Available online at www.sciencedirect.com
ScienceDirect
Computers & Fluids 37 (2008) 1126–1137
computers & fluids
www.elsevier.com/locate/compfluid

Parallel asynchronous iterations for the solution of a 3D continuous flow electrophoresis problem ☆

M. Chau [a], P. Spiteri [a,*], R. Guivarch [a], H.C. Boisson [b]

[a] *ENSEEIHT-IRIT/LIMA, 2 rue Camichel, B.P. 7122, F-31071 Toulouse-Cedex, France*
[b] *IMFT, Avenue du Professeur Camille Soula, F-31400 Toulouse-Cedex, France*

Received 25 May 2006; received in revised form 23 March 2007; accepted 28 June 2007
Available online 14 December 2007

Abstract

This paper deals with the parallel numerical solution of a 3D continuous flow electrophoresis problem governed by Navier–Stokes equations coupled with transport and potential equations. For this problem, using the properties of the discrete operators, the convergence of synchronous and asynchronous parallel Schwarz alternating methods is analysed. Finally, parallel solvers are implemented and the results of simulations are given.

1. Introduction

Continuous flow electrophoresis is a process for separating protein mixtures currently used for analysis but difficult to extend to a preparative scale [6–8]. The main difficulties consist in the migration distance at the collection plane and the thinness of the filament occupied by each protein species. Filaments undergo spreading due to a number of different phenomena, among which electrokinetics and electrohydrodynamics are known to be important. In the first of these phenomena, differences in migration velocity between the ionic species give rise to local variations in electrical conductivity near the protein filament. In the second, the local change in electrical conductivity distorts the electrical field, thus including shear stress in the liquid and creating a local flow pattern. More precisely density coupling phenomena involving thermal and solutal effects are very complex due to various mixed physical phenomena. This is why numerical simulations can bring useful information concerning the nature of expected effects.

A physical model was developed [6–8] to describe part of these phenomena when two or several proteins are separated. This model consists of the coupling of three evolutive boundary value problems defined on a bounded domain Ω in the three-dimensional space. Taking into account the classical shape of the electrophoresis chamber, we consider in the sequel that Ω is a parallelepiped. Thus the coupled equations describing the considered physical phenomena are

- the flow described by the Navier–Stokes equations with mixed boundary conditions,
- the transport of protein described by an evolution equation with mixed boundary conditions,
- the electric field described by a potential equation which corresponds to a generalized Laplacian with Dirichlet boundary conditions.

In order to discretize the Navier–Stokes equations, we consider classically four staggered grids. The time dependant equations just described can be discretized by the finite volume method in an implicit approach, using PISO algorithm (i.e. pressure implicit split operator) [18]. PISO is a time marching predictor–corrector algorithm based on the splitting of the solution of velocity equations and pressure equations. In order to compute the protein concentration

☆ Supported by IDRIS (Institut du Développement et des Ressources en Informatique Scientifique).
* Corresponding author.
E-mail address: pierre.spiteri@enseeiht.fr (P. Spiteri).

0045-7930/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compfluid.2007.06.006

and the electric potential, the evolution convection–diffusion equation, and the potential equation are discretized by appropriate finite difference methods using staggered grids of the flow equations. To summarize, seven large scale linear systems have to be solved at each time step. Then, in order to reduce the computational time and to obtain numerical results within reasonable elapsed time, parallelization is necessary.

Parallel computing generally requires synchronization between processors. However, suppressing synchronization is possible in some applications, in order to reduce elapsed time [1]. The aim of the present study is to test asynchronous Schwarz parallel algorithm with flexible communications [12,24] in order to solve the three-dimensional partial differential equations involved in continuous flow electrophoresis and to compare synchronous and asynchronous versions of this algorithm. Furthermore, our goal is to show the improved efficiency linked to the use of asynchronous algorithms. Note that, due to the modelization of the electrophoresis process, the linear systems at each time step are difficult to solve (large scale systems, non-symmetric matrices, ill conditioned systems, etc.).

From the mathematical point of view, we show in the present study, that all the linear systems derived from the discretization of the Navier–Stokes equations, the convection–diffusion equation, and the potential equation have a common property. Indeed we have easily verified that, if the appropriate discretization schemes are selected, the diagonal entries of all the matrices are strictly positive, the off-diagonal entries are non-positive and furthermore that all the matrices are irreducibly diagonal dominant. Thus, all these matrices are M-matrices [27]. According to theoretical results [12–15,24,25], these essential properties ensure the convergence of various variants of parallel Schwarz alternating method with synchronous or asynchronous communications.

This paper is organized as follows: the mathematical model of the physical problem is presented in Section 2. The discretization techniques of the boundary value problems arising in the electrophoresis problem are described in Section 3; then we derive the common property of the matrices arising in the discrete schemes. Due to this property, the convergence of parallel asynchronous Schwarz alternating method with or without flexible communications is studied in Section 4, where some details concerning the implementation are also given. Finally, parallel simulations for the solution of the electrophoresis problem are presented and discussed in Section 5.

2. The physical problem

2.1. The principle of continuous flow electrophoresis

This process takes place in a very thin parallelepipedic cell in which a solution flows slowly through (see Fig. 1). The solution consists of a protein stream to be separated.

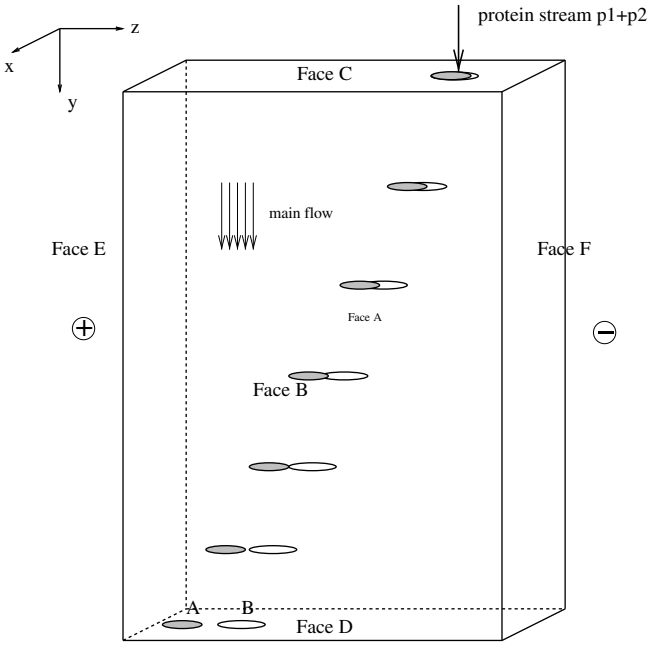

Fig. 1. The principle of continuous flow electrophoresis.

It is injected into a carrier flow by the face C of the cell with the shape of a sharp liquid filament. An electrical field is created through the cell by two electrodes located on both sides of the cell, on the faces E and F.

Due to the effect of the electrical field, they migrate with speed $\vec{W} = \vec{V} + \mu\vec{E}$, where $\vec{V}$ is the carrier flow velocity and $\mu\vec{E}$ is the migration velocity. The various species of protein having different electrical mobilities, they can be collected separately on the face D.

In the sequel, the flow is assumed to be isothermal and without chemical reaction; consequently the various physical coefficients arising in the phenomenon are constant.

The physical phenomena related to the present study concern the following:

- the main flow of the fluid in the three-dimensional space, which is perturbed by the electrohydrodynamics effects,
- the transport and the migration of the proteins,
- the electrokinetic effect, connected to the spatial changes of the conductivity due to the concentration of the various ionic species.

2.2. The physical problem

A physical model was developed by Clifton et al. [6–8]. Note that, in order to simplify the presentation, all variables, parameters, and equations given in the sequel are non-dimensional. In the relations governing the electrophoresis flow, the problem can be stated as follows: compute in the bounded domain Ω in the three-dimensional space the following:

- the velocity field $\vec{V} = (v_1, v_2, v_3)$,
- the pressure p,

- the electrical field $\vec{E} = (E_1, E_2, E_3)$,
- for each protein m, its concentration c_m,
- and the electrical potential Φ.

The physical parameters arising in the mathematical model are the following:

- the Reynolds number Re,
- the dielectric permittivity ϵ,
- the Peclet number associated to the transport of the protein m: Pe_m
- the electrical conductivity K,
- and the electrophoretic mobility of the protein m: μ_m.

In the studied problem we consider the flow of an incompressible viscous fluid in the domain Ω. The main flow is described by the 3D Navier–Stokes equation taking account the external strength field:

$$\frac{\partial v_i}{\partial t} - \frac{1}{Re}\Delta v_i + \sum_{j=1}^{3} v_j \frac{\partial v_i}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \epsilon \operatorname{div}(E_i \vec{E}), \quad i = 1, 2, 3, \tag{1}$$

$$\operatorname{div}(\vec{V}) = 0, \tag{2}$$

where

$$\operatorname{div}(E_i \vec{E}) = \sum_{j=1}^{3} \frac{\partial}{\partial x_j} E_i E_j.$$

The transport equation for a protein m is modelled by the following evolution convection–diffusion equation:

$$\frac{\partial c_m}{\partial t} - \frac{1}{Pe_m}\Delta c_m + \sum_{i=1}^{3} (v_i + \mu_m E_i) \frac{\partial c_m}{\partial x_i} = 0. \tag{3}$$

The potential Φ is governed by a generalized Poisson equation:

$$-\operatorname{div}(K \operatorname{grad} \Phi) = 0, \tag{4}$$

which can also be written as

$$-\sum_{j=1}^{3} \frac{\partial}{\partial x_j}\left(K \frac{\partial \Phi}{\partial x_j}\right) = 0.$$

The conductivity K is calculated using the concentrations of the n_p proteins, by the relation $K = K_0 + \sum_{m=1}^{n_p} \lambda_m c_m$, where for all m, $\lambda_m > 0$ and $\sum_{m=1}^{n_p} \lambda_m = 1$.

In the sequel, we consider the migration of only one protein. In this case, the expressions of concentration, conductivity and electrophoretic mobility become

$$c_m = c; \quad K = K_0 + c; \quad \mu_m = \mu, \tag{5}$$

where c is the concentration of the considered protein and μ its electrophoretic mobility.

Eq. (1) governing the flow is coupled with the potential equation (4) by the relation

$$\vec{E} = -\vec{\operatorname{grad}}(\Phi). \tag{6}$$

The partial differential equations (1)–(4) must be completed by the introduction of boundary values induced by physical considerations.

The fluid comes into the cell in the upper face C and comes out through the lower face D. Thus, we consider that the velocity fulfils non-homogeneous Dirichlet boundary conditions on the face C and homogeneous Neumann boundary conditions on the face D: $v_{1/C} = v_{3/C} = 0$, $v_{2/C} = v^0$; $\frac{\partial v_i}{\partial n/D} = 0$ for $i = 1, 2, 3$. Furthermore, the velocities v_1 and v_3 are zero on the other four faces; and therefore they fulfil homogeneous Dirichlet boundary conditions on the faces A, B, E and F: $v_{1/A} = v_{3/A} = 0$; $v_{1/B} = v_{3/B} = 0$; $v_{1/E} = v_{3/E} = 0$; $v_{1/F} = v_{3/F} = 0$. The axial velocity v_2 is zero on the faces A and B: $v_{2/A} = v_{2/B} = 0$, and on the faces E and F, it fulfils homogeneous Neumann boundary conditions: $\frac{\partial v_2}{\partial n/E} = \frac{\partial v_2}{\partial n/F} = 0$.

The proteins are injected into the cell through the face C; thus, on this upper face the concentration is known and it fulfils non-homogeneous Dirichlet boundary conditions: $c_{/C} = c^0$. Furthermore, the concentration is free on the other five faces of the cell; so we can consider that on these five faces the concentration fulfils homogeneous Neumann boundary conditions: $\frac{\partial c}{\partial n/A} = \frac{\partial c}{\partial n/B} = \frac{\partial c}{\partial n/E} = \frac{\partial c}{\partial n/F} = \frac{\partial c}{\partial n/D} = 0$.

Remark 1. Taking into account the shape of the cell, we assume that the proteins do not reach the electrodes on the faces E and F. Consequently, we could have considered that the concentration fulfils homogeneous Dirichlet boundary conditions on the faces E and F.

The potential fulfils Dirichlet boundary conditions on every face. As the potential is known and constant at every point of the electrodes, i.e. on the two lateral faces E and F: $\Phi_{/E} = \Phi^0$; $\Phi_{/F} = 0$. Furthermore, the boundary conditions on faces A and B are obtained by a linear interpolation between the values of the potential defined on the electrodes: $\Phi_{/A} = \Phi_{/B} = \frac{x_1}{L}\Phi^0$, where L is the width of the cell. On the horizontal faces C and D, the boundary conditions are obtained by the solution of the potential equation restricted to each upper and lower face; these boundary conditions are preliminary computed. As the concentration on the face C is constant, the potential on this face is computed only once: $\Phi_{/C} = \Phi_C$. On the other extremity D, the concentration changes with time. Then, at each time step the potential must be computed on this face in order to obtain the boundary condition: $\Phi_{/D} = \Phi_D(t)$.

3. Discretization of the partial differential equations

In order to solve numerically the coupled boundary value differential equations and particularly to ensure the convergence of the parallel iterative scheme, we consider in the sequel various well-known and well-suited discretization techniques. In order to analyse the behaviour of the numerical algorithms used in the solution of the large algebraic systems derived from the discretization, we also establish useful properties verified by the discrete operators.

3.1. Discretization of the Navier–Stokes equations

The Navier–Stokes equations are solved by the PISO algorithm (pressure implicit with split of operators) introduced by Issa [18] coupled with a discretization by the standard finite volume method [28]. In the case of incompressible flow, we recall that the PISO method is an implicit algorithm corresponding to a time marching predictor–corrector method based upon the splitting of the solution of velocity equations and pressure equations. This principle allows us to deal with the coupling of the variables $(\vec{V}, p)$ by dividing each time step into three substeps. The consistency and stability of this method were studied in [18]. It was shown that two corrector steps are sufficient to obtain a suitable accuracy compatible with the discretization scheme and round-of error propagation. Furthermore, this time marching scheme, based on Euler's method, is unconditionally stable.

The predictor step: This step computes implicitly the predicted velocity $\vec{V}^{\left(n+\frac{1}{3}\right)}$ by solving the momentum equation (1) when the pressure is fixed at its value obtained in the previous time step $p^{(n)}$. Then, the velocity components are computed as the solution of three systems derived from the discretization of (1) by the finite volume method applied on the staggered grids:

$$A_i \cdot v_i^{\left(n+\frac{1}{3}\right)} = -\Delta_i \cdot p^{(n)} + S_i(\vec{E}, v_i^{(n)}), \quad i = 1,2,3, \tag{7}$$

where Δ_i classically denotes the approximation of the space derivative $\frac{\partial}{\partial x_i}$ $(i = 1,2,3)$, S_i is a term that depends on $\vec{E}$ and on the velocity at the previous time step $\vec{V}^{(n)}$. Note that the right-hand side of Eq. (7) takes into account the integration over the control volumes.

Remark 2. The predictor step requires the solution of three uncoupled algebraic linear systems. Each system defines a component of the velocity; so these systems can be solved independently in parallel.

First corrector step: This first explicit step computes the velocity field $\vec{V}^{\left(n+\frac{2}{3}\right)}$ associated with the pressure $p^{\left(n+\frac{1}{2}\right)}$, verifying the discrete approximate Navier–Stokes equations and the discrete continuity equation. This step is based on the following approximation:

$$A_i.v_i^{\left(n+\frac{2}{3}\right)} \simeq D_i.v_i^{\left(n+\frac{2}{3}\right)} - H_i.v_i^{\left(n+\frac{1}{3}\right)}, \tag{8}$$

where D_i and $-H_i$ are, respectively, the diagonal entries and the off-diagonal entries of the matrix A_i.

According to [18], the first corrector step requires the following computations:

(a) solve the linear system

$$-\sum_{i=1}^{3} \Delta_i \cdot D_i^{-1} \cdot \Delta_i \cdot p^c = -\sum_{i=1}^{3} \Delta_i v_i^{\left(n+\frac{1}{3}\right)}, \tag{9}$$

where $p^c = p^{\left(n+\frac{1}{2}\right)} - p^{(n)}$;

(b) compute explicitly

$$v_i^{\left(n+\frac{2}{3}\right)} = v_i^{\left(n+\frac{1}{3}\right)} - D_i^{-1} \cdot \Delta_i \cdot p^c, \quad i = 1,2,3. \tag{10}$$

Second corrector step: This second explicit step computes the velocity field $\vec{V}^{(n+1)}$ associated with the pressure $p^{(n+1)}$, starting from the fields $\vec{V}^{\left(n+\frac{2}{3}\right)}$ and $p^{\left(n+\frac{1}{2}\right)}$. This step improves the velocities and the pressure obtained in the previous step. It is based on the same kind of approximation as (8):

$$A_i \cdot v_i^{(n+1)} \simeq D_i \cdot v_i^{(n+1)} - H_i \cdot v_i^{\left(n+\frac{2}{3}\right)}. \tag{11}$$

According to [18], the second corrector step requires the following computations:

(a) solve the linear system

$$-\sum_{i=1}^{3} \Delta_i \cdot D_i^{-1} \cdot \Delta_i \cdot p^{cc} = -\sum_{i=1}^{3} \Delta_i \cdot D_i^{-1} \cdot H_i \cdot \left(v_i^{\left(n+\frac{2}{3}\right)} - v_i^{\left(n+\frac{1}{3}\right)}\right), \tag{12}$$

where $p^{cc} = p^{(n+1)} - p^{\left(n+\frac{1}{2}\right)}$;

(b) compute explicitly

$$v_i^{(n+1)} = v_i^{\left(n+\frac{2}{3}\right)} + D_i^{-1} \cdot \left(H_i \cdot \left(v_i^{\left(n+\frac{2}{3}\right)} - v_i^{\left(n+\frac{1}{3}\right)}\right) - \Delta_i \cdot p^{cc}\right), \quad i = 1,2,3. \tag{13}$$

Remark 3. Note that the matrices of both corrector steps are the same and that the corrector equations (9) and (12) only differ by their right-hand sides.

3.1.1. Discretization of the flow equations for the predictor step

For the predictor step, the complete discretization of the Navier–Stokes equation leads to three uncoupled algebraic linear systems $A_i \cdot v_i = b_i$, $i = 1,2,3$, obtained from the momentum equations. Note that the three matrices are obtained using classical finite volume discretization scheme [28]. Fig. 2 shows the notation that we have adopted. The discretized equations are written on the following form, for a given discretization point M of the ith component of velocity field $(1 \leqslant i \leqslant 3)$:

$$a_{i,M}v_{i,M} + a_{i,B}v_{i,B} + a_{i,S}v_{i,S} + a_{i,W}v_{i,W} + a_{i,E}v_{i,E} + a_{i,N}v_{i,N} + a_{i,T}v_{i,T} = b_{i,M}, \tag{14}$$

where $a_{i,M}$ is a diagonal entry of A_i and $(a_{i,J})_{J \neq M}$ are the off-diagonal entries. The dimensions of a control volume are denoted δx_1, δx_2 and δx_3. The distance between two mesh points is denoted hx_1, hx_2 and hx_3. For the sake of simplicity, only uniform discretization steps are considered in each direction e_1, e_2 and e_3.

According to [28], the expression of the discretization matrix A_i is

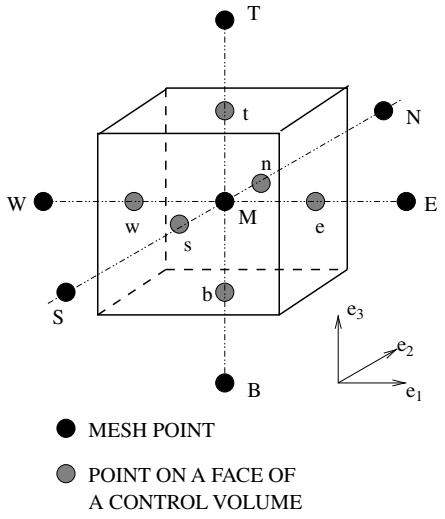

Fig. 2. Mesh points around a control volume.

$$\begin{cases} a_{i,B} = -\delta x_1 \delta x_2 \left(\mathscr{A}\left(\left| \frac{f_{i,b}}{v_{i,b}} \right| \right) v_{i,b} + \max(f_{i,b}, 0) \right), \\ a_{i,S} = -\delta x_1 \delta x_3 \left(\mathscr{A}\left(\left| \frac{f_{i,s}}{v_{i,s}} \right| \right) v_{i,s} + \max(f_{i,s}, 0) \right), \\ a_{i,W} = -\delta x_2 \delta x_3 \left(\mathscr{A}\left(\left| \frac{f_{i,w}}{v_{i,w}} \right| \right) v_{i,w} + \max(f_{i,w}, 0) \right), \\ a_{i,E} = -\delta x_2 \delta x_3 \left(\mathscr{A}\left(\left| \frac{f_{i,e}}{v_{i,e}} \right| \right) v_{i,e} + \max(-f_{i,e}, 0) \right), \\ a_{i,N} = -\delta x_1 \delta x_3 \left(\mathscr{A}\left(\left| \frac{f_{i,n}}{v_{i,n}} \right| \right) v_{i,n} + \max(-f_{i,n}, 0) \right), \\ a_{i,T} = -\delta x_1 \delta x_2 \left(\mathscr{A}\left(\left| \frac{f_{i,t}}{v_{i,t}} \right| \right) v_{i,t} + \max(-f_{i,t}, 0) \right), \\ a_{i,M} = \frac{\delta x_1 \delta x_2 \delta x_3}{\delta t} - \sum_{J \neq M} a_{i,J} \end{cases} \tag{15}$$

with

$$\begin{cases} f_{i,b} = v_{3,b}^{(n)}; \quad v_{i,b} = \frac{1}{Re \times hx_3}, \\ f_{i,s} = v_{2,s}^{(n)}; \quad v_{i,s} = \frac{1}{Re \times hx_2}, \\ f_{i,w} = v_{1,w}^{(n)}; \quad v_{i,w} = \frac{1}{Re \times hx_1}, \\ f_{i,e} = v_{1,e}^{(n)}; \quad v_{i,e} = \frac{1}{Re \times hx_1}, \\ f_{i,n} = v_{2,n}^{(n)}; \quad v_{i,n} = \frac{1}{Re \times hx_2}, \\ f_{i,t} = v_{3,t}^{(n)}; \quad v_{i,t} = \frac{1}{Re \times hx_3}. \end{cases} \tag{16}$$

In the previous relations the mapping $\mathscr{A}$ is defined in the Table 1 in order to define many discretization schemes according to [28].

The right-hand sides b_i are

$$b_{1,M} = \delta x_1 \delta x_2 \delta x_3 \left(\frac{1}{\delta t} v_{1,M}^{(n)} + \mathscr{F}_{1,M} \right) - \delta x_2 \delta x_3 (p_e^{(n)} - p_w^{(n)}) \tag{17}$$

$$b_{2,M} = \delta x_1 \delta x_2 \delta x_3 \left(\frac{1}{\delta t} v_{2,M}^{(n)} + \mathscr{F}_{2,M} \right) - \delta x_1 \delta x_3 (p_n^{(n)} - p_s^{(n)}) \tag{18}$$

Table 1
$\mathscr{A}(|\mathscr{P}|)$ with respect to discretization schemes

Scheme	Expression of $\mathscr{A}(\lvert\mathscr{P}\rvert)$
Centered differences	$1 - 0.5\lvert\mathscr{P}\rvert$
Upwind	1
Hybrid	$\max(0, 1 - 0.5\lvert\mathscr{P}\rvert)$
Power law	$\max(0, (1 - 0.5\lvert\mathscr{P}\rvert)^5)$
Exponential	$\lvert\mathscr{P}\rvert/(\exp(\lvert\mathscr{P}\rvert) - 1)$

$$b_{3,M} = \delta x_1 \delta x_2 \delta x_3 \left(\frac{1}{\delta t} v_{3,M}^{(n)} + \mathscr{F}_{3,M} \right) - \delta x_1 \delta x_2 (p_t^{(n)} - p_b^{(n)}) \tag{19}$$

where $\mathscr{F}$ denotes the external forces. Boundary conditions are discretized with standard techniques [28].

Then, concerning the predictor step and for all discretization schemes defined in Table 1, we have the following original results.

Proposition 1. *The matrices $(A_i)_{i=1,2,3}$ are diagonally dominant.*

Proof. Results directly from the values of the entries of the three matrices and from the standard treatment of boundary conditions. □

Proposition 2. *$(A_i)_{i=1,2,3}$ are M-matrices for all discretization schemes of Table 1, except for the central difference scheme. For the central difference scheme, if $|\mathscr{P}_i| \leqslant 2$, $(A_i)_{i=1,2,3}$ are also M-matrices.*

Proof. Indeed the diagonal entries of the matrices $(A_i)_{i=1,2,3}$ are strictly positive and their off-diagonal entries are non-positive. Furthermore the matrices $(A_i)_{i=1,2,3}$ are obviously irreducible [27]; since they are diagonally dominant, then the proof is achieved. □

3.1.2. Discretization of the equations for the corrector steps

For the two corrector steps the PISO method leads to the following systems to solve:

$$C \cdot p^c = b^c,$$
$$C \cdot p^{cc} = b^{cc}.$$

The matrix C is obtained by discretizing the conservation equation (2) on the mesh associated with the pressure. The discrete equations are written in the following

$$c_M p_M^\star + c_B p_B^\star + c_S p_S^\star + c_W p_W^\star + c_E p_E^\star + c_N p_N^\star + c_T p_T^\star = b_M^\star, \tag{20}$$

where the exponent ★ is equal to c or cc for each corrector steps. For each corrector step, the expressions (10) and (13) are substituted into the velocities in the discretized mass conservation equation in order to obtain algebraic equations where the unknowns are the pressures [28]. The following matrix is obtained

$$\begin{cases} c_B = -\delta x_1\delta x_2 \frac{1}{D_{3,b}} \delta x_1\delta x_2, \\ c_S = -\delta x_1\delta x_3 \frac{1}{D_{2,s}} \delta x_1\delta x_3, \\ c_W = -\delta x_2\delta x_3 \frac{1}{D_{1,w}} \delta x_2\delta x_3, \\ c_E = -\delta x_2\delta x_3 \frac{1}{D_{1,e}} \delta x_2\delta x_3, \\ c_N = -\delta x_1\delta x_3 \frac{1}{D_{2,n}} \delta x_1\delta x_3, \\ c_T = -\delta x_1\delta x_2 \frac{1}{D_{3,t}} \delta x_1\delta x_2, \\ c_M = -\sum\limits_{J\neq M} c_J, \end{cases} \tag{21}$$

where $D_{i,j}$ denotes the jth entry of the diagonal matrix D_i. The right-hand sides b^c and b^{cc} are obtained by integrating the right-hand sides of Eqs. (9) and (12) over the control volumes. Finally, the matrix C arising in the corrector steps has the same properties as the matrices arising in the predictor step.

On the boundaries, the pressures p^c and p^{cc} are defined as follows: on the face D, $p^c_{/D} = 0$ and $p^{cc}_{/D} = 0$; on the other faces, $\frac{\partial p^c}{\partial n}_{/X} = \frac{\partial p^{cc}}{\partial n}_{/X} = 0$, where $X = A, B, C, E, F$. Then we have the following original result

Proposition 3. *The matrix C is diagonally dominant.*

Proposition 4. *The matrix C is an M-matrix.*

The proofs of the previous propositions result from analogous considerations than those considered in the predictor step.

3.2. Discretization of the transport equation for proteins

The transport equation (3) is in fact a convection–diffusion equation. We consider a finite difference discretization of this equation. The diffusion term is discretized using the classical seven points discretization scheme. The convection terms can be discretized using either the central difference discretization scheme or either the upwind finite difference scheme. For theoretical convenience and in order to satisfy at each time step the convergence of the iterative algorithm (see Section 4), we consider only upwind finite difference scheme. According to the sign of the components $(v_i)_{i=1,2,3}$, a backward or forward scheme is used to discretize the convection terms. For example, if we consider the term $v_1 \frac{\partial c}{\partial x_1}$ of Eq. (3), the discretization is as follows

$$v_1 \frac{\partial c}{\partial x_1} = \begin{cases} v_1 \frac{c(x_1,x_2,x_3) - c(x_1-h_1,x_2,x_3)}{h_1} + \mathcal{O}(h_1), & \text{if } v_1 > 0, \\ v_1 \frac{c(x_1+h_1,x_2,x_3) - c(x_1,x_2,x_3)}{h_1} + \mathcal{O}(h_1), & \text{if } v_1 < 0. \end{cases}$$

Let us denote by T the associated discretization matrix. We obtain the following result:

Proposition 5. *The matrix T is an M-matrix.*

Remark 4. If we consider a central difference discretization scheme for the convection term, then, if the magnitude of the components of the velocity are small enough, then the matrix T is an M-matrix. More precisely, in the case where the discretization step h is uniform ($hx_1 = hx_2 = hx_3 = h$), if

$$|v_i| < \frac{2}{Pe_m h}, \quad i = 1, 2, 3$$

then, the matrix T is an M-matrix [16]. Nevertheless, from a practical point of view, these conditions are not interesting for the considered electrophoresis problem, because the components of the velocity form part of the unknowns.

3.3. Discretization of the potential equation

For the potential equation, we consider a finite difference discretization. The numerical scheme is the same for every term of the generalized Poisson equation (4). This scheme is obtained by taking the mean of two intermediate schemes. For example, let us consider the discretization of $-\frac{\partial}{\partial x_1}(K\frac{\partial \Phi}{\partial x_1})$ for x_2 and x_3 fixed, in a case where the discretization step is irregular; hx_{1-} and hx_{1+} denote, respectively, the distance between the points M and W, and the points M and E (see Fig. 2). Let us firstly consider the following two schemes:

- forward–backward scheme

$$-\frac{\partial}{\partial x_1}\left(K\frac{\partial \Phi}{\partial x_1}\right) = \frac{1}{hx_{1+}}\left(K_E\frac{\Phi_E - \Phi_M}{hx_{1+}} - K_M\frac{\Phi_M - \Phi_W}{hx_{1-}}\right),$$

- backward–forward scheme

$$-\frac{\partial}{\partial x_1}\left(K\frac{\partial \Phi}{\partial x_1}\right) = \frac{1}{hx_{1-}}\left(K_M\frac{\Phi_E - \Phi_M}{hx_{1+}} - K_W\frac{\Phi_M - \Phi_W}{hx_{1-}}\right).$$

Then, the final discretization scheme is obtained by taking the mean of both schemes. For the sake of simplicity, we consider in the sequel regular discretization steps in each direction x_1, x_2 and x_3.

Finally, the discretization matrix Π of the potential equation is

$$\begin{cases} \pi_B = -\frac{1}{2}\left(K_B\frac{1}{(hx_3)^2} + K_M\frac{1}{(hx_3)^2}\right), \\ \pi_S = -\frac{1}{2}\left(K_S\frac{1}{(hx_2)^2} + K_M\frac{1}{(hx_2)^2}\right), \\ \pi_W = -\frac{1}{2}\left(K_W\frac{1}{(hx_1)^2} + K_M\frac{1}{(hx_1)^2}\right), \\ \pi_E = -\frac{1}{2}\left(K_E\frac{1}{(hx_1)^2} + K_M\frac{1}{(hx_1)^2}\right), \\ \pi_N = -\frac{1}{2}\left(K_N\frac{1}{(hx_2)^2} + K_M\frac{1}{(hx_2)^2}\right), \\ \pi_T = -\frac{1}{2}\left(K_T\frac{1}{(hx_3)^2} + K_M\frac{1}{(hx_3)^2}\right), \\ \pi_M = -\sum\limits_{J\neq M}\pi_J, \end{cases} \tag{22}$$

where the discrete equations of the potential field have the following form:

$$\pi_M\phi_M + \pi_B\phi_B + \pi_S\phi_S + \pi_W\phi_W + \pi_E\phi_E + \pi_N\phi_N + \pi_T\phi_T = b^\pi_M. \tag{23}$$

It is a heptadiagonal-matrix and the approximation of the potential is obtained by solving the following linear system

$\Pi \cdot \phi = b^{\pi}$ where b^{π} is zero on every interior point of the domain. On the boundary points, b^{π} contains the values of the Dirichlet boundary conditions. The electrical conductivity K being a positive parameter, then the matrix Π is positive definite. Moreover the discretization error can be estimated by classical techniques and it is well known that this quantity tends to zero with the discretization step size. Thus, we have the following result.

Proposition 6. *The matrix Π is an M-matrix.*

Remark 5. The seven linear systems arising at each time steps are difficult to solve from a numerical point of view. Particularly, the concentration equation (3) is convection dominated and the two pressure correction equations (9) and (12) occurring in the PISO method are also stiff problems. Here is a classification of the difficulties in decreasing order:

- the two pressure correction equations (9) and (12),
- the concentration equation (3),
- generalized Poisson equation (4),
- the three momentum equations (7).

Note also that, except for the Poisson equation (4) the numerical solution can be improved by choosing a smaller time step size.

4. Solution of linear systems by the parallel asynchronous Schwarz alternating method

4.1. Principle of Schwarz alternating method

Domain decomposition methods, such as the Schwarz algorithm introduced by Lions [19–21], Dryja [9], Dryja and Widlund [10,11], are well suited to the parallel solution of partial differential equations [17]; for more details about Schwarz alternating method, the reader is referred to [2,29,30,33]. In these methods, in order to parallelize the computation, the domain of a partial differential equation is split into subdomains. Thus, in order to compute a solution of the global problem, smaller subproblems are solved on each processor of a parallel computer.

A major difficulty in the implementation of domain decomposition methods is the handling of the computational overhead due to the synchronizations required when processors communicate. Synchronizations occur during the communications at the beginning of each iteration, when a processor waits for a message. Parallel asynchronous algorithms are aimed at suppressing idle times without using load balancing techniques. The processors can work concurrently without any order or synchronizations. The convergence of asynchronous algorithms, firstly analysed by Chazan and Miranker for linear systems [5], have been established for non-linear systems in the frameworks of accretive operators and M-functions [15,22,23,25].

Note that in classical asynchronous algorithms, communication cannot occur during the solution of a subproblem. Then, theoretical studies have been carried out on the concept of flexible communications, which allows communications at any time of the computation. The convergence has been analysed by different approaches: partial ordering techniques [24] and contraction techniques [12].

Parallel asynchronous Schwarz algorithms can be formulated as follows: consider a boundary value problem $\Lambda \cdot u = f$ on a domain Ω with boundary condition $B \cdot u = g$ on $\partial\Omega$. For the sake of simplicity, we consider a decomposition in two subdomains Ω_1 and Ω_2. The parallel algorithm for two processors consists in solving at each iteration:

$$\begin{cases} \Lambda_1 \cdot u_1^{p+1} = f_1 & \text{on } \Omega_1, \\ B_1 \cdot u_1^{p+1} = g_1 & \text{on } \partial\Omega \cap \Omega_1, \\ u_1^{p+1} = \tilde{u}_2^{p} & \text{on } \partial\Omega_1 \cap \Omega_2 \end{cases} \quad \text{and} \quad \begin{cases} \Lambda_2 \cdot u_2^{p+1} = f_2 & \text{on } \Omega_2, \\ B_2 \cdot u_2^{p+1} = g_2 & \text{on } \partial\Omega \cap \Omega_2, \\ u_2^{p+1} = \tilde{u}_1^{p} & \text{on } \partial\Omega_2 \cap \Omega_1, \end{cases} \tag{24}$$

where $\tilde{u}_1^p$ and $\tilde{u}_2^p$ denote the available values of the components of the iterate vector (u_1, u_2) at the nth iteration. In the synchronous algorithm, $\tilde{u}_1^p = u_1^p$ and $\tilde{u}_2^p = u_2^p$. Besides, in the classical asynchronous algorithm [22], these components may be delayed as follows: $\tilde{u}_1^p = u_1^{p-\rho_1(p)}$ and $\tilde{u}_2^p = u_2^{p-\rho_2(p)}$ with $\rho_i(p) \geqslant 0$. Finally, in the case of asynchronous algorithm with flexible communications [24,12], $\tilde{u}_i^p$ are not necessarily associated to components that are labelled by an iteration number as communication may occur at any time. In the sequel, we will focus on linear Λ operators.

Theorem 1. *If the discretization matrix of the operator Λ is an M-matrix (irreducibly diagonal dominant), then synchronous and asynchronous parallel Schwarz iterations (24), (with or without flexible communications) converge to the solution of the boundary value problem $\Lambda \cdot u = f$ for any initial guess $(u_1^0,\ u_2^0)$.*

Proof. See [12,15,25]. □

In the previous section, the discretization of the boundary value problems governing the considered problem leads to the solution of seven linear algebraic systems. Furthermore we have shown that the matrices arising in these seven linear systems are all M-matrices. Thus, the parallel asynchronous Schwarz algorithm with flexible communications converge for the solution of all linear systems involved in the continuous flow electrophoresis simulation.

Remark 6. Synchronous and asynchronous Schwarz alternating methods were used for solving all the linear systems of the discrete problem and not directly the PDE. That is the reason why the M-Matrix property is necessary to prove convergence.

4.2. Implementation of the Schwarz alternating method

The Schwarz alternating method can be combined with various schemes of computation. An asynchronous iterative scheme with flexible communications and a synchro-

nous one have been implemented for the parallel numerical experiments with the 3D physical model. The domain Ω, where the boundary value problem is defined, is split into overlapping parallelepiped subdomains. Minimal values have been chosen for overlapping between subdomains (one mesh size). Thus, sequences of smaller subproblems are solved on each processor of the parallel computer in order to compute a solution of the global problem; practically more accuracy is obtained. Several subdomains, i.e. parallelepipeds, are assigned to each processor in order to implement a strategy which is close to the multiplicative strategy [2]. To obtain a faster convergence of the parallel computations, each processor handles contiguous subdomains, numbered according to red–black ordering; such ordering is more appropriate for parallel computations and, in this case the convergence result of Theorem 1 still holds [4].

Each processor updates the components of the iterate vector associated with its subdomains and computes the residual norm corresponding to the subdomains in order to participate to the convergence detection. A block relaxation method is used in order to solve each subproblem on each subdomain; this kind of method allows very flexible communications between the processors. Note also that a direct method is also suitable for this purpose (see [31,32]) but combersome with 3D domains since this kind of algorithm induces fill-in during the factorisation of the matrix.

Convergence of the parallel iterative process occurs when a given predicate on a global state is true; an usual predicate corresponds to the fact that, on every subdomain, the norm of the local residual remains under a given threshold after two successive updates of the component (see [1, p. 580], and [26]).

Due to termination and detection of the global state of the processes, the implementation of the asynchronous variant of the parallel Schwarz alternating method is more complex than the synchronous one. Let us recall that parallel asynchronous iterative algorithms with flexible communications are general iterative methods whereby iterations are carried out in parallel by several processors without any order nor synchronization (see [12,24]). The main feature of this class of parallel iterative methods is flexible data exchange between the processors. The value of the components of the iterate vector which is used in an updating phase may come from updates which are still in progress and which are not necessarily labelled by an outer iteration number. The efficiency of parallel algorithms strongly depends on the communication frequency within the computations as communications increase the overhead. Point to point communications between two processes have been implemented using persistent communications request and MPI (message passing interface) facilities in both version of Schwarz alternating methods. Message exchanges with the same argument list are repeatedly executed; they correspond to data transmission of successive values of the components of the iterate vector associated with a subdomain frontier. That is the reason why persistent communication requests are well suited. A persistent communication request can be thought of as a one way channel. This approach permits one to reduce the communication overhead between the process and the communication controller.

Remark 7. Between each step of the PISO method, synchronization is actually required, even in the asynchronous simulation.

For the sake of robustness, we have used a synchronous mode send operation, since ready mode is unsafe, and buffered mode may lead to overflow in the high communication frequency case. Note that the use of a synchronous mode send operation is not in contradiction with the implementation of asynchronous iterations since the implementation of communication layers and the type of parallel iterative computation scheme are independent.

If global convergence is detected, then computations can be terminated and resources can be freed. All persistent communication requests are cancelled. Note that cancellation of send requests must occur before cancellation of receive requests; otherwise data exchange based on rendezvous mechanism may fail. For more details on the implementation of asynchronous iterative schemes of computation, the reader is referred to [3,4].

Implementation of parallel synchronous iterative schemes of computation was based on the blocking reception of boundary values. The termination order of communications requests is totally handled with MPI facilities. It is not necessary to provide additional information about synchronous Schwarz alternating method, since its implementation and particularly, the optimization of the message passing between the processors is straightforward in this case. Reference is made to [3,16] for implementation details concerning parallel synchronous iterative algorithms.

5. Parallel numerical experiments

5.1. Numerical and physical parameters

Numerical simulations were carried out using a mesh constituted with 1,200,000 discretization nodes and the domain is splitted into 128 subdomains. Regular meshes have been used in all three-dimensions with different sizes ($N_1 = 400$, $N_2 = 150$, $N_3 = 20$ points).

The test case performed on this mesh concerns an electrophoresis cell of size $30 \times 0.5 \times 10$ cm. The velocity profile at entrance is imposed with a parabolic shape. The Reynolds number is defined on the width of the cell, the mean flow rate and the viscosity of the carrier flow. A protein with fixed concentration is injected at the carrier flow velocity (homocinetical injection) through a hole situated on the center of the face C. Physical parameters are shown in Table 2.

Table 2
Physical parameters

Reynolds number	Electrical field	Electrophoretic mobility	Filament radius
250	950 V m^{-1}	5.26×10^{-8} m^2 V^{-1} s^{-1}	1.5 mm

An example of the computed concentration is given in Figs. 3 and 4. We can notice that the protein filament is significantly deviated owing to its electrophoretic mobility. The croissant effect is clearly shown. In this example, the main stream direction is not significantly affected but secondary flows are observed owing to the coupling between Navier–Stokes, concentration and electrical field equations.

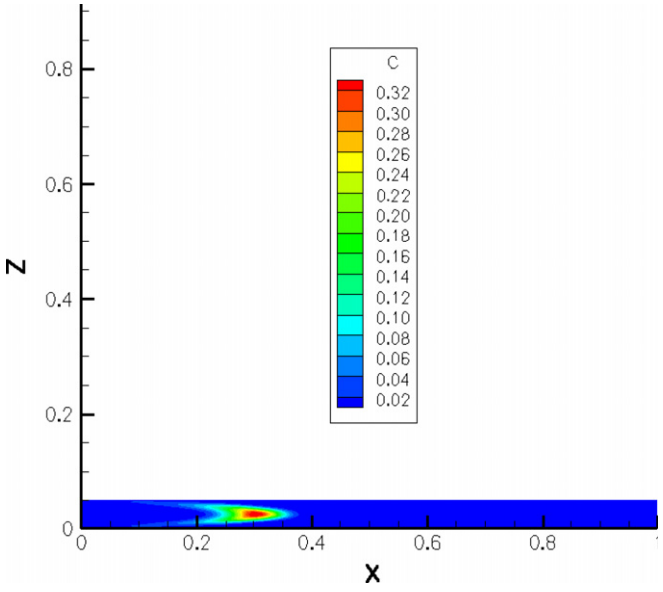

Fig. 3. Profile of the concentration in a transverse plane.

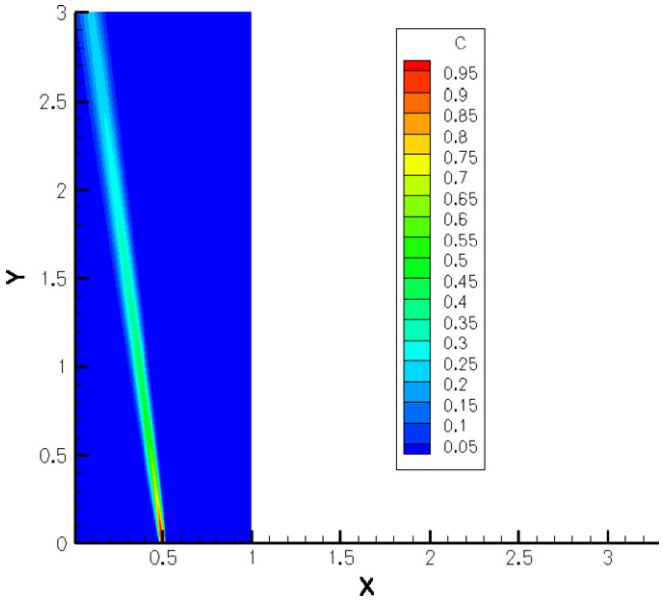

Fig. 4. A protein filament in the longitudinal plane.

Table 4
Elapsed times of parallelized parts of the simulations on p690+

Number of processors	Synchronous (s)	Asynchronous (s)
2	3135	3146
4	1670	1497
8	1041	769
16	709	499

Table 3
Elapsed times of parallel simulations on p690+

Number of processors	Synchronous (s)	Asynchronous (s)
1	5649	
2	3844	3954
4	2461	2285
8	1832	1555
16	1499	1285

Sequential simulations have been carried out using a Pentium 4 Personal Computer at 2.8 GHz; for 900 time steps, 46 h are necessary to achieve the computation. As a consequence, the parallelization of the computations is required.

5.2. The parallel computer

Parallel simulations have been carried out using an IBM SP4 located at IDRIS in Paris[1]; this machine is constituted by 12 SMP nodes of 32 P690+ processors (at 1.3 GHz) connected via a 1.6 Gbits/s federation network. The latency of Federation network is between 5 and 7 μs, and its bandwidth is 2 Gbits per second for each node (see [34]). Due to the limitation of the computation means in dedicated exploitation, the parallel simulations were achieved using only 20 time steps. Furthermore, because of the size of the linear systems to be solved, the number of processors used in the parallel simulations was limited to 16.

5.3. Results and discussion

Table 3 (respectively, 4) shows the elapsed time for the sequential, parallel simulations using synchronous and asynchronous Schwarz alternating methods with flexible communications (respectively, the parallelized part of the parallel simulations) with respect to the number of processors; indeed, the sequential part of a parallelized code being irreducible, the experimental results shown in Table 4 are very interesting to present here.

Fig. 5 displays the elapsed time of the simulations using synchronous and asynchronous Schwarz alternating method with flexible communications with respect to the

[1] Institut du Développement et des Ressources en Informatique Scientifique.

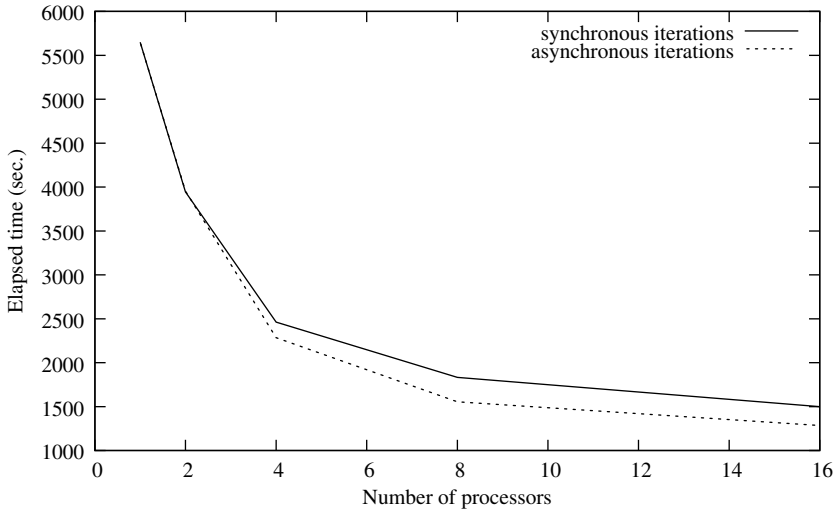

Fig. 5. Elapsed times of parallel simulations on p690+.

number of processors. Figs. 6 and 7 (respectively, 8 and 9) display the speed-up and the efficiency of the synchronous and asynchronous parallelizations of the numerical simulation (respectively, the parallelized parts) with respect to the number of processors.

We can note that, except when the number of processors is low, the parallel asynchronous Schwarz alternating method with flexible communications is more efficient than the synchronous one because in the asynchronous mode, the idle time due to synchronization between the processors are negligible. However, when two processors are used, the overhead of the parallelization is not sufficiently high to make asynchronism profitable. Note that in this case, asynchronism is not penalized since the elapsed times of the synchronous and asynchronous algorithms are very close.

When the number of processors increases, the efficiency of the synchronous algorithms decreases faster than the efficiency of the asynchronous one. The lack of synchronization points and the use of current values of the components of the iterate vector induce better performances for the asynchronous Schwarz alternating method with flexible communications, when many processors are used.

Note that in our implementation, two or more subdomains are assigned to each processor; thus, on each processor, this assignation allows a behaviour of the parallel algorithm as close as possible of the sequential multiplicative Schwarz alternating method [2]. From a practical point of view, parallel synchronous and asynchronous algorithms are more efficient in such subdomain distribution.

5.4. Conclusion on performance

Finally, it can be noted that above eight processors, the performances of both parallel synchronous and asynchronous versions of the Schwarz alternating method decrease. Indeed, in this case, the load of computation becomes too

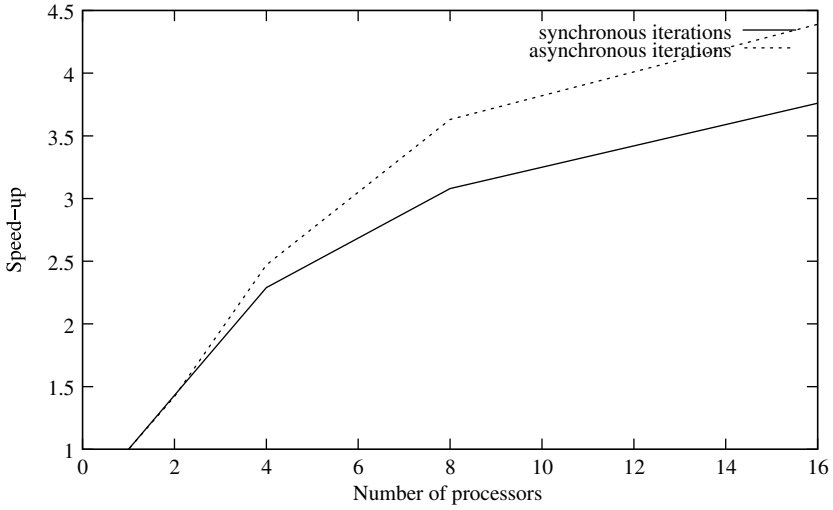

Fig. 6. Speed-up of parallel simulations on p690+.

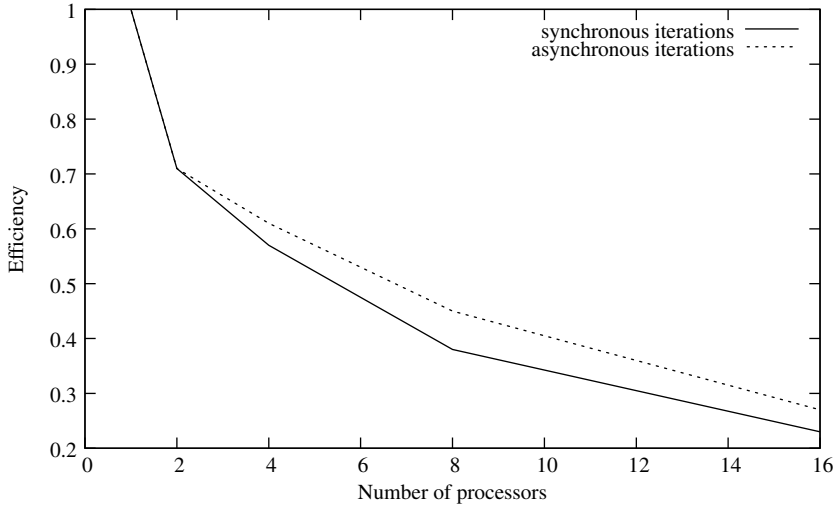

Fig. 7. Efficiency of parallel simulations on p690+.

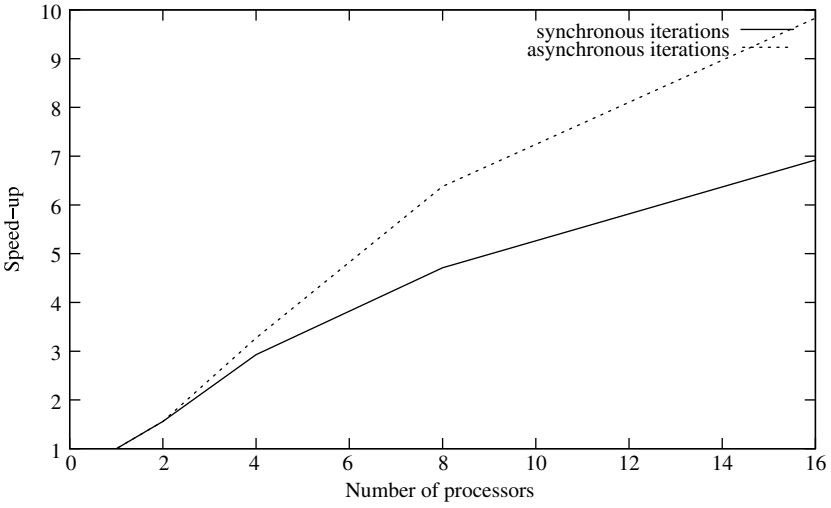

Fig. 8. Speed-up of parallelized parts of the simulations on p690+.

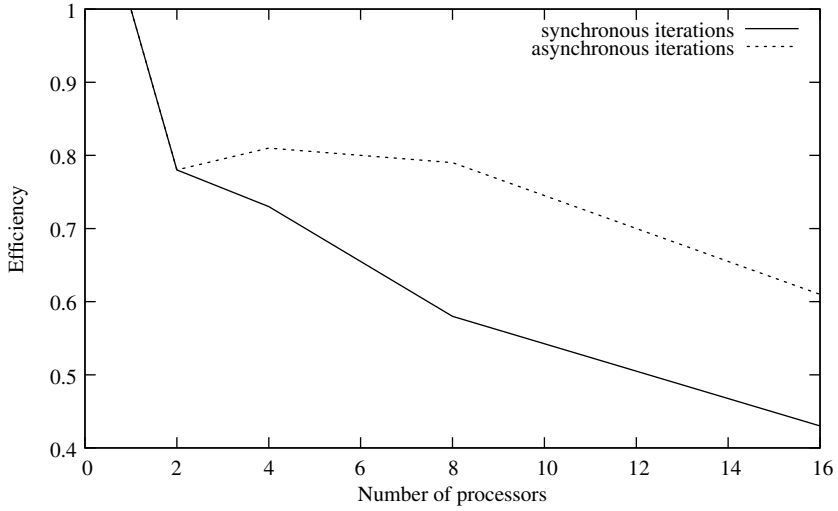

Fig. 9. Efficiency of parallelized parts of the simulations on p690+.

low compared with the overhead of the parallelization. The efficiencies of the considered parallel algorithms depend strongly on the size of the algebraic systems. If finer meshes are used, parallel simulations can be efficiently performed with more processors (see [3,4]), as the discretization will lead to larger linear systems.

6. Conclusion

We have fully tested asynchronous algorithms for Schwarz alternating methods with flexible communications, in the case of a complex fully coupled fluid dynamics problem. The numerical simulation of continuous flow electrophoresis was selected as a good candidate for this purpose. Recent results in applied mathematics have been used to ensure the convergence of the solutions of the associated linear systems. It was shown that asynchronism is efficient on a parallel computer; moreover the use of load balancing techniques is not necessary for such parallel asynchronous methods with flexible communications.

References

[1] Bertsekas DP, Tsitsiklis JN. Parallel and distributed computation, numerical methods. Englewood Cliffs, NJ: Prentice Hall; 1989.

[2] Chan T, Mathew T. Domain decomposition algorithms. Acta Numerica 1994:61–143.

[3] Chau M. Algorithmes parallèles asynchrones pour la simulation numérique. Thèse de Doctorat, Institut National Polytechnique de Toulouse, Laboratoire d'Informatique et de Mathématiques Appliquées (ENSEEIHT-IRIT); 2005.

[4] Chau M, El Baz D, Guivarch R, Spitéri P. MPI implementation of parallel subdomain methods for linear and nonlinear convection–diffusion problems. J Parallel Distr Comput 2007;67:581–91.

[5] Chazan D, Miranker W. Chaotic relaxation. Linear Algebra Appl 1969;2:199–222.

[6] Clifton MJ. Numerical simulation of protein separation by continuous-flow electrophoresis. Electrophoresis 1993;14:1284–91.

[7] Clifton MJ, Roux-de-Balmann H, Sanchez V. Electro-hydrodynamic deformation of the sample stream in continuous-flow electrophoresis with an ac electric field. Can J Chem Eng 1992;70:1055–62.

[8] Clifton MJ, Sanchez V. Continuous-flow electrophoresis: numerical simulation of electrokinetics and electrohydrodynamics. In: Proceedings of IAF-92-0908, 43rd Congress of the International Astronautical Federation, Washington DC, August 28–September 5; 1992.

[9] Dryja M. An additive Schwarz algorithm for two and three dimensional finite element elliptic problems. In: Chan T et al., editors. Domain decomposition methods. Philadelphia: SIAM; 1989. p. 147–53.

[10] Dryja M, Widlund OB. Some domain decomposition algorithms for elliptic problems. In: Hager L et al., editors. Proceedings of the conference on iterative methods for large linear systems. San-Diego, California: Academic Press; 1989. p. 273–91.

[11] Dryja M, Widlund OB. Toward a unified theory of domain decomposition algorithms for elliptic problems. In: Chan T et al., editors. Proceedings of the third international symposium on decomposition methods for partial differential equations. Philadelphia: SIAM; 1989. p. 3–21.

[12] El Baz D, Frommer A, Spitéri P. Asynchronous iterations with flexible communication: contracting operators. J Comput Appl Math 2005;176:91–103.

[13] El Tarazi MN. Some convergence results for asynchronous algorithms. Num Math 1982;39:325–40.

[14] Frommer A, Szyld D. On asynchronous iterations. J Comput Appl Math 2000;123:201–16.

[15] Giraud L, Spitéri P. Résolution parallèle de problèmes aux limites non linéaires. Math Model Numer Anal 1991;25:73–100.

[16] Guivarch R. Résolution parallèle de problèmes aux limites couplés par des méthodes de sous-domaines synchrones et asynchrones. Thèse de Doctorat, Institut National Polytechnique de Toulouse, Laboratoire d'Informatique et de Mathématiques Appliquées (ENSEEIHT-IRIT); 1997.

[17] Hoffman KH, Zou J. Parallel efficiency of domain decomposition methods. Parallel Comput 1993;19:1375–91.

[18] Issa RI. Solution of the implicitly discretised fluid flow equations by operator splitting. J Comput Phys 1986;62:40–65.

[19] Lions PL. On the Schwarz alternating method I. In: Glowinski R, Golub GH, Meurant GA, Periaux J, editors. First international symposium on domain decomposition methods for partial differential equations. Philadelphia: SIAM; 1988. p. 1–42.

[20] Lions PL. On the Schwarz alternating method II. In: Chan T, Glowinski R, Periaux J, Widlund O, editors. Domain decomposition methods. Philadelphia: SIAM; 1989. p. 47–70.

[21] Lions PL. On the Schwarz alternating method III: a variant for nonoverlapping subdomains. In: Chan T, Glowinski R, Periaux J, Widlund O, editors. Third international symposium on domain decomposition methods for partial differential equations, vol. 6. Philadelphia: SIAM; 1990. p. 202–23.

[22] Miellou JC. Algorithmes de relaxation chaotique à retards. RAIRO 1975;1:55–82.

[23] Miellou JC. Itérations chaotiques à retards, étude de la convergence dans le cas d'espaces partiellement ordonnés. CRAS Paris 1975;280:233–6.

[24] Miellou JC, El Baz D, Spitéri P. A new class of iterative algorithms with order intervals. Math Comput 1998;67:237–55.

[25] Miellou JC, Spitéri P. Un critère de convergence pour des méthodes générales de point fixe. Math Model Numer Anal 1985;19(4):645–69.

[26] Miellou JC, Spitéri P, El Baz D. Stopping criteria, forward and backward errors for perturbed asynchronous linear fixed point methods in finite precision. IMA J Numer Anal 2005;25:429–42.

[27] Ortega JM, Rheinboldt WC. Iterative solution of nonlinear equations in several variables. New York: Academic Press; 1970. reprinted on Philadelphia: SIAM; 2000.

[28] Patankar SV. Numerical heat transfer and fluid flow. McGraw Hill; 1980.

[29] Quarteroni A, Valli A. Domain decomposition methods for partial differential equations. Oxford: Oxford University Press; 1999.

[30] Smith B, Bjoerstad P, Gropp W. Domain decomposition: parallel multilevel methods for elliptic partial differential equations. Cambridge University Press; 1996.

[31] Spitéri P, Miellou JC, El Baz D. Asynchronous Schwarz alternating method with flexible communication for the obstacle problem. Calculateurs Parallèles, Réseaux et Systèmes Répartis 2001;13:47–66.

[32] Spitéri P, Miellou JC, El Baz D. Parallel Schwarz method for a nonlinear diffusion problem. Numer Algorithm 2003;33:461–74.

[33] Toselli A, Widlund OB. Domain decomposition methods – algorithms and theory. Series in computational mathematics, vol. 34. Berlin, Heidelberg: Springer-Verlag; 2005.

[34] <http://www.arcade-eu.org/overview/>.