

Study of Conservation on Implicit Techniques for Unstructured Finite Volume Navier-Stokes Solvers

Carlos Junqueira-Junior¹, Leonardo Costa Scalabrin², Edson Basso³, João Luiz F. Azevedo³

ABSTRACT: The work is a study of conservation on linearization techniques of time-marching schemes for the unstructured finite volume Reynolds-averaged Navier-Stokes formulation. The solver used in this work calculates the numerical flux applying an upwind discretization based on the flux vector splitting scheme. This numerical treatment results in a very large sparse linear system. The direct solution of this full implicit linear system is very expensive and, in most cases, impractical. There are several numerical approaches which are commonly used by the scientific community to treat sparse linear systems, and the point-implicit integration was selected in the present case. However, numerical approaches to solve implicit linear systems can be non-conservative in time, even for formulations which are conservative by construction, as the finite volume techniques. Moreover, there are physical problems which strongly demand conservative schemes in order to achieve the correct numerical solution. The work presents results of numerical simulations to evaluate the conservation of implicit and explicit time-marching methods and discusses numerical requirements that can help avoiding such non-conservation issues.

KEYWORDS: Computational fluid dynamics, Time marching methods, Flux vector splitting scheme, Conservative discretization.

INTRODUCTION

This work discusses issues associated with the coupling of implicit integration methods, for unstructured finite volume formulations, with the spatial discretization based on flux vector splitting schemes. The material discussed in the work is an extension of the work of Barth (1987). The original paper discussed the approximate local time linearizations of nonlinear terms for the finite difference formulation using Total Variation Diminishing (TVD) (Harten, 1983) and upwind algorithms. Here, the time conservation of the point-implicit integration for unstructured meshes is analyzed and discussed. Finite volume formulations have the tremendously important property of being conservative by construction. However, some time integration approaches may present numerical issues that can destroy this property, at least for unsteady applications or during the process of converging to a steady state. This shortcoming was observed when performing simulations of the flow inside closed systems in which there is no addition or extraction of fluid mass to/from the system. For such systems, the use of the point-implicit schemes discussed in this present paper has led to heat generation and non-conservation of mass in the interior of the computational domain. Results of simulations using explicit and implicit integration methods are presented in the present paper in order to better understand the non conservation issue of the time-marching methods. An analysis of the problem is also performed by a detailed study of the backward Euler method linearization.

¹Instituto Tecnológico de Aeronáutica – São José dos Campos/SP – Brazil ²EMBRAER S.A. – São José dos Campos/SP – Brazil ³Instituto de Aeronáutica e Espaço – São José dos Campos/SP – Brazil

Author for correspondence: João Luiz F. Azevedo | Instituto de Aeronáutica e Espaço | Praça Eduardo Gomes, 50 – Vila das Acácias | CEP: 12228-901 São José dos Campos/SP – Brazil | Email: joaoluiz.azevedo@gmail.com

Received: 04/24/2014 | **Accepted:** 08/19/2014

THEORETICAL FORMULATION

The formulation used in the present work is based on the Reynolds-averaged Navier-Stokes set of equations, also known by the Computational Fluid Dynamics (CFD) community as the Reynolds Averaged Navier-Stokes (RANS) equations. The RANS equations are obtained by time filtering the Navier-Stokes set of equations. The compressible RANS equations are written in the algebraic vector form as

$$\frac{\partial Q}{\partial t} + \nabla \cdot (F_e - F_v) = 0. \quad (1)$$

The conserved variables vector, Q , the inviscid flux vector, F_e , and the viscous flux vector, F_v , are given by

$$Q = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ e \end{bmatrix}, \quad (2)$$

$$F_e = \begin{bmatrix} \rho \mathbf{v} \\ \rho u \mathbf{v} + p \hat{i}_x \\ \rho v \mathbf{v} + p \hat{i}_y \\ \rho w \mathbf{v} + p \hat{i}_z \\ (e + p) \mathbf{v} \end{bmatrix}, \quad (3)$$

$$F_v = \begin{bmatrix} 0 \\ (\tau_{xi}) \hat{i}_i \\ (\tau_{yi}) \hat{i}_i \\ (\tau_{zi}) \hat{i}_i \\ (\beta_i) \hat{i}_i \end{bmatrix}, \quad (4)$$

in which ρ stands for density, $\mathbf{v} = \{u, v, w\}$ for the velocity vector in Cartesian coordinates, p for static pressure, τ for viscous stress tensor, q_H for heat flux vector, e for the total energy per unit volume and β_i is given by

$$\beta_i = \tau_{ij} u_j - q_{H_i}. \quad (5)$$

The \hat{i}_x , \hat{i}_y and \hat{i}_z terms are the Cartesian-coordinate orthonormal vector basis. It is very important to emphasize that field forces, such as gravity, are neglected here.

Other equations are necessary in order to close the system of equations given by Eq. (4). These additional equations are called constitutive relations. The first constitutive equation

presented to close the Navier-Stokes set of equations is known as the equation of state. This equation considers the perfect gas law, and it is written as

$$p = (\gamma - 1) \left[e - \frac{1}{2} \rho (u^2 + v^2 + w^2) \right], \quad (6)$$

in which the mean total energy per unit volume, e , is given by

$$e = \rho \left[e_i + \frac{1}{2} (u^2 + v^2 + w^2) \right], \quad (7)$$

and e_i stands for internal energy, defined as

$$e_i = C_v T, \quad (8)$$

in which T stands for the mean static temperature and C_v is the specific heat at constant volume. The heat flux from Eq. (5) is obtained from the Fourier law for heat conduction, and it is given by

$$q_{H_j} = -\frac{\gamma \mu}{Pr} \frac{\partial (e_i)}{\partial x_j}, \quad (9)$$

in which γ is the ratio of specific heats and Pr is the Prandtl number. Typically, for air, it is assumed that $\gamma = 1.4$ and $Pr = 0.72$. C_p is the gas specific heat at constant pressure and μ is the dynamic molecular viscosity coefficient, calculated as a function of the temperature by the Sutherland law equation (Anderson, 1991), written as

$$\mu = \mu_\infty \left(\frac{T}{T_\infty} \right)^{\frac{2}{3}} \frac{T_\infty + S}{T + S}, \quad (10)$$

where $S = 110K$, and μ_∞ is the dynamic molecular viscosity coefficient of the fluid at temperature T_∞ .

The components of the viscous stress tensor, for a Newtonian fluid, are given by

$$\tau_{ij} = \mu \left[\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{2}{3} \frac{\partial u_m}{\partial x_m} \delta_{ij} \right], \quad (11)$$

in which δ_{ij} stands for the Kronecker delta.

NUMERICAL FORMULATION

The numerical formulation applied in this work is briefly presented in this section. The study is performed using the

RANS equations discretized in the context of a cell-centered finite-volume. The solver used in the present work is Le Michigan Aerothermodynamics Navier-Stokes Solver (LeMANS). The original framework was developed by Scalabrin (2007) at the University of Michigan to simulate hypersonic flows over reentry capsules. It is a three dimensional (3-D) numerical solver that uses the finite volume formulation for unstructured meshes to solve the RANS equations coupled with non-equilibrium chemical reaction equations and the Spalart-Allmaras (SA) turbulence model (Spalart and Allmaras, 1992; 1994). The numerical flux is calculated using an upwind scheme based on the Steger-Warming flux vector splitting (Steger and Warming, 1981). The time integration is performed by point implicit and Runge-Kutta methods. This section describes the numerical formulations and discretizations applied in the present work.

FINITE VOLUME FORMULATION

The finite volume formulation is a numerical method applied to represent and evaluate partial differential equations. It is applied by the CFD community to find the solution of the RANS, equations. The method is obtained integrating the flow equations for each control volume within a given mesh. The RANS equations, written in the context of a cell-centered finite-volume formulation, are given by

$$\int_{V_i} \frac{\partial Q}{\partial t} dV + \int_{V_i} \nabla \cdot (Fe - Fv) dV = 0. \quad (12)$$

Considering a cell-centered formulation, V_i is a given cell of the given grid. After the integration it is possible to apply the Gauss theorem over the equation above, resulting in:

$$\int_{V_i} \frac{\partial Q}{\partial t} dV + \int_{S_i} (Fe - Fv) \cdot n dS = 0, \quad (13)$$

in which S_i is the outward-oriented area vector and it is defined as

$$S_i = \{S_x, S_y, S_z\}. \quad (14)$$

Considering the mean value of the conserved variables within the i -th control volume, one can write the first term of Eq. (13) as

$$Q_i = \frac{1}{V_i} \int_{V_i} Q dV_i. \quad (15)$$

The second term of Eq. (13) can be written as the sum of all faces of a cell

$$\int_{S_i} (Fe - Fv) \cdot n dS = \sum_{k=1}^{nf} (\vec{F}e_k - \vec{F}v_k) \cdot \vec{n}_k S_k, \quad (16)$$

in which the k subscript is the index of the cell face, and nf indicates the number of faces of the i -th volume. Finally, the RANS equations discretized with a finite volume approximation are given by

$$\frac{\partial Q_i}{\partial t} = -\frac{1}{V_i} \sum_{k=1}^{nf} (\vec{F}e_k - \vec{F}v_k) \cdot \vec{n}_k S_k. \quad (17)$$

For this formulation, the fluxes are computed at the faces of the control volume, and the conserved variables are computed in the cell.

INVISCID FLUX CALCULATION

The inviscid fluxes are calculated using a method based on a classical flux vector splitting formulation, the Steger-Warming Scheme (SW) (Steger and Warming, 1981). This method is an upwind scheme which uses the homogeneous property of the inviscid flux vectors to write

$$\vec{F}e_k \cdot \vec{n}_k = Fe_n = \frac{\partial Fe_n}{\partial Q} Q = AQ, \quad (18)$$

where Fe_n is the normal flux at the k -th face, and A is the Jacobian matrix of the inviscid flux which can be diagonalized by the matrices of its eigen vectors from the left and from the right, L and R

$$A = R \Lambda L, \quad (19)$$

and Λ is the diagonal matrix of the eigenvalues of the Jacobian matrix. The A matrix can be split into positive and negative parts as

$$A^+ = R \Lambda^+ L \text{ and } A^- = R \Lambda^- L. \quad (20)$$

The splitting separates the flux into two parts, the downstream and the upstream flux, in relation to the face orientation as:

$$\vec{F}e \cdot \vec{n} = Fe_n^+ + Fe_n^- = (A_{cl}^+ Q_{cl} + A_{cr}^- Q_{cr}), \quad (21)$$

where the cl and cr subscripts are the cells on the left and right sides of the face. The split eigen values of the Jacobian matrix are given by

$$\lambda^\pm = \frac{1}{2} (\lambda \pm |\lambda|), \quad (22)$$

In order to avoid sudden sign transition, and then discontinuous derivative, the split eigen values receive a small number, ϵ , turning Eq. (22) into

$$\lambda^\pm = \frac{1}{2} \left(\lambda \pm \sqrt{\lambda^2 + \epsilon^2} \right). \quad (23)$$

The soften sign transition turns the derivative continuous at the transition point.

Numerical studies performed in the present work indicated that this flux vector splitting is too dissipative and it can deteriorate the boundary layer profiles (Junqueira-Junior, 2012; Junqueira-Junior *et al.*, 2013; MacCormack and Candler, 1989). Therefore, a pressure switch is implemented to smoothly shift the Steger-Warming scheme into a centered one. Then, the artificial dissipation is controlled and the numerical stability is maintained, as presented in the following formulation:

$$\overrightarrow{F}e_k \cdot \overrightarrow{n}_k = F_k^+ + F_k^- = (A_{k+}^+ Q_{k+} + A_{k-}^- Q_{k-}). \quad (24)$$

in which

$$\begin{aligned} Q_{k+} &= (1-w)Q_{cl} + wQ_{cr}, \quad \text{and,} \\ Q_{k-} &= (1-w)Q_{cr} + wQ_{cl}. \end{aligned} \quad (25)$$

The switch, w , is given by

$$w = \frac{1}{2} \frac{1}{(\alpha \nabla p)^2 + 1} \quad \text{and} \quad \nabla p = \frac{|p_{cl} - p_{cr}|}{\min(p_{cl}, p_{cr})}, \quad (26)$$

where ∇p is a scalar number, a numerical approximation of the pressure gradient. For small ∇p , $w = (1-w) = 0.5$, the code runs with a centered scheme, and for larger values of ∇p , $w = 0$ and $(1-w) = 1$, the code runs with the original Steger-Warming scheme. For Eq. (26) one suggests $\alpha = 6$, but some problems may require larger values (Scalabrin, 2007).

The applied formulation was originally created with interest on studying flows over reentry capsules. For such particular cases, with very strong shock waves, it is very common to find solutions with numerical and non-physical structures such as carbuncles (Ramalho *et al.*, 2011). To avoid such numerical problems, artificial dissipation has to be added to the method. The dissipation was included into the split eigenvalues, Eq. (23), using an ϵ factor, which is given by:

$$\epsilon_k = \begin{cases} 0.3(a_k + |\overrightarrow{u}_k|) & d_k > d_0, \\ 0.3(1 - |\overrightarrow{n}_k \cdot \overrightarrow{m}_k|)(a_k + |\overrightarrow{u}_k|) & d_k < d_0, \end{cases} \quad (27)$$

where d_k is the distance of the k -th face, to the nearest wall boundary. d_0 is set by the user and must be smaller than the boundary layer thickness and larger than the shock stand-off distance. \overrightarrow{m}_k is the normal vector of the nearest wall, and \overrightarrow{n}_k is the normal vector to the k -th face. Equation (27) applies the term $(1 - |\overrightarrow{n}_k \cdot \overrightarrow{m}_k|)$ to decrease the value at the faces parallel to the wall inside the boundary layer (Scalabrin, 2007). This artificial dissipation model has shown an important role in the prediction of boundary layer profiles (Junqueira-Junior, 2012; Junqueira-Junior *et al.*, 2013).

VISCOUS FLUX CALCULATION

The viscous terms are based on derivative of properties on the faces. To build the derivative terms, two volumes are created over the face where the derivative is being calculated. At the center of each new volume, the derivative is calculated using the Green-Gauss theorem. This computation is used to find the derivative at the desired face.

A two dimensional (2-D) example is used in this section to better explain the derivative calculation. Consider the two cells, S1 and S2, in Fig. 1. Two new cells, S3 and S4, are created using node points, P1 and P3, and cell centered points, P2 and P4, to calculate the derivative on 1-3 face. The properties at the faces are calculated using simple averages. For example, in Figs. 1 and 2, the properties are given by

$$\begin{aligned} Q_{12} &= \frac{1}{2} (Q_1 + Q_2) \\ Q_{23} &= \frac{1}{2} (Q_2 + Q_3) \\ Q_{13} &= Q_{31} = \frac{1}{2} (Q_1 + Q_3) \\ Q_{34} &= \frac{1}{2} (Q_3 + Q_4) \\ Q_{41} &= \frac{1}{2} (Q_4 + Q_1). \end{aligned} \quad (28)$$

$$\int_V \nabla Q \, dV = \oint_S \overrightarrow{Q} \cdot \overrightarrow{n} \, dS. \quad (29)$$

Considering ∇Q as a constant over the cell, the equation above yields

$$\nabla \overline{Q} = \frac{1}{V} \oint_S \overrightarrow{Q} \cdot \overrightarrow{n} \, dS, \quad (30)$$

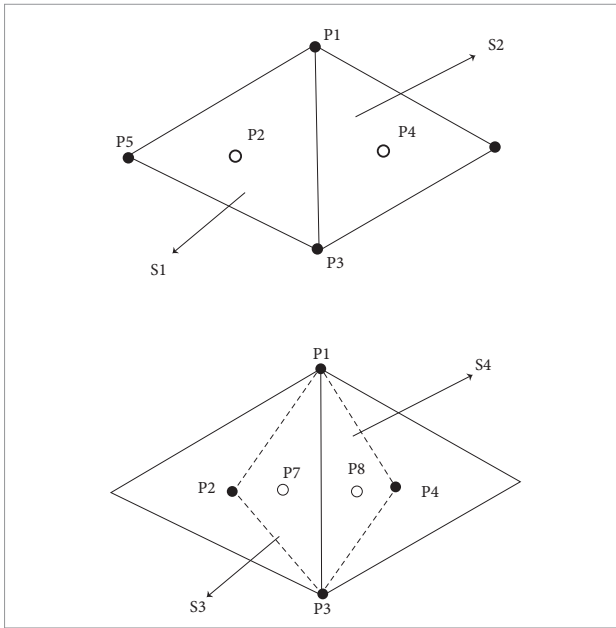


Figure 1. 2-D example of new volume creation.

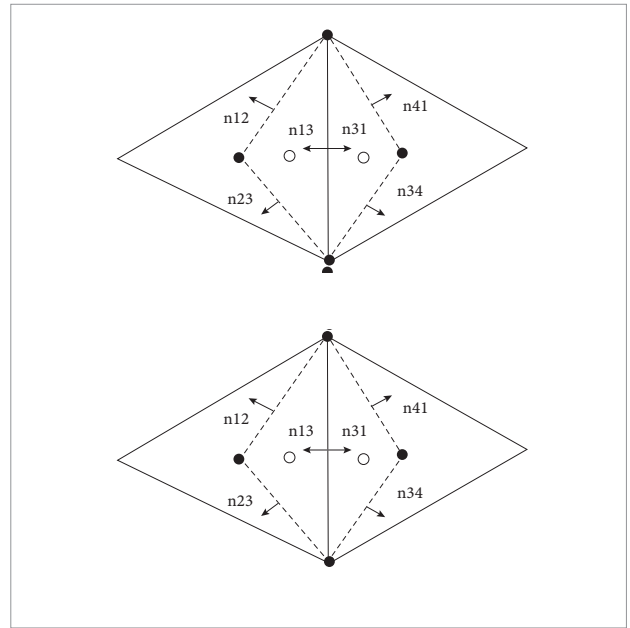


Figure 2. 2-D example of derivative calculation.

in which ∇Q is the constant cell-centered gradient. Using the derivatives in the S3 and S4 cells, the derivative at faces 1-3 is computed using

$$\nabla Q_{13} = \frac{V_3 \nabla Q_7 + V_4 \nabla Q_8}{V_3 + V_4} \quad (31)$$

The derivative computation for other types of element, 2-D or 3-D, is straightforward.

TIME INTEGRATION

The explicit second-order Runge-Kutta (Lomax *et al.*, 2001) and the point implicit scheme are the two time-marching methods applied in the present work.

The second-order Runge-Kutta integration method used in this work is given by

$$\begin{aligned} \Delta Q_{cl}^n &= -\frac{\Delta t}{V_{cl}} \left[\sum_{k=1}^{nf} (\vec{F}e_k - \vec{F}v_k) \cdot \vec{n}_k S_k \right]^n \\ Q_{cl}^{n+1} &= Q_{cl}^n + \Delta Q_{cl}^n \\ \Delta Q_{cl}^{n+1} &= -\frac{\Delta t}{V_{cl}} \left[\sum_{k=1}^{nf} (\vec{F}e_k - \vec{F}v_k) \cdot \vec{n}_k S_k \right]^{n+1} \\ Q_{cl}^{n+1} &= \frac{1}{2} (Q_{cl}^n + Q_{cl}^{n+1} + \Delta Q_{cl}^{n+1}). \end{aligned} \quad (32)$$

In order to simplify the forthcoming equations, the right hand side of Eq. (32) is written as

$$\begin{aligned} R_{cl}^n &= -\frac{\Delta t}{V_{cl}} \left[\sum_{k=1}^{nf} (\vec{F}e_k - \vec{F}v_k) \cdot \vec{n}_k S_k \right]^n \\ R_{cl}^{n+1} &= -\frac{\Delta t}{V_{cl}} \left[\sum_{k=1}^{nf} (\vec{F}e_k - \vec{F}v_k) \cdot \vec{n}_k S_k \right]^{n+1}, \end{aligned} \quad (33)$$

in which R_{cl} is the residue of the i -th cell.

The implicit integration applied in the present work is based on the backward Euler method, which is given by

$$\frac{\Delta Q_{cl}^n}{\Delta t} V_{cl} = \left[-\sum_{k=1}^{nf} (\vec{F}e_k - \vec{F}v_k) \cdot \vec{n}_k S_k \right]^{n+1} = R_{cl}^{n+1}. \quad (34)$$

One can linearize the residue at time $n + 1$ as a function of properties at time n .

$$\begin{aligned} \frac{\Delta Q_{cl}^n}{\Delta t} V_{cl} &= R_{cl}^n - \sum_{k=1}^{nf} \left\{ \left[\frac{\partial (\vec{F}e \cdot \vec{n})}{\partial Q} \right]_k^n \Delta Q_{cl}^n \right. \\ &\quad \left. - \left[\frac{\partial (\vec{F}v \cdot \vec{n})}{\partial Q} \right]_k^n \Delta Q_{cl}^n \right\} \cdot S_k. \end{aligned} \quad (35)$$

From the spatial discretization, the inviscid terms can be written as

$$\left[\frac{\partial (\vec{F}e \cdot \vec{n})}{\partial Q} \right]_k \Delta Q_{cl} = \left[\frac{\partial (\vec{F}e^+ \cdot \vec{n})}{\partial Q} \right]_k \Delta Q_{cl} + \left[\frac{\partial (\vec{F}e^- \cdot \vec{n})}{\partial Q} \right]_k \Delta Q_{cl}. \tag{36}$$

It is a common practice to assume

$$\frac{\partial (\vec{F}e^\pm \cdot \vec{n})}{\partial Q} = A^\pm, \tag{37}$$

and then write

$$\begin{aligned} \left[\frac{\partial (\vec{F}e^+ \cdot \vec{n})}{\partial Q} \right]_k \Delta Q_{cl} &= A_{k^+}^+ \Delta Q_{cl} \\ \left[\frac{\partial (\vec{F}e^- \cdot \vec{n})}{\partial Q} \right]_k \Delta Q_{cl} &= A_{k^-}^- \Delta Q_{cr,k}, \end{aligned} \tag{38}$$

which is not true and different from the real definition of the Jacobian matrix written in Eq. (20). Using the approximate form, Eq. (38) to calculate the inviscid terms may decrease the numerical stability of the method. Hence, in this work, the true Jacobian matrices of the split fluxes, given by Eq. (20), are implemented in order to calculate the implicit operator. The approximate Jacobian matrices, which are calculated using Eqs. (37) and (38), are implemented at the right hand side of the linear system. Issues involving the true Jacobians matrices have a major importance in the context of numerical stability for computational methods (Anderson *et al.*, 1986; Hirsch, 1990; Steger and Warming, 1981).

The viscous terms can be written in the same form as

$$\left[\frac{\partial (\vec{F}v \cdot \vec{n})}{\partial Q} \right]_k \Delta Q_{cl} = \left[\frac{\partial (\vec{F}v^- \cdot \vec{n})}{\partial Q} \right]_k \Delta Q_{cl} - \left[\frac{\partial (\vec{F}v^+ \cdot \vec{n})}{\partial Q} \right]_k \Delta Q_{cl}. \tag{39}$$

In this work, the viscous Jacobian matrices are represented by B . Hence, the Jacobian matrix splitting is written as

$$\left[\frac{\partial (\vec{F}v \cdot \vec{n})}{\partial Q} \right]_k \Delta Q_{cl} = B_{k^-}^- \Delta Q_{cr,k} - B_{k^+}^+ \Delta Q_{cl}. \tag{40}$$

One can write the system as

$$\begin{aligned} \left[\frac{V_{cl}}{\Delta t} + \sum_{k=1}^{nf} (A_{k^+}^+ + B_{k^+}^+) S_k \right] \Delta Q_{cl}^n + \\ \left[\sum_{k=1}^{nf} (A_{k^-}^- - B_{k^-}^-) S_k \Delta Q_{cr,k}^n \right] = R_{cl}^n. \end{aligned} \tag{41}$$

It is, then, possible to write

$$M_{cl} \Delta Q_{cl}^n + \sum_{k=1}^{nf} N_k^- \Delta Q_{cr,k}^n = R_{cl}^n, \tag{42}$$

with

$$M_{cl} = \frac{V_{cl}}{\Delta t} + \sum_{k=1}^{nf} N_k^+, \tag{43}$$

$$N_k^+ = (A_{k^+}^+ + B_{k^+}^+) S_k. \tag{44}$$

and

$$N_k^- = (A_{k^-}^- - B_{k^-}^-) S_k. \tag{45}$$

As the code is an unstructured solver, this system of equations results in a sparse block matrix, where each block is a square matrix of size equal to the number of equations to be solved in each control volume. The solution of such system is typically very expensive and, depending on the size of the mesh, it is not even practical. A less expensive implicit method is applied in the present work, the implicit point integration (Gnoffo, 2003; Venkatakrishnan, 1995; Wright, 1997).

The main idea of the implicit point integration is to move all the off-diagonal terms to the right hand side and to solve the resulting system iteratively, i.e.,

$$M_{cl} \Delta Q_{cl}^{n+1,p} = R_{cl}^n - \sum_{k=1}^{nf} N_k^- \Delta Q_{cr,k}^{n+1,p-1}. \tag{46}$$

It is assumed that $\Delta Q^{n+1,0} = 0$ and four iterations are taken in the process as suggested in the literature (Wright, 1997). The given sparse linear system illustrates the point

$$\begin{bmatrix} \square & & & & \\ & \square & & & \\ & & \square & & \\ & & & \square & \\ & & & & \square \end{bmatrix} \begin{bmatrix} \Delta Q^{(n+1)} \end{bmatrix} = \begin{bmatrix} R^n \end{bmatrix}$$

$$\begin{bmatrix} \square & & & & \\ & \square & & & \\ & & \square & & \\ & & & \square & \\ & & & & \square \end{bmatrix} \begin{bmatrix} \Delta Q^{(n+1),p} \end{bmatrix} =$$

$$\begin{bmatrix} R^n \end{bmatrix} - \begin{bmatrix} & & & \square & \\ & & & & \square \\ & & \square & & \\ & & & \square & \\ & & & & \square \end{bmatrix} \begin{bmatrix} \Delta Q^{(n+1),p-1} \end{bmatrix}$$

where each \square is a 5×5 block matrix.

The time step is computed by

$$\Delta t = CFL \frac{\ell}{\|\vec{v}\| + a}, \tag{47}$$

in which CFL is a parameter set to ensure the stability of the time integration method, ℓ is the size of the cell and $\|\vec{v}\| + a$ is the largest wave speed in the cell.

BOUNDARY CONDITIONS

The boundary conditions are implemented using ghost cells. The solver creates the ghost cells to hold properties that satisfy the correct flux calculation at the boundaries. The implementation assigns properties that satisfy the Euler boundary conditions to calculate the inviscid fluxes, and properties that satisfy the Navier-Stokes boundary conditions to calculate the viscous fluxes. Therefore, the ghost volumes store two different types of fluxes for the correct computation of the RANS equations.

WALL INVISCID BOUNDARY CONDITIONS

The ghost cells hold the properties in the same manner to calculate the inviscid fluxes at the wall and at the symmetry boundaries. Mass and energy fluxes should yield zero, and the momentum flux is equal to the pressure flux. This is accomplished by setting the normal velocity component to the boundary face zero. In the present work, the interior domain

is defined as the properties at the left side of the boundary face and the ghost cells are defined as the properties at the right side of the boundary.

In order to simplify the implementation, the properties at the left side of the boundary face are rotated to the face coordinates using

$$Q_{cl}^{rot} = \mathbb{R} Q_{cl}. \tag{48}$$

The vector of conservative properties, Q , is written as

$$Q = [\rho \quad \rho u \quad \rho v \quad \rho w \quad e]^T \tag{49}$$

In Eq. (48), \mathbb{R} is the rotation matrix given by,

$$\mathbb{R} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & n_x & n_y & n_z & 0 \\ 0 & \mathbf{t}_x & \mathbf{t}_y & \mathbf{t}_z & 0 \\ 0 & r_x & r_y & r_z & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{50}$$

and the \vec{n} , \vec{t} , \vec{r} vectors define the face-based reference frame. The properties at the ghost cells are set to

$$\begin{aligned} \rho_{cr} &= \rho_{cl} \\ \rho_{cr} u_{cr,n}^{rot} &= -\rho_{cl} u_{cl,n}^{rot} \\ \rho_{cr} u_{cr,t}^{rot} &= \rho_{cl} u_{cl,t}^{rot} \\ \rho_{cr} u_{cr,r}^{rot} &= \rho_{cl} u_{cl,r}^{rot} \\ (e_i)_{cr} &= (e_i)_{cl} \end{aligned} \tag{51}$$

One can write in the matrix form as

$$Q_{cr}^{rot} = \mathbf{W} Q_{cl}^{rot}, \tag{52}$$

in which \mathbf{W} is the inviscid wall matrix given by

$$\mathbf{W} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{53}$$

Therefore, the boundary condition can be written as

$$Q_{cr} = \mathbb{R}^{-1} \mathbf{W} \mathbb{R} Q_{cl}, \tag{54}$$

in which the R^{-1} matrix is given by

$$\mathbb{R}^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & n_x & \mathbf{t}_x & r_x & 0 \\ 0 & n_y & \mathbf{t}_y & r_y & 0 \\ 0 & n_z & \mathbf{t}_z & r_z & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \quad (55)$$

It returns the properties to the Cartesian coordinate frame.

VISCOUS WALL BOUNDARY CONDITION WITH SPECIFIED TEMPERATURE

Viscous wall with specified temperature does not have necessarily zero heat conduction at the boundary face. For this boundary condition the user provides the wall temperature, T_{wall} , and the wall pressure is extrapolated from the interior in order to satisfy the zero normal wall pressure gradient condition Schlichting (1978). Hence,

$$\begin{aligned} T_{wall} &= T_{cl} \\ p_{wall} &= p_{cl}. \end{aligned} \quad (56)$$

One can compute the conservative properties at the wall boundary using the Cartesian components of the wall velocity, u_{wall} , v_{wall} and w_{wall} , which are set by the user.

$$\begin{aligned} \rho_{wall} &= \Re \frac{\rho_{wall}}{T_{wall}} \\ (\rho u)_{wall} &= \rho_{wall} u_{wall} \\ (\rho v)_{wall} &= \rho_{wall} v_{wall} \\ (\rho w)_{wall} &= \rho_{wall} w_{wall} \\ e_{wall} &= \rho_{wall} [C_v T_{wall} \\ &\quad + \frac{1}{2} (u_{wall}^2 + v_{wall}^2 + w_{wall}^2)] \end{aligned} \quad (57)$$

Then, the ghost cell conservative properties are calculated using an average between the left and right cells

$$\begin{aligned} \rho_{cr} &= 2\rho_{wall} - \rho_{cl} \\ \rho u_{cr} &= 2\rho u_{wall} - \rho u_{cl} \\ \rho v_{cr} &= 2\rho v_{wall} - \rho v_{cl} \\ \rho w_{cr} &= 2\rho w_{wall} - \rho w_{cl} \\ e_{cr} &= 2e_{wall} - e_{cl}. \end{aligned} \quad (58)$$

IMPLICIT BOUNDARY CONDITIONS

Implicit boundary conditions are necessary in order to obtain a truly implicit time-marching method. The use of explicit boundary

conditions can limitate substantially the stability of the numerical method in the marching procedure for the solution convergence.

BASIC IMPLICIT FORMULATION

A simplified form of the implicit equation, Eq. (42), is written in this section in order to detail the implementation of implicit boundary conditions for flux vector splitting schemes,

$$\begin{aligned} \left[\frac{V_{cl}}{\Delta t} + (A_{k^+}^+ + B_{k^+}^+) S_k \right] \Delta Q_{cl} \\ + \left[(A_{k^-}^- - B_{k^-}^-) S_k \right] \Delta Q_{cr,k} = R_{cl}^n \end{aligned} \quad (59)$$

where the repeated k index, in the second term in left hand side of the equation, indicates summation over all the k faces of the control volume. The equation above is written only to present the relation between an internal cell, cl , and a boundary cell, cr, k . In the original formulation, Eq. (42), the cl -th cell has contributions from other faces, which may or may not be boundaries.

As discussed in the present work, the ghost cells hold different values for inviscid and viscous calculations. Hence, using the splitting definition, presented in section *Inviscid Fluxes Calculation*, Eq. (59) can be written as

$$\begin{aligned} \left[\frac{V_{cl}}{\Delta t} + (A_{k^+}^+ + B_{k^+}^+) S_k \right] \Delta Q_{cl} + \\ A_{k^-}^- S_k \Delta Q_{cr,k,inv} - B_{k^-}^- S_k \Delta Q_{cr,k,visc} = R_{cl}^n. \end{aligned} \quad (60)$$

The contributions of the boundary face can be expressed in terms of the internal cell corrections as

$$\begin{aligned} \Delta Q_{cr,k,inv} &= F_{k,inv} \Delta Q_{cl} \\ \Delta Q_{cr,k,visc} &= F_{k,visc} \Delta Q_{cl}. \end{aligned} \quad (61)$$

Hence, Eq. (60), can be rewritten as

$$\begin{aligned} \left[\frac{V_{cl}}{\Delta t} + (A_{k^+}^+ + F_{k,inv} A_{k^-}^- \right. \\ \left. - F_{k,visc} B_{k^-}^- + B_{k^+}^+) S_k \right] \Delta Q_{cl} = R_{cl}^n. \end{aligned} \quad (62)$$

The viscous Jacobians are calculated using primitive variables, then the corrections are set for the primitive variables and applied directly at the calculation of the viscous Jacobians. The matrix $B_{k^+}^+$ already includes the contribution from the boundary. The $A_{k^+}^+$, $A_{k^-}^-$, $B_{k^+}^+$ and $B_{k^-}^-$ are presented in the work of Scalabrin (2007).

INVISCID WALL MATRICES FOR IMPLICIT BOUNDARY CONDITIONS

The matrix for an inviscid wall or for a symmetry boundary is the same matrix presented in Eq. (54),

$$F_{k, inv, wall} = \mathbb{R}^{-1} \mathbf{W} \mathbb{R}. \tag{63}$$

The matrices are applied to ΔQ for the implicit boundary condition according to Eq. (61).

VISCOUS WALL WITH SPECIFIED TEMPERATURE MATRIX FOR IMPLICIT BOUNDARY CONDITIONS

The viscous Jacobians are created using primitive variables. Hence, the implementation of implicit viscous matrices is performed using primitive variables. They are applied directly at the calculation of the Jacobians matrices.

The code was originally created for flow simulation with chemical reactions, hence the mass fraction, Y , is present in the primitive variable vector, which is given by

$$\mathbf{V} = [Y \quad u \quad v \quad w \quad T]^T. \tag{64}$$

In the present work, it is always considered that there is only one species in the flow, then, $Y = 1$.

The implicit F matrix for wall boundary with specified temperature is derived from the average between the left and right cells,

$$z_{cr} = 2z_{wall} - z_{cl}, \tag{65}$$

in which z is given property. The velocity components and temperature at the wall are considered constants in time, hence,

$$\begin{aligned} \Delta u_{i, wall}^n &= 0 \\ \Delta T_{wall}^n &= 0 \end{aligned} \tag{66}$$

The mass fraction, Y , at the wall, is given by the left cell value, $Y_{wall} = Y_{cl}$. Therefore, the implicit matrix for wall boundary with specified temperature is given by

$$F_{k, visc} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 \end{bmatrix}. \tag{67}$$

CONSERVATION ANALYSIS OF IMPLICIT METHODS

GENERAL CONSERVATION ANALYSIS

After this brief review, the backward Euler time marching method, Eq. (42), is re-written using a simplified notation

$$[M] \Delta Q_{cl}^n + \frac{V_{cl}}{\Delta t} [I] \Delta Q_{cl}^n = R_{cl}^n. \tag{68}$$

The integration is applied on a generic grid, which is a representative 2-D finite volume mesh using nine quad-cells, as illustrated in Fig. 3, in order to present the conservative issues of such linearization. The $[M]$ matrix, for the mesh used here, is represented by

$$[M] = \begin{bmatrix} \sum A_1^+ & A_{1,2}^- & & A_{1,4}^- & & & & & & \\ A_{2,1}^- & \sum A_2^+ & A_{2,3}^- & & & & & & & \\ & A_{3,2}^- & \sum A_3^+ & & & & & & & \\ A_{4,1}^- & & & & \sum A_4^+ & & & & & \\ & A_{5,2}^- & & & A_{5,4}^- & & & & & \\ & & & A_{6,3}^- & & & & & & \\ & & & & & & A_{7,4}^- & & & \\ & & & & & & & & & \\ & A_{2,5}^- & & & & & & & & \\ & & A_{3,6}^- & & & & & & & \\ A_{4,5}^- & & & A_{4,7}^- & & & & & & \\ \sum A_5^+ & & A_{5,6}^- & & A_{5,8}^- & & & & & \\ A_{6,5}^- & & \sum A_6^+ & & & & A_{6,9}^- & & & \\ & & & \sum A_7^+ & & A_{7,8}^- & & & & \\ A_{8,5}^- & & & A_{8,7}^- & & \sum A_8^+ & & A_{8,9}^- & & \\ & & & & & & A_{9,8}^- & & \sum A_9^+ & \\ & & & & & & & & & A_{9,6}^- \end{bmatrix} \tag{69}$$

The formulation can be considered conservative when the sum of $V_i \Delta Q_i^n$ terms, for all cells, yields zero, i.e.,

$$\sum [V_i \Delta Q_i^n] = 0, \tag{70}$$

or yet

$$\sum [V_i Q_i^n] = \sum [V_i Q_i^{n+1}]. \tag{71}$$

Solving the linear system for the converged solution, which means that $R_i^n = 0$, one obtains

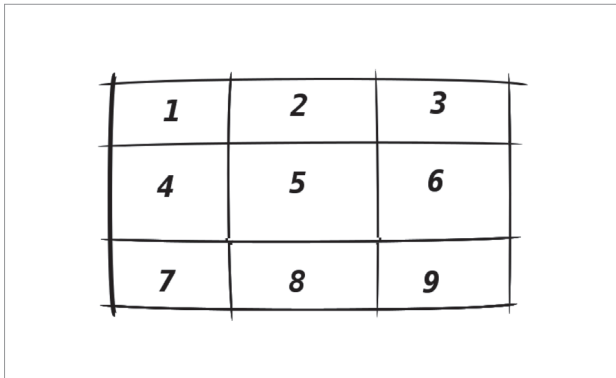


Figure 3. Representative 2-D finite volume mesh.

$$\begin{aligned}
 & \left[\frac{V_1}{\Delta t} + \sum A_1^+ + A_{2,1}^- + A_{4,1}^- \right] \Delta Q_1 + \\
 & \left[\frac{V_2}{\Delta t} + A_{1,2}^- + \sum A_2^+ + A_{3,2}^- + A_{5,2}^- \right] \Delta Q_2 + \\
 & \left[\frac{V_3}{\Delta t} + A_{2,3}^- + \sum A_3^+ + A_{6,3}^- \right] \Delta Q_3 + \\
 & \left[\frac{V_4}{\Delta t} + A_{1,4}^- + \sum A_4^+ + A_{5,4}^- + A_{7,4}^- \right] \Delta Q_4 + \\
 & \left[\frac{V_5}{\Delta t} + A_{2,5}^- + A_{4,5}^- + \sum A_5^+ + A_{6,5}^- + A_{8,5}^- \right] \Delta Q_5 + \\
 & \left[\frac{V_6}{\Delta t} + A_{3,6}^- + A_{5,6}^- + \sum A_6^+ + A_{9,6}^- \right] \Delta Q_6 + \\
 & \left[\frac{V_7}{\Delta t} + A_{4,7}^- + \sum A_7^+ + A_{8,7}^- \right] \Delta Q_7 + \\
 & \left[\frac{V_8}{\Delta t} + A_{5,8}^- + A_{7,8}^- + \sum A_8^+ + A_{9,8}^- \right] \Delta Q_8 + \\
 & \left[\frac{V_9}{\Delta t} + A_{6,9}^- + A_{8,9}^- + \sum A_9^+ \right] \Delta Q_9 = 0.
 \end{aligned} \tag{72}$$

It is possible to simplify this equation in order to write

$$\left[\frac{V_1}{\Delta t} + [C_1] \right] \Delta Q_1 + \left[\frac{V_2}{\Delta t} + [C_2] \right] \Delta Q_2 + \dots = 0, \tag{73}$$

where the $[C_i]$ matrix is the sum of the inviscid and viscous Jacobian matrices in each column of the $[M]$ matrix. Comparing Eqs. (71) and (73), one can easily state that, in order to preserve the conservative property of the finite volume formulation, the $[C_i]$ matrices should be zero and Δt_i should be the same for all the cells in the domain, i.e.,

$$\left[\frac{V_1}{\Delta t} \right] \Delta Q_1 + \left[\frac{V_2}{\Delta t} \right] \Delta Q_2 + \dots = \frac{1}{\Delta t} \sum [V_i \Delta Q_i^n] = 0. \tag{74}$$

POINT IMPLICIT CONSERVATION ANALYSIS

If one solves the linear system for the same mesh presented in Fig. 3 using the point-implicit integration, after a converged solution is obtained, i.e., $R_i^n = 0$, one could finally write

$$\begin{aligned}
 & \left\{ \left[\frac{V_1}{\Delta t} + \sum A_1^+ \right] \Delta Q_1^p + [A_{2,1}^- + A_{4,1}^-] \Delta Q_1^{p-1} \right\} + \\
 & \left\{ \left[\frac{V_2}{\Delta t} + \sum A_2^+ \right] \Delta Q_2^p + [A_{1,2}^- + A_{3,2}^- + A_{5,2}^-] \Delta Q_2^{p-1} \right\} + \\
 & \left\{ \left[\frac{V_3}{\Delta t} + \sum A_3^+ \right] \Delta Q_3^p + [A_{2,3}^- + A_{6,3}^-] \Delta Q_3^{p-1} \right\} + \\
 & \left\{ \left[\frac{V_4}{\Delta t} + \sum A_4^+ \right] \Delta Q_4^p + [A_{1,4}^- + A_{5,4}^- + A_{7,4}^-] \Delta Q_4^{p-1} \right\} + \\
 & \left\{ \left[\frac{V_5}{\Delta t} + \sum A_5^+ \right] \Delta Q_5^p + [A_{2,5}^- + A_{4,5}^- + A_{6,5}^- + A_{8,5}^-] \Delta Q_5^{p-1} \right\} + \\
 & \left\{ \left[\frac{V_6}{\Delta t} + \sum A_6^+ \right] \Delta Q_6^p + [A_{3,6}^- + A_{5,6}^- + A_{9,6}^-] \Delta Q_6^{p-1} \right\} + \\
 & \left\{ \left[\frac{V_7}{\Delta t} + \sum A_7^+ \right] \Delta Q_7^p + [A_{4,7}^- + A_{8,7}^-] \Delta Q_7^{p-1} \right\} + \\
 & \left\{ \left[\frac{V_8}{\Delta t} + \sum A_8^+ \right] \Delta Q_8^p + [A_{5,8}^- + A_{7,8}^- + A_{9,8}^-] \Delta Q_8^{p-1} \right\} + \\
 & \left\{ \left[\frac{V_9}{\Delta t} + \sum A_9^+ \right] \Delta Q_9^p + [A_{6,9}^- + A_{8,9}^-] \Delta Q_9^{p-1} \right\} = 0.
 \end{aligned} \tag{75}$$

One can state that, as long the sub-iterations of the point implicit method have not achieved the convergence for ΔQ_i^p and/or Δt is not the same for all the cells in the domain, it is not possible to find a general relation for any ΔQ_i^p and ΔQ_i^{p-1} such that $\sum [V_i \Delta Q_i^n] = 0$. Therefore, the conservation property for the point-implicit integration can be achieved only if the $[C_i]$ matrices are zero, the solution of ΔQ_i^p for the sub-iterations in p is converged, and Δt is constant over the entire mesh.

However, achieving convergence of the point-implicit sub-iterations can be as expensive as performing the fully implicit integration. Hence, all practical numerical solvers perform a limited number of sub-iterations. The authors, usually, perform up to 10 sub-iterations in p for the point-implicit integration and, then, move on to the next time step. Typically, this is not enough to achieve convergence for ΔQ_i^p , as discussed the forthcoming section of the paper.

RESULTS AND DISCUSSION

Results for the so-called “rigid body simulation” problem are presented in this section in order to expose the effects of the time-marching scheme on the mass conservation. The rigid body problem consists of the simulation of the flow contained between two concentric cylinders, in which both walls rotate at the same angular velocity. Therefore, after convergence, the fluid in the domain is rotating at the same angular velocity as if it were a rigid body. Here, the problem is addressed as a 2-D flow. The present work performed simulations using the point-implicit and the Runge-Kutta time marching methods.

An analysis of the use of a constant CFL number on all mesh cells is also presented here, in comparison to the use of a constant Δt throughout the mesh.

The two dimensional geometry and the detailed 360,000 cell mesh, used in the simulations, are presented in Fig. 4. The external and internal cylinders are rotating walls with fixed angular velocity, ω , and fixed temperature, T_0 . Air with zero velocity and T_0 temperature are considered as initial conditions. All simulations are conducted with the same initial and boundary conditions. Each simulation is performed using a different time marching method, as presented in Table 1.

Figures 5 and 6 present the total amount of mass (per unit of length in the cylinder axial direction) inside the computational domain, as a function of the iteration number, for the rigid body simulations. It is clear from the figures that the point implicit time marching method can significantly deteriorate the important conservative property of the finite volume formulation. Moreover, both simulations with the point-implicit time-marching scheme presented exactly the same non-conservative behavior. Moreover, for the point-implicit time integration test cases, i.e., test cases 1 and 2, or cases shown in Fig. 5 (a) and (b), there is almost no influence of the selection of constant CFL or of constant Δt in the time march. In other words, one could state that, for these test cases, the non-conservation effects of using a constant CFL number are far less significant than the effects of using the point implicit integration. Furthermore, it is correct to state that both simulations diverged after some time (not shown in Fig. 5 (a) and (b)).

In contrast to that behavior, simulations performed using the explicit Runge-Kutta scheme and a constant time step throughout the domain perfectly conserve the mass in the computational domain, as one would expect from a finite volume code. The results for this test case (case 4) are shown in Fig. 6 (b). On the other hand, results in Fig.6 (a) indicate that, even with an explicit scheme, there is no mass conservation if a variable time step, or constant CFL number, is used in the time integration. This is a serious problem since most convergence acceleration procedures typically employed in aerospace CFD codes are based on the use of implicit integration or on variable time stepping, or both. The present results are clearly demonstrating that, for such cases, there is no mass conservation during the transient process of converging to a steady state solution.

Moreover, all results presented in Figs. 5 and 6 reinforce the previous analysis performed in this work. To assure the conservative property of a time marching method, the $[C_i]$

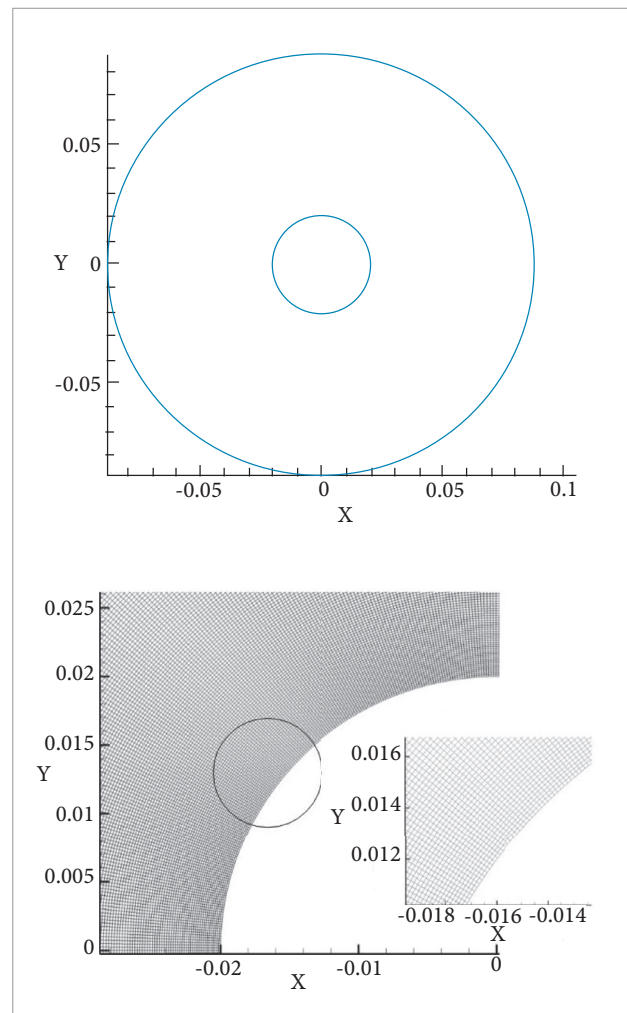


Figure 4. Rigid body geometry and mesh detail.

Table 1. Definition of the test case configurations for the numerical simulations.

Case	Time marching scheme	Constant Δt or CFL
1	Point implicit	Constant CFL
2	Point implicit	Constant Δt
3	Runge-Kutta	Constant CFL
4	Runge-Kutta	Constant Δt

matrix should be zero. This is automatically enforced by the explicit integration schemes by construction. Therefore, for the point implicit integration, the convergence of the sub-iterations has to be achieved in order to obtain a conservative scheme. Furthermore, all the mesh cells have to advance in time using

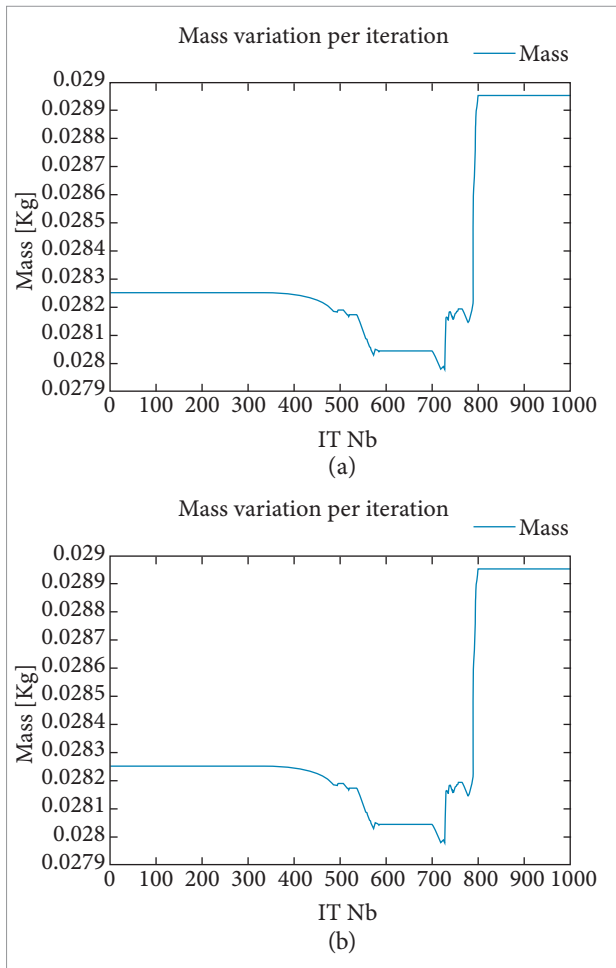


Figure 5. Effects of implicit time marching scheme on total mass conservation; (a) Constant CFL calculation for point-implicit scheme; (b) Constant Δt calculation for point-implicit scheme.

the same Δt value in order to maintain the conservative property of the finite volume method. This is true even for the explicit time marching methods.

CONCLUSIONS

A discussion on the issues associated with the coupling of implicit integration methods, for unstructured finite volume formulations, with the spatial discretization based on flux vector splitting schemes, is presented in this work. The linearization of inviscid and viscous Jacobians may result in a non-conservative method during the transient phase of the flow simulation, even for the finite volume formulation, which is supposed to be

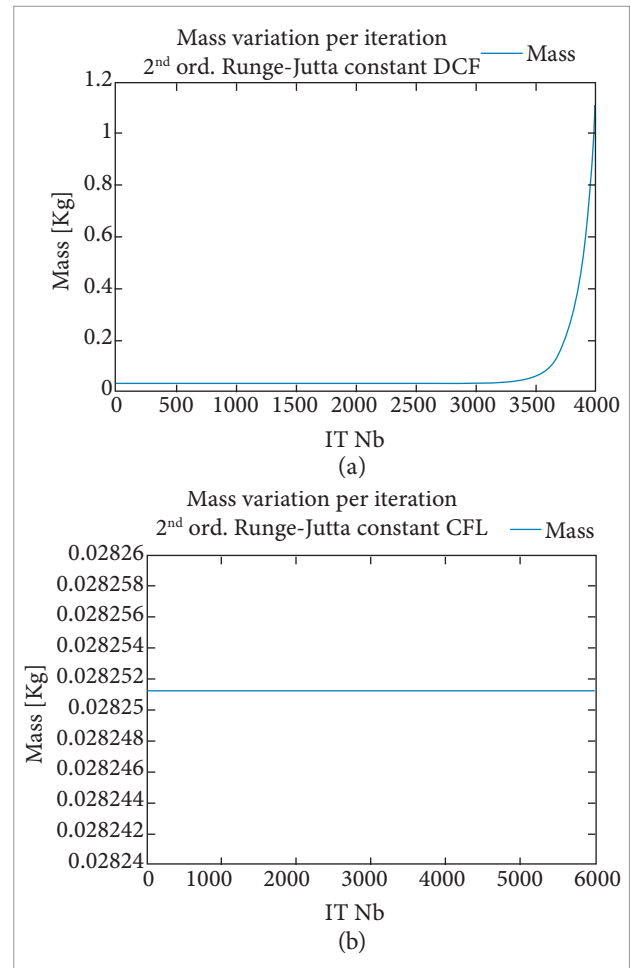


Figure 6. Effects of explicit time marching scheme on total mass conservation; (a) Constant CFL calculation for explicit RK scheme; (b) Constant Δt calculation for explicit RK scheme.

conservative by construction. The present analysis of the numerical formulation and of the computational results obtained has indicated that the time integration can be considered conservative only if the sum of the Jacobian matrices in each column of the linear system matrix is zero and all mesh cells have the same Δt value. Moreover, very popular approximate methods used in many CFD codes, to solve sparse linear systems, such as the point implicit integration, need to achieve convergence of the sub-iterations in order to be conservative during the transient portion of the simulation. This is a very expensive proposition and it can make such approximate solvers as expensive as those which perform the direct solution of the full implicit linear system.

The important conclusion is that care must be exercised in the linearization of time-marching methods for simulations which demand the conservative property. It is important to

be aware of the effects of such issue on the physical problem of interest. Physical problems with incoming and outgoing flow boundary conditions, very common in simulations for aerospace applications, do not necessarily need a conservative scheme during the transient portion of a steady state calculation. The amount of variation in the flow properties, during one time step, is negligible compared to the flux of the same properties that is crossing the open boundaries of the domain. Moreover, for many steady-state external aerodynamic applications, the initial conditions are strictly numerical, in the sense that they cannot be physically realized as implemented in the solver. In other words, they are typically associated to impulsively started flows. For such cases, the aspect of the lack of mass conservation in the entire computational domain, as the solution evolves from a non-physical initial condition to a physically relevant converged steady state condition, is not an issue. On the other hand, the conservative property is absolutely essential for closed systems, in which the total mass, and other properties, must always be

conserved in order to achieve a physically relevant solution. In these critical cases, approximate numerical methods should not be used in order to solve the full implicit linear systems.

ACKNOWLEDGEMENTS

The authors would like to acknowledge Conselho Nacional de Desenvolvimento Científico e Tecnológico, CNPq, which partially supported the project under the Research Grant No. 309985/2013-7. Partial support for the present research was also provided by Fundação de Amparo à Pesquisa do Estado de São Paulo, FAPESP, under Research Grants No. 2013/07375 and No. 2013/21535-0. Further partial support was provided by Fundação Coordenação de Aperfeiçoamento de Pessoal de Nível Superior, CAPES, through a Ph.D. scholarship to the first author, which is also gratefully acknowledged.

REFERENCES

- Anderson, J.D.J., 1991, "Fundamentals of Aerodynamics", McGraw-Hill International Editions, New York, NY, USA.
- Anderson, W.K., Thomas, J.L. and Van Leer, B., 1986, "A Comparison of Finite Volume Flux Vector Splittings for the Euler Equations", *AIAA Journal*, Vol. 24, No. 6, pp. 1453-1460.
- Barth, T.J., 1987, "Analysis of Implicit Local Linearization Techniques for Upwind and TVD Algorithms", Proceedings of the 25th AIAA Aerospace Sciences Meeting, AIAA Paper No. 87-0595, Reno, NV, USA.
- Gnoffo, P.A., 2003, "Computational Aerothermodynamics in Aeroassist Applications", *Journal of Spacecraft and Rockets*, Vol. 40, No. 3, pp. 305-312. doi: 10.2514/2.3957.
- Harten, A., 1983, "High Resolution Schemes for Hyperbolic Conservation Laws", *Journal of Computational Physics*, Vol. 49, No. 3, pp. 357-393. doi: 10.1016/0021-9991(83)90136-5.
- Hirsch, C., 1990, "Numerical Computation of Internal and External Flows", Vol. 2, Computational Methods for Inviscid and Viscous Flows, John Wiley and Sons, New York, USA.
- Junqueira-Junior, C.A., 2012, "A Study on the Extension of an Upwind Parallel Solver for Turbulent Flow Applications" Masters Thesis, Instituto Tecnológico de Aeronáutica, São José dos Campos, SP, Brasil.
- Junqueira-Junior, C.A., Azevedo, J.L.F., Scalabrin, L.C. and Basso, E., 2013, "A Study of Physical and Numerical Effects of Dissipation on Turbulent Flows Simulations", *Journal of Aerospace Technology and Management*, Vol. 5, No. 2, pp. 145-168. doi: 10.5028/jatm.v5i2.179.
- Lomax, H., Pulliam, T.H. and Zingg, D.W., 2001, "Fundamentals of Computation Fluid Dynamics", Springer, NY, USA.
- MacCormack, R.W. and Candler, G.V., 1989, "The Solution of the Navier-Stokes Equations Using Gauss-Seidel Line Relaxation", *Computer and Fluids*, Vol. 17, No. 1, pp. 135-150. doi: 10.1016/0045-7930(89)90012-1.
- Ramalho, M.V.C., Azevedo, J.L.F. and Azevedo, J.H., 2011, "Further Investigation into the Origin of the Carbuncle Phenomenon in Aerodynamic Simulations", 49th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition, AIAA Paper No. 2011-1184, Orlando, FL, USA.
- Scalabrin, L.C., 2007, "Numerical Simulation of Weakly Ionized Hypersonic Flow Over Reentry Capsules", PhD Thesis, Department of Aerospace Engineering, The University of Michigan, Michigan, USA.
- Schlichting, H., 1978, "Boundary-Layer Theory", 7th Edition, McGraw Hill, New York, USA.
- Spalart, P.R. and Allmaras, S.R., 1992, "A One-Equation Turbulence Model for Aerodynamic Flows", Proceedings of the 30th AIAA Aerospace Sciences Meeting and Exhibit, AIAA Paper No. 92-0439, Reno, NV, USA.
- Spalart, P.R. and Allmaras, S.R., 1994, "A One-Equation Turbulence Model for Aerodynamic Flows", *La Recherche Aérospatiale*, Vol. 1, pp. 5-21.
- Steger, J.L. and Warming, R.F., 1981, "Flux Vector Splitting of the Inviscid Gasdynamic Equations with Application to the Finite-Difference Method", *Journal of Computational Physics*, Vol. 40, No. 2, pp. 263-293. doi: 10.1016/0021-9991(81)90210-2.
- Venkatakrisnan, V., 1995, "Implicit Schemes and Parallel Computing in Unstructured Grid CFD", Technical Report, ICASE Report No. 95-28, ICASE.
- Wright, M.J., 1997, "A Family of Data Parallel Relaxation Methods for the Navier-Stokes Equations", PhD Thesis, University of Minnesota, Minneapolis, MN, USA.

ABBREVIATIONS

2-D: Two dimensional
 3-D: Three dimensional
 CFD: Computational Fluid Dynamics
 CFL: Courant-Friedrichs-Lewy number
 LeMANS: Le Michigan Aerothermodynamics Navier- Stokes Solver
 RANS: Reynolds Averaged Navier-Stokes
 SA: Spalart-Allmaras Turbulence Model
 SW: Steger-Warming Scheme
 TVD: Total Variation Diminishing

LIST OF SYMBOLS**English Characters**

a : Speed of the sound
 A : Jacobian matrix of the inviscid flux
 B : Jacobian matrix of the viscous flux
 C : Sum of Jacobian matrices
 C_p : Specific heat at constant pressure
 C_v : Specific heat at constant volume
 d_k : Distance of k -th face to the nearest wall
 e : Total energy per unit volume
 e : Internal energy
 F_e : Inviscid flux vector
 F_v : Viscous flux vector
 $\hat{i}_x, \hat{i}_y, \hat{i}_z$: Cartesian-coordinate orthonormal vector basis
 I : Identity matrix
 L : Matrix of eigenvectors from the left
 m : Normal vector to the nearest wall
 n : Normal vector
 n_f : Number of faces of a given volume
 p : Static pressure
 Pr : Prandtl number
 qH : Heat transfert vector
 Q : Conserved variable vector
 R : Matrix of eigenvectors from the right
 R : Residue
 \mathbb{R} : Rotation matrix
 \mathcal{R} : Gas constant
 S : Outward-oriented area vector
 S : Constant for the Sutherland law equation
 t : Time

T : Temperature
 $v = \{u, v, w\}$: Velocity vector in the cartesian coordinate
 V : Volume
 \mathbf{V} : Primitive variables vector
 W : Inviscid wall matrix
 w : Switch of the Steger and Warming scheme
 Y : Mass fraction

Greek Characters

α : Switch factor
 β : Viscous force work and heat transfer term
 δ_{ij} : Kronecker delta
 F : Implicit matrix
 ϵ : Artificial dissipation
 γ : Ratio of specific heats
 κ : Thermal conductivity coefficient
 λ : Eigenvalue of the Jacobian matrix
 Λ : Diagonal matrix of the eigenvalues of the Jacobian
 μ : Dynamic molecular viscosity coefficient
 ν : Kinematic molecular viscosity coefficient
 ρ : Density
 τ_{ij} : Shear-stress tensor

Subscripts

cl : Cell on the left side of the face
 cr : Cell on the right side of the face
 ∞ : Free-stream property
 inv : Inviscid property
 k : Index of the cell face
 n : Normal property at a given face
 $wall$: Property at the wall
 $visc$: Viscous property

Superscripts

n : Index of iteration in time
 p : Index of iteration for the point-implicit time integration
 $+$: Positive part of a matrix or vector
 $-$: Negative part of a matrix or vector
 rot : Rotated property