Asociación Argentina
de Mecánica Computacional

Mecánica Computacional Vol XXIX, págs. 3535-3552 (artículo completo)
Eduardo Dvorkin, Marcela Goldschmit, Mario Storti (Eds.)
Buenos Aires, Argentina, 15-18 Noviembre 2010

IMPROVEMENTS IN NUMERICAL SIMULATIONS OF ISOTACHOPHORESIS BY USING ADAPTIVE MESH REFINEMENT

Pablo A. Kler[a], Gustavo A. Rios Rodríguez[a] and Fabio A. Guarnieri[a,b]

[a]*CIMEC, INTEC (UNL-CONICET), PTLC, El Pozo, 3000, Santa Fe, Argentina.*

[b]*Fac. de Bioingeniería, UNER, 3100, Oro Verde, Argentina.*

Keywords: Microfluidic chips, Isotachophoresis, Adaptive Mesh Refinement, PETSC-FEM.

Abstract. Isotachophoresis (ITP) belongs to a wide group of analytical techniques named as Electrophoretic separations, which also includes capillary electrophoresis, isoelectric focusing and free flow electrophoresis. Electrophoretic separations are based on the dissimilar mobility of ionic species under the action of an external electric field. In particular in ITP assays, the sample is introduced between a fast leading electrolyte and a slow terminating electrolyte. After applying a difference of electric potential, a low electrical field is created in the leading electrolyte and a high electrical field in the terminating electrolyte. The constituents will completely separate from each other, and concentrate at an equilibrium point, surrounded by sharp electrical field differences. These sharpnesses in the electric field give place to spurious oscillations due to the high local values of the Pèclet number. In order to avoid this oscillations, a large amount of nodes in the mesh are required. The additions of points have to be cleverly done in order to avoid a substantial increase in the computational costs. This is achieved by using an h-adaptive mesh refinement (AMR) method. In this work, we present two examples of ITP in microfluidic chips, in order to show the advantages of using AMR in these kind of problems. Examples were solved by using a generalized numerical model of electrophoresis on microfluidic devices previously presented. The model is based on the set of equations that governs electrical phenomena (Poisson equation), fluid dynamics (Navier-Stokes equations), mass transport (Nerst-Planck equation) and chemical reactions. The numerical simulations were carried out by using PETSc-FEM (Portable Extensible Toolkit for Scientific Computation - Finite Elements Method), in a Python environment using high performance parallel computing and solving techniques based on domain decomposition methods.

1 INTRODUCTION

Electrophoretic separation techniques are based on the mobility of ions under the action of an external electric field. These techniques, which are widely used in chemical and biochemical analysis, have been miniaturized in the last 20 years and now represent one of the most important applications of lab-on-a-chip (LOC) technology (Manz et al., 1990; Tian and Finehout, 2008). The LOC's are instruments that integrate one or several laboratory functions on a single chip of only millimeters to a few square centimeters (Wikipedia, 2008). The LOC's have provided a platform to conduct chemical and biochemical analysis in a variety of scientific areas as genetics, molecular biology, pharmacology, environment, among others. Electrophoretic separations carried out by LOC's technology comprise a group of different techniques such as: capillary zone electrophoresis (CZE), isoelectric focusing (IEF) and isotachophoresis (ITP) among others (Landers, 2007). In particular, ITP works using a discontinuous electrical field to create sharp boundaries between the sample constituents. ITP is a powerful sample preconcentration (stacking) technique which is useful in the analysis of low abundance species.

As microchips for electrophoresis are becoming increasingly complex, simulation tools are required to numerically prototype the devices, as well as to control and optimize manipulations (Erickson, 2005). However, there are difficulties due to the several orders of magnitude of the relevant length scales involved: the electric double layer (EDL) thickness (in nm), microchannels width (in μm), and microchannels length (in mm). At the same time, the most challenging and interesting aspect of computational simulation of microfluidic chips is the multiphysics nature, which combines fluidics, mass transport, electromagnetics and chemical reaction kinetics.

The first mathematical model of electrophoresis, including ITP, was developed by Saville and Palusinski (1986). More complex models of conventional electrophoretic separations were later reported (Hruska et al., 2006; Thormann et al., 2007; Bercovici et al., 2009). Numerical simulations aimed to LOC technology involving fluid flow and species transport were early addressed to electrokinetic focusing and sample dispensing techniques, by employing finite volume method in structured grids (Patankar and Hu, 1998; Ermakov et al., 1998, 2000). Due to its non-continuous character, ITP represents an extra challenge from the numerical point of view, and in consequence, a few works in the area were reported, such as Hruska et al. (2006) and Thormann et al. (2007) in 1D domains, and by Shim et al. (2007) in 2D domains. Adaptive mesh refinement (AMR) in numerical simulations of ITP were previously reported by Bercovici et al. (2009) in 1D finite differences.

The need to use AMR in ITP simulations is based in the sharp variations of electric field between the leading electrolyte and sample, between sample and the terminating electrolyte, and also, between sample components, that generates high local values of the Pèclet number. Adaptive mesh refinement has well known benefits when different length scales, great magnitude advective terms and discontinuities are present in the solution field. Among the different mesh adaptation techniques, region subdivisions (h-adaptation), has also shown to be the most widely accepted in the scientific community when solving unsteady problems (Löhner and Baum, 1992; Berzins and Speares, 1997; Usmani, 1999; Remacle et al., 2002; Waltz, 2004; Alauzet et al., 2007). The AMR helps to reduce the numerical instabilities near high gradient regions and better define the aforementioned discontinuities without increasing the computational costs. AMR tool used in this work was previously presented in solving compressible flow problems (Ríos Rodriguez et al., 2005, 2006, 2008, 2009b).

In this paper, a previously presented model for generalized electrophoretic processes in microfluidic chips (Kler et al., 2010) is employed to simulate ITP assays. The use of AMR in

ITP numerical simulations allows a better exploitation of computational resources in model resolution. Two application examples are included in order to show AMR advantages.

2 MODELLING

This section summarizes the mathematical model previously presented (Kler et al., 2010). First the fluid dynamics and the electric field are discussed, and then the mass transport balance for all species considered and the chemistry involved are described.

2.1 Flow field

Fluid velocity u and pressure p are governed by the following equations (Probstein, 2003; Li, 2004):

$$-\nabla \cdot \mathbf{u} = 0, \tag{1}$$

$$\rho(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u}) = \nabla \cdot (-p\mathbf{I} + \mu(\nabla \mathbf{u} + \nabla \mathbf{u}^T)) + \rho \mathbf{g} + \rho_e \mathbf{E}\,. \tag{2}$$

Equation 1 expresses the conservation of mass for incompressible fluids. Equation 2 expresses the conservation of momentum for Newtonian fluids of density ρ, viscosity μ, subjected to gravitational field of acceleration $\mathbf{g}$ and electric field strength $\mathbf{E}$. The last term on the right hand side of Eq. 2 represents the contribution of electrical forces to the momentum balance, where $\rho_e = F \sum_j z_j c_j$ is the electric charge density of the electrolyte solution, obtained as the summation over all type-j ions, with valence z_j and concentration c_j, and F is the Faraday constant. The treatment of this term deserves a complex analysis (see Kler et al. (2010)) due to the strong coupling between the fluid dynamics, the electric field and the concentrations fields. Physicochemical processes related to the coupling are located at the solid-liquid interface were the Electric Double Layer (EDL) forms (Hunter, 2001). One common approximation is the thin EDL approximation. This approximation is employed in this work, and briefly described above.

2.2 Thin EDL approximation

The thickness of the EDL is quantified through Debye length (Probstein, 2003). For the ionic concentrations normally used in practice, this thickness is in the order of $1-10\,nm$, while cross-sectional channel dimensions are in the $10 - 100\ \mu m$ range. Consequently, in most of the flow domain $\rho_e \approx 0$, except in the close vicinity of charged interfaces. When an external electric field $\nabla\phi_a$ is applied tangent to the interface, the electric forces acting on excess ions in the EDL drag the surrounding liquid, and thus EOF develops. For thin EDL in relation to the channel width, the effect is confined to a certain plane parallel to the channel wall, also designated shear plane, where the surface potential is the electrokinetic potential (ζ). Under these conditions, the electro-osmotically driven flow can be regarded as the result of an electrically-induced slip velocity, the magnitude of which is:

$$\mathbf{u}_{EO} = -\frac{\epsilon\zeta}{\mu}\,\nabla\phi_a \tag{3}$$

Further, the last term on the RHS of Eq. (2) vanishes, and the EOF is considered by using Eq. (3) as a boundary value at channel walls.

2.3 Electric Field

The electric potential ϕ is calculated from the charge conservation equation in steady state (Probstein, 2003):

$$\nabla \cdot (-\sigma \nabla \phi - F \sum_{j=1}^{N} z_j D_j \nabla c_j + \rho_e \mathbf{u}) = 0 \tag{4}$$

where D_j is the diffusion coefficient, and σ is the electrical conductivity of the electrolyte solution,

$$\sigma = F^2 \sum_{j=1}^{N} {z_j}^2 \Omega_j c_j \tag{5}$$

where Ω_j is the ionic mobility. In fact, the terms between brackets in Eq. 4 constitute the electric current density i, which accounts for the ion fluxes due to fluid convection, electrical forces, and Brownian diffusion.

2.4 Mass transport and chemistry

The mass transport of weakly concentrated sample ions and buffer electrolyte constituents can be modelled by a linear superposition of migrative, convective and diffusive transport mechanisms, plus a source term due to chemical reactions. In a non-stationary mode, the concentration of each j-type species, is governed by (Probstein, 2003):

$$\frac{\partial c_j}{\partial t} + \nabla \cdot (-z_j \Omega_j \nabla \phi c_j + \mathbf{u} c_j - D_j \nabla c_j) - r_j = 0 \tag{6}$$

were r_j is the reaction term. Different electrolytes (acids, bases and ampholytes), analytes, and particularly the hydrogen ion have to be considered. In electrolyte chemistry the processes of association and dissociation are much faster than the transport electrokinetic processes, hence, it is a good approximation to adopt chemical equilibrium constants to model the reactions of weak electrolytes (Arnaud et al., 2002), while strong electrolytes are considered as completely dissociated.

2.4.1 Acid-base reactions

For the general case, reactions associated to an ampholyte A are

$$\mathrm{AH} \rightleftharpoons \mathrm{A^-} + \mathrm{H^+} \tag{7}$$

$$\mathrm{AH_2^+} \rightleftharpoons \mathrm{AH} + \mathrm{H^+} \tag{8}$$

then the equilibrium state is characterized by,

$$\frac{k_{a2}}{k_{a1}} = \frac{[\mathrm{A^-}][\mathrm{H^+}]}{[\mathrm{AH}]} = K_a \tag{9}$$

$$\frac{k_{b2}}{k_{b1}} = \frac{[\mathrm{AH}][\mathrm{H^+}]}{[\mathrm{AH_2^+}]} = K_b \tag{10}$$

where the square brackets represent concentration (mol/m^3) of the given specie. The corresponding expressions of r_j are obtained as follows,

$$r_{\mathrm{A}^-} = -k_{a1}[\mathrm{A}^-][\mathrm{H}^+] + k_{a2}[\mathrm{AH}] \tag{11}$$
$$r_{\mathrm{AH}} = k_{a1}[\mathrm{A}^-][\mathrm{H}^+] - k_{a2}[\mathrm{AH}] - k_{b1}[\mathrm{AH}][\mathrm{H}^+] + k_{b2}[\mathrm{AH}_2^+] \tag{12}$$
$$r_{\mathrm{AH}_2^+} = k_{b1}[\mathrm{AH}][\mathrm{H}^+] - k_{b2}[\mathrm{AH}_2^+] \tag{13}$$
$$r_{\mathrm{H}^+} = -k_{a1}[\mathrm{A}^-][\mathrm{H}^+] + k_{a2}[\mathrm{AH}] - k_{b1}[\mathrm{AH}][\mathrm{H}^+] + k_{b2}[\mathrm{AH}_2^+] \tag{14}$$

In Eq. 14 the water dissociation term is not included due to the fact that this reaction is several orders of magnitude faster than reactions 7 and 8 (Arnaud et al., 2002), then $[OH^-]$ can be calculated directly as

$$[OH^-] = \frac{K_w}{[\mathrm{H}^+]} \tag{15}$$

were $K_w = 10^{-14}$.

2.4.2 Effective charge and mobility of analytes

When the concentration of analytes is much lower than that of buffer constituents, its effect on the pH is negligible. In these cases, considering all ionic species represents a high computational cost. However the influence of pH on the analytes must be taking into account. Thus the transport equation of these analytes includes $r_j = 0$, and the product $z_j\Omega_j$ as a function of pH. For example, if the specie is an ampholyte that obeys a reaction scheme like the one shown in Eqs. 7 and 8, $z_j\Omega_j$ is included in Eq. 6 as an effective charge-mobility product ($z_{eff(j)}\Omega_{eff(j)}$; Chatterjee (2003)). This product is calculated as $(\alpha_0 - \alpha_2)\Omega_j$, where $\alpha_0 = [A^-]/[A_T]$ and $\alpha_2 = [AH_2^+]/[A_T]$ with $A_T = [AH_2^+] + [AH] + [A^-]$, are the degrees of dissociation of anions and cations, respectively, which are written in terms of $[H^+]$ as,

$$\alpha_0 = \frac{\frac{K_a K_b}{[\mathrm{H}^+]^2}}{1 + \frac{K_b}{[\mathrm{H}^+]} + \frac{K_a K_b}{[\mathrm{H}^+]^2}} \tag{16}$$
$$\alpha_2 = \frac{1}{1 + \frac{K_b}{[\mathrm{H}^+]} + \frac{K_a K_b}{[\mathrm{H}^+]^2}} \tag{17}$$

Therefore the governing equation for the sample plug results,

$$\frac{\partial c_j}{\partial t} + \nabla \cdot [-(\alpha_0 - \alpha_2)\Omega_j \nabla\phi c_j + \mathbf{u}c_j - D_j \nabla c_j] = 0 \tag{18}$$

where it is observed that the physical motion of analytes is coupled to the degree of dissociation at a given pH.

3 MESH ADAPTATION

3.1 Overview

The adaptation strategy used to solve the problem is that introduced in a series of previous works (Ríos Rodriguez et al., 2005, 2006, 2008, 2009b). It is based on an h- refinement / derefinement algorithm that allows to adapt unstructured linear finite element meshes made up of triangles, quadrangles, tetrahedrons and hexahedron. By now the strategy does not support meshes of mixed-type elements nor the unrefinement of the starting mesh (i.e. the mesh used to start the simulation). The procedure generates refined meshes where the elements corresponding to the finer levels are completely nested within the coarser ones, until a maximum level of refinement is reached. Figure (1) shows a detail of a mesh made up of quadrilaterals with four levels of refinement. The boundary conditions for the problem are specified on the starting mesh and are automatically updated by the adaptation procedure. A solution projection is realized to get an initial condition in order to restart calculations.

3.2 Basic requirements, refinement cases and refinement constraints

Since a great number of refinements and derefinements of the mesh are needed in order to "follow" the structures of interest through the solution domain, the adaption algorithm should introduce little memory overhead to the solution procedure and should also be fast. Another desirable feature is to minimize the geometrical quality degradation of the starting mesh. To address these issues, an h-refinement strategy uniquely based on the 1:4 and 1:8 *regular* subdivision patterns for triangles, quadrangles and hexaedra are adopted (figures 2.a, 2.b and 2.c). Because no regular 1:8 subdivision exists for a tetrahedron, refining it by joining the midpoints of its edges and choosing then the shortest diagonal of the inner octaedron (figure 2.d) to refine it and get the remaining tetrahedra has shown, through a series of numerical experiments, a good tradeoff between the required computational effort and the geometrical quality of the resulting tetrahedra (Ríos Rodriguez et al., 2009b).

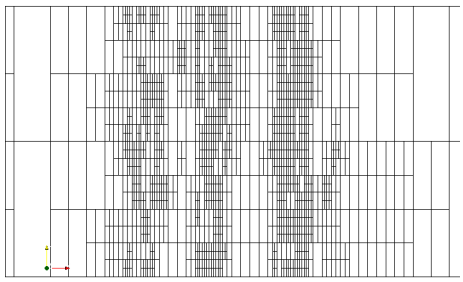


Figure 1: Detail of a refined quadrilateral mesh.

No transition elements are used to match zones with different levels of refinement so that hanging nodes appear and the refined meshes are non-conforming. The well known 1-irregular vertex refinement constraint introduced by (Babuska and Rheinboldt, 1978) and already used by numerous authors (Greaves, 2004; Popinet, 2003; Remacle et al., 2002) is adopted as a smoothing procedure (i.e. to ensure a uniform element size distribution). This rule states that *no more than one hanging node should be shared among neighbour elements through the common edge to which the hanging node belongs* (see fig.(3)). Also, an extension to it for tetrahedra

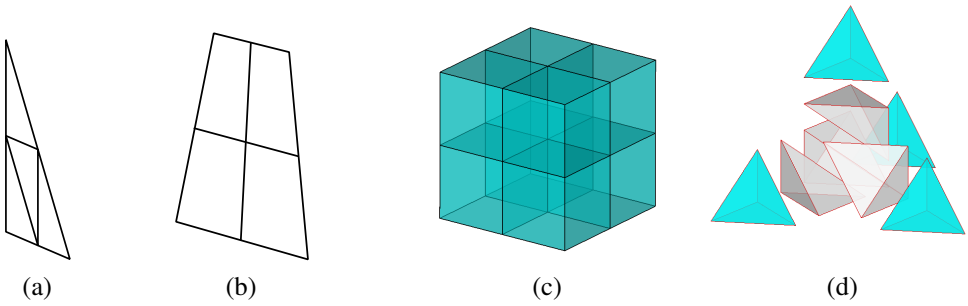


Figure 2: 1:4 and 1:8 subdivision patterns for bi- and three-dimensional elements.

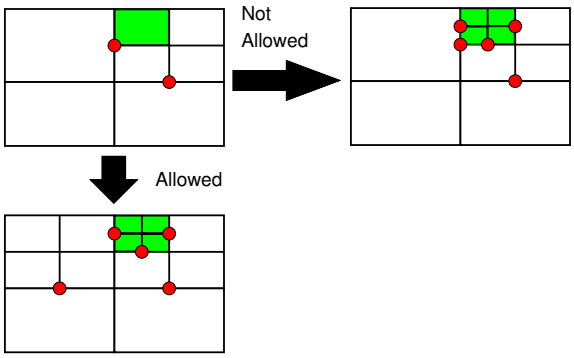


Figure 3: 1-irregular vertex refinement constraint for 2-D meshes.

meshes is presented in (Ríos Rodriguez et al., 2009b). In this manner, the resulting refinement strategy eliminates the effort of managing a great number of transition templates (Staten, 1996; Löhner and Baum, 1992; Remacle et al., 2002) and also keeps bounded the quality degradation of the mesh (Ríos Rodriguez et al., 2009b).

Also, since continuous finite element functions are used in the numerical formulation, constraints for the solution at irregular nodes must be applied. We enforce the solution at these hanging nodes to the average of the corresponding values at the nodes which define the edges or faces to which the hanging nodes belong.

3.3 Selection criterion

The regions of the base mesh that need to be refined are selected based on the gradient's magnitude of the electrical conductivity field σ. All the elements whose gradient magnitude times its size is equal to or greater than a percentage of the maximum corresponding value for all the elements in the mesh are marked to be refined

$$c_1 \leq \frac{\parallel \nabla_i \sigma \parallel \cdot h_i}{\max_{i=1,\ldots,Nel}(\parallel \nabla_i \sigma \parallel \cdot h_i)} \tag{19}$$

where c_1 is a constant set beforehand by the user, h_i is a measure of the element size (the longest edge is chosen in this case), $\parallel \nabla_i \sigma \parallel$ is the magnitude of the σ gradient computed for element-i and Nel is the number of elements in the mesh. The accurate choice of c_1 mostly depends on the user's experience and was set equal to 0.1 for both simulations. This is a classical procedure to tag the cells that need to be refined, commonly used in compressible flow problems (Mavriplis,

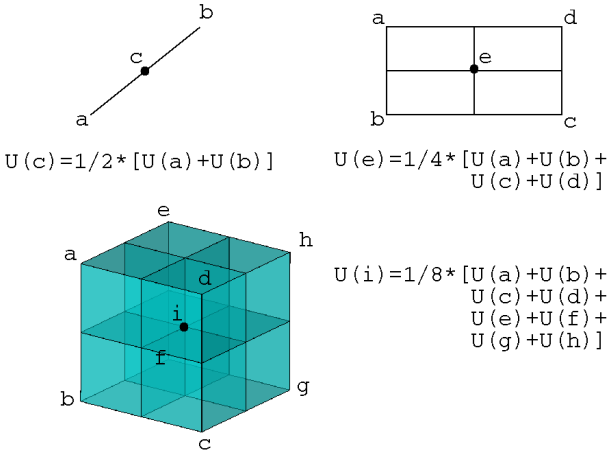


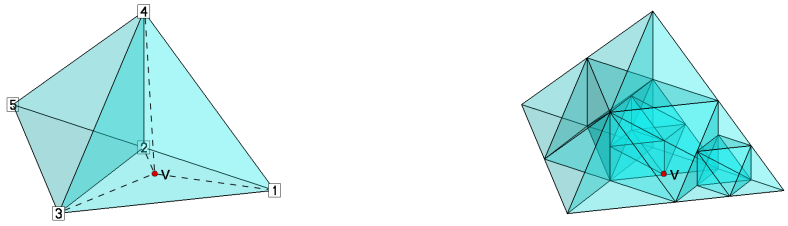
Figure 4: Solution interpolation at the first adaptation step.

1995; Berzins and Speares, 1997; Waltz, 2004; Young and Kwon, 2005).

3.4 Adaptation strategy in unsteady problems

When the base mesh is refined for the first time, the last state computed on it by the solver is linearly interpolated to the new vertices added in the refinement (figure 4). When the maximum level of refinement allowed by the user is attained, this interpolated state is used as the initial condition to resume the solution of the problem equations.

After the solution is advanced $nsteps$ time steps, the selection criterion given by eq.(19) is applied again to the last computed solution and elements are marked to be refined. We assume in this work that all the elements that are not selected for refinement should be unrefined up to the base mesh (coarsest) level. Also, since a maximum level of refinement is imposed, only those elements that do not belong to the maximum level of refinement are finally included in the list of elements to be actually refined. On the other hand, an element is unrefined if only *all* its brothers (7 in three dimensions and 3 in two dimensions) are also marked to be unrefined. If this is so, they are replaced by their *parent* element in the mesh. This *search parent* procedure is recursively applied on the data structure that stores the hierarchical relationship of the elements in the mesh until the coarsest mesh level is attained. It must be taken into account that although some elements are initially marked to be unrefined, the subsequent application of the refinement and unrefinement constraints may not allow it. After the mesh is updated for the current adaptation step, the state needs to be projected again. In this case, and for all the forthcoming adaptation steps, the projection is done as follows: given a vertex V in the currently adapted mesh (adaptation step n) it is required to find the element in the previously adapted mesh (adaptation step $n-1$) that contains it. This is done by building first the list of elements that share vertex V in the current mesh. A recursive search is then applied on this list to find the parents of this elements in the base mesh whereupon they are replaced by their descendants in the previously adapted mesh. These elements are "candidates" for containing V since they are in its neighbourhood. Then, it is possible to find the element that effectively contains vertex V by iterating on this list and computing the volume (3-D) or area (2-D) coordinates $N_j^{(e_i)}$, $j = 1, \ldots, ne$ for each one of the candidates. Here ne is the number of nodes per element and j is the index of the local vertex in the element. If any of the volume / area coordinates $N_j^{(e_i)}$ is less than zero, point V is outside of element e_i and the next element in the list is checked. Only if all the volume coordinates for an element are greater than or equal



(a) Vertex position at adaption step n-1

(b) Vertex position at adaption step n

Figure 5: State projection - Finding vertex position in the previous mesh.

to zero it can be stated that V is inside element e_i or over one of its faces or edges. In this case traversal of the list is interrupted and the state vector is computed as follows

$$\mathbf{U}_V^{(n)} = \sum_{j=1}^{Ne} N_j^{(e_i)}(\mathbf{x}_V) \cdot \mathbf{U}_j^{(n-1)} \tag{20}$$

where $\mathbf{U}_j^{(n-1)}$ is the state vector defined at local vertex j of the element e_i at adaptation step $n-1$, $\mathbf{x}_V$ is the coordinate vector for vertex V and $\mathbf{U}_V^{(n)}$ is the state vector for vertex V at adaptation step n. Dashed lines in figure 5.a) illustrate the volume coordinates at vertex V while figure 5.b) shows the situation for the current adaptation step. If quadrangles or hexaedra are used, they have to be previously splitted into two triangles or five tetrahedra correspondingly. Special care has to be taken in this case for highly deformed elements (Löhner, 2008).

Now, a few words about the frequency for the adaptation of the mesh would like to be mentioned, since this is set constant for the whole simulation. First of all, the time step size Δt for the numerical problem is updated after every mesh adaptation in order to satisfy the Courant condition

$$\Delta t = CFL \min_{j=1,\ldots,Nel}(h_j/|u_j^*|) \tag{21}$$

so the time simulated between two successive adaptations of the mesh is not constant. In eq.(21) CFL is the Courant number, h_j is the length of the shortest edge for element-*j* and u_j^* is the maximum value for the sum of the advection and the maximum migration speeds overall species, for all the vertices in element-*j* (see eq.(6)). Because the most refined regions of the mesh are expected to be at discontinuities, the time step size will be dictated by the size of those elements (i.e. the combination of smaller elements and higher speeds is more probable to happen at discontinuities). This helps to prevent the discontinuities to move outside of the most refined regions until the mesh is adapted again.

The proper choice of the adaptation frequency depends on various factors. Several authors (Waltz, 2004; Ripley et al., 2004; Remacle et al., 2002) and us (Ríos Rodriguez et al., 2009a) have found in practice that the adaptation of the mesh takes just a small fraction of the overall simulation time in 3-D compressible problems (approximately less than 5 per cent). This result induce us to choose a high updating frequency for the mesh for not compromising the overall performance of the adaptive solution procedure. If the time required by the adaptation of the mesh were found to be a greater percentage of the overall simulation time, then a lower updating frequency should be chosen. However, in this latter case a bigger cost would be transferred

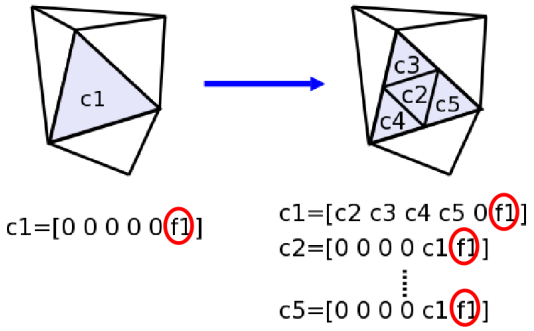


Figure 6: Inheritance of entity properties during refinement.

to the flow computation stage since the refined regions of the mesh would need to be wider to ensure that discontinuities will be kept inside them until the mesh is adapted again. Choosing a higher frequency for adapting the mesh enables to use narrower refined regions around discontinuities and the fluid flow problem is less expensive to solve.

3.5 Management of mesh entities properties

To carry on the numerical simulation with mesh adaptation, it is necessary to properly manage its boundary conditions and other properties associated to the mesh entities (i.e. elements, faces, edges and vertices). This is achieved by assigning a property identifier or flag (illustrated as $f1$ in figure 6) to the entities of the base mesh. During refinement, this flag is inherited from a parent entity to its children by assigning them the same value. The flag is defined by the user and can describe a set of features of different "nature" for an entity (e.g. the identifier assigned to an edge could mean that a fixation for a particular degree of freedom of the problem has to be enforced on that edge and maybe that it also belongs to a curved boundary of the domain). Before starting the solution procedure, the user must supply a list of vertices which define the entities of the mesh that have a particular set of properties. In this way the flag is only assigned to an entity provided certain conditions on that list of vertices are satisfied (e.g. if a set of properties is to be applied to faces then this condition might be that *all* the vertices of a face should be in the list for the identifier to be assigned to that face). After refinement, the entities with the same properties are identified in order to update the boundary conditions supplied to the flow solver.

4 SIMULATION TOOLS

4.1 Software

All numerical simulations presented were performed within a Python programming environment built upon *MPI for Python* (Dalcín et al., 2008), *PETSc for Python* (Dalcín, 2005-2008), and *PETSc-FEM* (Sonzogni et al., 2002). PETSc-FEM is a parallel multiphysics code primarily targeted to 2-D and 3-D finite elements computations on general unstructured grids. PETSc-FEM is based on MPI and PETSc (Balay et al., 2008), it is being developed since 1999 at the *International Center for Numerical Methods in Engineering* (CIMEC), Argentina. PETSc-FEM provides a core library in charge of managing parallel data distribution and assembly of residual vectors and Jacobian matrices, as well as facilities for general tensor algebra computations at the level of problem-specific finite element routines. Additionally, PETSc-FEM provides a suite of specialized application programs built on top of the core library but targeted

to a variety of problems (e.g., compressible/incompressible Navier–Stokes and compressible Euler equations, general advective-diffusive systems, weak/strong fluid-structure interaction). In particular mass transport, chemistry and fluid flow computations presented in this article are carried out within the Navier–Stokes module available in *PETSc-FEM*. This module provides the required capabilities for simulating mass transport and incompressible fluid flow through a monolithic *SUPG/PSPG* (Tezduyar et al., 1992; Tezduyar and Osawa, 2000) stabilized formulation for linear finite elements. Electric Computations are carried out with the Laplace's and the Charge Conservation modules.

4.2 Hardware

Simulations were carried out using a Beowulf cluster *Aquiles* (Storti, 2005-2008). The hardware consists of 82 disk-less single processor computing nodes with Intel Pentium 4 Prescott 3.0GHz 2MB cache processors, Intel Desktop Board D915PGN motherboards, Kingston Value RAM 2GB DDR2 400MHz memory, and 3Com 2000ct Gigabit LAN network cards, interconnected with a 3Com SuperStack 3 Switch 3870 48-ports Gigabit Ethernet.

5 NUMERICAL EXAMPLES

In what follows, two different numerical examples are presented. Both were carried out experimentally in the literature. First a capillary ITP assay (Khurana and Santiago, 2008) where the method is validated in 1D. Then a 2D simulation of a LOC process in which two electrophoretic separations are coupled: ITP + CZE.

5.1 Capillary ITP

In this example a capillary ITP is simulated. Computational domain consist in a $6\ mm$ long capillary in which Acetic acid and 3-Phenilpropionic acid are separated. Also 3 fluorescent markers are added in order to carry out the mobility markers technique developed by Khurana and Santiago (2008). Physicochemical properties of buffer constituents and analytes are listed in Table 1.

Component	pK_a	pK_b	**Mobility** (m^2/Vs)	**Diffusivity** (m^2/s)	**Initial concentration**
Hydrochloric acid	-2.0	–	$7.91\ 10^{-8}$	$2.03\ 10^{-9}$	$5\ mM$
Tetraphenylborate	5.0	–	$1.8\ 10^{-8}$	$4.6\ 10^{-10}$	$5\ mM$
Tris	–	8.076	$2.95\ 10^{-8}$	$7.6\ 10^{-10}$	$75\ mM$
Acetic acid	4.756	–	$4.2\ 10^{-8}$	$1.08\ 10^{-9}$	$380\ \mu M$
3-Phenylpropionic acid	4.664	–	$2.65\ 10^{-8}$	$6.8\ 10^{-10}$	$190\ \mu M$
Oregon Green carboxilic acid	4.7	–	$4.3\ 10^{-8}$	$1.10\ 10^{-9}$	$1.9\ \mu M$
Fluorescein	6.7	–	$3.3\ 10^{-8}$	$4.2\ 10^{-10}$	$1.9\ \mu M$
Bodipy	5.0	–	$2.0\ 10^{-8}$	$5.1\ 10^{-10}$	$1.9\ \mu M$

Table 1: Physicochemical properties of buffer constituents and analytes from Bercovici et al. (2009).

The leading electrolyte (LE) is composed of Tris-HCl and the terminating electrolyte is composed of Tris-Tetraphenylborate. Tris concentration is determined to achieve an initial pH of 9.2. Initial concentrations for sample and buffer constituents are shown in Fig. 7. Conductiv-

ity field is used as reference for refinement process, its aforementioned discontinuities are also shown in Fig. 7. Potentials are applied to obtain a current density around 300 A/m^2.

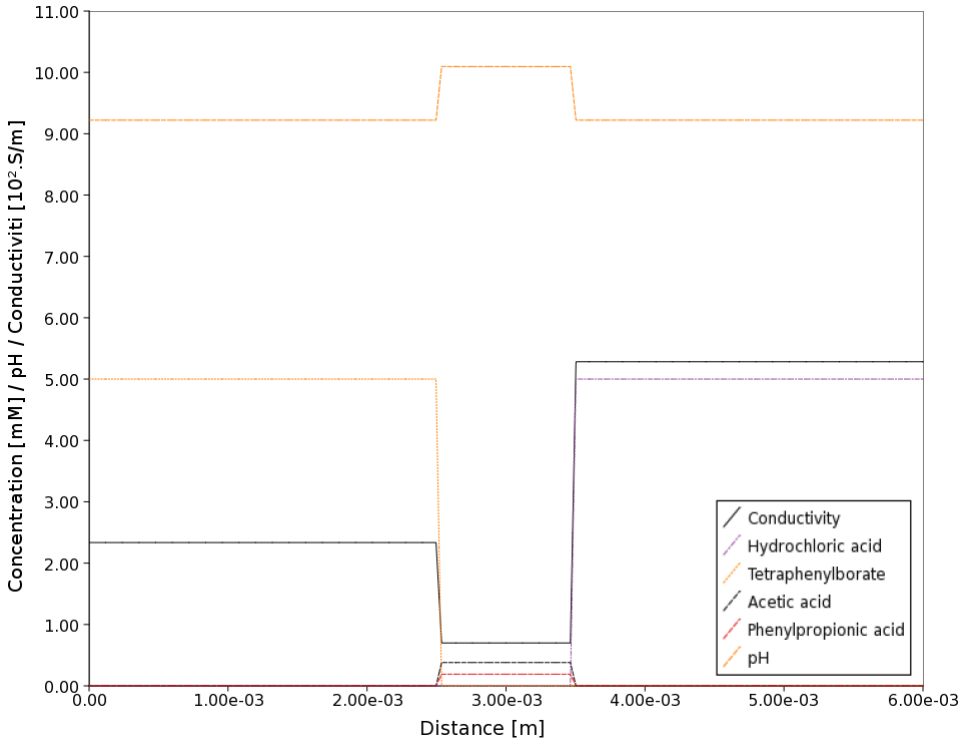


Figure 7: Initial conditions for pH, conductivity, and, sample and buffer constituents.

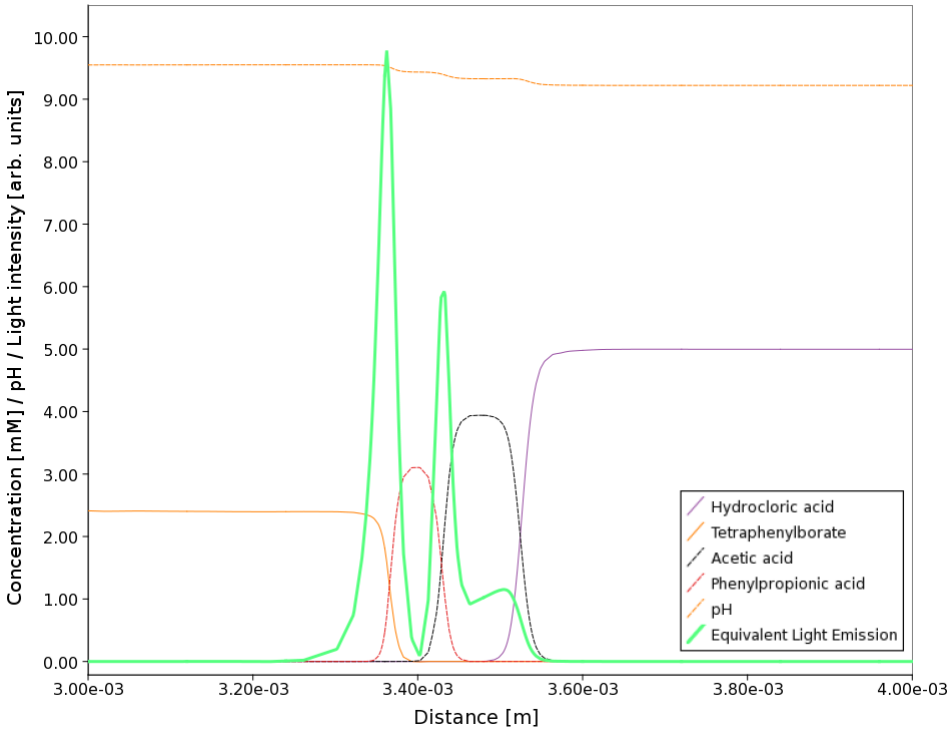
The initial uniform grid consist in 150 nodes. In order to solve correctly this problem with a uniform mesh at least 800 nodes are required. In this solution, the maximum number of points is around 450. Stationary state is achieved after 5 seconds. Separation results as equivalent light intensity, analytes and markers distribution, and conductivity perfectly match with those reported in literature and are shown in Fig.8.

5.2 Two Dimensional Electrophoresis:ITP + CZE

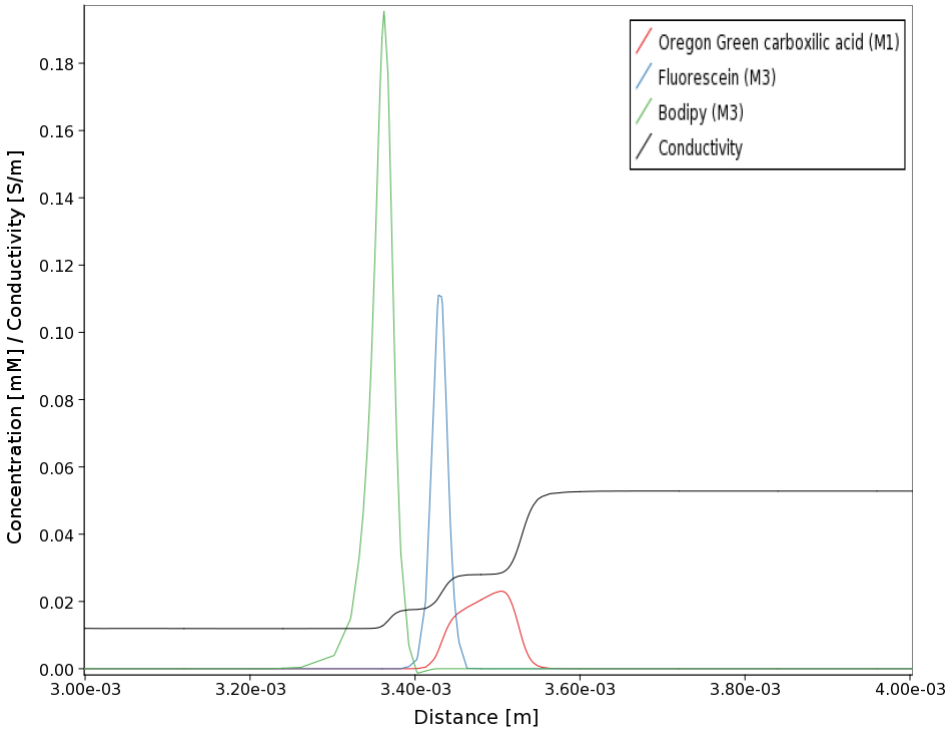
In this example a 2DE experimental assay from the literature is simulated. Two-dimensional electrophoretic (2DE) separations consist of two independent mechanisms that are employed sequentially (O'Farrell, 1975). Each mechanism provides a selective relative displacement between analytes based on different properties of such analytes. The separation efficiency is estimated as the product of the independent efficiency of each method, provided the methods are uncoupled (orthogonality). ITP and CZE satisfy orthogonality and have been extensively employed in the analysis of complex protein samples in conventional devices (Herr et al., 2003).

Particularly, in this case the two mechanisms are ITP and CZE. The chip consist in a cross-shape network ($28 \times 0.2\ mm^2$, for each branch), were in the horizontal axis ITP is carry out, and then in an asynchronous process the CZE is developed in the vertical channel. Geometry, conductivity, and details about the refined mesh are shown in Fig. 9. In this assay three compounds are separated: FITC-Ovoalbumin, FITC-Dextran, and Recombinant green fluorescent protein (GFP). Phosphoric acid is used as leading electrolyte and NaOH as terminating electrolyte.

After focusing, species arrive the intersection, and voltages are switched in order to inject sample into the vertical channel to carry out the CZE. This asynchronous process asserts the



(a) Sample, electrolytes distribution and equivalent light signal.



(b) Markers concentration and conductivity.

Figure 8: Stationary state results for the capillary ITP at t=5s.

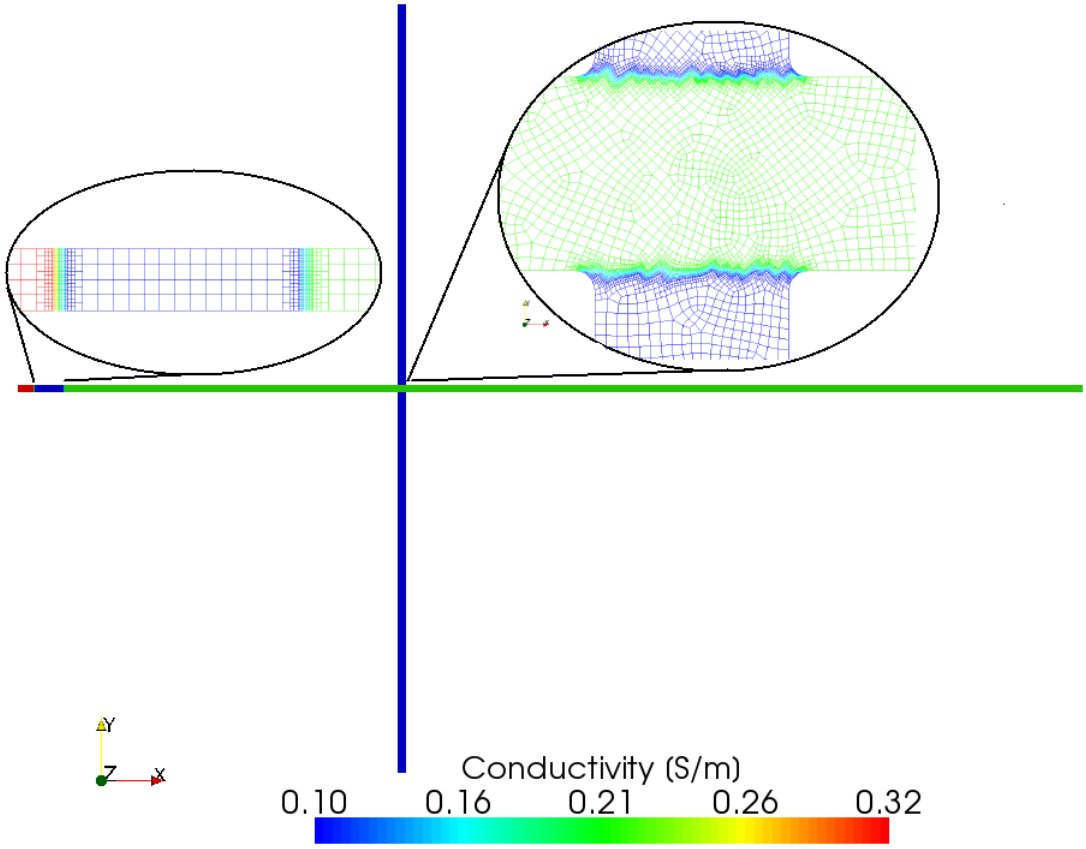


Figure 9: Initial conditions for conductivity and mesh refinement details.

orthogonality of the methods employed. Results of the sample distributions at different values of time are shown in Fig. 10.

6 CONCLUSIONS

In this work, a mathematical model for electrophoretic separations in microfluidic devices previously presented was used, employing AMR, in solving ITP problems. Numerical implementation of the model is carried out by using FEM with high performance parallel computing techniques.

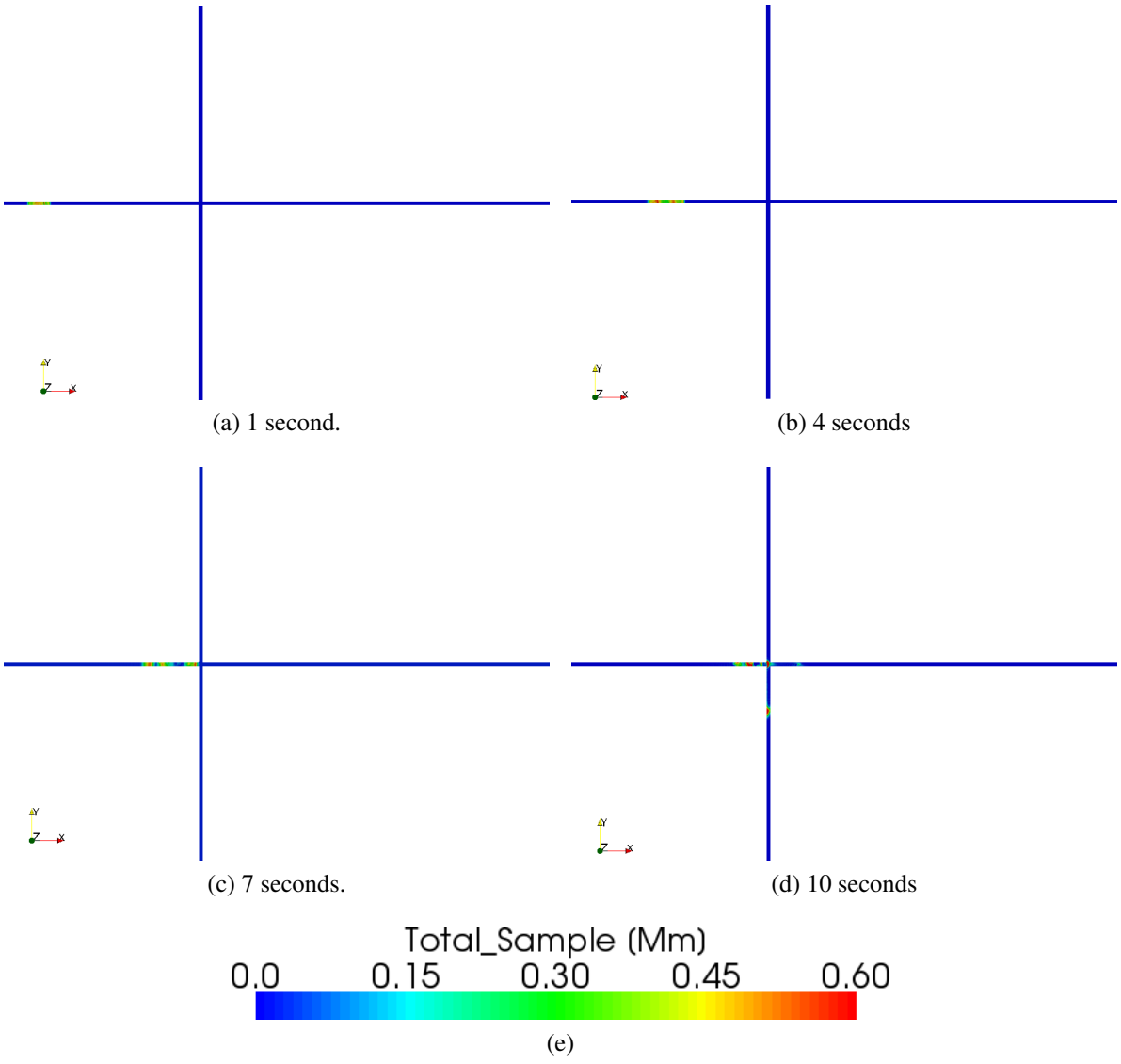
In to show model + AMR capabilities, two numerical experiments were carried out. These experiments were taken from the literature and provide an excellent experimental validation for the method. In both cases results agree with those from experiments.

Combining the model with AMR, provides reduction of computational costs allowing to solve complex problems with high gradients and Pèclet numbers, that are frequently found in ITP and others electrophoretic assays.

Finally, solving a 2DE experiment denotes the capability of the tools presented to test and design state-of-the-art electrophoretic chips.

7 ACKNOWLEDGEMENTS

This work has received financial support from Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET, Argentina), Universidad Nacional del Litoral (UNL, Argentina) and



(a) 1 second.

(b) 4 seconds

(c) 7 seconds.

(d) 10 seconds

(e)

Figure 10: Total sample concentration at different times.

Agencia Nacional de Promoción Científica y Tecnológica (ANPCyT, Argentina). Authors made extensive use of freely distributed software as GNU/Linux OS, MPI, PETSc, GCC compilers, Octave, Paraview, Python, VTK, among many others.

REFERENCES

Alauzet F., Frey P., George P., and Mohammadi B. 3-D transient fixed point mesh adaptation for time-dependent problems: Applications to CFD simulations. *Journal of Computational Physics*, 222:592–623, 2007.

Arnaud I., Josserand J., Rossier J., and Girault H. Finite element simulation of off-gel buffering. *Electrophoresis*, 23:3253–3261, 2002.

Babuska I. and Rheinboldt W. Error estimates for adaptive finite element computations. *SIAM Journal on Numerical Analysis*, 15(4):736–754, 1978.

Balay S., Buschelman K., Gropp W.D., Kaushik D., Knepley M.G., McInnes L.C., Smith B.F., and Zhang H. PETSc Web page. 2008. http://www.mcs.anl.gov/petsc.

Bercovici M., Lele S.K., and Santiago J.G. Open source simulation tool for electrophoretic stacking, focusing, and separation. *J. Chromatogr. A*, 1216:1008 – 1018, 2009.

Berzins M. and Speares W. A 3D Unstructured Mesh Adaption Algorithm for Time Dependent Shock Dominated Problems. *Int. Journal for Num. Methods in Fluids*, 25:81–104, 1997.

Chatterjee A. Generalized numerical formulations for multi-physics microfluidics-type applications. *J. Micromech. Microeng.*, 13:758–767, 2003.

Dalcín L. PETSc for Python. 2005-2008. http://petsc4py.googlecode.com/.

Dalcín L., Paz R., and D'Elia M.S.J. MPI for Python: Performance improvements and MPI-2 extensions. *J. Parallel Distr. Com.*, 68(5):655–662, 2008.

Erickson D. Towards numerical prototyping of labs-on-chip: modeling for integrated microfluidic devices. *Microfluid. Nanofluid.*, 1(4):301–318, 2005.

Ermakov S., Jacobson S., and Ramsey J. Computer simulations of electrokinetic transport in microfabricated channel structures. *Anal. Chem.*, 70(21):4494–4504, 1998.

Ermakov S., Jacobson S., and Ramsey J. Computer simulations of electrokinetic injection techniques in microfluidic devices. *Anal. Chem.*, 72(15):3512–3517, 2000.

Greaves D. A quadtree adaptive method for simulating fluid flows with moving interfaces. *J. Comput. Phys.*, 194(1):35–56, 2004. ISSN 0021-9991. doi:http://dx.doi.org/10.1016/j.jcp.2003.08.018.

Herr A., Molho J., Drouvalakis K., Mikkelsen J., Utz P., Santiago J., and Kenny T. On-chip coupling of isoelectric focusing and free solution electrophoresis for multidimensional separarions. *Anal. Chem*, 75:1180–1187, 2003.

Hruska V., Jaros M., and Gas B. Simul 5 - free dynamic simulator of electrophoresis. *Electrophoresis*, 27:984–991, 2006.

Hunter R. *Foundations of Colloid Science*. Oxford University Press, second edition, 2001.

Khurana T.K. and Santiago J.G. Preconcentration, separation, and indirect detection of nonfluorescent analytes using fluorescent mobility markers. *Anal. Chem.*, 80(1):279–286, 2008.

Kler P., Berli C., and Guarnieri F. Modeling and high performance simulation of electrophoretic techniques in microfluidic chips. *Microfluidics and Nanofluidics*, pages 1–12, 2010. ISSN 1613-4982. 10.1007/s10404-010-0660-x.

Landers J.P. *Handbook of capillary and microchip electrophoresis and associated microtechniques*. CRC Press, third edition, 2007.

Li D. *Electrokinetics in Microfluidics*. Elsevier Academic Press, 2004.

Löhner R. *Applied Computational Fluid Dynamics Techniques. An Introduction Based on Finite*

Element Methods. John Wiley & Sons, second edition edition, 2008.

Löhner R. and Baum J. Adaptive h-refinement on 3-D unstructured grids for transient problems. *Int. Journal for Num. Methods in Fluids*, 14:1407–1419, 1992.

Manz A., Miyahara Y., Miura J., Watanabe Y., Miyagi H., and Sato K. Design of an open-tubular column liquid chromatograph using silicon chip technology. *Sensors and Actuators B: Chemical*, 1(1-6):249 – 255, 1990.

Mavriplis D. Unstructured mesh generation and adaptivity. Report 95-26, ICASE - NASA Langley Research Centre, 1995.

O'Farrell P.H. High resolution two-dimensional electrophoresis of proteins. *J. Biol. Chem.*, 250(10):4007–4021, 1975.

Patankar N. and Hu H. Numerical simulation of electroosmotic flow. *Anal. Chem.*, 70(9):1870–1881, 1998.

Popinet S. Gerris: a tree-based adaptive solver for the incompressible Euler equations in complex geometries. *Journal of Computational Physics*, 190:572–600, 2003.

Probstein R. *Physicochemical Hydrodynamics. An Introduction*. Wiley-Interscience, second edition, 2003.

Remacle J., Li X., Chevaugeon N., and Shephard M. Transient mesh adaptation using conforming and non conforming mesh modifications. In *Proc. 11th Int. Meshing Roundtable, Sandia*. 2002.

Ríos Rodriguez G., López E., Nigro N., and Storti M. Refinamiento adaptativo homogéneo de mallas aplicable a problemas bi- y tridimensionales. 2005.

Ríos Rodriguez G., Storti M., and Nigro N. Refinamiento adaptativo en problemas no estacionarios. 2006.

Ríos Rodriguez G., Storti M., and Nigro N. H-adaptive non-conformal 3-D unstructured mesh strategy for unsteady compressible flows. 2008.

Ríos Rodriguez G., Storti M., and Nigro N. Adaptive refinement of unstructured finite element meshes for compressible flows. 2009a.

Ríos Rodriguez G., Storti M., and Nigro N. An h-adaptive unstructured mesh refinement strategy for unsteady problems. *Latin American Applied Research*, 39:137–143, 2009b.

Ripley R., Lien F., and Yovanovich M. Adaptive unstructured mesh refinement of supersonic channel flows. *Int. Journal of Comp. Fluid Dynamics*, 18:189–198, 2004.

Saville D. and Palusinski O. Theory of electrophoretic separations. part i: Formulation of a mathematical model. *AICHE J.*, 32(2):207–214, 1986.

Shim J., Dutta P., and Ivory C. Finite-volume methods for isotachophoretic separation in microchannels. *Numerical Heat Transfer, Part A: Applications*, 52(5):441–461, 2007.

Sonzogni V.E., Yommi A.M., Nigro N.M., and Storti M.A. A parallel finite element program on a Beowulf cluster. *Adv. Eng. Softw.*, 33(7–10):427–443, 2002.

Staten M. *Selective Refinement of Two and Three-Dimensional Finite Element Meshes.* Master's Thesis, Department of Civil Engineering, Brigham Young University, 1996.

Storti M.A. Aquiles cluster at CIMEC. 2005-2008. http://www.cimec.org.ar/aquiles.

Tezduyar T., Mittal S., Ray S., and Shih R. Incompressible flow computations with stabilized bilinear and linear equal order interpolation velocity pressure elements. *Comput. Method. Appl. M.*, 95:221–242, 1992.

Tezduyar T. and Osawa Y. Finite element stabilization parameters computed from element matrices and vectors. *Comput. Method. Appl. M.*, 190(3-4):411–430, 2000.

Thormann W., Caslavska J., and Mosher R. Modeling of electroosmotic and electrophoretic mobilization in capillary and microchip isoelectric focusing. *J. Chromatogr. A*, 1155(2):154

– 163, 2007.

Tia S. and Herr A. On-chip technologies for multidimensional separations. *Lab Chip*, 9:2524–2536, 2009.

Tian W.C. and Finehout E. *Microfluidics for Biological Applications*. Springer, first edition, 2008.

Usmani A. Solution of steady and transient advection problems using an h-adaptive finite element method. *Int. Journal of Comp. Fluid Dynamics*, 11:249–259, 1999.

Waltz J. Parallel Adaptive Refinement for Unsteady Flow Calculations on 3-D Unstructured Grids. *Int. J. Numer. Meth. Fluids*, 46:37–57, 2004.

Wikipedia t.f.e. Lab-on-a-chip. 2008. http://http://en.wikipedia.org/wiki/Lab_on_a_chip/.

Young M. and Kwon O. A Parallel Unstructured Dynamic Mesh Adaption Algorithm for 3-D Unsteady Flows. *International Journal for Numerical Methods in Fluids*, 48:671–690, 2005.