AN IMPLICIT MONOLITHIC FINITE ELEMENT METHOD TO SOLVE 3D INCOMPRESSIBLE NAVIER-STOKES EQUATIONS USING FRACTIONAL STEP METHOD AND PARALLEL EDGE-BASED IMPLEMENTATION

Alessandro R. E. Antunes, areantunes@yahoo.com.br
Paulo R. M. Lyra, prmlyra@ufpe.br
Rogério S. Da Silva, rogsoares@yahoo.com.br
Ramiro B. Willmersdorf, ramiro@willmersdorf.net
Federal University of Pernambuco
Department of Mechanical Engineering
Av. Acadêmico Hélio Ramos S/N, Recife-PE, Brazil

Abstract. *The aim of present work is to describe a mathematical formulation using a stabilized Finite Element Method and Fractional Step Method for the simulation of three-dimensional laminar incompressible Navier-Stokes flow problems written in primitive variables. A second order stable monolithic formulation is obtained. An edge-based data structure was adopted and the computational system developed was implemented using FORTRAN 90 and parallel computing concepts, allowing the computation of large scale 3D CFD problems. Several applications were analyzed in order to verify and validate the computational system developed. The obtained results are in good agreement with the experimental, theoretical and numerical data available in the literature.*

Keywords*: Parallel Computing, Finite Element Method, Fractional Step, Navier-Stokes Equations, Incompressible Flow*

1. INTRODUCTION

Several different approaches to deal numerically with problems involving incompressible flows have been developed over the last decades. Among those are monolithic procedures, artificial incompressibility, preconditioned Navier-Stokes equations and Fractional Step Methods. Each one has advantages and disadvantages (Gresho and Sani, 1998, vol 1 & 2).

In this work we adopted an implicit, monolithic formulation based on algebraic fractioning of the Navier-Stokes equations, which is developed in order to get pressure stability (Antunes, 2008, Codina, 2002, Soto *et al.*, 2004), preserving spatial second order approximation and a segregated solution scheme. This formulation was implemented using the finite element method written using an edge-based data structure, which is attractive from the computational point of view as the necessary matrices required during the resolution procedure are built during a pre-processing stage and kept in memory a priori to the processing step. The vectors for the final algebraic system of equations, i.e. LHS (Left Hand Side) and RHS (Right Hand Side), are computed through loops over the mesh edges, accumulating each edge contribution. The final system of equations is solved using PETSc (Portable, Extensible Toolkit for Scientific Computation). The parallel implementation adopted the domain decomposition paradigm and the domain partitions were obtained with ParMetis (Parallel Graph Partitioning and Fill-reducing Matrix Ordering) (Karypis *et al.*, 2003) and MPI (Pacheco, 1997) to manage communications among processes.

The rest of this paper is organized such that the second section describes the mathematical-numerical formulation, at section 3 some issues related to the computational implementation and the main tools or packages used are briefly presented, section 4 shows the solution of a model problem used to verify the general performance of the developed computational system and, finally, at section 5 some conclusions are drawn.

2. MATHEMATICAL AND NUMERICAL FORMULATION

The incompressible Navier-Stokes Equations, without thermal effects, in continuous form can be written like

$$\frac{\partial \mathbf{u}}{\partial t}+\left(\mathbf{u}\cdot\nabla\right)\mathbf{u}-\upsilon\Delta\mathbf{u}+\nabla p=\mathbf{f} \qquad \text{in } \Omega\times\left(0,t\right) \tag{1}$$

$$\nabla\cdot\mathbf{u}=0 \qquad \text{in } \Omega\times\left(0,t\right) \tag{2}$$

where Ω is the spatial fluid domain, t is the time variable, *(0,t)* is the time interval, $\mathbf{u}$ is the velocity field, υ is the kinematic viscosity, p is the pressure, $\mathbf{f}$ is the external force vector, ∇ is the gradient operator and Δ is the Laplacian operator. The physical boundary is divided in two non overlapping parts Γ_{du} and Γ_{nu} in which the Dirichlet and

Neumann boundary conditions are prescribed for each equation, respectively. The Dirichlet and Neumann boundary conditions as

$$\mathbf{u} = \overline{\mathbf{u}} \quad \text{in } \Gamma_{du}, \qquad p = \overline{p} \qquad \text{and} \qquad \mathbf{n}\cdot\boldsymbol{\sigma} = \overline{\mathbf{t}} \qquad \text{in } \Gamma_{nu} \tag{3}$$

where $\boldsymbol{\sigma}$ is the viscous stress tensor, $\mathbf{n}$ is the unit outward normal vector, and $\overline{t}$ is the surface stress or traction. An upper bar refers to a prescribed value. Finally, initial conditions must be known in the whole domain at the initial time.

2.1 Fractional Step Method

A popular approach to deal with transient incompressible Navier-Stokes equations considers the so called Fractional Step Method (Donea and Huerta, 2003), that advances the solution on time as a sequence of two or more sub-steps. These methods have been developed independently by Chorin (1967) and Teman (1969). Decomposition is adopted to circumvent the numerical difficulties associated with variational formulation due to the incompressibility. Basically, the complex numerical operator that arises from the original problem is replaced by a set of simpler sub-operators that are easier to be solved. There are several ways to fraction the system of equations and so there are several possible fractional step methods. In this context, generally, the time discretization is performed first than the spatial discretization, and so there is some controversy referring to the required boundary conditions at each time step to keep the problem mathematically well-posed, as some additional variables result from the fractioned system. Another characteristic of this method is the temporal accuracy, which can be of first or higher order depending on the adopted fractional strategy.

The fractional step method adopted here (Antunes, 2008, Codina, 2002), considers an algebraic fractional partitioning of the system of equations obtained through an LU factorization of the resultant matrix system. In what follows we present the central idea of this method, which is similar to that presented by Chorin (1967) and Teman (1969), that uses the Helmholtz decomposition of a vector into its solenoidal and gradient portions. An advantage of this strategy is the fact that the resultant variables do not require boundary conditions, as they do not have physical meaning (Codina, 2002).

By discretizing implicitly in time the convective, diffusion and pressure terms of Eq. (1), using the "Euler Backward" and the incompressibility condition (Eq. 2), we get:

$$\frac{\mathbf{u}^{n+1}}{\Delta t} + \mathbf{u}^{n+1}\cdot\nabla\mathbf{u}^{n+1} - \upsilon\nabla^2\mathbf{u}^{n+1} + \nabla p^{n+1} = \frac{\mathbf{u}^{n}}{\Delta t} \tag{4}$$

$$\nabla\cdot\mathbf{u}^{n+1} = 0 \tag{5}$$

Adding to Eq. (4) the term $\left(\nabla p^n - \nabla p^n\right)$, which is used to get the pressure increment, and re-writing Eqs. (4) and (5) using a matrix notation gives

$$\begin{bmatrix} \mathbf{A} & \mathbf{G} \\ \mathbf{D} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{u}^{n+1} \\ \mathbf{q} \end{bmatrix} = \begin{bmatrix} \mathbf{r}^{n} \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{bc}_1 \\ \mathbf{bc}_2 \end{bmatrix} \tag{6}$$

where: $\mathbf{A}$ is the implicit operator for mass, advection and diffusion terms, $\mathbf{G}$ is the gradient operator, $\mathbf{D}$ is the divergent operator, $\mathbf{r}^n$ is the load forces modified by pressure contribution, $\mathbf{q}$ is the pressure variables, with $q = p^{n+1} - p^{n}$, $\mathbf{bc}_1$ is the boundary conditions for momentum equations, and $\mathbf{bc}_2$ is the boundary conditions for incompressibility.

The matrix $\mathbf{A}$ structure depends on the time integration scheme, the mesh and the finite element adopted. The most common formulation kept the convective term explicitly and the diffusion term implicitly. Doing so Eqs. (4) and (5) would have to be replaced by

$$\frac{\mathbf{u}^{n+1}}{\Delta t} - \upsilon\nabla^2\mathbf{u}^{n+1} + \nabla p^{n+1} = \frac{\mathbf{u}^{n}}{\Delta t} - \mathbf{u}^{n}\cdot\nabla\mathbf{u}^{n} \tag{7}$$

$$\nabla\cdot\mathbf{u}^{n+1} = 0 \tag{8}$$

If we adopted an explicit advection, Eq. (7), then

$$\mathbf{A} = \frac{\mathbf{I}}{\Delta t} - \upsilon\mathbf{L} \tag{9}$$

On the other hand, if we consider an implicit advection discretization, Eq. (4), we get

$$\mathbf{A} = \frac{\mathbf{I}}{\Delta t} - \upsilon\mathbf{L} + \mathbf{N} \tag{10}$$

where $\mathbf{L} \rightarrow$ Laplacian operator, and $\mathbf{N} \rightarrow$ linearized advection operator.

Using an implicit convection discretization the obtained matrix is no longer symmetrical and block diagonal, due to the presence of the $\mathbf{N}$ operator. However, even using implicit convection, the $\mathbf{N}$ term can be approximately defined such that $\mathbf{A}$ is still block diagonal, with the consequence of getting a reduction in the temporal accuracy to first order.

We must observe that the difficulty in solving the system of Eq. (6) arises from the existence of a null sub-matrix in the global matrix (Antunes, 2008). To get a proper rank for the global matrix, the order of interpolation adopted for velocity and pressure must be chosen respecting some stability conditions (Zienkiewicz and Taylor, 1988). Alternatively, the original system can be decomposed into smaller sub-systems, which is the central idea of the projection methods.

Considering the system given by Eq. (6), applying the LU factorization leads to

$$\begin{bmatrix} \mathbf{A} & \mathbf{G} \\ \mathbf{D} & \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{D} & -\mathbf{DA}^{-1}\mathbf{G} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{A}^{-1}\mathbf{G} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \tag{11}$$

where

$$\begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{D} & -\mathbf{DA}^{-1}\mathbf{G} \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ \mathbf{Iq} \end{bmatrix} = \begin{bmatrix} \mathbf{r}^{n} \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{bc}_1 \\ \mathbf{bc}_2 \end{bmatrix} \tag{12}$$

Replacing $\mathbf{A}^{-1}$ by $\mathbf{H}_1$ and by $\mathbf{H}_2$ at the inferior triangular and superior triangular matrix given in Eq. (11), respectively, we have

$$\begin{bmatrix} \mathbf{A} & \mathbf{G} \\ \mathbf{D} & \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{D} & -\mathbf{DH}_1\mathbf{G} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{H}_2\mathbf{G} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{AH}_2\mathbf{G} \\ \mathbf{D} & \mathbf{D}(\mathbf{H}_2 - \mathbf{H}_1)\mathbf{G} \end{bmatrix} \tag{13}$$

and different choices of approximations for $\mathbf{A}^{-1}$, or $\mathbf{H}_1$ and $\mathbf{H}_2$, leads to different types of projection methods, such as:

$\mathbf{H}_1 = \mathbf{H}_2 = \mathbf{A}^{-1}$ - Original system (14)

$\mathbf{H}_1 = \mathbf{H}_2 \neq \mathbf{A}^{-1}$ - Just mass is conserved (15)

$\mathbf{H}_1 \neq \mathbf{H}_2; \mathbf{H}_2 = \mathbf{A}^{-1}$ - Just momentum is conserved (16)

$\mathbf{H}_1 \neq \mathbf{H}_2 \neq \mathbf{A}^{-1}$ - No physical variable is conserved (17)

In this work we took the option of conserve mass (Eq. 15) despite the perturbation in the momentum equation, and the same inverse of matrix $\mathbf{A}$ was employed for all equations of the global system.

Taking $\mathbf{A}^{-1} = \mathbf{H}_1 = \mathbf{H}_2 = \Delta t\mathbf{I}$ we get the "Classic Fractional Step Method", and no computation of an inverse matrix is required, and the system can then be rewritten as

$$\begin{aligned} &\mathbf{Aw} = \mathbf{r}^{n} + \mathbf{bc}_1 \Rightarrow \mathbf{w} = \mathbf{H}_1\mathbf{r}^{n} - \mathbf{H}_1\mathbf{bc}_1 \\ &\mathbf{DH}_1\mathbf{Gq} = \mathbf{Dw} - \mathbf{bc}_2 \\ &\mathbf{u}^{n+1} = \mathbf{w} - \mathbf{H}_2\mathbf{Gq} \end{aligned} \tag{18}$$

In this formulation we assume

$$\tilde{\mathbf{u}} = \mathbf{w} \tag{19}$$

$$\mathbf{r}^{n} = \mathbf{f}^{n} \tag{20}$$

$$\mathbf{A}\tilde{\mathbf{u}} = \mathbf{r}^{n} + \mathbf{bc}_{1} \tag{21}$$

$$\mathbf{DH}_{1}\mathbf{Gq} = \mathbf{D}\tilde{\mathbf{u}} - \mathbf{bc}_{2} \tag{22}$$

$$\tilde{\mathbf{u}}^{n+1} = \tilde{\mathbf{u}} - \mathbf{H}_{1}\mathbf{Gq} \tag{23}$$

By eliminating the boundary conditions, i.e. setting $\mathbf{bc}_1 = \mathbf{bc}_2 = \mathbf{0}$, the system given by Eq. (18), using the definitions given at Eqs. (19-23), can be expressed, analytically, as

$$\frac{\partial \tilde{\mathbf{u}}}{\partial t} + \tilde{\mathbf{u}}^{n+\theta} \cdot \nabla \tilde{\mathbf{u}}^{n+\theta} + \nabla p^{n+1} - \upsilon \nabla^{2} \tilde{\mathbf{u}}^{n+\theta} = \mathbf{f}^{n+\theta} \tag{24}$$

$$\Delta t \nabla^{2} \left(p^{n+1} - p^{n} \right) = \nabla \cdot \tilde{\mathbf{u}}^{n+1} \tag{25}$$

$$\frac{\mathbf{u}^{n+1} - \tilde{\mathbf{u}}^{n+1}}{\Delta t} + \nabla \left(p^{n+1} - p^{n} \right) = 0 \tag{26}$$

for a temporal discretization using two-step trapezoidal method, in which θ controls the approximation adopted ($\theta = 1$, leads to first order "Euler Backward", and $\theta = 1/2$, gives second order "Crank-Nicholson" approach).

Taking

$$\mathbf{A}^{-1} = \Delta t \mathbf{I} \qquad \text{and} \qquad \mathbf{A} = \frac{\mathbf{I}}{\Delta t} - \upsilon \mathbf{L} + \mathbf{N} \tag{27}$$

implies that

$$\mathbf{DA}^{-1}\mathbf{Gq} = \mathbf{A}^{-1}\mathbf{DGq} = \mathbf{A}^{-1}\nabla^{2}\mathbf{q} \tag{28}$$

and

$$\mathbf{q} = p^{n+1} - p^{n} \tag{29}$$

$$\mathbf{w} = \tilde{\mathbf{u}} \tag{30}$$

$$\mathbf{Dw} = \nabla \cdot \tilde{\mathbf{u}}^{n+1} \tag{31}$$

$$\mathbf{u}^{n+1} - \mathbf{w} = -\mathbf{A}^{-1}\mathbf{Gq} \Rightarrow \mathbf{u}^{n+1} - \tilde{\mathbf{u}}^{n+1} = -\Delta t \mathbf{I} \nabla \mathbf{q} \Rightarrow \frac{\mathbf{u}^{n+1} - \tilde{\mathbf{u}}^{n+1}}{\Delta t} + \nabla \left(p^{n+1} - p^{n} \right) = 0 \tag{32}$$

In this way we get a Fractional Step formulation, in which the super indices n and θ refer to the time level and the adopted temporal method adopted, respectively, and $\tilde{\mathbf{u}}$ is the intermediate velocity, introduced to allow the momentum equations to be fractioned.

2.2 Variational Formulation

The final discrete variational formulation, including stabilization of the equations [Antunes, 2008; Soto *et al.*, 2001], can be written as: Given $\mathbf{u}_h^{n}$, find $\left(\mathbf{u}_h^{n+1}, p_h^{n+1}, \boldsymbol{\pi}_h^{n+1}, \boldsymbol{\xi}_h^{n+1} \right)$ in $\mathbf{V}_h \times Q_h \times \tilde{\mathbf{V}}_h \times \tilde{\mathbf{V}}_h$ such that

$$\begin{aligned} &\frac{1}{\delta t}\left(\mathbf{u}_h^{n+1,i} - \mathbf{u}_h^{n}, \mathbf{v}_h \right) + \left(\mathbf{u}_h^{n+1,i-1} \cdot \nabla \mathbf{u}_h^{n+\theta,i}, \mathbf{v}_h \right) + \left(\upsilon \nabla \mathbf{u}_h^{n+\theta,i}, \nabla \mathbf{v}_h \right) + \left(\nabla p_h^{n+1,i-1}, \mathbf{v}_h \right) \\ &+ \left(\tau \left(\mathbf{u}_h^{n+\theta,i-1} \cdot \nabla \mathbf{u}_h^{n+\theta,i} - \beta^{n+1,i-1} \boldsymbol{\pi}_h^{n+\theta,i-1} \right), \mathbf{u}_h^{n+\theta,i-1} \cdot \nabla \mathbf{v}_h \right) = \left(\mathbf{f}^{n+\theta}, \mathbf{v}_h \right) + \left(\boldsymbol{\sigma}^{n+\theta,i-1} \cdot \mathbf{n}, \mathbf{v}_h \right)_{\Gamma_{nu}} \end{aligned} \tag{33}$$

$$\delta t \left(\nabla p_h^{n+1,i} - \nabla p_h^{n+1,i-1}, \nabla q_h \right) + \left(\tau \left(\nabla p_h^{n+1,i} - \beta^{n+1,i-1} \boldsymbol{\xi}_h^{n+1,i-1} \right), \nabla q_h \right) = \text{-} \left(\nabla \cdot \mathbf{u}_h^{n+1,i}, q_h \right) \tag{34}$$

$$\left(\boldsymbol{\pi}_h^{n+\theta,i}, \tilde{\mathbf{v}}_h\right) = \left(\mathbf{u}_h^{n+\theta,i} \cdot \nabla \mathbf{u}_h^{n+\theta,i}, \tilde{\mathbf{v}}_h\right) \tag{35}$$

$$\left(\boldsymbol{\xi}_h^{n+1,i}, \tilde{\mathbf{v}}_h\right) = \left(\nabla p_h^{n+1,i}, \tilde{\mathbf{v}}_h\right) \tag{36}$$

$$\forall\left(\mathbf{v}_h, q_h, \tilde{\mathbf{v}}_h, \tilde{\mathbf{v}}_h\right) \in \mathbf{V}_h \times Q_h \times \tilde{\mathbf{V}}_h \times \tilde{\mathbf{V}}_h \tag{37}$$

where $\mathbf{u}_h^{n+1}, p_h^{n+1}, \boldsymbol{\pi}_h^{n+1}, \boldsymbol{\xi}_h^{n+1}$ are the velocity, pressure, the projection of the convective term and the projection of the pressure gradient into the finite element space, respectively, and $\mathbf{V}_h \times Q_h \times \tilde{\mathbf{V}}_h \times \tilde{\mathbf{V}}_h$ are the spaces corresponding to the previously cited variables. The superscripts *n* and *i* refer, respectively, to the time step and Gauss-Seidel iteration counter in each time step. The subscript *h* stands for the discrete variables, δt is the time step size, β is a numerical switch in the presence of strong gradients, computed as a function of the pressure gradient and within the interval *I=[0 1]*, with values close to 1 in regions where the pressure field is smooth and close to zero near strong gradient regions (Löhner, 2001; Soto *et. al.*, 2004), τ is a parameter that controls stability and convergence behavior of the system, θ controls the time integration method, $\boldsymbol{\sigma}$ is the viscous stress tensor, **f** is the vector of external loads, Γ_{nu} is the portion of the boundary with Neumann boundary condition, υ is the cinematic fluid viscosity, $\mathbf{v}_h$, are the velocity weighting functions, $\tilde{\mathbf{v}}_h$ are weighting functions for the projection of the convection and pressure gradient terms, and q_h are weighting functions for pressure, respectively. It must be observed that $\boldsymbol{\pi}$ and $\boldsymbol{\xi}$ are auxiliary variables that do not require boundary and initial conditions.

The final discrete system given by Eqs. (33-37) is solved through the following procedure:

```
While the maximum number of iterations is not achieved, do
    Compute the time step;
    While the residual is greater than the adopted tolerance, do
            Step 1: solve the momentum equations in a segregated way, for each Cartesian component, Eq. (33);
            Step 2: solve the pressure equation using the velocity field obtained at Step 1, Eq. (34);
            Step 3: solve convection projection equations, in a segregated way, using the velocity field obtained at
                    Step 1, Eq. (35);
            Step 4: solve pressure projection equations, in a segregated way, using the pressure field obtained at
                    Step 2, Eq. (36);
            Step 5: check for convergence of the Gausse-Siedel iterations, by computing both pressure and
                    velocity residuals.
```

2.3 Stabilization and Other Numerical Issues

2.3.1 SUPG Terms for Spatial and Temporal Stabilization

Due to the difficulty associated with the numerical solution of the Navier-Stokes equations, it was necessary to incorporate a SUPG-like (Streamline Upwind Petrov-Galerkin) stability term (Antunes, 2008), similar to that obtained for the pressure equation through the fractional formulation, and this will be further detailed in what follows.

Stabilization of the Poison Equation for Pressure

The stabilization term of the pressure equation, Eq. (34), is given by

$$\left(\tau\left(\nabla p_h^{n+1,i} - \beta^{n+1,i-1}\boldsymbol{\xi}_h^{n+1,i-1}\right), \nabla q_h\right) \tag{38}$$

Which was obtained algebraically trough the fractional step method. It must be observed that this term, in discrete form, would result on a symmetric positive-definite matrix, acting here as a pre-conditioning for the pressure equation. The amount of stabilization is controlled through the artificial parameter β , which determines the regions of strong pressure gradients, controlling the system conditioning.

For consistency reasons, the parameter that controls pressure stability and convergence, τ , in Eq. (38), has to be computed for each node, and not for each edge. However, this would have destroyed the symmetry of the system. As the pressure stabilization terms are of at least 4th order, meaning that their effect on the final system accuracy is small, we follow the stabilization proposed in Soto *et al.*, (2004), which preserves symmetry and conservation at the price of losing consistency, computed as

$$\tau_{ij} = \frac{h_i^2 + h_j^2}{4\left(\upsilon_i + \upsilon_j\right) + 2\left(\left|\mathbf{a}_i\right| h_i + \left|\mathbf{a}_j\right| h_j\right)} \tag{39}$$

where h_i is the minimum length of the edges surrounding the nodal point *i*, υ_i is the nodal value of the kinematic viscosity and $\mathbf{a}_i$ is the convective velocity at node *i*. Note that the edge term is obtained as an arithmetic average of the nodal stabilizing terms.

Stabilization of the Momentum Equations

The stabilization term of the momentum equations, Eq. (33), is given by

$$\left(\tau\left(\mathbf{u}_h^{n+\theta,i-1} \cdot \nabla \mathbf{u}_h^{n+\theta,i} - \beta^{n+1,i-1} \boldsymbol{\pi}_h^{n+\theta,i-1}\right), \mathbf{u}_h^{n+\theta,i-1} \cdot \nabla \mathbf{v}_h\right) \tag{40}$$

Observe that this term is the same that appears when a SUPG method is adopted [Brooks and Hughes, 1982; Cebral *et al.*, 2001]. The addition of this term implies in adding diffusion to the non-smooth portion of the solution, dropping the approximation just to first order, and keeping second order in the regions of a smooth solution. The amount of diffusion is controlled by the parameter τ and β, with β being exactly the same parameter used in the stabilization term for the pressure equation (38).

The τ in Eq. (40) is computed for each node, with consistency guaranteed (Codina, 2000). Here this parameter is computed as

$$\tau_i = \frac{h_i^2}{4\upsilon_i + 2\left|\mathbf{a}_i\right| h_i} \tag{41}$$

where the variables in this equation have the same meaning as in Eq. (39).

2.3.2 Important Numerical Issues

Some further discussion about the parameter β and the time step computation is presented now. These parameters are computed automatically, without user interference, based on the physical and numerical parameters of the analysed problem and on the flow solution at each time step.

Monotonicity Preserving Term

The final scheme presented in Eqs. (33-37) represents a high-order formulation, which in principle is not able to avoid spurious oscillations for convective dominant flows. Apart from being non-physical, those oscillations would lead to deterioration of the convergence of the formulation, and the extra parameter β that detects regions with strong gradients was incorporated. This parameter is computed by the difference between the pressure gradient and its projection onto the finite element space (Antunes, 2008; Soto *et al.*, 2004). At each iteration and direction, β is computed for the *ij* as

$$\beta_{k,ij} = 1 - \left| \frac{\left|p_i - p_j\right| + 0.5 l_{k,ij} (\xi_{k,i} + \xi_{k,j})}{\left|p_i - p_j\right| + \left|0.5 l_{k,ij} (\xi_{k,i} + \xi_{k,j})\right|} \right| \tag{42}$$

where p_i is the pressure at node i, $l_{k,ij}$ is the ij edge dimension in the *k* direction, and $\xi_{k,i}$ is the *k* component of the pressure gradient projection at nodal point i. It can be verified that $0 \le \beta \le 1$, taking value close to 1 for smooth pressure Field and close to 0 for strong pressure gradients regions. This parameter acts by controlling the higher-order SUPG-like and the pressure stabilizing terms in Eqs. (33) and (34).

Time Step Computation

The time step length is calculated following the expression given by Zienkiewicz and Taylor (1988), as

$$\delta t = \frac{\delta t_\sigma \delta t_v}{\delta t_\sigma + \delta t_v} \qquad \text{with} \qquad \delta t_\sigma = \frac{h_E}{|u_E|} \qquad \text{and} \qquad \delta t_v = \frac{h_E^2}{2\mu} \tag{43}$$

in which δt_σ and δt_v represent the stability limit for pure adjective an pure diffusive phenomena, respectively. As we adopt an edge-based data structure, then h_E, which represents a characteristic length, is taken as the edge length, $|u_E|$ is the absolute velocity at the edge and μ is the dynamic viscosity. This time step is used just as a reference value as the presented formulation is fully implicit and much larger time increment can in general be adopted.

3. SOME IMPLEMENTATION ISSUES

3.1 Edge Based Data Structure

The matrices resulting from the finite element discretization are assembled edge-by-edge (Löhner, 2001), which has the advantage of being computationally more efficient than their element-based counterparts. It demands a pre-processor stage, executed only once for a fixed mesh, where some coefficients, based on geometric mesh data, are computed and associated to the edges. Then, during the simulation these coefficients are used to assemble the global matrices. The edge-based implementation can guarantee discrete local conservation and symmetry (Soto *et al.*, 2004).

3.2 Parallel Implementation

Parallel computers make possible the solution of large scale problems which otherwise would be impossible to be solved using a computer with a single processor due to the limited amount of available memory or to the excessive total time of simulation. It also represents an important tool to simulate complex physics with accuracy that could not be achieved using a machine with a unique processor. Within this framework, large scale problems can be simulated using meshes with a high level of refinement to obtain very accurate results. Here, the domain decomposition's paradigm is used to divide the computational domain into sub-domains and associate each one to a process that will be executed by one processor. The parallel library ParMetis (Karypis *et al.*, 2003) has been used as a mesh partitioning tool to reduce communication among processors. The mesh partition is based on its edges in such way that any edge at the partition interface has a remote copy. Another parallel library, PETSc, has been used to define distributed structures as vector and matrices. Different solvers and pre-conditioners were also available by PETSc and used to solve the systems of linear equations that results from the adopted formulation. Some tasks not supported by ParMetis or PETSc were performed using the MPI (Pacheco, 1997) routines specially to change data among processors.

4. Numeric Applications

In this section, the results of a laminar incompressible flow simulation are compared with some results found on literature.

Back-Facing Step

This example consider a viscous and incompressible flow through a channel with a "back-facing step" (Antunes, 2008; Armaly *et al.*, 1983; Soto *et al.*, 2004; Williams e Baker, 1997). Figure (1) shows the adopted mesh with 77163 tetrahedrals, 96506 edges and 14617 nodes. The mesh has been partitioned among two processors and each color corresponds to a subdomain associated to a processor. The mesh was refined near the step.

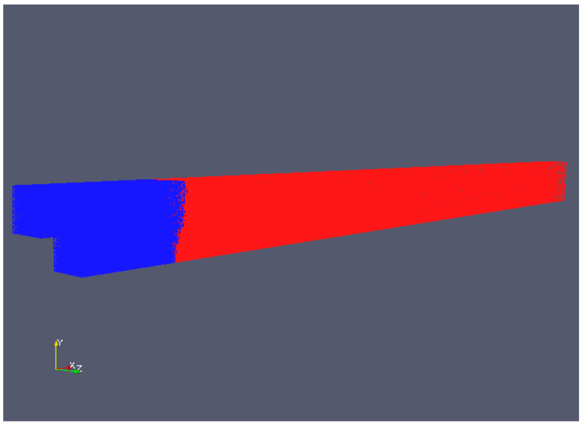


Figure 1. Domain of interest.

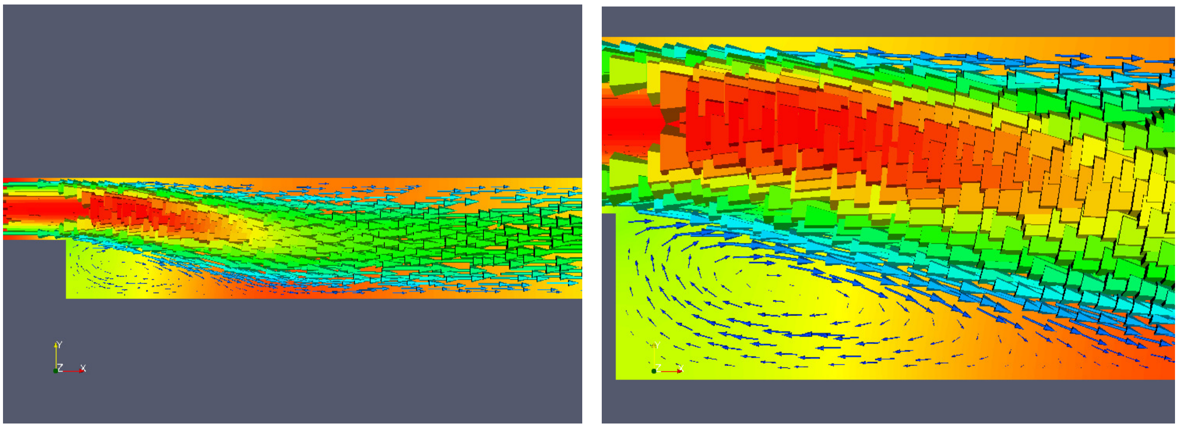
The geometry is: inlet length 1m; height, 94m; total length, 31m.
The boundary conditions are:

- Inlet: parabolic profile for prescribed velocity;
- Outlet: p=0 and "pseudo-tractions" set to zero;
- Non slip condition on walls: $\mathbf{u} = \mathbf{0}$;
- On front and back boundaries the viscous tensor on x and z directions are set to zero and the y velocity component was also set to zero.

Simulation was performed with Reynolds number 100, with $U_{\max}$ being the maximum inlet velocity set to obtain the desired Reynolds, which is calculated as (Soto *et al.*, 2004; Williams and Baker, 1997):

$$\mathrm{Re} = \frac{(2/3)U_{\max}\,2h}{\upsilon} \tag{44}$$

Figure 2 shows the velocity field highlighting the vortex created by the step.



(a)

(b)

Figure 2. (a) Velocity field, (b) a zoom in vortex region.

On this example, we compare with other authors the vortex length. Here, a vortex of 2.6m was formed, as it can be seen in Fig. 3(a). This value is quite near that found in literature (Armaly *et al.*, 1983, Williams and Baker, 1997) with 2.8m. Although a convergence mesh study had not been performed, the results can be considered good.

.

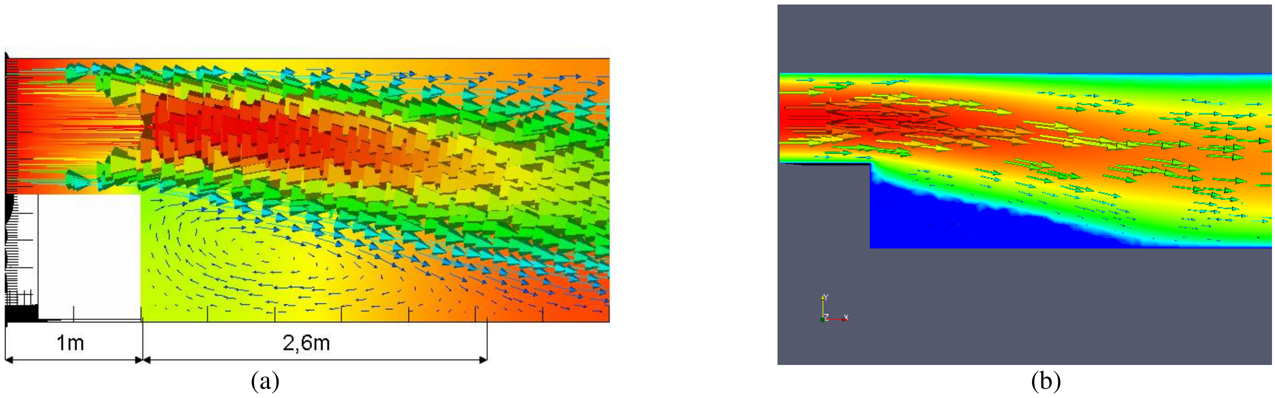


Figure 3. (a) Pressure and velocity field showing vortex length, (b) velocity color map for flow direction.

Figure 3 (b) shows in blue the portion of the domain with negative velocity component at the flow direction. This is one way to estimate the vortex length.

Figure 4 (a) and (b) shows in a quality way the velocity field for Reynolds number 1000 highlighting the second vortex created on top of the channel and the pressure field showing the reduction of the pressure at the same location, respectively.

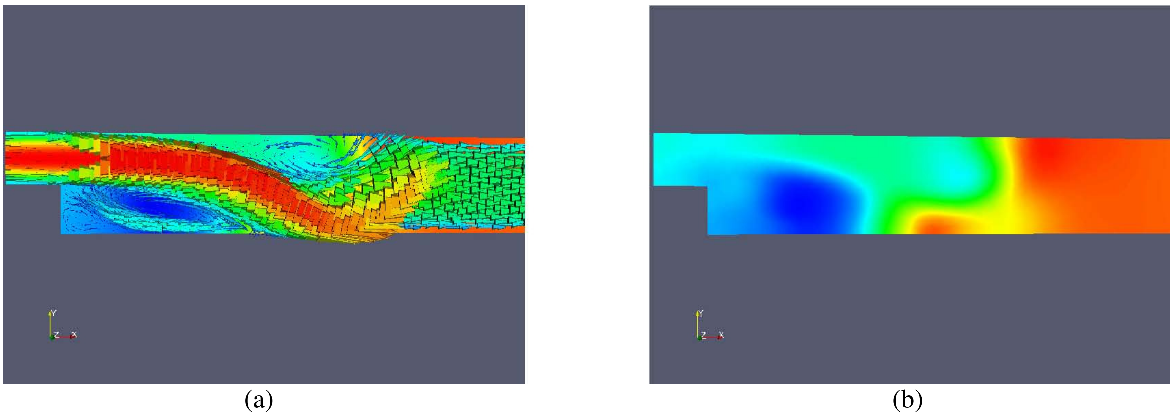


Figure 4. Velocity and pressure field for Reynolds number 1000.

5. CONCLUSIONS

A monolithic implicit FEM formulation has been presented to solve Navier-Stokes equations for laminar incompressible tridimensional flows. The formulation also includes stability properties for pressure and convection and a Fractional Step procedure. An edge-based data structure was used to assembly all global matrices involved in the computational model and a parallel implementation, using domain decomposition, was developed allowing simulation on distributed memory machines. The implicit formulation presented is very attractive for the analysis of transient problems in which the time step can assume larger values than those required for explicit formulation, which are severely restricted by stability criteria.

Numerical results for classical model problems were obtained and compared with others found in literature showing good agreement when simulating laminar flows at different Reynolds numbers. Details about the methodology adopted for a parallel, edge-based implementation and a study on the computational efficiency will be presented in future works.

6. ACKNOWLEDGEMENTS

The authors would like to thank the Brazilian Research Council (CNPq), the Pernambuco State Research Council (FACEPE) and the Petroleum National Agency (ANP-PRH26) for the financial support and the Center for Parallel Computations (NACAD) for the computational support.

7. REFERENCES

Antunes, A. R. E., 2008, “Um Sistema Computacional Utilizando uma Formulação do Tipo Fractional Step e o Método dos Elementos Finitos por Arestas para a Análise de Escoamentos Incompressíveis 3D Usando Computação Paralela”, D.Sc. Thesis, Recife, Brazil (*in Portuguese*)

Armaly, B. F., Durst, F., Pereira, J. C. F. and Schönung, B., 1983, “Experimental and theoretical investigation of backward-facing step flow”, Journal Fluid Mech., vol. 127, p. 473-496.

Brooks, A. N. and Hughes, T. J. R., 1982, “Streamline Upwind/Petrov-Galerkin Formulations for Convection Dominated Flows with Particular Emphasis on the Incompressible Navier-Stokes Equations”, Computer Methods in Applied Mechanics and Engineering, vol. 32, pp. 199-259.

Cebral, J.R., Yim, P. J., Löhner, R., Soto, O., Marcos, H., Choyke, P. L., 2001, “New methods for computational fluid dynamics of carotid artery from magnetic resonance angiography”, in: Proceedings of SPIE Medical Imaging, vol. 4321, paper No 22, San Diego, California, February.

Chorin, A. J., 1967, “A Numerical Method for Solving Incompressible Viscous Problem”, J. Comput. Phys., vol. 2.

Codina, R., 2000. “Stabilization of incompressibility and convection trough orthogonal sub-scales in finite element methods”, Compt. Meth. Applied Mechanics and Engineering, v. 190, p. 1579-1599.

Codina, R., 2002. “Stabilized finite element approximation of transient incompressible flows using orthogonal subscales”, Comp. Meth. Applied mechanics and Engineering, vol 191, p. 4295-4321.

Donea, J. and Huerta, A., 2003, “Finite Element Methods for Flow Problems”, John Wiley & Sons Ltd., 362p.

Gresho, P. M. and Sani, R. L., 1998. “Incompressible flow and the finite element method”, John Wiley & Sons, vol. 1 and 2.

Karypis, G., Schgloegel, K. and Kumar, V., 2003, “PARMETIS – Parallel Graph Partitioning and Sparse Matrix Ordering Library”, Reference Manual, University of Minnesota

Löhner, R., 2001. “Applied CFD techniques - An introduction based on finite element methods”, john Wiley & Sons Ltd, 376p.

Pacheco, P., “Parallel Programming with MPI”, Morgan Kaufmam Publishers Inc, 1997, 418p.

PETSc Online Manual, Web page. http:/www-unix.mcs.anl.gov/petsc/petsc-as/documentation.

Soto, O., Löhner, R. and Cebral, J., 2001, “An implicit monolithic time accurate finite element scheme for incompressible flow problems”, In: 15th AIAA Computational Fluid Dynamics Conference, Anaheim, EUA, June.

Soto, O., Löhner. R., Cebral, J. and Camelli, F. 2004. “A stabilized edge-based implicit incompressible flow formulation”, Comp. Meth. Applied Mechanics and Engineering, vol. 193, p. 2139-2154.

Temam, R., 1969, “Sur l’approximation de la solution des équaciones de Navier-Stokes par la méthode dês pás fractionaires (I)”, Arch. Ration. Mech. Anal., vol. 32 (*in French*).

Williams, P. T. and Baker, A. J., 1997, “Numerical simulations of laminar flow over a 3D backward-facing step”, Int. Journal Num. Meth. In Fluids, vol 24, p. 1159-1183.

Zienkiewicz, O. C. and Taylor, R. L., 1988, “The finite element method: basic formulation and linear problems”, vol. 1, McGraw-Hill, 4ª edition, 648p.

8. RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.