A mortar element method for fluids

Yves ACHDOU[1] and Jean-Claude HONTAND[2] and Olivier PIRONNEAU[3]

ABSTRACT

Our interest is turned towards the Navier-Stokes equations. The general trend in computer architecture being towards message passing coarse grain MIMD parallelism, we wish to develop robust, fast and user-friendly algorithms for such machines.

Robustness is achieved by selecting well tested methods with known stability and error properties. Speed is achieved by block decomposition, fast solvers within blocks and minimal communication between blocks. Versatility is obtained by using non matching meshes between subdomains and user-friendliness will be obtained by using a high level language to drive the software as done in FreeFEM (see Pironneau,1994).

1 PROBLEM STATEMENT

The Navier-Stokes equations for incompressible flows are

$$D_t u + \nabla p - \nabla.\nu\nabla u = 0 \quad \nabla.u = 0$$

or, in stream function-vorticity formulation (for constant siscosity ν)

$$D_t\omega - \nu\Delta\omega = 0 \quad -\Delta\psi = \omega$$

where

$$D_t = \partial_t + u\cdot\nabla.$$

In most application ν is found through a turbulence model such as the $k-epsilon$ model:

$$D_t k - \nabla.\nu\nabla k - \frac{k^2}{\varepsilon}|\nabla U + \nabla U^T|^2 + \varepsilon = 0,$$

[1] CMAP,Ecole Polytechnique 91128 Palaiseau cedex: achdou@cmapx.polytechnique.fr
[2] University of Paris 6:hontand@ann.jussieu.fr
[3] University of Paris 6:pironneau@ann.jussieu.fr

Domain Decomposition Methods in Sciences and Engineering, edited by R. Glowinski *et al.*

$$D_t\varepsilon - \nabla.\nu\nabla\varepsilon - c_1 k|\nabla U + \nabla U^T|^2 + c_2\frac{\varepsilon^2}{k} = 0.$$

2 GENERAL DESCRIPTION OF THE ALGORITHM

2.1 Time discretization

Total derivatives are approximated by a finite difference formula in space and time, leading to a Eulerian-Lagrangian method:

$$D_t w = \partial_t w + u\cdot\nabla w \approx \frac{1}{\delta t}[w^{m+1} - w^m oX^m] \quad \text{where} \quad w^m oX(x) \approx w^m(x - \delta t u^m(x)).$$

For the velocity pressure formulation of the Navier-Stokes equations, a projection method for the pressure leads to

Step 1: solve

$$\Delta p^{m+1} = \frac{1}{\delta t}\nabla.u^m oX^m + \nabla.(\nabla.\nu\nabla u^m + f) \text{ in } \Omega, \quad \partial_n p|_{\partial\Omega} = 0.$$

Step 2: Solve

$$\frac{u^{m+1}}{\delta t} - \nabla\nu\nabla u^{m+1} = \frac{u^m oX^m}{\delta t} + \nabla p + f.$$

2.2 Spatial discretization

A finite element method with a feasible set of finite element space for velocity and pressure is used (LBB condition) or a stabilized term is added in the pressure equation if the $P^1 - P^1$ couple is used (Franca et al (1992)).

2.3 The Language Gfem

Such algorithms are fairly standard by now and the difficulty is not so much in the problem rather than in the infinite possibilities for coupling with other equations such as turbulence, temperature, electromagnetism.... We are experimenting with a macro-description of the problem with a dedicated language called Gfem. It used two basic blocks: an elliptic solver called by the key word "solve" and a convection operator instanced by the key word "convect". To demonstrate the power of the language we send the reader to http://www.ann.jussieu.fr/freefem/gfem.html

3 TWO BASIC BLOCKS

The first basic block is a general solver for linear second order PDE

$$au + b\partial_x u + c\partial_y u - \nabla.M\nabla u = f \quad \text{in} \quad \Omega \subset R^d,$$
$$u = u_0, \quad \text{or } \alpha u + \partial_\nu u = g \quad \text{on} \quad \partial\Omega.$$

So far only the symmetric case is implemented in parallel with mortars:

$$au - \nabla.M\nabla u = f \text{ in } \Omega, \quad u = u_0 \text{ or } \alpha u + \partial_\nu u = g \text{ on } \partial\Omega$$

where $M = M^T$ is an $R^d \times R^d$ matrix, not necessarily positive definite.

The second block is the convection operator

$$w(x) \rightarrow w(X(x))$$

where X(x) is the solution at $t - \delta t$ of

$$\frac{dX}{dt} = u(X,t), \quad X(t) = x.$$

4 NON MATCHING MESHES

Quadrilateral subdomains are used with cartesian meshes inside each subdomain. Q^1 or P^1 discretization inside each macro-element is used with possible discontinuities at interfaces. We do not use the mortar element in its generality, but restrict the macro-mesh to be geometrically conforming.

The subdomains will be large and few of them must be allowed to have at most one curved boundary if necessary. In this case we require the knowledge of a parametric description of the curved boundary from which we can then build a map G which transforms the unit square (the reference element R) into the element $Q = G(R)$.

Meshes in each macro elements are chosen independently. They do not match at the interfaces beetween subdomains. Therefore, in order to build a finite element space approaching $H^1(\Omega)$ one has to write a weak continuity constraint at the subdomain interfaces.

Within each macro element some adaption is done via a small number of parameters such as the mesh aspect ratio. Of course such mesh do not have the flexibility of unstructured meshes as used in adaptive FEM. But they allow fast linear solvers in each block.

5 CHARACTERISTICS IN PARALLEL

The problem is mapped to the reference macro element:

Given $x = G(y)$ u and $w(.)$ find $w(X(x)$ with $X(x) = G(Y(t - \delta t))$ and $M = G'^{-1}$

$$\frac{dY}{d\tau} = Mu(Y,\tau), \quad Y(t) = y.$$

A time stepping procedure such as Runge-Kutta or backward Euler is used:

$$Y^{k+1} = Y^k - \frac{\delta t}{K} Mu^k(Y^k),$$

where the time step δt is subdivided into K steps $\frac{\delta t}{K}$. To evaluate $Mu(Y^k, t^k)$ is not difficult because the mesh being cartesian the values of u can be found once the element which contains Y^k is known and that can be found by integer divisions.

However during these sub time steps, Y^k on the curve $Y(t)$ may leave the reference macro-element. Instead of continuing immediately the integration process on the neighbor element which contains the rest of $Y(t)$ it is much better for parallelism to store the boundary point $\{t^k, Y(t^k)\}$ for later use.

When all the characteristics have been partially computed in parallel, subdomain by subdomain, a new loop is made with for initial points all the $\{t^k, Y(t^k)\}$ corresponding to interrupted characteristics.

We summarize the process for a mesh made of L macro elements, each having $M_l \times N_l$ elements $Q_{m,n}$, with quadrature points $y_{m,n,l}$:

Set $t_{m,n,l} = \delta t$ for all m, n, l
while some $t_{m,n,l} > 0$
  do // l=1 to L
    for m = 1 to M_l and n = 1 to N_l do
      Loop on doing
        $y_{m,n,l} = y_{m,n,l} - \frac{\delta t}{K} Mu^k(y_{m,n,l})$
        $t_{m,n,l} = t_{m,n,l} - \frac{\delta t}{K}$
      till $t_{m,n,l} < 0$ or $y_{m,n,l}$ not in $]0,1[^2$

6 A PARALLEL ELLIPTIC SOLVER

6.1 The mortar element method

Consider the model elliptic problem

$$u - \mu\Delta u = f|_\Omega, \quad au + \partial_n u = g|_{\partial\Omega}.$$

Two finite element spaces are chosen, one for u and the other one for the Lagrange multipliers of the weak continuity constraint. In our case the functions restricted to a macro element Ω_k are continuous functions piecewise bilinear on the quadrilateral mesh of Ω_k : we call $Q_h(\Omega_k)$ this space of functions and we introduce the product

$$Q_h = \prod_{k=1}^{K} Q_h(\Omega_k),$$

which has an obvious canonical basis of shape functions.

Since the meshes do not match at the interfaces, we need to introduce a Lagrange multiplier space for the weak continuity constraint: for each interface $\Gamma_{kl} = \Omega_k \cup \Omega_l$ the Lagrange multiplier space is called $X_h(\Gamma_{kl})$ and the weak continuity constraint is

$$\int_{\Gamma_{kl}} (u_{h|\Omega_k} - u_{h|\ \Omega_l})\lambda_h = 0 \quad \forall \lambda_h \in X_h(\Gamma_{kl}).$$

The space $X_h(\Gamma_{kl})$ may be constructed either from $Q_h(\Omega_k)$ or $Q_h(\Omega_l)$. Assuming it is built from $Q_h(\Omega_k)$, let $(\lambda^i)_{i\in\{0,\ldots,s_{kl}\}}$ be the traces of the shape functions of $Q_h(\Omega_k)$ on Γ_{kl}, ordered according the abcissa on Γ_{kl} of the mesh nodes with which they are associated. The convergence proof of the method tells us that a good choice

for $X_h(\Gamma_{kl})$ is

$$X_h(\Gamma_{kl}) = Span\{\lambda^0 + \lambda^1, \lambda^2, ..., \lambda^{s_{kl}-2}, \lambda^{s_{kl}-1} + \lambda^{s_{kl}}\},$$

(the space $Span(\lambda^i)_{i\in\{0,...,s_{kl}\}}$ would be too large for preserving the LBB condition).

Calling U the vector of coordinates of an element u_h of Q_h, the weak continuity constraint yields a set of linear equations for U:

$$BU = 0.$$

Similarly the weak form of the PDE is discretized by

$$\sum_k \int_{\Omega_k} [u_h w_h + \mu \nabla u_h \nabla w_h] + \int_{\partial\Omega} a u_h w_h = \sum_l \int_{\Omega_l} f w_h + \int_{\partial\Omega} g w_h$$

for all w_h with $BW = 0$.

This method has the same order of accuracy as its conforming analogon (Bernardi-Maday-Patera[1991], Le Tallec[1992]).

6.2 Solution of the linear system

At the discrete level the problem is of the type : find $U \in R^N$, such that $BU = 0$ and

$$W^T A U = F^T W, \quad \forall W \in R^N \text{ with } BW = 0,$$

or equivalently : find $U \in R^N$ and $\Lambda \in R^S$, $(S = \sum_{k,l}(s_{kl} - 2))$ such that

$$\begin{pmatrix} A & B \\ B^T & 0 \end{pmatrix} \begin{pmatrix} U \\ \Lambda \end{pmatrix} = \begin{pmatrix} F \\ 0 \end{pmatrix}.$$

Iterative solution

The system above is solved by an iterative method in a subspace of constraints. Preconditionning is necessary for efficiency (see Achdou-Kuznetsov in these proceedings).

7 APPLICATION TO THE NAVIER-STOKES EQUATIONS

The Navier-Stokes equations in their stream function-vorticity formulation are

$$D_t\omega - \nu\Delta\omega = 0, \quad -\Delta\psi = \omega, \quad \psi|_{\partial\Omega} = \psi_\Gamma, \quad \frac{\partial\psi}{\partial n} = g.$$

We discretize them in time by

$$\frac{1}{\delta t}[\omega^{m+1} - \omega^m o X^m] - \nu\Delta\omega^{m+1} = 0, \quad -\Delta\psi^{m+1} = \omega^{m+1}.$$

So at each time step we must solve

$$\omega - \mu\Delta\omega = f, \quad -\Delta\Psi = \omega, \quad \frac{\partial\Psi}{\partial n}| = g_1, \quad \Psi|_\Gamma = g_2.$$

As this is not in the standard form of our PDE block we use the following splitting:
decompose ω and Ψ as $\omega = \omega^1 + \omega^0$, $\Psi = \Psi^1 + \Psi^0$ with

$$\omega^0 - \mu\Delta\omega^0 = f, \quad \omega^0|_\Gamma = 0, \text{ and } -\Delta\Psi^0 = \omega^0, \quad \Psi|_\Gamma = \psi_\Gamma,$$

and

$$\omega^1 - \mu\Delta\omega^1 = 0, \quad -\Delta\Psi^1 = \omega^1 \text{ and } \frac{\partial\Psi^1}{\partial n} = g - \frac{\partial\Psi^0}{\partial n}, \quad \Psi^1|_\Gamma = 0.$$

The first set of equations are 2 second order elliptic problems which can be solved successively and the last equations are a fourth order problem homogeneous in space. The traces of Ψ^1 and ω^1 can be obtained by a boundary element method (see Achdou, Pironneau). Alternatively, for laminar flows at high Reynolds number the following approximation can be used

$$\omega^1 - \mu\Delta\omega^1 = 0, \quad \omega^1|_\Gamma = -\frac{1}{\sqrt{\mu}}[g - \frac{\partial\Psi^0}{\partial n}] \text{ and } -\Delta\Psi^1 = \omega^1, \quad \Psi^1|_\Gamma = 0.$$

8 NUMERICAL TEST

The program is written in C and parallelized with PVM; all test have been made on a network of HP 735 connected by Ethernet.

8.1 Test 1

Test Parallelism efficiency on a single problem of Helmholtz type, (conjugate gradient without preconditioning for the Schur complement system on Λ).

$$\omega - \Delta\omega = 1 + \sin(x)\sin(y)$$

$$\omega|_{\partial\Omega} = \sin(x)\cos(y).$$

The grid has 36 macro elements and a total of 21 000 vertices. On a network of 6 machines the following has been measured:

processors	cpu time	% com / cpu
2	58 s	3 %
3	39 s	8 %
4	28 s	14 %
6	19 s	26 %

8.2 Test 2: The Navier-Stokes equations

A stream function-vorticity formulation is used with a decomposition at the boundary. The flow past a cylinder at $Re = 300$ is computed with 40 macro-elements, 20558 vertices on 3 processors (see figure 1 and 2).

The conjugate gradient method for the Schur complement system for Λ is much too slow; so a preconditionner developped by Achdou-Kuznetsov(1995) has been used.

The following performances are obtained with the grid of 36 macro elements and a total of 90 000 vertices. The number of iteration for the iterative algorithm (PCG) is 36. In the table below, the column *matrix* indicates the time needed for assembling the matrix and for factorizing the subdomains blocks, the column *pcg* indicates the time needed for solving one linear system.

proc	cpu	com	% com/cpu	matrix	precond.	pcg
2	486 s	4 s	1 %	416 s	7 s	67 s
3	330 s	6 s	2 %	289 s	20 s	38 s
4	262 s	8 s	3 %	227 s	28 s	32 s
6	197 s	28 s	14%	151 s	32 s	31 s

9 CONCLUSION

The Mortar element method allows the use of nonmatching meshes which is a key feature for time dependent problems with sliding subdomains and for parallel solvers with domain decomposition. It remains to see if this will make our implicit algorithms faster than their explicit competitors with unstructured mesh adaption (Mohammadi (1995) for instance), but it is well worth the programming effort to try it. A 3D implementation of the same ideas for the Navier Stokes equations is underway. With modifications, the same method can applied to the equations of electromagnetism and numerical tests will be presented in a forthcoming paper.

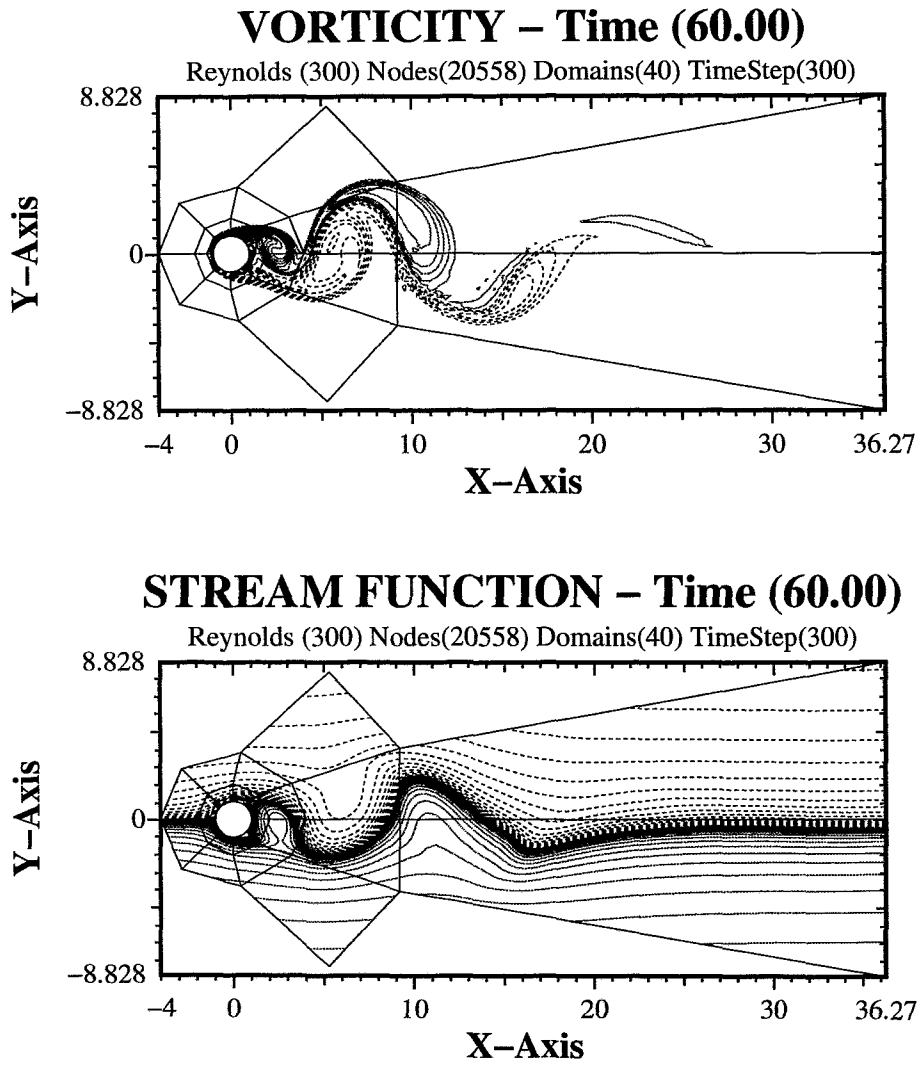

Figure 1 unsteady flow around a cylinder at Reynolds 300

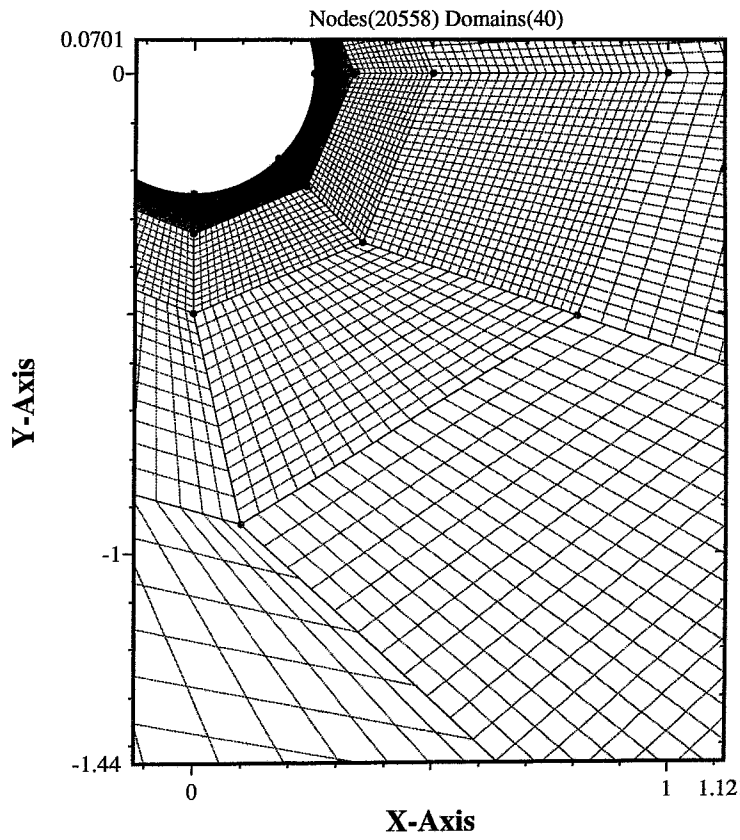

Figure 2 mesh used for computation of stream function and vorticity solutions

References

Y.Achdou, O.Pironneau, *Integral Equation for the Generalized Stokes Operator with Applications to Boundary layer Matching*, Comptes Rendus Acad. Sci. Paris 315 Série I (1992),pp 91-96.

Y.Achdou, O.Pironneau, *A fast solver for Navier-Stokes equations in the laminar regime using mortar finite element and boundary element method.*, SIAM journal of Numerical Analysis. Vol 32, No4, pp985-1016 (1995).

Y.Achdou, Yu.A.Kuznetsov,*Algorithms for the Mortar Element Methods* these proceedings.

O.Pironneau,*FreeFEM: A finite element software with a language and an integrated environment.* Colloque d'Analyse Numérique. Aussoix, 1994. http://www.ann.jussieu.fr/freefem/gfem.html.

B.Mohamadi,*Unstructured explicit solver for Navier-Stokes equations*, INRIA report (1993).