions, which hold a stationary gas at two different states, left and right. The diaphragm is suddenly removed allowing the gas, at different states, to interact and a flow starts to develop. The structure of this one-dimensional flow turns out to be very interesting with the typical solution consisting of four constant states separated by three elementary waves: a linear degenerate contact discontinuity wave and two non-linear waves, each of which might be either a shock or a rarefaction wave depending on the left and right initial states (Hirsch, 2010). It represents a significant test case for the validation of any numerical algorithm developed for the solution of inviscid compressible flows.

The initial conditions considered for this classical Riemann problem are the following: left: $\rho = 1.0$, $\rho u_i = 0$, $\rho\varepsilon = 2.5$; right: $\rho = 0.125$, $\rho u_i = 0$, $\rho\varepsilon = 0.25$. The three dimensional domain adopted has length of 1 and 0.1 in both width and height. An uniform mesh with 15.988 nodes and 86.806 tetrahedra elements was used. The adopted boundary conditions are far field condition at left and right surfaces and solid slip wall for all the other four tube surfaces. Finally, we considered: $\gamma = 1.4$, $CFL = 0.45$, $\delta_k = 0.1$, a global time-step procedure and min-mod limiter function.



(a)

(b)

Figure 2. Shock Tube problem: a) density map; b) density along the tube using first order and higher order approaches.

In Figure 2a we can identify the four constant density states and three distinctive waves, which are better seen from Fig. 2b, which presents the density distribution along the tube. Both solutions, obtained with first and higher order schemes, are free from oscillations, capturing all relevant features with good accuracy. The higher order scheme gives sharper shock, contact discontinuity and rarefaction waves representation as expected. These results compare well with 1-D and 2-D solutions obtained with similar formulation (Lyra and Morgan, 2002), being slightly more diffusive.

Flow past a flat plate

The second example consists in the investigation of a regular shock reflection formed when a non-viscous and steady flow impinges over a flat plate with Mach number of 2.0 and angle of attack of minus 10° relative to the flat plate. This simple steady-state test case is another useful problem to verify computer algorithms, because there is an analytical solution calculated by similarity, Lyra and Morgan (2002), allowing an insight into the performance of the schemes extended in this work for three–dimensional simulations. The analytical solution of this 2-D problem consists in two constant states connected by a shock which has a theoretical slope of 29.3 degrees with the flat plate wall.

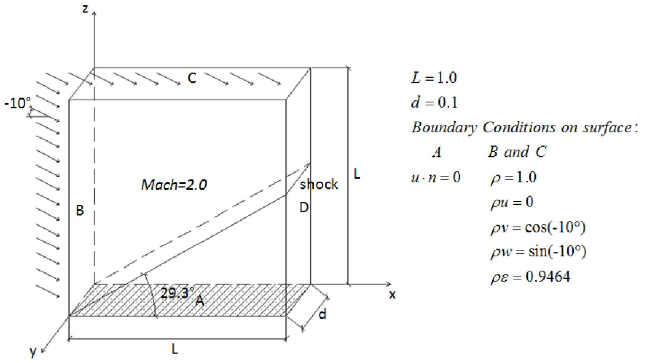
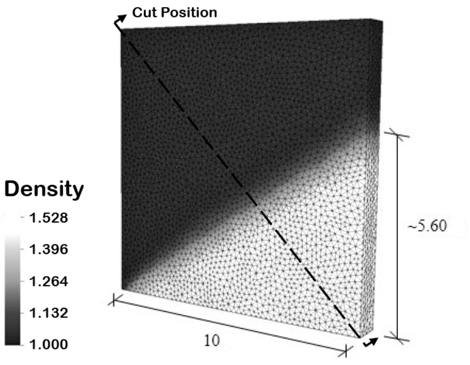
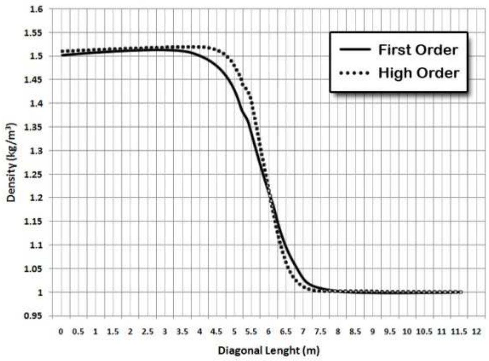


Figure 3. Description of flow past a flat plate problem and sketch of the analytical solution.

Figure 3 describes the whole problem, i.e. geometry, initial and main boundary conditions (i.e. surface A is a solid wall, B and C are inflow far field surface and D is a free far field surface). The frontal and back surfaces have symmetric boundary condition. The other parameters for this case are the following: $\gamma = 1.4$, *CFL* = 0.45, d_K = 0.1, and 10^{-4} as the tolerance to reach steady state, using density residual L_2-norm and min-mod limiter function. An uniform unstructured tetrahedra mesh with 108.322 elements and 20.340 nodes was adopted.



(a)



(b)

Figure 4. Flow past a flat plate problem: a) mesh and density map values using a higher order approach; b) density profile using first and higher order schemes.

The techniques presented to increase the robustness of the algorithm were implemented in the numerical solver, but were not required for these two simple applications presented here. A proper investigation as to their effective necessity in challenging 3-D problems has not yet been fully made, and since a local extended 1-D "structured" stencil is used for each edge of the mesh, the 3-D model approximation would involve even more "artificial" diffusion than its 2-D counterpart. So, it is expected that some of those techniques might be deactivated for certain applications that where required for 2-D models (Lyra and Morgan, 2002), without compromising the solution.

Concluding Remarks

A successful higher-order 3-D unstructured mesh, inviscid compressible flow solver has been fully described. Apart from the core FEM-MUSCL/LED formulation, several important numerical and computational ingredients were added to render the final scheme robust and ready to be tested for the simulation of challenging high-speed flow applications. The benchmark problems analyzed demonstrate the effectiveness of the developed code. The presented formulation and corresponding serial code can be extended for distributed memory parallel computation using similar methodology as previously adopted by the authors with similar formulation for incompressible flow studies (Antunes, 2008). The code can also be further extended to deal with more complex physical-mathematical models, such as: laminar flows and transient regimes, using improved time integration techniques, by extending the 2-D approaches (Lyra, 1994).

Acknowledgements

The authors acknowledge the financial support given by the Brazilian research councils CNPq, CAPES and FACEPE.

References

Antunes, A.R.E., 2008, "A Computational System Using Edge-based Finite Element Method and Fractional Time Step Formulation for the Solution of 3D Incompressible Flows on Parallel Computers" (in Portuguese), Ph.D. Thesis, Federal University of Pernambuco, Recife, Brazil, 163p.

Catabriga, L., Coutinho, A.L.G.A., 2002, "Implicit SUPG Solution of Euler Equations Using Edge-Based Data Structures", *Comput. Methods Appl. Mech. Engng.*, Vol. 191, pp. 3477-3490.

FMDB, <http:// www.scorec.rpi.edu/ FMDB/>.

Giles, M., 1987, “Energy Stability Analysis of Multi-Step Methods on Unstructured Meshes”, Technical Report CFDL-TR-87.1, M.I.T. CFD Laboratory Report.

Gmsh, http://geuz.org/gmsh/;

Godunov, S.K., 1959, “A Difference Scheme for Numerical Computation of Discontinuous Solutions of Hydrodynamic Equations. Math. Sbornik”, 47, pp. 271-306.

Hirsch, C., 2010, “Numerical Computation of Internal and External Flows”, Vol. 2 Wiley, 2nd ed.

Jameson, A., 1993, “Artificial Diffusion, Upwind Biasing, Limiters and their Effect on Accuracy and Multigrid Convergence in Transonic and Hypersonic Flows”. Technical Report 93-3359, AIAA Paper.

Kusmin, D., Löhner, R., Turek, S. (Eds), 2005, “Flux-Corrected Transport: Principles, Algorithms and Applications”, Springer.

Löhner, R., 2008, “Applied CFD Techniques: An Introduction Based on Finite Element Methods”, Wiley, 2nd. Ed.

Lyra, P.R.M., 1994, “Unstructured Grid Adaptive Algorithms for Fluid Dynamics and Heat Conduction”. Ph.D. Thesis, University of Wales at Swansea, Swansea, UK, 333 p.

Lyra, P.R.M. and Morgan, K., 2000, “A Review and Comparative Study of Upwind Biased Schemes for Compressible Flow Computation”. Part II: 1-D High-Order Schemes. Archives of Computational Methods in Engineering. Vol. 7, No. 3, pp. 333-377.

Lyra, P.R.M. and Morgan, K., 2002, “A Review and Comparative Study of Upwind Biased Schemes for Compressible Flow Computation”. Part III: Multidimensional Extension on Unstructured Grids. Archives of Computational Methods in Engineering. Vol. 9, No.3, pp. 207-256.

Paraview, <http://www.paraview.org/>.

Peraire, J., Peiró, J. & Morgan, K., 1993, “Finite Element Multigrid Solution of Euler Flows Past Installed Aero-engines”, Computational Mechanics, Springer-Verlag.

Roe, P.L., 1981, “Approximate Riemann Solvers, Parameter Vectors and Difference Schemes”, *Journal of Computational Physics*, Vol. 43, pp. 357-372.

Smolarkiewicz, P.K., Szmelter, J., 2005, MPDATA: “An edge-based unstructured-grid formulation”, *Journal of Computational Physics*, Vol. 206-2, pp. 624-649.

Van Leer, B., 1979, “Towards the Ultimate Conservative Difference Scheme V. A Second Order Sequel to Godunov's Method”, *Journal of Computational Physics*, Vol. 32, pp. 101-136.

Ventura, D.M., 2008, “Tridimensional Analysis of Inviscid Compressible Flows Using an Edge-Based Finite Element Method” (in Portuguese), Master Dissertation, Federal University of Pernambuco, Recife, Brazil, 71p.

VisIt, <http://www.llnl.gov/visit/home.html>.

Yee, H.C., 1989, “A Class of High-Resolution Explicit and Implicit Shock-Capturing Methods”. Technical Memorandum 101088, NASA.

Zienkiewicz, O.C., Taylor and P. Nithiarasu R.L., 2006, “The Finite Element Method for Fluid Dynamics”, Elsevier, 6ª edition.