

# A SEMI-LAGRANGIAN SCHEME FOR FLUID FLOW SIMULATIONS WITH A ZIENKIEWICZ-TYPE FINITE ELEMENT INTERPOLATION

Gustavo Charles P. de Oliveira, [tavolesliv@gmail.com](mailto:tavolesliv@gmail.com)  
Norberto Mangiavacchi, [norberto.mangiavacchi@gmail.com](mailto:norberto.mangiavacchi@gmail.com)

University of the State of Rio de Janeiro, Rio de Janeiro, Brazil

José Pontes, [jopontes@metalmat.ufrj.br](mailto:jopontes@metalmat.ufrj.br)

Federal University of Rio de Janeiro, Rio de Janeiro, Brazil

**Abstract.** We develop a high-order finite element interpolation scheme for use in incompressible fluid flow simulations with a semi-lagrangian description of movement. Attention is given to obtain a best fit evaluation of the relevant advected variables from departure points of particle trajectories, evolving according to evolution laws of the continuum mechanics, like the Navier-Stokes equations. Third order polynomials along with gradients evaluated at the nodes of the adopted Zienkiewicz-type nonconforming element define the corresponding set of degrees of freedom that assure a high order representation of the sought solution. By including a numerical residual in the shape functions we seek to minimize deviations observed on the approximation of linear functions in the domain. Our numerical experiments show that the Zienkiewicz-type element used in our work might compromise the ability to approximate linear functions on the benefit of a better representation of more complex functions. We present the first numerical results concerning the simulation of the evolution of a three-dimensional Lamb-Oseen vortex as a step on the validation of the proposed scheme. The numerical experiments were made on a tetrahedral grid of elements.

**Keywords:** Semi-Lagrangian, Interpolation, Finite Element

## 1. INTRODUCTION

Semi-Lagrangian Methods, also named Characteristic Methods, were established from meteorologic sciences in the framework of numerical climatic predictions, due to its applicability to atmospheric fluid flow simulations and the ability of building on a fixed computational meshing arrangement. These methods gather the potencialities existing between the Lagrangian and Eulerian movement descriptions according to (Staniforth and Côté, 1991) and (Smolarkiewicz and Pudykiewicz, 1992). Ever since, the insertion of Semi-Lagrangian methods into another scientific ramifications turned into a feasible experimental task. Combination with the Finite Element Method for solving problems governed by differential equations which encompass advective properties, for instance, dates back to the pioneering work introduced by (Pironneau, 1982). Since these results were very cohesive, the implementation of computational codes binding the Navier-Stokes equations to the semi-lagrangian approach could be, later, treated in a more specific way, with emphasis given to the interpolation mechanisms (Xiu and Karniadakis, 2001). It is well known that one of the main characteristics of successful semi-lagrangian schemes is indeed an interpolation scheme suitable to evaluate variables (scalar or vectorial) at the *departure points* (inherent to the method), or “foot” of each trajectory. Considering that the characteristics are the curves of solution of differential equation  $\frac{d\mathbf{X}}{d\tau}(\mathbf{x}, \tau; t) = \mathbf{u}(\mathbf{x}, \tau; t)$ , where  $\mathbf{x} = (x_1, x_2, x_3)$  is a point of domain  $\Omega = \mathbb{R}^3$ ,  $\mathbf{u} = (u, v, w)$  is the velocity field distributed over this domain,  $\tau = \cup_{n=0}^{N-1} [nt, (n+1)t]$  is the time and  $t$  is a particular time of  $\tau$ . The departure point of the each characteristic is the point  $\mathbf{x}_d$  occupied by the particle  $\chi$  when  $\tau = nt$ ,  $n = 0, 1, 2, \dots, N-1$ . Semi-Lagrangian Methods are then concerned to perform a *backward-in-time* integration along the characteristic, so that the particle  $\chi$  occupying the position at  $\mathbf{X}(\mathbf{x}_d, \tau; nt)$  is advected to the position at  $\mathbf{X}(\mathbf{x}_a, \tau; (n+1)t)$ , where  $\mathbf{x}_a$  belongs to the fixed computational mesh. In the effect of the backward integration,  $\mathbf{x}_d$  might not match a mesh point, hence, an interpolation is needed to evaluate the variables at that point.

## 2. FINITE ELEMENT DISCRETIZATION AND MATHEMATICAL BACKGROUND

In bidimensional space, the Zienkiewicz element is a triangular element that uses the values of the function and their first order derivatives at vertices as degrees of freedom. According to (Ciarlet, 2002), the expansion of this element to tridimensional space is the Hermite's element TE16 with a particular interpolation space made up of cubic polynomials. TE16 has been the terminology used in (Wang and Xu, 2006a) and (Wang and Xu, 2006b) to explain that the directional gradients are calculated on the edges of tetrahedra from a vertex and pointing to the adjacents by invoking sixteen points of evaluation on each element, of which twelve are gradients and four are function values. Z-Type (Zienkiewicz-Type, for brevity) elements form a class of nonconforming finite elements and they have been explored for applications of high-order problems, like the fourth-order elliptical ones. Nonconforming elements have been conceiving a strategy to reduce the order of polynomials and the degrees of freedom responsible for the high computational cost of practical applications implemented with conforming elements. Continuity requirements influence this expense as well as the informations about

gradients of the functions. By using gradients information, we intend find a higher interpolation to the semi-lagrangian scheme developed here.

Let  $\Omega = \mathbb{R}^3$  be our problem domain,  $\Gamma = \partial\Omega$  its boundary,  $\mathcal{T}$  a tetrahedralization of  $\Omega$ ,  $T \in \mathcal{T}$  the mesh elements with vertices  $v_i$ ,  $i = 1, 2, 3, 4$ . Face centroids are the points  $v_{ijk} = (v_i + v_j + v_k)/3$  (with cyclic permutation on  $i, j, k$ ); however we mention the face centroids here just for understanding. They will not be used directly in our calculations, but the centroid of element, to be introduced later. Finally,  $f$  is an arbitrary (at first glance) 3-fold function - we will later associate  $f$  to the fluid flow velocity field - to be approximated. Velocity is the most important variable in our implementation.

Each  $T \in \mathcal{T}$  suffers an affine map to a tetrahedral pattern element  $\hat{T}$ , with barycentric coordinates  $\lambda_i$  according to the common coordinate change process in finite elements codes. Following the compact notation from (Ciarlet, 2002), we denote by:  $\hat{v}_i$ , the vertices of the pattern tetrahedra;  $f(\hat{v}_i)$ , the evaluation of function  $f$  at the vertices and  $\nabla f(\hat{v}_i)$  the evaluation of the gradients at the vertices, with  $i = 1, 2, 3, 4$  in all. Intentionally,  $\hat{v}_1 = (1, 0, 0)$ ,  $\hat{v}_2 = (0, 1, 0)$ ,  $\hat{v}_3 = (0, 0, 1)$  and  $\hat{v}_4 = (0, 0, 0)$ .

TE16 is the element  $(\mathcal{K}, \mathcal{P}, \mathcal{N})$  that may be identified as: 1)  $\mathcal{K} = \mathcal{T}$ ; 2)  $\mathcal{P} = \Pi'_3$ ; 3)  $\mathcal{N} = \{f(v_i), \nabla f(v_i), 1 \leq i \leq 4\}$ , where the polynomial space  $\Pi'_3 := p \in \Pi_3(T) \mid \psi_{ijk}(p) = 0$ . In turn,  $\Pi_3(T)$  is the set of the all polynomials with degree  $\leq 3$  over  $T$  and:

$$\psi_{ijk}(f) = 6f(v_{ijk}) - 2 \sum_{l=i,j,k} f(v_l) + \sum_{l=i,j,k} \nabla f(v_l)(v_l - v_{ijk}), \quad \forall f \in C^1 T. \quad (1)$$

Once we are interested in directional gradients, we look at this element. Particularly, our focus is in the shape functions raising its correspondent interpolant:

$$\mathcal{I}_T f = \sum_{i=1}^4 (3\lambda_i^2 - 2\lambda_i^3 + 2\lambda_i \sum_{\substack{1 \leq j < k < 4 \\ j, k \neq i}} \lambda_j \lambda_k) f(v_i) + \frac{1}{2} \sum_{1 \leq i \neq j < 4} \lambda_i \lambda_j (1 + \lambda_i - \lambda_j) \nabla f(v_i)(v_j - v_i), \quad \forall f \in C^1(T) \quad (2)$$

Shape functions associated to derivatives hold the propriety of vanishing on each vertice of tetrahedra. Moreover, they are cubic, propping a considerable order for interpolation.

Now, we address ourselves to explain about our mesh element: the MINI 3D. Strongly used for solving problems coupling velocity and pressure, it has tetrahedral geometry like TE16 one, however it differs on the degrees of freedom set and on the shape functions space. For now, we will postpone more details about this element, suggesting reading of (Arnold *et al.*, 1984). We could describe it as the element  $(\mathcal{K}_{mini}, \mathcal{P}_{mini}, \mathcal{N}_{mini})$  with  $\mathcal{K}_{mini} \equiv \mathcal{K}$  and  $\mathcal{P}_{mini} \neq \mathcal{P}$ . In contrast, it is enough know the degrees of freedoms set of MINI 3D element, i.e.,  $\mathcal{N}_{mini} = \{f(v_i), 1 \leq i \leq 5; p(v_i), 1 \leq i \leq 4\}$ . In the MINI 3D, velocity is evaluated at five points, all the four vertices more the centroid (of element), whereas the pressure  $p$  is evaluated only at the vertices. Our interpolation scheme lifts up from a structuring linking pieces of these two elements. It would be as if we coupled each other, in the sense of handling their capabilities. From the MINI element, we take the geometry, while from TE16 we take a part of its interpolant. Figure 1 shows us a simple configuration of elements coupling. Concentration is a scalar field present in the simulations.

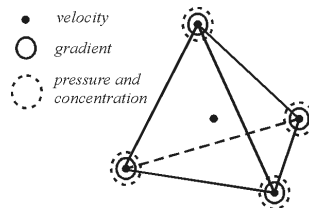


Figure 1. 3D MINI element with evaluation of gradients

### 3. BACKWARD-IN-TIME INTEGRATION AND INTERPOLATION SCHEMES

As we have introduced in the principle, to integrate the particle trajectory backward-in-time we start from an arrival point  $\mathbf{x}_a$  whose velocity at that position is the reference information that we have to proceed to the approximation. As the simulation runs, we evaluate the velocity field on early time to obtain it on a future time. Semi-lagrangian approach is embedded in treatment of convective term type  $\mathbf{v} \cdot \nabla \mathbf{v}$ . We will see ahead the material derivative of simulated equations claim to be approximated by:

$$\frac{D\mathbf{v}}{Dt} = \frac{\mathbf{v}^{(n+1)t} - \mathbf{v}_d^{nt}}{\Delta t}, \quad (3)$$

where the subscript  $d$  indicates the evaluation of velocity at the departure point. This is the heartwood of discrete version of Semi-Lagrangian approach.

Our integration scheme computes the trajectory by doing successive corrections through internal iterations, pursuing the original curve by means of retilinear approximations. In the work by (dos Anjos, 2007), the semi-lagrangian integration was made through a simple straight line on each step time. Here, we complement the idea, but remaking it several times on space as far as a convergence criterion can be satisfied. Basically, the computational algorithm works as follows:

- i) For each point belonging to the mesh, find a departure point;
- ii) Evaluate the velocity at departure point;
- iii) Evaluate the mean velocity between the mesh point velocity and the departure point one;
- iv) Repeat internal iterations as far as the convergence criterion can be satisfied;
- v) Go ahead to the next step time.

Considering  $\mathbf{x}$  a point of some  $T \in \mathcal{T}$ ;  $t_n = n\Delta t$  the step time and  $\mathbf{a}(\mathbf{x}, t_n)$  the velocity at point  $\mathbf{x}$  and time  $t_n$ , if we use a mathematical lexicon, we expound the previous algorithm in the following context:

- i) Find  $\mathbf{x}_d^k \equiv \mathbf{x}_{n,k}^\gamma \in T^\gamma$  such as  $\mathbf{x} = \mathbf{x}_d^k + \mathbf{a}(\mathbf{x}, t_n^k)\Delta t$ , where the parameter  $\gamma$  assumes values  $i$ ,  $e$  or  $v$  just to identify, in this sequel, if each departure point  $\mathbf{x}_d$  either is a point from interior of element  $T^\gamma$ , or an edge point, or if it matches a vertex;  $k$  is referred to as a current internal iteration;
- ii) Calculate  $\mathbf{a}(\mathbf{x}_d, t_n^k)$  by interpolation (discussed below);
- iii) Calculate the mean velocity  $\bar{\mathbf{a}}(\mathbf{x}_{mean}, t_n^k) = \frac{1}{2}[\mathbf{x}(\mathbf{x}_d, t_n^k) + \mathbf{x}(\mathbf{x}, t_n^k)]$  at the midpoint  $\mathbf{x}_{mean}$ ;
- iv) Do  $k = k + 1$  while holds:  $\|\bar{\mathbf{x}}(\mathbf{x}_{mean}, t_n^{k+1}) - \bar{\mathbf{x}}(\mathbf{x}_{mean}, t_n^k)\| < \delta$ , for  $\delta$  small. Nicely, we have chosen  $\delta = 10^{-8}$ ;
- v) Take  $\bar{\mathbf{x}}(\mathbf{x}_{mean}, t_n^K) \equiv \mathbf{a}(\mathbf{x}_d, t_n)$  after  $K$  internal iterations as the final velocity obtained on the step time  $n\Delta t$  to proceed to the velocity field calculus  $\mathbf{a}(\mathbf{x}, t_{n+1}) = \mathbb{M}_q \mathbf{a}(\mathbf{x}_d, t_n)$  on the next step, given all the matrices  $\mathbb{M}_q$  operating according to the finite element formulation. (To be exposed in Sec. 4).

The sketch of Fig. 2 shows us how the internal iterations work for approximating the trajectories. We select a mesh partition at an arbitrary time step. Tetrahedra with coarser edges represents  $T^\gamma$  with its vertices. We have omitted the mesh partition centroids for visual clarity, leaving  $v_5$  as centroid of the highlighted element. The dash line traveling from the vertex  $v_1$  and ending at the point  $\mathbf{x}_a$  is the particle trajectory at that interval time  $\Delta\tau$ ; the bold points upon the vertices represent the velocity evaluations at that vertices; the points filled in white represent the successive departure points found after carrying out several internal iterations. Internal iterations happen during one step time as far as the convergence. Numbers 1, 2 and 3 at the line boards to the left indicate three internal iterations corresponding to their respective backward integration: dash-point line (first); dash-point-point line (second) and dash-point-point-point line (third). Each line of these is a correction for the original trajectory in order to pursue the maximum adjust for it. After all, it is important mention that our scheme uses only spacial approximation, do not taking advantage of velocity evaluations at any intermediary times, but at intermediary positions. In (Bonaventura, 1997), the fractional way of thinking the evaluations in intermediary times is very well studied under the finite differences scope.

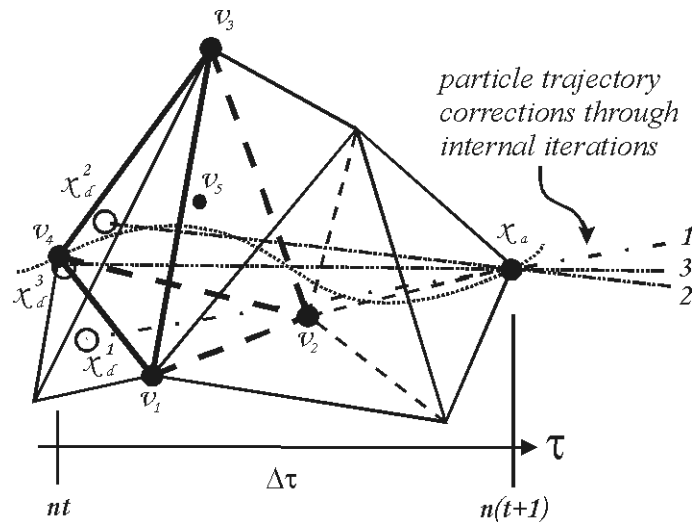


Figure 2. Internal iterations being performed on a tetrahedral mesh.

We consider our approach a thread from what is provided by the introduction of gradient recovery and post-processing techniques into finite elements approximations as it is declared by (Silva *et al.*, 2004). In that paper, the applications were addressed to structures and stress studies, however the references therein showed itself theoretically and historically useful. Higher order polynomials assume a strong relation with those techniques in order to yield good interpolation schemes.

Mapping each  $T \in \mathcal{T}$  to  $\hat{T}$  require a Jacobian transformation. Once the barycentric coordinates  $\lambda_i$  are dominant linear

functions over  $\hat{T}$ , their gradients will be given by  $\frac{\partial \lambda_l}{\partial x_i}$ , with  $1 \leq l \leq 4, 1 \leq i \leq 3$ , following the variables change due to the transformation. By interplaying the usual finite element linear interpolation and the shape functions from the second term of TE16 element  $\mathcal{I}_T$  interpolant, we developed a new interpolant,  $\mathcal{I}_3$ , performing the sum:

$$\mathcal{I}_3 f = \sum_{l=1}^4 f(v_l) \lambda_l + \sum_{i=1}^3 \sum_{j=1}^4 \sum_{k=1}^4 \left( \frac{\partial f}{\partial x_i}(v_j) - \sum_{l=1}^4 f(v_l) \frac{\partial \lambda_l}{\partial x_i} \right) \varphi(i, j, k) \quad (4)$$

where

$$\varphi(i, j, k) = \frac{1}{2} (x_{ik} - x_{ij}) (\lambda_j \lambda_k + \lambda_j^2 \lambda_k - \lambda_j \lambda_k^2) \quad (5)$$

is the same shape function as the second term of TE16 element; the labels  $x$  with their subscripts are associated with the positions of each vertex  $v$ , e.g.,  $v_l = (x_{l1}, x_{l2}, x_{l3})$ .

The l.h.s. of Eq. 4 has a subscript 3 recalling the Hermite's cubics figuring on the r.h.s. The r.h.s., in turn, is drawn up by a linear part more a term relating to the gradients. Then, we can rewrite the Eq. 4 like:

$$\mathcal{I}_3 f = \psi_{lin}(f) + \psi_{grad}(f) \quad (6)$$

In case of the function to be interpolated has a linear behavior,  $\psi_{grad}(f)$  vanishes, since the gradients will be constant and the difference inside parenthesis will be null. Thus  $\psi_{lin}(f)$  undertakes carrying out the interpolation. Otherwise, when the gradients are nonzero,  $\psi_{grad}(f)$  acts interpolating the function by adjusting the approximation up to third order.

Other interpolation scheme investigated here results from a approach much more convincent relating to the Z-type expression of finite elements. In (Andreev and Racheva, 2010), a new Z-type triangular element was proposed, in which they modified the Zienkiewicz element by adding an integral value to its degrees of freedoms set. However, the shape functions of the Zienkiewicz triangle still were declared there. We extended these functions to tridimensional space applying them to the tetrahedra.

#### 4. NUMERICAL RESULTS

Our interpolation scheme was developed to be applied for solving the 3D incompressible Navier-Stokes equations coupled to the transport equation of a chemical species, written compactly in adimensional vectorial form as:

$$\nabla \cdot \mathbf{v} = 0 \quad (7)$$

$$\frac{D\mathbf{v}}{Dt} = -\frac{1}{\rho} \nabla p + \frac{1}{Re} + \nu \nabla \cdot (\nabla \mathbf{v} + \nabla \mathbf{v}^T) + \frac{1}{Fr^2} \mathbf{g} \quad (8)$$

$$\frac{Dc}{Dt} = \frac{\kappa}{ReSc} \nabla^2 c \quad (9)$$

Above,  $\mathbf{v} = (u, v, w)$  is the velocity field,  $\rho$  the density,  $p$  the pressure field,  $\nu$  the viscosity and  $c = c(\mathbf{x})$  the chemical species concentration field spread over domain. Coefficients  $Re$ ,  $Sc$ ,  $Fr$  and  $\kappa$  are, respectively, the Reynolds, Schmidt, Froude numbers and the chemical species diffusion. For simulations we will neglect the gravitational term  $\mathbf{g}$ .

Taking account of all tools concerning to a finite element formulation, it is relevant to do some definitions more. Thus, let  $\mathcal{H}^1$  and  $\mathcal{L}^2$  be an Hilbert and Lebesgue, in this sequel, functional spaces so that:

$$\mathcal{H}^1(\Omega) := \{ \mathbf{w} \in \mathcal{L}^2(\Omega) \mid \frac{\partial \mathbf{w}}{\partial x_i} \in \mathcal{L}^2(\Omega), i = 1, 2, 3 \}$$

and

$$\mathcal{L}^2(\Omega) := \left\{ \mathbf{w} : \Omega \rightarrow \mathbb{R}^3 \mid \left( \int_{\Omega} |\mathbf{w}|^2 d\Omega \right)^{1/2} < \infty \right\}.$$

These functional spaces are necessary to create approximation subspaces and they figure in Adams (1975). Now let  $\mathcal{S}$  and  $\mathcal{V}$  be the *trial functions* space and the *weight functions* space, respectively, given by:

$$\mathcal{S} := \{ \mathbf{u} \in \mathcal{H}^1(\Omega) \mid \mathbf{u} = \mathbf{u}_D \text{ on } \Gamma_D \}$$

and

$$\mathcal{V} := \{ \mathbf{w} \in \mathcal{H}^1(\Omega) \mid \mathbf{w} = \mathbf{0} \text{ on } \Gamma_D \}$$

with  $\Gamma_D$  being a Dirichlet-type boundary.

After discretizing the equations by finite elements and weakening them through Galerkin's method, we are conducted to a matricial system of ordinary differential equations. Writting it on bilinear form, we declare the statement: *we should find solutions*  $\mathbf{v}(\mathbf{x}, t) \in \mathcal{V} \subset \mathcal{H}^1(\Omega)$ ,  $p(\mathbf{x}, t) \in \mathcal{P} \subset \mathcal{L}^2(\Omega)$  and  $c(\mathbf{x}, t) \in \Psi \subset \mathcal{L}^2(\Omega)$  so that for any  $\mathbf{w} \in \mathcal{V}_0$ ,  $q \in \mathcal{P}_0$  and  $r \in \Psi_0$  obtain:

$$\mathbf{m}\left(\frac{D\mathbf{v}}{Dt}, \mathbf{w}\right) + \frac{1}{Re} \mathbf{k}(\nu; \mathbf{v}, \mathbf{w}) + \mathbf{g}(p, \mathbf{w}) = 0 \quad (10)$$

$$\mathbf{d}(q, \mathbf{v}) = 0 \quad (11)$$

$$\mathbf{m}\left(\frac{Dc}{Dt}, r\right) + \frac{1}{ReSc} \mathbf{k}(\kappa; c, r) = 0 \quad (12)$$

with subscript 0 meaning that the set functions must vanish on the boundary and the matrices given by

$$\mathbf{m}\left(\frac{D\mathbf{v}}{Dt}, \mathbf{w}\right) = \int_{\Omega} \left[ \frac{D\mathbf{v}}{Dt} \right] \cdot \mathbf{w} \, d\Omega$$

$$\mathbf{k}(\nu; \mathbf{v}, \mathbf{w}) = \int_{\Omega} [\nu(\nabla \mathbf{v} + \nabla \mathbf{v}^T)] : \mathbf{w} \, d\Omega$$

$$\mathbf{g}(p, \mathbf{w}) = \int_{\Omega} p [\nabla \cdot \mathbf{w}] \, d\Omega$$

$$\mathbf{d}(q, \mathbf{v}) = \int_{\Omega} [\nabla \cdot \mathbf{v}] q \, d\Omega$$

$$\mathbf{m}\left(\frac{Dc}{Dt}, r\right) = \int_{\Omega} \left[ \frac{Dc}{Dt} \right] r \, d\Omega$$

$$\mathbf{k}(\kappa; c, r) = \int_{\Omega} [\kappa \nabla c] \nabla r^T \, d\Omega$$

We emphasize that Galerkin's method is obtained by doing successive integration by parts in the weighted equations, at which the weight functions space is taken as the same as the trial functions space one.

For simulations, we have chosen a finite element mesh similar to a flatten cube. Mesh adimensional sizes are given by  $l \times l \times h$ , where  $l$  in respecting to  $x$ ,  $y$  and  $z$  directions.  $z$  is taken to be thin equivalent to a unique layer of tetrahedras. In order to compare details of trajectory between two semi-lagrangian methods - developed in our research group -, we have selected the *Lamb-Oseen* vortex (Delbende *et al.*, 2004), that is a solution for the Navier-Stokes equations. Curvilinear behavior of trajectories help us to visually identify improvements of approximation. Vortex velocity field drives to steady-state after a reasonable time has elapsed. In a cylindrical coordinates frame, the vortex velocity profiles read:

$$v_r(r, \tau) = 0$$

$$v_{\theta}(r, \tau) = \frac{\gamma}{2\pi r} \left[ 1 - \exp\left(-\left(\frac{r}{r_c(\tau)}\right)^2\right) \right] \quad r_c(\tau) = \sqrt{4\nu\tau}$$

$$v_z(r, \tau) = 0$$

Above,  $\tau$  represents time,  $\gamma$  the vortex circulation,  $\nu$  the cinematic viscosity of fluid under issue and  $r_c$  the marked radius at the vortex centre. Vortex is inserted as initial condition at the mesh centre. Outflow boundary conditions were imposed so that vortex disperse freely. Mathematically, we have  $\Upsilon = \Omega \times \tau$ , with  $\Omega = [0, l] \times [0, l] \times [0, h]$  and  $\tau = [t_0, t]$ . Regions  $\mathcal{R}_v$  and  $\mathcal{R}_c$  spanning mesh interior points receive velocity values and chemical concentration values, i.e.,  $\mathbf{x} \in \mathcal{R}_v : d(\mathbf{x}, \mathbf{x}_c) \leq r_{ref}$  belonging to the circle with radius  $r_{ref}$  and center  $\mathbf{x}_c$  set velocity initial condition, whereas  $\mathbf{x} \in \mathcal{R}_c : \frac{(x-x_c)^2}{a^2} + \frac{(y-y_c)^2}{b^2} = 1$   $a > b$  is a concentration elliptical cylinder immersed inside vortex. We term  $\mathbb{S}L_{old}$  (the old one) and  $\mathbb{S}L_{new}$  (the new one) our two developed numerical approximation Semi-Lagrangian methods. Relative error calculations were performed with the discretized  $\mathcal{L}_2$ -norm formulae:

$$Err_{rel} = \left\{ \int_{\Sigma_n} \frac{\|\mathbf{u} - \mathbf{u}_h\|_n^2}{\|\mathbf{u}\|_n^2} \right\}^{\frac{1}{2}}$$

where  $\mathbf{u}$  and  $\mathbf{u}_h$  are the velocity fields evaluations of exact and approximated solutions for each node  $n$  of mesh.

Results follow in the sequel. Each pair of figures is the comparison between the two methods. The new method is represented by the inferior ones. On a mesh with 3200 vertices and 9120 elements, we have simulated the L-O vortex keeping the Reynolds number fixed at 100 and varying the CFL number as 1,2,3 and 5. We take shots of some iterations among a set of 100. Main difference between the methods is nicely observed from the colors.

Figures (3) and (4) are related to CFL = 1; (5) and (6) are related to CFL = 2; (7) and (8) are related to CFL = 3; (9) and (10) are related to CFL = 5:

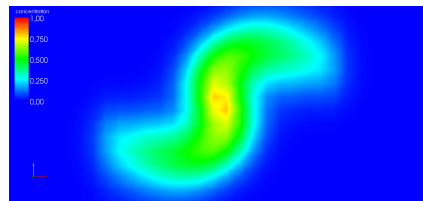


Figure 3.  $CFL = 1$ ,  $Re = 100$ , iteration 34.

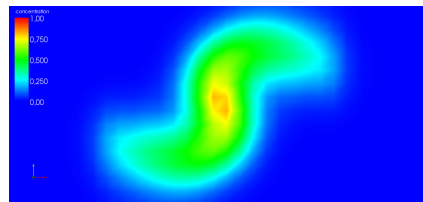


Figure 4.  $CFL = 1$ ,  $Re = 100$ , iteration 34.

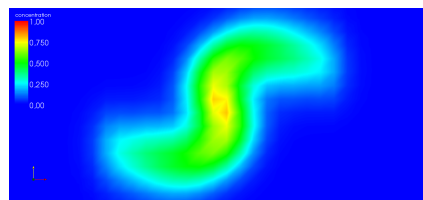


Figure 5.  $CFL = 2$ ,  $Re = 100$ , iteration 16.

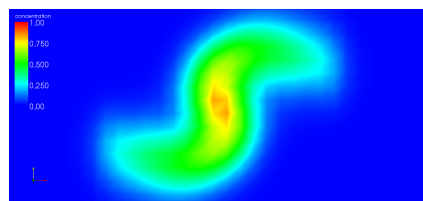


Figure 6.  $CFL = 2$ ,  $Re = 100$ , iteration 16.

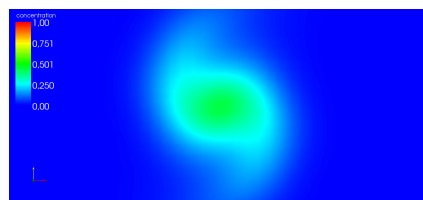


Figure 7.  $CFL = 3$ ,  $Re = 100$ , iteration 47.

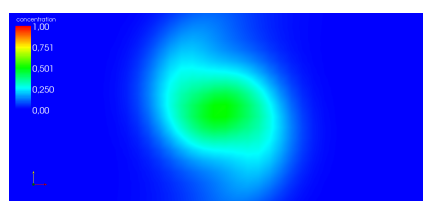


Figure 8.  $CFL = 3$ ,  $Re = 100$ , iteration 47.

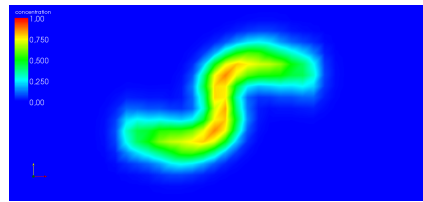


Figure 9.  $CFL = 5$ ,  $Re = 100$ , iteration 4.

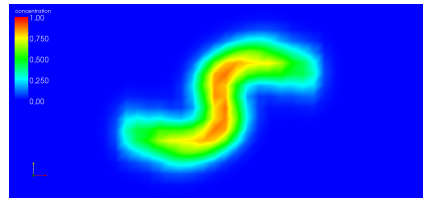


Figure 10.  $CFL = 5$ ,  $Re = 100$ , iteration 4.

## 5. ACKNOWLEDGEMENTS

The authors thank for CNPq-Brazil.

## 6. REFERENCES

- Adams, R.A., 1975. *Sobolev Spaces*. Academic Press, New York.
- Andreev, A.B. and Racheva, M.R., 2010. "A zienkiewicz-type finite element applied to fourth-order problems". *Journal of Computational and Applied Mathematics*, Vol. 235, pp. 348–357.
- Arnold, D.N., Brezzi, F. and Fortin, M., 1984. "A stable finite element for the stokes equations". *Calcolo*, Vol. 21, No. 4, pp. 337–344.
- Bonaventura, L., 1997. "A second order scheme for semi-lagrangian advection with accurate approximation of trajectories". In *Proceedings of the 10th International Conference on Numerical Methods in Laminar and Turbulent Flow*. Swansea, United Kingdom, Vol. 1, pp. 1–12.
- Ciarlet, P.G., 2002. *The Finite Element Method for Elliptic Problems*. SIAM, New York, United States.
- Delbende, I., Gomez, T., Josserand, C. and Nore, C., 2004. "Various aspects of fluid vortices". *Comptes Rendus Mecanique*, Vol. 332, No. 9, pp. 767–781.
- dos Anjos, G.R., 2007. *Hydrodynamics Field Solution of Electrochemical Cells Through Finite Element Method*. Ph.D. thesis, Federal University of Rio de Janeiro, Rio de Janeiro.
- Pironneau, O., 1982. "On the transport-diffusion algorithm and its applications to the navier-stokes equations". *Numerische Mathematik*, Vol. 38, No. 3, pp. 309–332.
- Silva, R.C.C., Loula, A.F. and Guerreiro, J.N.C., 2004. "Local gradient and stress recovery for triangular elements". *Computers & Structures*, Vol. 82, No. 23-26, pp. 2083–2092.
- Smolarkiewicz, P.K. and Pudykiewicz, J.A., 1992. "A class of semi-lagrangian approximations for fluids". *Journal of the Atmospheric Sciences*, Vol. 49, No. 22, pp. 2082–2096.
- Staniforth, A. and Côté, J., 1991. "Semi-lagrangian integration schemes for atmospheric models-a review". *Monthly Weather Review*, Vol. 119, pp. 2206–2223.
- Wang, M. and Xu, J., 2006a. *A New Class of Zienkiewicz-Type Nonconforming Element in Any Dimensions*. University of PennState, Pennsylvania.
- Wang, M. and Xu, J., 2006b. *Some Tetrahedron Nonconforming Elements for Fourth Order Elliptic Equations*. University of PennState, Pennsylvania.
- Xiu, D. and Karniadakis, G.E., 2001. "A semi-lagrangian high-order method for navier-stokes equations". *Journal of Computational Physics*, Vol. 172, No. 2, pp. 658–684.