A fully parallel mortar finite element projection method for the solution of the unsteady Navier–Stokes equations

A. BEN ABDALLAH[1] **and J.-L. GUERMOND**[2]

Abstract. This paper describes the parallel implementation of a mortar finite-element projection method to compute incompressible viscous flows. The basic idea in the derivation of this method is that the appropriate functional setting for projection methods must accommodate two different spaces for representing the two velocity fields calculated in the viscous and incompressible half steps of the method. The velocity calculated in the viscous step is chosen in the space constrained by the mortar elements; that is, a weak continuity through the subdomain interfaces is enforced, whereas in the projection step, the weak continuity is relaxed. As a result, the projection step is fully parallel. The numerical solutions of a series of test problems in two dimensions calculated by the proposed method compare quite satisfactorily with the reference solutions.

1 INTRODUCTION

The projection method of Chorin [4, 5] and Temam [14] (see also [13] and [12]) is the most frequently employed technique for the numerical solution of the primitive variable Navier–Stokes equations. This method is based on a peculiar time-discretization of the equations governing viscous incompressible flows, in which the viscosity and the incompressibility of the fluid are dealt within two separate steps.

A functional analytical setting which properly accounts for the different character of the equations of the two half-steps has been proposed recently by Guermond [8, 9] and Guermond-Quartapelle [10]. The aim of this work is to describe the parallel implementation of a finite-element projection method which fully exploits the different mathematical structure of the two half-steps. The spatial discretization is based on a mortar finite-element approximation. In the first step, the viscous velocity is approximated by means of the mortar finite element technique, whereas in the second step the constraint enforced by the mortar element is relaxed, *i.e.* the weak continuity through the subdomains' interface is not enforced for the projected velocity. This algorithm is shown to be unconditionally stable and to converge provided the time step is small enough (basically $\delta t = h^\alpha$, with $\alpha > 1/2$). The original point is that the projection step amounts to solving as many independent linear systems as subdomains; this step is fully parallel since the subproblems are independent.

This technique has been tested on a network of workstations and a CRAY 3D by using MPI. Accuracy tests together with measures of speed-up have been performed and are reported at the end of this paper.

2 THE UNSTEADY STOKES PROBLEM

2.1 Hypotheses and notations

Let Ω be an open connected bounded domain of $\mathbb{R}^d$ ($d = 2$ or 3 in applications) with a smooth boundary $\partial\Omega$; say $\partial\Omega$ is Lipschitz and Ω is locally on one side of its boundary.

In order to formulate the time-dependent Stokes problem in a variational setting, we define the following Hilbert spaces:

$$X = H_0^1(\Omega)^d, \qquad M = L^2(\Omega)/\mathbb{R}, \tag{1}$$

and

$$V = \{v \in X,\, div\ v = 0\}, \qquad H = \{v \in V,\, v.n_{|\partial\Omega} = 0\}. \tag{2}$$

We now consider the following variational problem. For $f \in L^2(0,T;L^2(\Omega)^d)$, and $u_0 \in H$, find $u \in C^0(0,T;H) \cap L^2(0,T;X)$ and $p \in L^2(0,T;M)$ so that $u_{|t=0} = u_0$ and

$$\left\{ \begin{array}{l} \forall v \in X, \quad \left(\dfrac{\partial u}{\partial t}, v\right) + (\nabla u, \nabla v) - (p, div\ v) = (f,v), \\ \forall q \in M, \quad (div\ u, q) = 0, \end{array} \right. \tag{3}$$

In the sequel we assume that u, p are smooth solutions of the problem above and that at the initial time all the compatibility conditions implied by the required smoothness are satisfied. For instance, such conditions are satisfied if the initial datum is zero and the source term is regularized at $t = 0$.

[1] ASCI et LIMSI, UPR-CNRS 3251, BP 133, 91403, Orsay, France (adnene@asci.fr)

[2] LIMSI, UPR-CNRS 3251, BP 133, 91403, Orsay, France (guermond@limsi.fr)

ECCOMAS 96.
Published in 1996 by John Wiley & Sons, Ltd.

2.2 The spatial discretization

The mortar element technique have been developed by Bernardi, Maday and Patera [1], [2]. In this section we recall some aspects of this technique and we formulate our time dependent Stokes problem within this discrete setting.
We assume that we have at hand a partition of Ω into N non-overlapping polygonals:

$$\overline{\Omega} = \cup_{n=1}^{N} \overline{\Omega}^{n}, \text{ and } \qquad \forall n \neq m,\ \Omega^{n} \cap \Omega^{m} = \emptyset. \tag{4}$$

For sake of simplicity we assume also that Ω is two-dimensional, though the fractional step technique we develop is dimension independent. We restrict ourself to polygonal sub-domains. Furthermore, we assume that the decomposition of Ω is geometrically conformal; that is to say, the intersection of two subdomains is an edge, a vertex or empty. This hypothesis simplifies the implementation of the technique but is not a limitation of the mortar method. The interface between subdomain Ω^m and Ω^n is denoted by $\partial\Omega^{mn}$.
For each subdomain Ω^n, we define $\mathcal{F}_h^n$ a regular mixed triangulation (*e.g.* P_1-iso-P_2/P_1 or P_2/P_1 see Girault-Raviart [6] or Brezzi [3] for other details). We denote by X_h^n and M_h^n the linear spaces spanned by the velocity and the pressure triangulation of Ω^n respectively. Note that we do not require the grids of each subdomains to match; the weak continuity through the subdomains' interfaces is enforced by mortar functions. We denote by W_h^{mn} the space of the mortar functions associated to the interface $\partial\Omega^{mn}$; for a clear definition of this space we refer to [1]. The space of the mortar functions is defined by

$$W_h = \prod_{n=1}^{N} W_h^{mn}. \tag{5}$$

For a function μ_h in W_h we denote by μ_h^{mn} the components of μ_h in W_h^{mn}.
We now define, X_h, the subspace of $X_h^1 \times \ldots \times X_h^N$ so that

$$\begin{aligned} X_h \quad = \quad & \{(v_h^1, \ldots, v_h^N) | \forall m, n, \forall \mu_h \in W_h, v^n_{h|\partial\Omega} = 0, \\ & \int_{\partial\Omega^{mn}} (\gamma_0^m v_h^m - \gamma_0^n v_h^n).\mu_h^{mn} = 0\}, \end{aligned} \tag{6}$$

where γ_0^n denotes the trace operator from $H^1(\Omega^n)$ onto $H^{1/2}(\partial\Omega^n)$. The space X_h is equipped with the following scalar product:

$$(u_h, v_h)_{X_h} = \sum_{n=1}^{N} \int_{\Omega^n} u_h^n.v_h^n + \int_{\Omega^n} \nabla u_h^n.\nabla v_h^n. \tag{7}$$

We also introduce the scalar product $(.,.)_{L_h}$ so that

$$(u_h, v_h)_{L_h} = \sum_{n=1}^{N} \int_{\Omega^n} u_h^n.v_h^n. \tag{8}$$

We use this scalar product to define the dual norm of X_h; the dual of X_h equipped with the dual norm is hereafter denoted by X_h'. Finally, we define $\pi_{X_h'} : L^2(\Omega)^d \longrightarrow X_h'$ the L^2 projection onto X_h'.
The pressure will be approximated in M_h so that

$$M_h = \prod_{n=1}^{N} M_h^n. \tag{9}$$

M_h is equipped with the scalar product

$$(p_h, q_h)_{M_h} = \sum_{n=1}^{N} \int_{\Omega^n} p_h^n q_h^n. \tag{10}$$

Let us also introduce the continuous bilinear form $a_h : X_h \times X_h \longrightarrow \mathbb{R}$, so that for all (u_h, v_h) in $X_h \times X_h$,

$$a_h(u_h, v_h) = \sum_{n=1}^{N} \int_{\Omega^n} \nabla u_h^n.\nabla v_h^n \tag{11}$$

It can be shown that a_h is coercive with respect to the norm $|.|_{X_h}$ of X_h.

$$\exists c > 0,\ \forall u_h \in X_h, \qquad a_h(u_h, u_h) \geq c|u_h|_{X_h} \tag{12}$$

We associate with a_h the linear continuous operator $A_h : X_h \longrightarrow X_h'$ so that, for all (u_h, v_h) in $X_h \times X_h$, we have $(A_h u_h, v_h)_{L_h} = a_h(u_h, v_h)$.
We now introduce the continuous bilinear form $b_h : X_h \times M_h \longrightarrow \mathbb{R}$ so that

$$\forall v_h \in X_h, \forall q_h \in M_h,\ b_h(v_h, q_h) = -\sum_{n=1}^{N} \int_{\Omega^n} q_h^n\, div\, v_h^n \tag{13}$$

We associate with b_h the continuous linear operator $B_h : X_h \longrightarrow M_h$ and its transpose $B_h^t : M_h \longrightarrow X_h'$ so that for every couple (v_h, q_h) in $X_h \times M_h$ we have $(B_h v_h, q_h)_{M_h} = b_h(v_h, q_h)$ and $(v_h, B_h^t q_h)_{L_h} = b_h(v_h, q_h)$. It can be shown that B_h is onto; that is to say, there is a constant $c > 0$ (independent of h) so that

$$\forall q_h \in M_h, \qquad |B_h^t q_h|_{X_h'} \geq c|q_h|_{M_h}. \tag{14}$$

In the functional framework defined above, the spatially discretized time-dependent Stokes problem can be reformulated as follows. For $f \in L^2(0,T; L^2(\Omega)^d)$ and $u_{0,h} \in ker(B_h)$ find $u_h \in L^2(0,T; X_h)$ and $p_h \in L^2(0,T; M_h)$ so that:

$$\left\{ \begin{array}{l} \dfrac{du_h}{dt} + A_h u_h + B_h^t p_h = \pi_{X_h'} f \\ B_h u_h = 0 \\ u_{h|t=0} = u_{0,h} \end{array} \right. \tag{15}$$

where $u_{0,h} \in ker(B_h)$ is an approximation of u_0 in X_h. The discrete counterpart of the source term f is hereafter denoted by f_h for simplicity.

The problem (15) can be shown to be well posed. We hereafter assume that the solution of this semi-discretized problem converges in the appropriate sense to that of (3); the convergence analysis is very classical. In the following we are interested only in approximating the time-dependent problem by means of a projection technique.

3 THE FRACTIONAL-STEP PROJECTION ALGORITHMS

3.1 The discrete setting

In order to uncouple the incompressibility constraint from the time evolution problem, we are led to introduce additional tools (see [8] [9] for other details).
We define Y_h the finite dimensional linear space so that

$$Y_h = \{(v_h^1, \ldots, v_h^N) \in X_h^1 \times \ldots \times X_h^N \mid v^n_{h|\partial\Omega} = 0\}. \quad (1)$$

We equip Y_h with the norm of $L^2(\Omega)^d$ and we denote this norm by $|.|_{Y_h}$. It is clear that X_h is a subspace of Y_h (in terms of linear space) and we denote by i_h the continuous injection of X_h into Y_h. Actually, X_h is composed of the functions of Y_h which are weakly continuous across the subdomains' interfaces.
We introduce another discrete version of the divergence operator; let $C_h : Y_h \longrightarrow M_h$ be so that

$$\forall (v_h, q_h) \in Y_h \times M_h, (C_h v_h, q_h) = -\sum_{n=1}^{N} \int_{\Omega^n} (div\, v_h^n, q_h^n) \quad (2)$$

The relation between B_h and C_h is brought to light by

Proposition 1 *C_h is an extension of B_h, and $i_h^t C_h^t = B_h^t$.*

A consequence of this proposition is that C_h is also necessarily onto, for B_h is onto. As a consequence, if we set $H_h = ker\, C_h$, we have a discrete counterpart of the classical decomposition $L^2(\Omega)^d = H \oplus \nabla(H^1(\Omega))$:

Corollary 1 *We have the orthogonal decomposition:*

$$Y_h = H_h \oplus C_h^t(M_h). \quad (3)$$

3.2 The projection scheme

We introduce a partition of the time interval $[0, T]$: $t^k = k\delta t$ for $0 \leq k \leq K$ where $\delta t = T/K$, and define two series of approximate velocities $\{\tilde{u}_h^k \in X_h\}$ and $\{u_h^k \in Y_h\}$ and one series of approximate pressures $\{p_h^k \in M_h\}$ so that

$$\left\{ \; \frac{\tilde{u}_h^{k+1} - i_h^t u_h^k}{\delta t} + A_h u_h^{k+1} = f_h^{k+1} - B_h^t p_h^k \right. \quad (4)$$

and

$$\left\{ \begin{array}{l} \dfrac{u_h^{k+1} - i_h \tilde{u}_h^{k+1}}{\delta t} + C_h^t(p_h^{k+1} - p_h^k) = 0 \\ C_h u_h^{k+1} = 0 \end{array} \right. \quad (5)$$

The series $\{u_h^k\}$ is initialized by $u_h^0 = u_{0,h}$ and assuming that $p_h \in C(0, T; M_h)$ the series $\{p_h^k\}$ is initialized by $p_h^0 = p_{h|t=0}$.
Remark 3.1. The problem (4) is well posed since A_h is X_h-elliptic. The problem (5) is also well posed thanks to corollary 1: indeed the couple $(u_h^{k+1}, \delta t(p_h^{k+1} - p_h^k))$ is the decomposition of $\tilde{u}_h^{k+1}$ in $H_h \oplus C_h^t(M_h)$; *i.e.*, $u_h^{k+1} = P_{H_h} \tilde{u}_h^{k+1}$ where P_{H_h} is the orthogonal projection of Y_h onto H_h.
Remark 3.2. Note that since no weak continuity through the subdomains' interfaces is enforced on u_h^{k+1} and p_h^{k+1}, the projection problem (5) reduces to a series of N completely independent problems that can be solved in parallel.
Remark 3.3. In practice the projected velocity u_h^k is eliminated from the algorithm as follows (see [8]). Replace u_h^k in (4) by its definition which is given by (5) at the k-th time step; note that $i_h^t C_h^t = B_h^t$, as already mentioned. In (5), u_h^{k+1} is eliminated by applying C_h to the first equation and by noting that C_h is an extension of B_h. The algorithm which is implemented reads, for $k \geq 1$,

$$\frac{\tilde{u}_h^{k+1} - \tilde{u}_h^k}{\delta t} + A_h \tilde{u}_h^{k+1} = f_h^{k+1} - B_h^t(2p_h^k - p_h^{k-1}), \quad (6)$$

and

$$C_h C_h^t(p_h^{k+1} - p_h^k) = \frac{B_h \tilde{u}_h^{k+1}}{\delta t}. \quad (7)$$

Remark 3.4. Higher accuracy in time can be obtained if we replace the two-level backward Euler step of first order by a backward three-level Euler step of second order as follows

$$\frac{3\tilde{u}_h^{k+1} - \tilde{u}_h^k - i_h^t(3u_h^k - u_h^{k-1})}{2\delta t} + A_h \tilde{u}_h^{k+1} = f_h^{k+1} - B_h^t p_h^k, \quad (8)$$

and

$$\left\{ \begin{array}{l} \dfrac{3u_h^{k+1} - u_h^k - i_h(3\tilde{u}_h^{k+1} - \tilde{u}_h^k)}{2\delta t} + C_h^t(p_h^{k+1} - p_h^k) = 0, \\ C_h u_h^{k+1} = 0. \end{array} \right. \quad (9)$$

Of course, the algorithm can be implemented in a more convenient form by eliminating the end-of-step velocity, as follows:

$$\frac{3\tilde{u}_h^{k+1} - 4\tilde{u}_h^k + \tilde{u}_h^{k-1}}{2\delta t} + A_h \tilde{u}_h^{k+1} = f_h^{k+1} - B_h^t(2p_h^k - p_h^{k-1}), \quad (10)$$

and

$$C_h C_h^t(p_h^{k+1} - p_h^k) = \frac{B_h^t(3\widetilde{u}_h^{k+1} - \widetilde{u}_h^k)}{2\delta t}. \quad (11)$$

It can be shown that for a fixed mesh size h, this algorithm yields second order accuracy in time in the natural norms defined below.

3.3 Stability and convergence analysis

The projection algorithm introduced above is stable in the following sense:

$$\max_{0 \leq k \leq K} |u_h^k|_{L_h} + \left[\delta t \sum_{k=1}^{K} |p_h^k|^2_{M_h} + |\tilde{u}_h^k|^2_{X_h} \right]^{1/2} \leq c(u_0, p_0, f), \quad (12)$$

where $c(u_0, p_0, f)$ is a function of the data of the problem. The fact that the projection step is parallel is paid by the fact that the discrete gradient operator is not optimally stable; indeed if we denote by ρ_h an interpolation operator on M_h, we only have

$$\forall q \in H^1(\Omega), \qquad |C_h^t \rho_h q|_{L_h} \leq \frac{c}{h^{1/2}} |q|_1. \tag{13}$$

This default of stability, yields a conditional convergence rate; indeed, we have been able to prove

$$\max_{0\leq k\leq K} |u(t^k) - \tilde{u}_h^k|_{L_h} + \left[\delta t \sum_{k=1}^{K} |p(t^k) - p_h^k|^2_{M_h} + |u(t^k) - \tilde{u}_h^k|^2_{X_h}\right]^{1/2} \leq c(h + \delta t/h^{1/2}). \tag{14}$$

Hence we have convergence provided $\delta t = h^\alpha$ with $\alpha > 1/2$. Actually we have convergence of order h if $\delta t = h^{3/2}$.

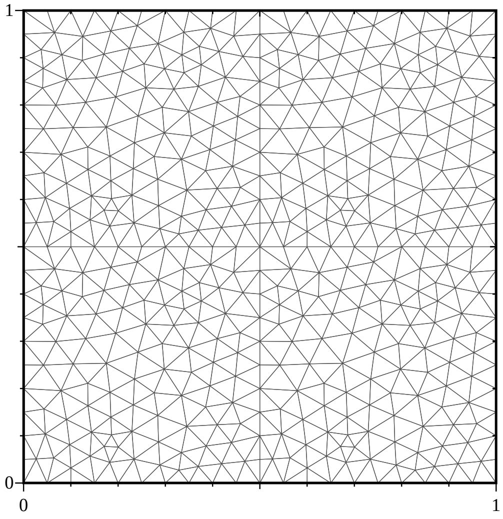

Figure 1. Display of the four subdomains together with their unstructured P_1 mesh.

4 NUMERICAL TESTS AND DISCUSSIONS

4.1 Spatial discretization and solution of linear systems

The fractional-step method described in the previous section has been implemented using P_1-iso-P_2 / P_1 finite element meshes. In each subdomain, a P_1 Delaunay grid is generated.

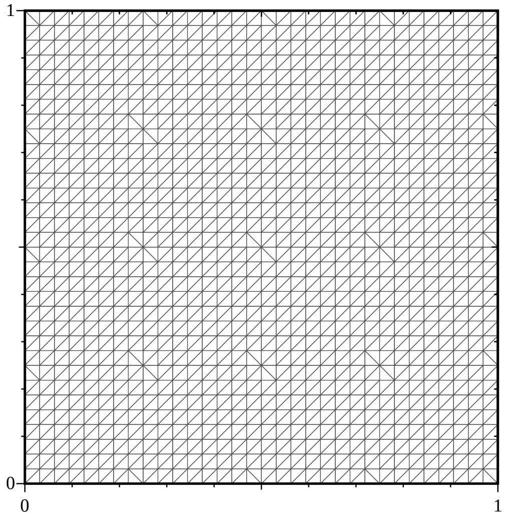

Figure 2. Display of the sixteen subdomains together with their P_1 mesh.

The finer mesh P_1-iso-P_2 is then obtained from the coarse one by splitting each coarse triangle into four equal small triangles introducing the mid-side nodes on all sides of the coarse mesh.

The integration over the triangles is performed by means of numerical quadrature using a three-point Gauss formula. This assures the exact evaluation of all scalar products including those which involve the nonlinear convection term. The values of the Jacobian determinant and of the weighting function derivatives at Gauss points of all elements are evaluated once an for all at the beginning of the calculation and stored in arrays for subsequent use.

The projection algorithm requires to solve sparse linear systems of algebraic equations for both the velocity and the pressure in each subdomain Ω^n, $n = 1, \ldots, N$. To obtain the velocity components, we have to solve a saddle-point problem

$$\begin{cases} A_h u + R_h^t \lambda = F \\ R_h u = 0, \end{cases} \tag{1}$$

where R_h is the mortar elements matrix and A_h is a bloc diagonal matrix, each bloc is factorised locally on each subdomain by means of Cholesky algorithm. The hole system is solved by means of conjugate gradient applied to the equation

$$R_h A_h^{-1} R_h^t \lambda = R_h A_h^{-1} F \tag{2}$$

The Poisson-like pressure problem require to solve systems of type

$$C_h I_h^{-1} C_h^t P = F \tag{3}$$

where I_h is a mass matrix. This system is solved iteratively by means of a conjugate gradient preconditioned by $C_h \tilde{I}_h^{-1} C_h^t$ where $\tilde{I}_h$ is the lumped mass matrix.

4.2 Implementation

The code is written in FORTRAN 90 and uses the MPI message passing library. It is run on a CRAY-T3D which has 128 Dec Alpha processors. Each processor can deliver $150 Mflops/s$ but, still now, because of hardware implementation (mutilated memory caches), hardly 10% of the peak $150 Mflops/s$ can be used. (recall that T3D is an experimental machine, hopefully, these problems will be solved with the T3E).

4.3 Test problems

In order to illustrate the second order algorithm described above,(10)-(11), we provide convergence tests on a test problem. We consider the following exact solution on the square $\Omega =]0,1[^2$,

$$\begin{aligned} u_x &= \cos(\pi t)x + \sin(\pi t)y + \sin^2(\pi t) + 3\sin(\pi t), \\ u_y &= \sin^2(\pi t)x - \cos(\pi t)y + \cos(\pi t), \\ p &= \sin(\pi t)(y - 1/2) \\ &\quad -[sin(2\pi t) + 3\cos(\pi t)](x - 1/2). \end{aligned}$$

We have chosen linear functions in space to avoid spending much time on error calculation.

The domain Ω is divided into sixteen subdomains as shown in figure 2. Fore sake of simplicity, we used structured grid on subdomains. We solve the time-dependent Stokes problem on the time interval $0 \leq t \leq 1$ with a source term corresponding to our chosen solution.

In figure 3, we have reported the errors on the velocity in the norms $l^\infty(0,T;L^2(\Omega)^d)$ and $l^2(0,T;H^1(\Omega)^d)$ as functions of the time step δt for different global mesh sizes. As expected we observe a second order slope for moderate δt but we can see that the estimation constant seems to be dependent of h in the rate of h^{-1}. This result is not surprising since we have proven convergence of order $h + \delta t/h^{1/2}$ for the first order scheme (6)-(7); hence, convergence in time of order $\delta t^2/h$ for (10)-(11) seems reasonable. (but has yet to be proven).

The error on pressure in the norm of $l^2(0,T;L^2(\Omega))$ is reported as a function of δt in figure 4. The same second order slope and mesh size dependency as that obtained on the velocity are observed on the pressure.

To illustrate the mesh size influence on the convergence, we have reported in figure 5 the errors on both velocity and pressure as functions of h while δt is kept equal to $ch^{3/2}$. We note a second order slope on the velocity in the norm $l^\infty(0,T;L^2(\Omega)^d)$. First order slope is obtained in the $l^2(0,T;H^1(\Omega)^d)$ norm (practically we obtained better than first order for the term in $h^2 = \delta t^2/h$ is still dominant). The same observations can be made for the pressure.

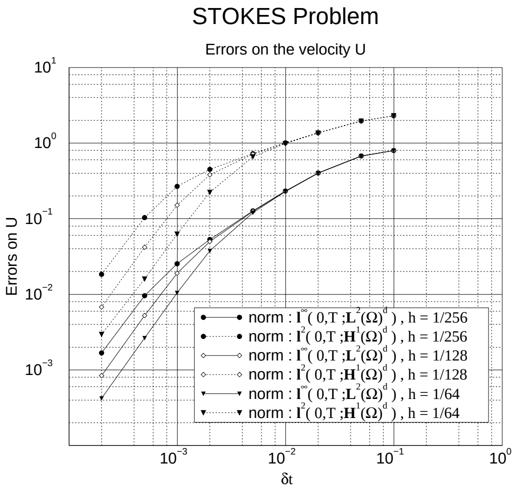

Figure 3. $\max_{0 \leq k \leq K} |u(t^k) - \tilde{u}_h^k|_{L^2(\Omega)^d}$ and $[\delta t \sum_{k=1}^{K} |u(t^k) - \tilde{u}_h^k|_{H^1(\Omega)^d}]^{1/2}$ versus δt.

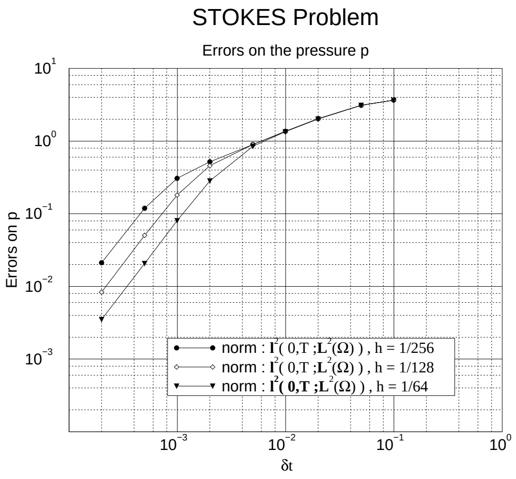

Figure 4. $[\delta t \sum_{k=1}^{K} |p(t^k) - p_h^k|^2_{L^2(\Omega)}]^{1/2}$ versus δt.

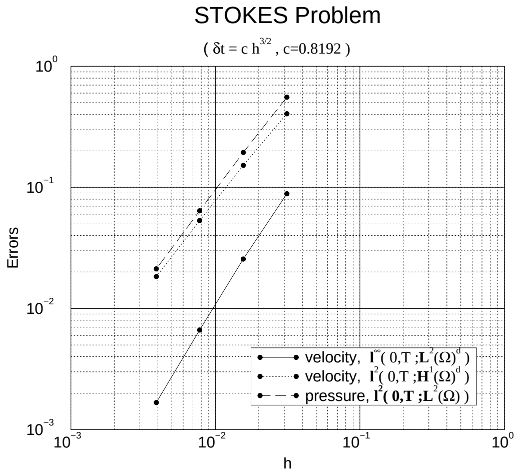

Figure 5. $\max_{0\leq k\leq K} |u(t^k) - \tilde{u}_h^k|_{L^2(\Omega)^d}$, $[\delta t \sum_{k=1}^{K} |u(t^k) - \tilde{u}_h^k|_{H^1(\Omega)^d}]^{1/2}$ and $[\delta t \sum_{k=1}^{K} |p(t^k) - p_h^k|^2_{L^2(\Omega)}]^{1/2}$ versus h with $\delta t = ch^{3/2}$.

4.4 Speed-Up

The numerical tests reported in this section were performed with different numbers of processors for a global mesh size $h = 1/256$ $(66049$ nodes) and with $\delta t = 0.002$. In table 1, we can see for different numbers PE_s of processors the elapse time T_t for one time step iteration, the mono-processor equivalent time $PE_s * T_t$, the elapse time and the number of Conjugate Gradient iterations N_bCG_{it} (resp. the time and the number of Preconditioned Conjugate Gradient iterations N_bPCG_{it}) for one prediction (resp. projection) step. At first sight, we can notice that the algorithm has almost the right speed-up (from $PE_s = 16$ to $PE_s = 32$ we have a global speed-up of 1.9). But if we look at the projection step we notice better results; recall that this step amounts to solve a Poisson problem. If we look at the prediction step, where the communications between processors occur, the speed-up between $16PE_s$ and $32PE_s$ (resp. $32PE_s$ and $64PE_s$) is about 1.8 (resp. 1.75). Actually, if we consider the speed-up for one CG iteration of the prediction step we obtain 2.28 (resp. 2.0). There are two reasons for this. First, in the prediction step, the linear system is solved by means of a Conjugate Gradient algorithm which is not preconditioned. As a result, the total number of iterations that is needed to reach convergence is dependent of both the problem size and δt (the condition number of $R_hA_h^{-1}R_h^t$ depends on δt). The second reason is that the cost of one CG iteration is not of order N but rather between N and N^2 (we use sparse matrix techniques, see for instance [7]). The efficiency rate has not been calculated since the problem is too big to be solved on one single processor $(N_{nodes} = 66049)$.

Table 1. Time inventory for one time step iteration.

			Prediction step		Projection step	
PE_s	T_t	⋄	♢	♠	♢	♣
16	6.1	98.4	4.27	65	1.88	9
32	3.2	102.3	2.36	82	0.83	8
64	1.7	111.5	1.35	94	0.39	8

⋄ : $PE_s * T_t$ in seconds, ♢ : elapse time in seconds,
♠ : N_bCG_{it}, ♣ : N_bPCG_{it}.

ACKNOWLEDGEMENTS

The authors are grateful to Y. Achdou, Y. Maday and O. Pironneau for helpful discussions and remarks that improved the content of this paper. The present work has been supported by ASCI (Applications Scientifiques du Calcul Intensif, UPR-CNRS 9029, Orsay). The computing resource has been provided by IDRIS. We would like to thank all of the members of the IDRIS parallel computing team for their support. We are especially grateful for J. Chergui's tireless assistance for the CRAY/T3D parallel programming.

REFERENCES

[1] C. Bernardi and Y. Maday, Coupling spectral and finite element methods for the Poisson equation: a review, in the proceedings of the seventh International Conference on Finite Elements in Fluids, T. Chung, G. Karr editors, UAH Press, Huntsville, 1989.

[2] C. Bernardi, Y. Maday, and A. Patera, A new non-conforming approach to domain decomposition: the mortar element technique, in the proceedings of the seventh International Conference on Finite Elements in Fluids, T. Chung, G. Karr editors, UAH Press, Huntsville, 1989, 269–286.

[3] F. Brezzi, On the existence uniqueness and approximation of saddle-point problems arising from Lagrangian multipliers, *R.A.I.R.O.*, **R.2**, 1974, 129–151.

[4] A. J. Chorin, Numerical solution of the Navier–Stokes equations, *Math. Comp.*, **22**, 1968, 745–762.

[5] A. J. Chorin, On the convergence of discrete approximations to the Navier–Stokes equations, *Math. Comp.*, **23**, 1969, 341–353.

[6] V. Girault and P.-A. Raviart, *Finite Element Methods for Navier–Stokes Equations*, Springer Series in Computational Mathematics, **5**, Springer-Verlag, 1986.

[7] J. A. George and J. W.-H. Liu, *Computer Solution of Large Sparse Positive Definite Systems*, Prentice-Hall, Englewood Cliffs, N. J., 1981.

[8] J.-L. Guermond, Some practical implementation of projection methods for Navier–Stokes equations, LIMSI report **94-07**, *Modél. Math. Anal. Numer.* (M^2AN), **30**, 3, 1996.

[9] J.-L. Guermond, Sur l’approximation des équations de Navier–Stokes instationnaires par une méthode de projection, *C. R. Acad. Sc. Paris*, Série I, **319**, 1994, 887–892.

[10] J.-L. Guermond and L. Quartapelle, On the approximation of the unsteady Navier–Stokes equations by finite element projection methods, LIMSI report **95-14**, submitted to *Numer. Math.*

[11] O. Pironneau, *Méthode des ĺéments finis pour les fluides*, RMA 7, Masson, 1988.

[12] L. Quartapelle, *Numerical Solution of the Incompressible Navier–Stokes Equations*, ISNM **113**, Birkhäuser, Basel, 1993.

[13] R. Temam, *Navier–Stokes Equations*, Studies in Mathematics and its Applications, 2, North-Holland, 1977.

[14] R. Temam, Une méthode d’approximation de la solution des équations de Navier–Stokes, *Bull. Soc. Math. France*, **98**, 1968, 115–152.