Available online at www.sciencedirect.com
SCIENCE DIRECT®
Comput. Methods Appl. Mech. Engrg. 195 (2006) 4604–4616
Computer methods in applied mechanics and engineering
www.elsevier.com/locate/cma

A computational multilevel approach for solving 2D Navier–Stokes equations over non-matching grids

E. Aulisa [a], S. Manservisi [a,b], P. Seshaiyer [a,*]

[a] Department of Mathematics and Statistics, Texas Tech University, Box 41042, Lubbock, TX 79409-1042, United States
[b] DIENCA, University of Bologna, Italy

Received 2 November 2004; received in revised form 27 September 2005; accepted 7 October 2005

Abstract

A multilevel approach with parallel implementation is developed for obtaining fast solutions of the Navier–Stokes equations solved on domains with non-matching grids. The method relies on computing solutions over different subdomains with different multigrid levels by using multiple processors. A local Vanka-type relaxation operator for the multigrid solution of the Navier–Stokes system allows solutions to be computed at the element level. The natural implementation on a multiprocessor architecture results in a straightforward and flexible algorithm. Numerical computations are presented, using benchmark applications, in order to support the method. Parallelization is discussed to achieve proper accuracy, load balancing and computational efficiency between different processors.

Keywords: Finite element method; Domain decomposition; Vanka solver; Navier–Stokes; Non-matching grids

1. Introduction

In recent years, computational fluid dynamics (CFD) has been gaining acceptance as a design tool in industry but in spite of the current successes it is necessary to enhance the prediction capability of the simulations. In fluid dynamics, it is very common to have regions with different flows and therefore, there is a great interest in developing domain decomposition techniques for coupling non-matching grids in order to compute solutions efficiently in different parts of the domain. One popular domain decomposition approach that has been considered in the recent years is the mortar finite element method (see e.g., [6,4,26] and references therein) which allows the coupling of different subdomains with non-matching grids and discretization techniques within a mathematical and understandable framework [4,25,27]. This method was originally introduced for spectral methods [6,7] and then was extended to the coupling of spectral and finite elements as well as to the coupling of primal and dual methods [3,26,33]. The basic idea is to replace the strong continuity condition at the interfaces between the different subdomains by a weaker one using Lagrange multipliers and making precise choices for the solution and multiplier spaces to obtain optimal error estimates [5]. Given the matching conditions at the interface, there are two different approaches to obtain the discrete solution: imposing the weak continuity on the global discrete space or satisfying this condition by means of Lagrange multipliers. The first technique results in a nonconforming method, where the discrete space is not contained in H^1 Sobolev space (defined later), and the discrete spaces associated with a nested sequence of triangulations are not nested. The second approach gives a saddle point problem and requires the solution of local problems in each subdomain in each iteration step. Other examples of domain decomposition methods in the literature (some defined only at the inter-element, rather than the inter-sub-domain level) can be found in [24,29,11].

* Corresponding author. Tel.: +1 806 742 2568; fax: +1 806 742 1112.
E-mail addresses: eugenio.aulisa@ttu.edu (E. Aulisa), sandro.manservisi@ttu.edu (S. Manservisi), padmanabhan.seshaiyer@ttu.edu (P. Seshaiyer).

0045-7825/$ - see front matter
doi:10.1016/j.cma.2005.10.011

An important issue in these coupling techniques is the construction of efficient iterative solvers for the resulting algebraic linear system [2,10,19]. In the past few years, several researchers have considered multigrid methods as an option for mortar finite elements [9,12,33]. The general idea in many of these papers is to guarantee that the iterate is contained in a subspace where the saddle point problem is positive definite. In many cases this approach requires the exact solution of a modified Schur complement system within each smoothing step which may be too expensive in the multigrid algorithm.

The purpose of this paper is to introduce a flexible multilevel multigrid algorithm for the Navier–Stokes equation which can be used in conjunction with multiprocessor architectures without any major revision or introduction of "ad hoc" structures. For this purpose a Vanka-type solver is used since the local Schur complement relaxation operator relaxes on the single finite element only and therefore is very suitable for parallel computation and subgrid divisions. By using this method it is possible to relax on a single element on the finest grid where we desire accurate solutions and simultaneously relax on a coarse grid level in other parts of the domain, where the solution is not of particular interest. Different physical processes can be modeled independently over different subdomains with different grids with less effort in computation using the proposed methodology.

The paper is organized as follows. In Section 2, the formulation of the problem is presented and in Section 3, the finite element discretization is formulated. In Section 4, numerical results for the parallel computational algorithm, applied to various benchmark applications, are presented and tested.

2. Weak formulation of the Navier–Stokes

For $1 \leqslant p < \infty$ the Sobolev space $W^{s,p}(O)$ is defined as the closure of $C^\infty(O)$ in the norm

$$\|f\|^p_{W^{s,p}(O)} = \sum_{|\alpha|\leqslant s} \int_O \left| \left(\frac{\partial}{\partial x}\right)^\alpha f(x) \right|^p \mathrm{d}x.$$

The closure of $C^\infty_0(O)$ under the norm $\|\cdot\|_{W^{s,p}(O)}$ will be denoted by $W^{s,p}_0(O)$. For $p = 2$, we denote by $H^s(O)$, $s \in \mathbb{R}$, the standard Sobolev space of order s with respect to the set O, which is either the flow domain Ω, or its boundary Γ, or part of its boundary. Hence, we associate with $H^s(O)$ its natural norm $\|f\|_{s,O}$. Whenever possible, we will neglect the domain label in the norm. For vector-valued functions and spaces, we use boldface notation. For example, $\mathbf{H}^s(\Omega) = [H^s(\Omega)]^n$ denotes the space of $\mathbb{R}^n$-valued functions such that each component belongs to $H^s(\Omega)$. Also we denote the space of square integrable functions having zero mean over Ω by

$$L^2_0(\Omega) = \left\{ p \in L^2(\Omega) \,\middle|\, \int_\Omega p \,\mathrm{d}\vec{x} = 0 \right\}.$$

For $\Gamma_1 \subset \Gamma$ with nonzero measure, we also consider the subspace

$$\mathbf{H}^1_{\Gamma_1}(\Omega) = \{\vec{v} \in \mathbf{H}^1(\Omega) | \vec{v} = \vec{0} \text{ on } \Gamma_1\}.$$

Also, we write $\mathbf{H}^1_0(\Omega) = \mathbf{H}^1_\Gamma(\Omega)$. Let $(\mathbf{H}^1_{\Gamma_1})^*$ denote the dual space of $\mathbf{H}^1_{\Gamma_1}$. Note that $(\mathbf{H}^1_{\Gamma_1})^*$ is a subspace of $\mathbf{H}^{-1}(\Omega)$, where the latter is the dual space of $\mathbf{H}^1_0(\Omega)$. The duality pairing between $\mathbf{H}^{-1}(\Omega)$ and $\mathbf{H}^1_0(\Omega)$ is denoted by $\langle\cdot,\cdot\rangle$.

Let $\vec{g}$ be an element of $\mathbf{H}^{1/2}(\Gamma)$. It is well known that $\mathbf{H}^{1/2}(\Gamma)$ is a Hilbert space with norm

$$\|\vec{g}\|_{1/2,\Gamma} = \inf_{\vec{v}\in\mathbf{H}^1(\Omega);\gamma_\Gamma\vec{v}=\vec{g}} \|\vec{v}\|_1,$$

where γ_Γ denotes the trace mapping $\gamma_\Gamma : \mathbf{H}^1(\Omega) \to \mathbf{H}^{1/2}(\Gamma)$. We let $(\mathbf{H}^{1/2}(\Gamma))^*$ denote the dual space of $\mathbf{H}^{1/2}(\Gamma)$ and $\langle\cdot,\cdot\rangle_\Gamma$ denote the duality pairing between $(\mathbf{H}^{1/2}(\Gamma))^*$ and $\mathbf{H}^{1/2}(\Gamma)$.

In order to define a weak form of the Navier–Stokes equations, we introduce the continuous bilinear forms

$$a(\vec{u},\vec{v}) = 2\nu \sum_{i,j=1}^{2} \int_\Omega D_{ij}(\vec{u}) D_{ij}(\vec{v}) \,\mathrm{d}\vec{x} \quad \forall \vec{u},\vec{v} \in \mathbf{H}^1(\Omega) \tag{1}$$

and

$$b(\vec{v}, r) = -\int_\Omega r \nabla\cdot\vec{v} \,\mathrm{d}\vec{x} \quad \forall r \in L^2_0(\Omega), \ \forall \vec{v} \in \mathbf{H}^1(\Omega) \tag{2}$$

and the trilinear form

$$c(\vec{w};\vec{u},\vec{v}) = \int_\Omega \vec{w}\cdot\nabla\vec{u}\cdot\vec{v} \,\mathrm{d}\vec{x} = \sum_{i,j=1}^{2} \int_\Omega w_j \left(\frac{\partial u_i}{\partial x_j}\right) v_i \,\mathrm{d}\vec{x} \quad \forall \vec{w},\vec{u},\vec{v} \in \mathbf{H}^1(\Omega). \tag{3}$$

For details concerning the function spaces we have introduced, one may consult [1,30] and for details about the bilinear and trilinear forms and their properties, one may consult [17,30].

Let Ω be an open domain with boundary Γ. On part of the boundary $\Gamma_1 \subset \Gamma$ we apply Dirichlet boundary conditions. Let $\vec{f} \in \mathbf{L}^2(\Omega)$ be the body force and $\vec{g} \in \mathbf{H}^{1/2}(\Gamma_1)$ be the prescribed velocity over Γ_1 satisfying the divergence free compatibility condition. The velocity, pressure and the stress vector fields $(\vec{u}, p, \vec{\tau}) \in \mathbf{H}^1(\Omega) \times L_0^2(\Omega) \times \mathbf{H}^{-1/2}(\Gamma)$ satisfy the weak-form of the Navier–Stokes equations

$$\begin{aligned} &a(\vec{u},\vec{v}) + c(\vec{u};\vec{u},\vec{v}) + b(\vec{v},p) + \langle \vec{\tau},\vec{v}\rangle_{\Gamma} = \langle \vec{f},\vec{v}\rangle, \\ &b(\vec{u},r) = 0, \\ &\langle \vec{u},\vec{s}\rangle_{\Gamma_1} = \langle \vec{g},\vec{s}\rangle_{\Gamma_1} \end{aligned} \tag{4}$$

for all $(\vec{v}, r, \vec{s}) \in \mathbf{H}^1(\Omega) \times L_0^2(\Omega) \times \mathbf{H}^{-1/2}(\Gamma_1)$. We note that the system (4) must be solved for the stress vector $\vec{\tau} = -\partial\vec{u}/\partial\vec{n} + \vec{n}p \in \mathbf{H}^{-1/2}(\Gamma)$ too and its computation is an important issue. The interested reader can consult [17] and citations therein.

We now partition the domain Ω into m non-overlapping subdomains $\{\Omega^i\}_{i=1}^m$ such that $\partial\Omega^i \cap \partial\Omega^j$ $(i \neq j)$ is either empty, a vertex, or a collection of edges of Ω^i and Ω^j. In the latter case, we denote this interface by Γ^{ij} which consists of individual common edges from the domains Ω^i and Ω^j. We assume zero stress tensor whenever non-Dirichlet boundary conditions are imposed, namely $\langle \vec{\tau},\vec{v}\rangle_{\Gamma\setminus\Gamma_1} = 0$. The velocity, pressure and stress fields $(\vec{u}^i, p^i, \vec{\tau}^{ij}) \in \mathbf{H}^1(\Omega^i) \times L_0^2(\Omega^i) \times \mathbf{H}^{-1/2}(\Gamma^{ij})$ must satisfy the Navier–Stokes equations

$$\begin{aligned} &a(\vec{u}^i,\vec{v}^i) + c(\vec{u}^i;\vec{u}^i,\vec{v}^i) + b(\vec{v}^i,p^i) + \langle \tau^{ij},\vec{v}^i\rangle_{\Gamma^{ij}} = \langle \vec{f},\vec{v}^i\rangle, \\ &b(\vec{u}^i,r^i) = 0, \\ &\langle \vec{u}^i,\vec{s}^i\rangle_{\Gamma_1} = \langle \vec{g}^i,\vec{s}^i\rangle_{\Gamma_1}, \\ &\langle \vec{u}^i - \vec{u}^j,\vec{s}^{ij}\rangle_{\Gamma^{ij}} = 0 \end{aligned} \tag{5}$$

for all $(\vec{v}^i, r^i, \vec{s}^i, \vec{s}^{ij}) \in \mathbf{H}^1_{\Gamma_1}(\Omega^i) \times L_0^2(\Omega_i) \times \mathbf{H}^{-1/2}(\Gamma_1) \times \mathbf{H}^{-1/2}(\Gamma^{ij})$ and $i = 1, 2, \ldots, m$. The problem in (5) is clearly equivalent to (4) and therefore existence and uniqueness results for solutions of the system (5) are well known; see, e.g., [14–16].

Note that problem (5) can be discretized in different ways but our approach will be to pursue discretizations which can assure regularity, accuracy, easy implementations and compatibility with the most common techniques used by fluid dynamics codes. We remark that the computation of $\vec{\tau}^{ij} \in \mathbf{H}^{-1/2}(\Gamma^{ij})$ cannot be in general accurate, especially at corners or singular points, due to its poor regularity [27]. The reader interested in the numerical computation of the stress vector can consult [13]. However it is possible to compute $\vec{\tau}^{ij}$ from $\vec{u}^i$ by using the concept of extended function and extended domain $\hat{\Omega}^i$ of Ω^i as $\Omega = \hat{\Omega}^i$. The solution $\vec{u}^i$ over the domain Ω^i can be extended by using the standard theory [16]. In the rest of the paper we write $\vec{u}^i$ to denote the function over Ω^i and $\hat{u}^i$ its extension to Ω. The extension of $\vec{u}^i$ over Ω satisfies (4) over $\Omega - \Omega^i$ and the stress $\vec{\tau}^{ij}$ can be computed by

$$\langle \vec{\tau}^{ij},\hat{v}^i\rangle_{\Gamma^{ij}} = -a(\hat{u}^i,\hat{v}^i) - c(\hat{u}^i;\hat{u}^i,\hat{v}^i) - b(\hat{v}^i,\hat{p}^i) + \langle \vec{f},\hat{v}^i\rangle \tag{6}$$

for all $\hat{v}^i \in \mathbf{H}^1_{\Gamma_1}(\Omega \setminus \Omega^i)$, which also gives an expression for the computation of the Lagrange multipliers $\vec{\tau}^{ij}$. The representation of $\vec{\tau}^{ij}$ in (6) can be used to speed up and simplify the solution of the system (4) by computing extended solutions through the multigrid method.

3. Finite element discretization

3.1. Introduction

Let us introduce a finite element discretization in each subdomain Ω^i through the mesh parameter h which tends to zero. We consider our discretized domain Ω_h to be partitioned into m non-overlapping subdomains Ω_h^i with boundary $\partial\Omega_h^i$. Now, by starting at the multigrid coarse level l_0, we subdivide Ω_h^i and consequently Ω_h into triangles or rectangles by families of meshes T_h^{i,l_0}. At this coarse level l_0, as at the generic multigrid level l, the triangulation over all Ω_h^i obeys to finite element compatibility constraints along the interfaces Γ_h^{ij}. Based on the simple element midpoint refinement different multigrid levels can be built to reach the finite element mesh $T_h^{i,l}$ over the entire domain Ω_h at the top finest multigrid level n_t. For details on multigrid levels and their construction one may consult [8,28,31]. Now we have meshes at each multigrid level in a standard finite element fashion with compatibility enforced across all the element interfaces built over midpoint refinements. In every macro domain Ω_h^i the Navier–Stokes equation can be solved over a different level l_i generating a solution mesh over Ω_h consisting of different meshes over each subdomain. Let us denote $\Omega_{h_{l_i}}^i$ to be the subdomain i where the solution will be computed at the multigrid level l_i, with h_{l_i} denoting the maximum size of the triangulation of subdomain. It should be noted that the multigrid levels at which the solution is computed over individual subdomains Ω_h^i and Ω_h^j may be different from each other, with no compatibility enforced across the interface Γ_h^{ij}.

Finite element approximation spaces can be generated regularly, as function of the characteristic length h_l over each multigrid level l resulting in different approximation spaces over the solution mesh Ω_h^i. Note that on the solution mesh, we compute the velocity field $\vec{u}_h^i$ at the level l over $\Omega_{h_l}^i$ but the extended function $\hat{u}_h^i$ is defined over all Ω_h with the same basis functions and over each level l in a standard and regular way. There may be parts of the domain where the solution is not computed at the top level but a projection operator from the coarser level can always be used to approximate the solution over the extended domain Ω_h and therefore an approximation to the extended function $\hat{u}_h^i$ is always available. This extended function has the same value at those nodes in the coarser mesh that are included in the finest mesh. This is always the case if the different levels are generated by successive midpoint refinements.

3.2. Multi-level approximation

Let us choose the families of finite-dimensional spaces $\mathbf{X}_{h_l} \subset \mathbf{H}^1(\Omega_h)$ and $S_{h_l} \subset L_0^2(\Omega_h)$. We make the following assumptions on $\mathbf{X}_{h_l}$ and S_{h_l} for the extended functions defined over Ω_h (see, e.g., [15]):

(a) *Approximation hypotheses:* For each multigrid level l there exists an integer l_1 and a constant C, independent of $h_l, \hat{u}$ and p, such that

$$\inf_{\hat{u}_{h_l} \in \mathbf{X}_{h_l}} \|\vec{u} - \hat{u}_{h_l}\|_1 \leqslant C h_l^k \|\vec{u}\|_{k+1} \quad \forall \vec{u} \in \mathbf{H}^{k+1}(\Omega) \cap \mathbf{H}_0^1(\Omega), \;\; 1 \leqslant k \leqslant l_1,$$

$$\inf_{\hat{p}_{h_l} \in S_{h_l}} \|p - \hat{p}_{h_l}\| \leqslant C h_l^k \|p\|_k \quad \forall p \in H^k(\Omega) \cap L_0^2(\Omega), \;\; 1 \leqslant k \leqslant l_1.$$

(b) *Inf–sup or LBB condition:* There exists a constant C', independent of h_l, such that

$$\inf_{0 \neq \hat{q}_{h_l} \in S_{h_l}} \sup_{0 \neq \hat{u}_{h_l} \in \mathbf{X}_{h_l}} \frac{\int_\Omega \hat{q}_{h_l} \nabla \cdot \hat{u}_{h_l} \, \mathrm{d}\vec{x}}{\|\hat{u}_{h_l}\|_1 \|\hat{q}_{h_l}\|_0} \geqslant C' > 0$$

for all multigrid levels $l \leqslant n_t$. This condition assures the stability of the Navier–Stokes discrete solutions.

(c) Let $\mathbf{P}_{h_l} = \mathbf{X}_{h_l}|_\Gamma$, i.e., $\mathbf{P}_{h_l}$ consists of all the restrictions of functions belonging to $\mathbf{X}_{h_l}$ to the boundary Γ. For the subspaces $\mathbf{P}_{h_l}$ we assume the approximation property:
For each multigrid level l there exists an integer l_1 and a constant C, independent of $\vec{s}$ such that

$$\inf_{0 \neq \vec{s}_{h_l} \in \mathbf{P}_{h_l}} \|\vec{s} - \vec{s}_{h_l}\|_{-\frac{1}{2},\Gamma} \leqslant C h_l^k \|\vec{s}\|_{k-\frac{1}{2}} \quad \forall \vec{s} \in \mathbf{H}^{k-\frac{1}{2}}(\Gamma), \;\; 1 \leqslant k \leqslant l_1.$$

See [15,13] for details concerning the approximation on the boundary. With these hypotheses we can build regular conforming approximations over each grid while the approximate solution belongs to $\mathbf{X}_{h_l}$ corresponding to the subdomain $\Omega_{h_l}^i$.

Essential elements of a multigrid algorithm are the velocity and pressure prolongation operators and the velocity and pressure restriction operators. Since we would like to use conforming Taylor–Hood finite element approximation spaces we have the nested finite element hierarchies $\mathbf{X}_{h_0} \subset \mathbf{X}_{h_1} \subset \cdots \subset \mathbf{X}_{h_l}$ and $S_{h_0} \subset S_{h_1} \subset \cdots \subset S_{h_l}$ and the canonical prolongation maps can then be easily obtained. For details and properties one can consult [20,8] and citations therein.

To solve the coupled problem over the domains Ω_h^i $(i = 1, 2, \ldots, m)$ on the level l_i surrounded by the domains Ω_h^j with $j \in I_i$ (I_i is the set of the neighboring regions of i), we discretize (5) to yield the following: Given $\vec{f} \in \mathbf{L}^2(\Omega)$ and $\vec{g} \in \mathbf{H}^{1/2}(\Gamma)$, find $(\vec{u}_{h_{l_i}}^i, p_{h_{l_i}}^i, \vec{\tau}_{h_{l_i}}^{ij}) \in \mathbf{X}_{h_{l_i}}(\Omega_h^i) \times S_{h_{l_i}}(\Omega_h^i) \times \mathbf{P}_{h_{l_i}}(\Gamma_h^{ij})$ satisfying the weak form of the Navier–Stokes equations

$$\begin{aligned}
&a\left(\vec{u}_{h_{l_i}}^i, \vec{v}_{h_{l_i}}^i\right) + c\left(\vec{u}_{h_{l_i}}^i; \vec{u}_{h_{l_i}}^i, \vec{v}_{h_{l_i}}^i\right) + b\left(\vec{v}_{h_{l_i}}^i, p_{h_{l_i}}^i\right) + \left\langle \vec{\tau}_{h_{l_i}}^{ij}, \vec{v}_{h_{l_i}}^i \right\rangle_{\Gamma_h^{ij}} = \left\langle \vec{f}, \vec{v}_{h_{l_i}}^i \right\rangle, \\
&b\left(\vec{u}_{h_{l_i}}^i, r_{h_{l_i}}^i\right) = 0, \\
&\left\langle \vec{u}_{h_{l_i}}^i, \vec{s}_{h_{l_i}}^i \right\rangle_{\Gamma_{1h}^i} = \left\langle \vec{g}_{h_{l_i}}^i, \vec{s}_{h_{l_i}}^i \right\rangle_{\Gamma_{1h}^i}, \\
&\left\langle P_{l_i,l_k}\left(\vec{u}_{h_{l_i}}^i\right) - P_{l_j,l_k}\left(\vec{u}_{h_{l_i}}^j\right), \vec{s}_{h_{l_k}}^{ij} \right\rangle_{\Gamma_h^{ij}} = 0
\end{aligned} \tag{7}$$

for all $(\vec{v}_{h_{l_i}}^i, r_{h_{l_i}}^i, \vec{s}_{h_{l_i}}^i, \vec{s}_{h_{l_k}}^{ij}) \in [\mathbf{X}_{h_{l_i}} \cap \mathbf{H}_{\Gamma_1}^1(\Omega_h^i)] \times S_{h_{l_i}}(\Omega_h^i) \times \mathbf{P}_{h_{l_i}}(\Gamma_{1h}^i \cap \Omega_h^i) \times \mathbf{P}_{h_{l_k}}(\Gamma_h^{ij})$ for $n = 1, 2, \ldots, N$ and $i = 1, 2, \ldots, m$ where $j \in I_i$, $\Gamma_{1h}^i = \Gamma_{1h} \cap \partial\Omega_h^i$ and $l_k = \max\{l_i, l_j\}$ over the multigrid levels available at the boundary Γ_h^{ij}.

In order to have equivalence between the formulation in (5) and (7) the equation $\langle \vec{u}^i - \vec{u}^j, \vec{s}^{ij} \rangle_{\Gamma^{ij}} = 0$ is replaced by $\langle K_h^i \vec{u}_h^i - K_h^j \vec{u}_h^j, \vec{s}_h^{ij} \rangle_{\Gamma_h^{ij}} = 0$ where the operator K_h^i is the projection operator from the trace space $\mathbf{P}_{h_i}$ into itself and must

converge to the identity operator in the continuous limit. In this particular multilevel formulation we write $K_h^i = K_{l_i,l_k}$ to stress the two levels l_i and l_k which are involved in the computation.

In order to ensure maximum accuracy the projection operator $K_{l_i,l_k}(\vec{u})$ projects the velocity from the level l_i to the level l_k which is the finest grid present on the boundary Γ_h^{ij}. With this hypotheses, the Lagrange multipliers $\vec{\tau}_h^{ij}$ can also be discretized and computed on the finest grid available on Γ_h^{ij}. Since the mesh on the multigrid are quite openly constructed, the mesh between the domain Ω_h^i and the neighboring subdomain Ω_h^j can be quite different, as mentioned earlier. Moreover, very fine meshes on the region of interest and coarse meshes elsewhere can be handled with little effort.

Introducing the corresponding finite element matrices the problem (7) is equivalent to

$$\begin{pmatrix} A_l + C_l & B_l^{\mathrm{T}} & D_l \\ B_l & 0 & 0 \\ E_l & 0 & 0 \end{pmatrix} \begin{pmatrix} \vec{u}_{h_l} \\ \vec{p}_{h_l} \\ \vec{\tau}_{h_l}^{ij} \end{pmatrix} = \begin{pmatrix} \vec{F}(\hat{u}_{h_l}^i) \\ 0 \\ \vec{T}(\hat{u}_{h_l}^j) \end{pmatrix}, \tag{8}$$

where D_l is the surface integral matrix, E_l, the velocity constraint at the interface between different meshes, $\vec{F}(\hat{u}_{h_l}^i)$ and $\vec{T}(\hat{u}_{h_l}^j)$ are vectors that represent all terms that do not involve the unknown variables.

We solve the coupled system (8) using an iterative method. Multigrid solvers for coupled velocity/pressure system compute simultaneously the solution for both pressure and velocity and they are known to be one of the best class of solvers for laminar Navier–Stokes equations (see for examples [28,31]). An iterative coupled solution of the linearized and discretized incompressible Navier–Stokes equations requires the approximate solution of large and sparse saddle point problems. In order to optimally solve this system involving the unknown variable $\vec{\tau}_h^{ij}$, we choose Vanka smoother class of solvers which can be considered as block Gauss–Seidel methods where one block consists of a small number of degrees of freedom (for details see [32,21,22,31,23]). The characteristic feature of this type of smoother is that in each smoothing step a large number of small linear systems of equations has to be solved. In the Vanka-type smoother, a block corresponds to all degrees of freedom which are connected to few elements. For example, for conforming finite elements the block may consist of all the elements containing some pressure vertices. Thus, in this case a smoothing step with this Vanka smoother consists of a loop over all the blocks, solving only the equations involving the degrees of freedom related to the elements that are around the pressure vertices. The velocity and pressure variables are updated many times in one smoothing step.

The Vanka smoother employed to solve our problem involves the solution of the minimal number of degrees of freedom for standard conforming Taylor–Hood finite elements but meets enough requirements of robustness. Over the internal region Ω_h^i which is solved at the level l_i and for Taylor–Hood finite elements with linear pressure and the quadratic velocity we solve for all the degrees of freedom of an element and the neighboring midpoints as shown in Fig. 1. Our block consists of four vertex points and 12 middle points to be solved for a total of 36 unknowns. We have also used different blocks with different performances but we have found this particular block to be very robust and reliable even at high Reynolds numbers. Examples of computations with this kind of solvers can be seen in [28,31]. The fact that the solution is searched locally allows us to solve for $\vec{\tau}_h^{ij}$ only near the surface with different meshes and to solve by standard techniques inside the domain.

Note that the solution for $\vec{\tau}_h^{ij}$ may not be well represented in the case in which $\vec{\tau}_h^{ij}$ is strictly in the space $\mathbf{H}^{-1/2}(\Gamma_h^{ij})$. Consider Γ_h^{ij}, the boundary between the domain Ω_h^i solved on the level l_i and Ω_h^j solved on the level l_j. We focus our attention on solution computation on the domain Ω_h^i. If singularities are present it is more convenient computing the $\vec{\tau}_h^{ij}$ by using the prolongation on the finest grid of the velocity $\hat{u}_{h_l}^i$ and pressure $\hat{p}_{h_l}^i$ than solving the system directly. The local nature of the solver allows us to solve exactly $\hat{u}_{h_l}^i$ inside the domain and match the prolongation $\hat{u}_{h_l}^i$ computed by a projection/prolongation from the level l_j to the level l_i. The velocity computed by prolongation over Ω_h^j defines the boundary conditions for the problem in Ω_h^i which does not need any Lagrange multiplier computations. We remark that the Lagrange multiplier $\vec{\tau}_h^{ij}$ does not disappear but is computed from the extended solutions.

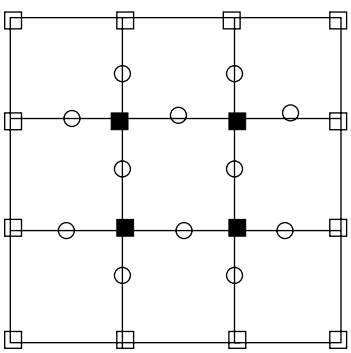

Fig. 1. Relaxation block for the Vanka smoother for standard Taylor–Hood finite elements: four vertex points (square) and 12 middle points (circle).

4. Computational experiments

In this section we present three computational experiments, which employ the multilevel method previously discussed in a parallel multiprocessor architecture. The first example, a standard 2D benchmark computation of flow around a cylinder [28], shows that accurate results can be achieved by using this multilevel method over nonmatching grids. In the second example we show that the same method can be used to investigate domains with different scale geometries where different resolutions are needed. In the third example we show that this multilevel strategy coupled with Vanka-type solvers with parallel implementation can be computationally efficient by reducing the cpu time.

4.1. Numerical experiment 1

The possibility to compute flows over different regions by using different nonmatching meshes is explored in this first test. This involves a standard benchmark application of flow around a cylinder with circular cross-section. The geometry, as shown in Fig. 2, and the boundary conditions for this application are standard and can be found in [28]. The inflow condition is

$$\vec{u}(0,y) = (4U_{\mathrm{m}} y(H-y)/H^2, 0)$$

on Γ_1 where the maximum velocity U_{m} equals 0.3 m/s and $H = 0.41$ m for a Reynolds number equal to 20. The boundary conditions on Γ_2, Γ_3 and Γ_5 are Dirichlet boundary conditions. In this test we compute and compare the drag and the lift force for matching and nonmatching grids in order to show the performance of the coupling technique for nonmatching grids. The drag and the lift forces are defined by [28]

$$F_{\mathrm{D}} = \int_{\Gamma_5} \left(\mu \frac{\partial \vec{u}_t}{\partial \vec{n}} n_y - p n_x \right) \mathrm{d}S, \quad F_{\mathrm{L}} = -\int_{\Gamma_5} \left(\mu \frac{\partial \vec{u}_t}{\partial \vec{n}} n_x + p n_y \right) \mathrm{d}S, \tag{9}$$

where $\vec{u}_t$ is the tangential velocity, $\vec{n} = (n_x, n_y)$ is the inward unit norm and $\vec{t} = (n_y, -n_x)$ the unit tangent. The drag and lift coefficients are

$$C_{\mathrm{D}} = \frac{2F_{\mathrm{D}}}{\rho \bar{u}^2 D}, \quad C_{\mathrm{L}} = \frac{2F_{\mathrm{L}}}{\rho \bar{u}^2 D}$$

with the mean velocity $\bar{u} = 2\vec{u}(0, H/2)/3$. As a further reference value the pressure difference $\Delta P = P(x_a, y_a) - P(x_e, y_e)$ is defined, where the front and the end point of the cylinder $(x_a, y_a) = (.15, .2)$ and $(x_e, y_e) = (.25, .2)$ respectively [28].

In order to show the advantage of the use of nonmatching methods proposed we discretize the domain Ω_h in two main subdomains Ω_1 and Ω_2 as shown in Fig. 2. We discretize Ω_h at level l_0 with coarse triangulation and we define the subsequent levels l_1, l_2, l_3 and l_4 by standard midpoint refinements. We remark that over Γ_5 the midpoint refinement must be performed to conserve the form of the circular boundary. The mesh is a coarse mesh of isoparametric quadrangular finite elements for Q_2/Q_1 velocity/pressure representation. The computations in the region close to the cylinder Ω_1 should be accurate and therefore must be solved on the finest grid l_4. The levels l_4, l_3 or l_2 are considered for computations over Ω_2. Fig. 3 shows, from the top to the bottom, the mesh grid over the finer level l_4 denoted as case A and the mesh grid for the domain-decomposition problem when the level l_3 and the level l_2 are coupled in the subregions Ω_2 (cases B and C). These decompositions do not introduce any supplementary work since they are strictly embedded in the multigrid construction. With Vanka-type solvers we can partition the domain and the processor load at the element block level and therefore in a very efficient and flexible way. The computations are performed on a multiprocessor machine but in this example we focus on the accuracy of the results. The results are reported in Table 1 and can be compared with the existing ones in [28,18]. The test compared in this experiment is the two-dimensional steady case. The focus in this paper is not on the accuracy of the quantities but on the possibility to reach the same results by using coarse meshes in parts of the domain which are not of interest. In order to compare these results the polygon approximating cylinder surface is used as cylinder

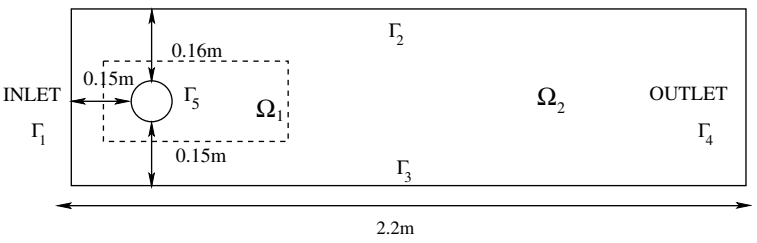

Fig. 2. Geometry of the numerical experiment 1.

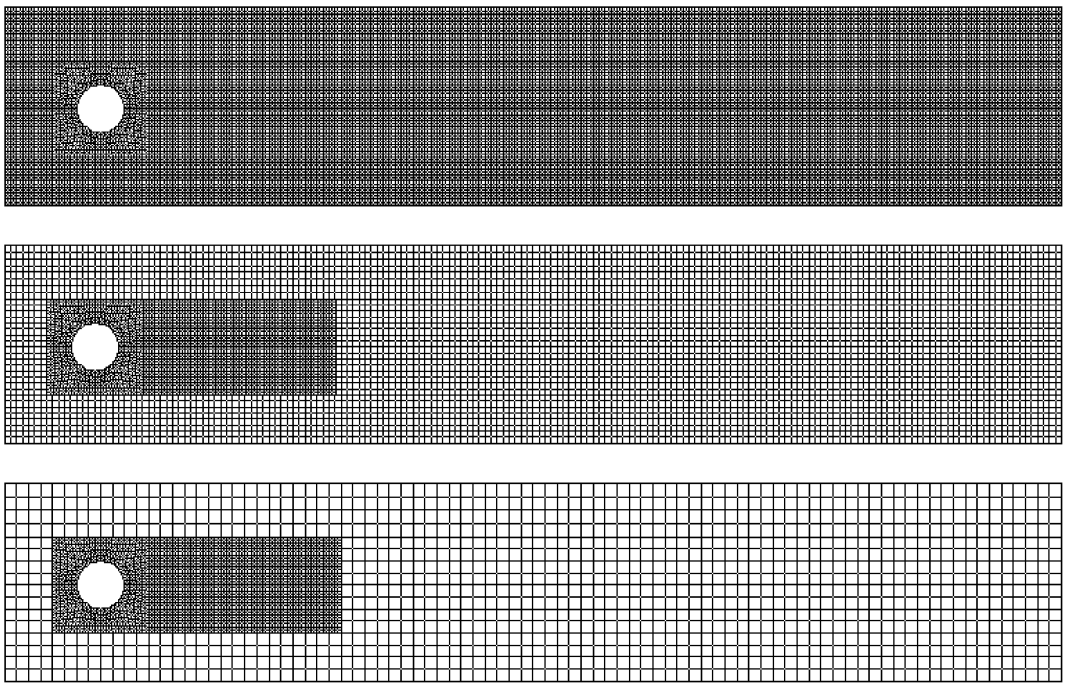

Fig. 3. Different subgrid configurations. Mesh for coupled levels $l_4 - l_4$ on the top (case A), for coupled levels $l_4 - l_3$ in the middle (case B) and for coupled levels $l_4 - l_2$ on the bottom (case C).

Table 1
Results for the computational experiment 1

Grids	Unknowns	C_D	C_L	ΔP	Rel. cpu time
$l_4 - l_4$	112,704	5.59012	0.01100	0.11759	1
$l_4 - l_3$	39,204	5.59015	0.01109	0.117604	0.391
$l_4 - l_2$	20,682	5.59502	0.01123	0.117255	0.1875
$l_3 - l_3$	28,512	5.62150	0.01309	0.115156	0.285
$l_2 - l_2$	7296	5.63402	0.01411	0.115848	0.071

surface at the lower levels. For this reason some small differences between these results and those in [28,18] due to the refinement algorithm of the cylinder surface for different mesh resolutions can be observed. The computation of case A on level $l_4 - l_4$ gives results that confirm with those in literature and can be considered as the benchmark result for the comparison with the multilevel computation of cases B and C. We notice that the results in Table 1 over the level $l_4 - l_3$ (case B) and $l_4 - l_2$ (case C) are very close to those obtained at the level $l_4 - l_4$. The region around the cylinder has the same grid level and therefore almost the same accuracy is expected for the different multilevel computations. However the number of nodes by using nonmatching grid decreases enormously and so the necessary cpu time. It is not easy to compare the cpu time due to the parallel implementation but the computation time on the level $l_4 - l_4$ is approximately five times this of case C. The computations on level $l_3 - l_3$ and $l_2 - l_2$ in Table 1 show different values from the benchmark results over $l_4 - l_4$ and clearly suggest that the nonmatching levels $l_4 - l_3$ and $l_4 - l_2$ can help to achieve better accuracy than this reached by using low level uniform grids.

4.2. Numerical experiment 2

In the second numerical experiment we illustrate an example in which the multilevel method can be efficiently applied to problems in fluid dynamics. We consider challenging domains where small regions of fluids are coupled with large ones, for example the geometry in Fig. 4 where a L-shape domain Ω_0 is shown with eight unitary small square cavities Ω_i, $i = 1, \ldots, 8$. Here the use of a single grid level leads to a very cumberstone implementation since the cavity flow must be solved with different resolution. As in the previous test the first multigrid level l_0 is the coarse mesh designed to contain all the relevant information such as boundary conditions and geometric details. The boundary conditions for this problem are inflow boundary conditions on the bottom of the first branch with parabolic profile (max vel. 1 m/s) and outflow boundary conditions on Γ_2. Dirichlet boundary conditions are applied in the rest of the boundary. The computations are perform in

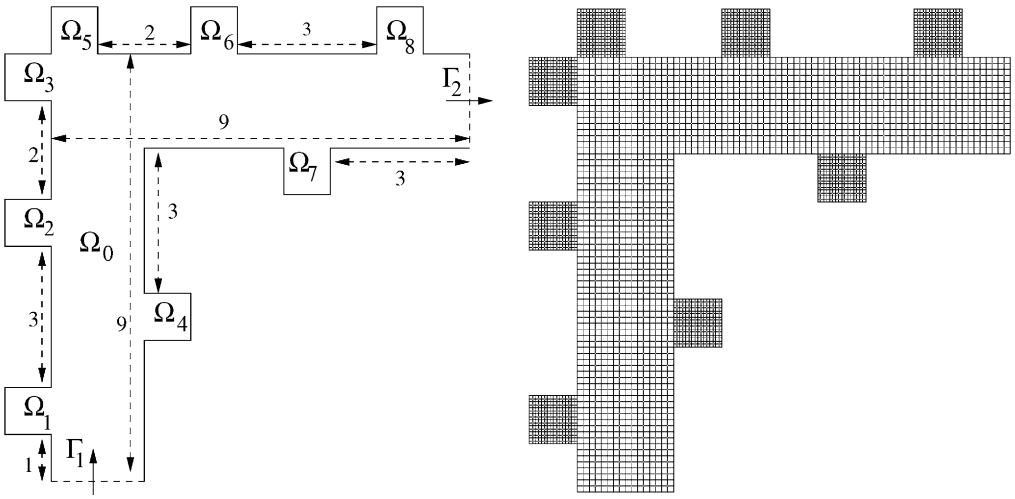

Fig. 4. Geometry for the numerical experiment 2 on the left and domain decomposition with coupled levels $l_4 - l_3$ (case B) on the right.

laminar regime for Reynolds number equals 20 and rectangular finite elements Q_2/Q_1 velocity/pressure are used. The other levels l_i $(i = 1,2,3,4)$ are generated by midpoint refinement. The computations in the cavity regions Ω_i $(i = 1,2,3,\ldots,8)$ should be accurate and therefore solved on the finest grid l_4. The levels l_4, l_3, l_2 or l_1 are considered for computation over Ω_0. In Figs. 4 and 5 we can see the multilevel configurations on nonmatching grids considered in this test when the mesh level l_4 over the cavities Ω_i is coupled with the mesh levels l_3, l_2 and l_1 over Ω_0 (cases B, C and D respectively). The solution, over the uniform level l_4 (case A), is taken as reference solution and is obtained by standard V-cycle multigrid with residual norm in velocity and pressure approximately 10^{-13}. The reference velocity is 1 m/s which is the maximum velocity of the parabolic inflow profile at the inlet. The procedure computes the Lagrange multipliers $\vec{\tau}_h^{ij}$, the boundary stresses, implicitly. In this case the solution $\vec{u}_{h_{l_i}}$ is projected by the standard finite element projection operator (the same of the standard multigrid) over the finest grid at level l_4 obtaining the extended solution for $\hat{u}_h$ over Ω_0. The extended solution generates the boundary conditions for the computation of the solution on the finest grid, which is the union of all Ω_i $(i = 1,2,\ldots,8)$.

The solution along the channel centerlines is reproduced accurately in both branches for all cases A, B, C and D. The solution over all the regions Ω_i for $i = 1,2,\ldots,8$ cannot be captured with a low resolution mesh (for example level l_2) and therefore the multilevel technique becomes a powerful tool in such configurations. Even with very coarse mesh in the channel the fine grid on the cavity allows a good and accurate simulation of the cavity flows. As example in Fig. 6 the v-component of the velocity field is plotted as across the cavities in the regions Ω_1, Ω_2 as a function of the x-coordinate and Ω_6, Ω_7 as a function of the y-coordinate from the top to the bottom and from left to right. The velocity field computations over the different configurations A, B and C cannot be distinguished. The solution for the case D shows that the solution is starting to be different due to the very coarse mesh matching.

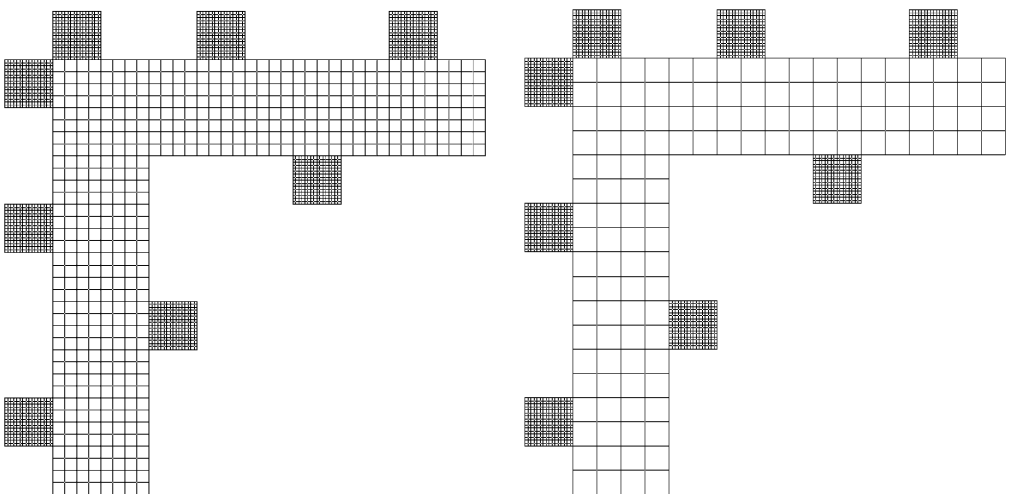

Fig. 5. Multilevel decomposition with coupled levels $l_4 - l_2$ (case C) on the left and with coupled levels $l_4 - l_1$ (case D) on the right.

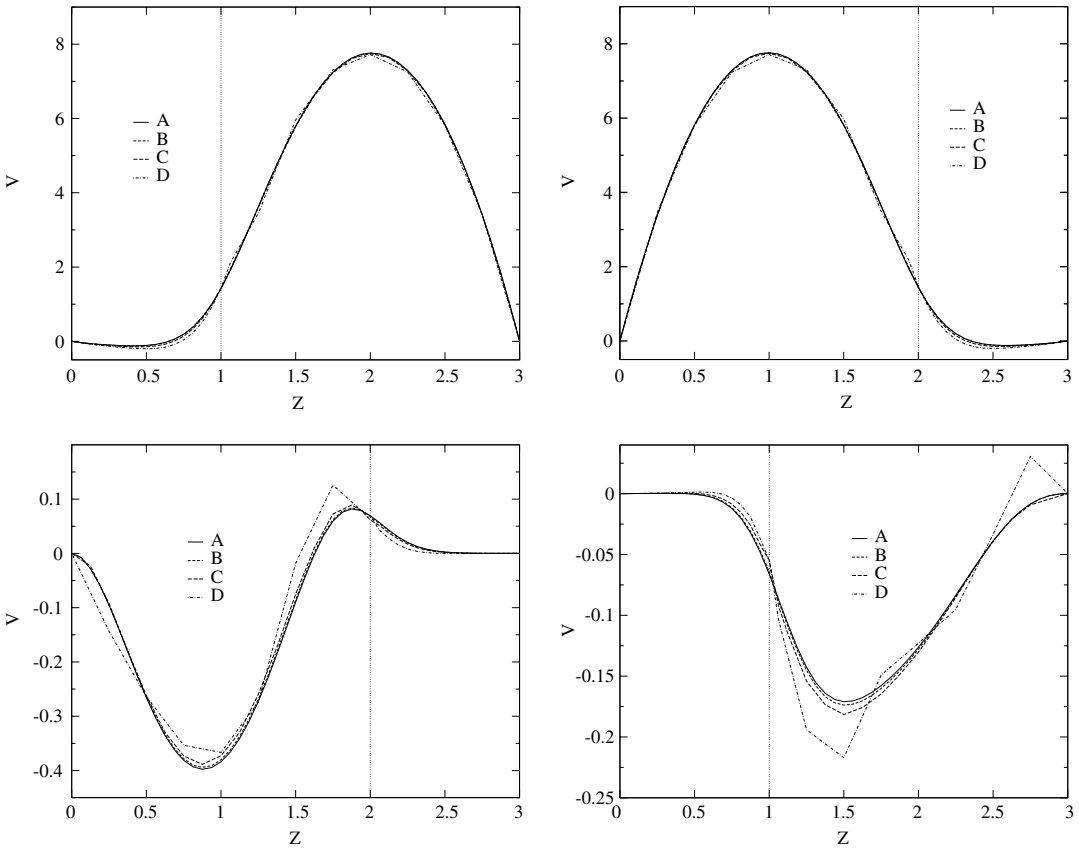

Fig. 6. V-component along the midline in the cavity regions Ω_1 (top left), Ω_2 (top right) as a function of the x-coordinate and Ω_6 (bottom left), Ω_7 (bottom right) as a function of the y-coordinate for the different cases A, B, C and D.

4.3. Numerical experiment 3

In this third numerical experiment parallel computations of a flow through a L-shape channel are investigated. The inflow is a parabolic profile with maximum velocity equal to 1 m/s along the bottom side of the element 4–3. Outflow boundary conditions are prescribed at the outlet, the right side of the elements 31–32 and Dirichlet boundary conditions are applied to the rest of the boundary. All the computations are performed at 50 Reynolds number in the steady laminar regime. This multilevel non matching grid computation can be used to optimize the parallel implementation since different nonmatching grids can be used in different parts of the domain where they are needed. For the application of the multilevel decomposition solution with multiprocessor architecture the processor domain decomposition can be done in several ways. However it is convenient to introduce a decomposition strictly embedded in the multigrid technique that does not introduce any supplementary work by using one processor over suitable block of unknowns. As one can see in Fig. 7 an L-shaped channel Ω_h is divided in four subdomains Ω_i ($i = 1, 2, 3, 4$). On the coarse grid the subdomain Ω_1 consists of four elements (1–4), Ω_2 consists of six elements (17–22), Ω_3 of twelve (5–16) and Ω_4 of ten (23–32). Again rectangular finite elements for Q_2/Q_1 velocity/pressure are used and the finer levels l_i ($i = 1, 2, 3$) are generated by midpoint refinements. The computations in the inlet region Ω_1 and in the region Ω_3 should be accurate and therefore solved on the finest grid l_3. The levels l_3, l_2, l_1 or l_0 are considered for computations over $\Omega_2 \cup \Omega_4$.

Computation on the finer level l_3 over all the domain is denoted by case A and it is considered as benchmark case. In Fig. 7 on the right, the mesh grid for the multilevel problem over level l_3 in the subregions $\Omega_1 \cup \Omega_3$ and over level l_2 in $\Omega_2 \cup \Omega_4$ (case B) is shown. Fig. 8 shows on the left the mesh level l_3 over $\Omega_1 \cup \Omega_3$ coupled with the mesh level l_1 over $\Omega_2 \cup \Omega_4$ (case C) and on the right the level l_3 over $\Omega_1 \cup \Omega_3$ coupled with level l_0 over $\Omega_2 \cup \Omega_4$ (case D). These decompositions are among the numerous possibilities but they do not introduce any supplementary work since they are strictly embedded in the multigrid technique.

With Vanka-type solvers we can partition the domain and the processor load at the coarse element block level and therefore in a very efficient and flexible way. The processors are distributed in different ways for different cases in order to balance the load and to speeding up the computations. In case A all the processors are distributed uniformly over the 32 elements. To optimize the communication time we use only 16 processors and assign the element pairs along the L-shape

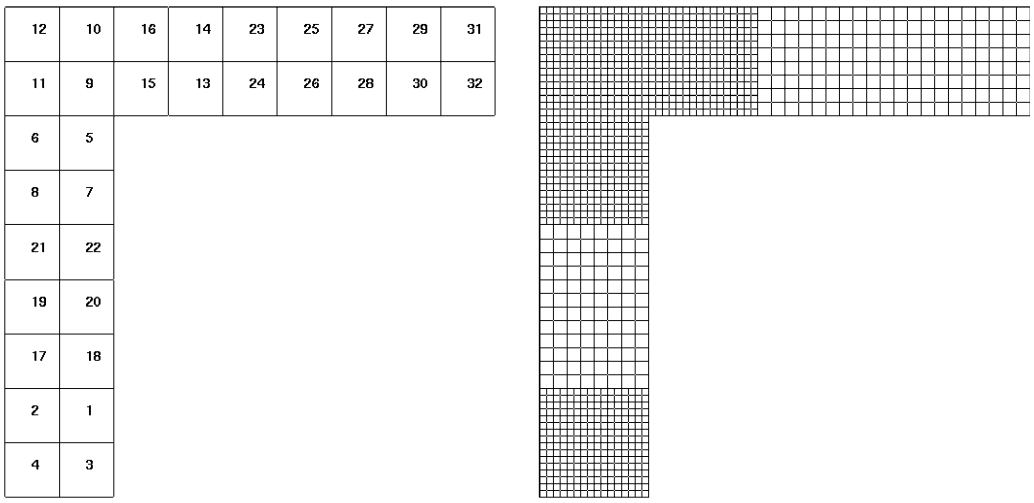

Fig. 7. Element configuration for the numerical experiment 3 at level l_0 on the left and mesh for coupled level $l_2 - l_3$ (case B) on the right.

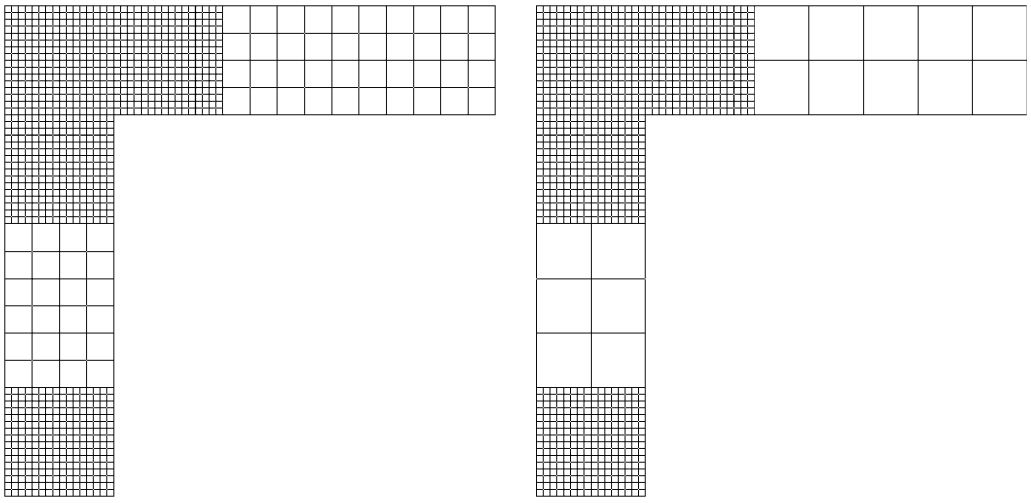

Fig. 8. Different subgrid configurations. Mesh for coupled levels $l_1 - l_3$ on the left (case C) and for coupled levels $l_0 - l_3$ on the right (case D).

to only one processor. We compute and compare the solution of the problem obtained by using 1, 2, 4, 8, and 16 processors respectively. For the cases B, C and D the same distribution of processors, as in the case A, is considered during the V-cycle on the coarse grid (levels l_2, l_1 and l_0) but we rearrange the processor during the computation on the fine grid (level l_3). The computation on level l_3 is over $\Omega_1 \cup \Omega_3$ only and therefore the other processors should be rearranged over this domain. Only 16 elements are available in this case: one for each processor. In the configuration proposed in the case A the solution is obtained at the level l_3 by a standard V-cycle multigrid over all the available levels and it is stopped when the residual of the linear system is 10^{-13} for the velocity and pressure. In the cases B, C and D the multigrid V-cycle is regularly applied over the coarse grid at the level l_2, l_1 and l_0 respectively.

In the Vanka relaxation approach the solution of the multigrid algebraic system requires the solution, block by block, of several small algebraic systems and the iterative update of the solution. We call the solution of this small algebraic systems a block relaxation since this operation gives the solutions and the new update for the block unknowns. Since our solving block is based on an element the update between different processor regions can be performed in many ways. In order to minimize the communication among processors the necessary data exchange for the update during the global relaxation can be performed after a fixed number of block relaxations. In our computations we have explored the different possibilities by computing the two limiting cases: the data exchange is performed after every element block relaxation or the data exchange is performed at the end of a global relaxation (grid relaxation). In Fig. 9 we show the cpu time (on the left) and the relative speeding up (on the right) for the parallel computations for different domain decomposition configurations (cases A, B, C and D) in the limiting case in which the communication of data is performed at the end of each block relaxation. The case A, in which the mesh is uniform over all the domain, is taken as comparison case for speeding up

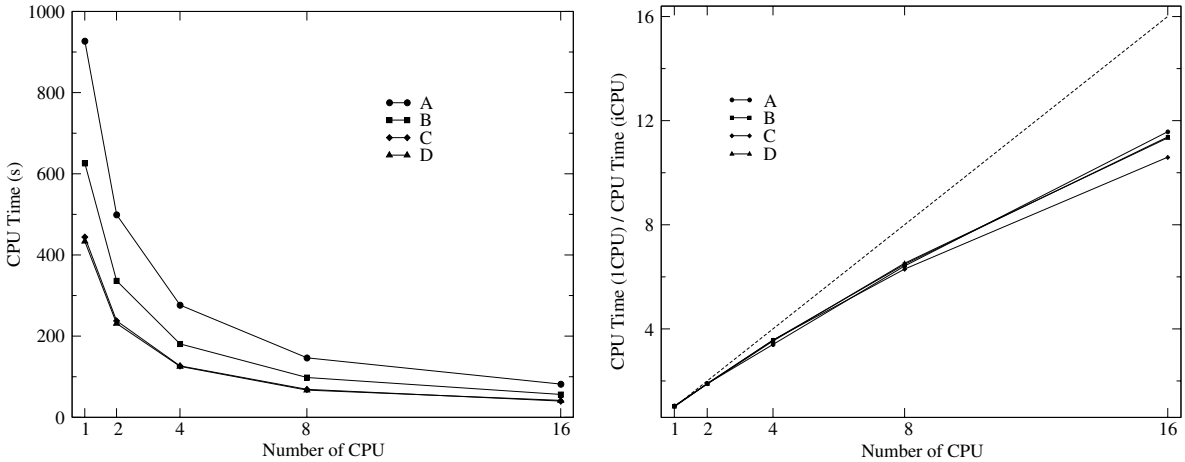

Fig. 9. Cpu time (on the left) and speeding up (on the right) with fixed norm residual (10^{-13}) as a function of the number of the processors for the different cases A, B, C and D with data exchange at the end of each block relaxations.

Table 2
Number of cycles to reach convergence

No. of cpu	1	2	4	8	16
$l_4 - l_4$	11	11	11	11	11
$l_4 - l_3$	12	12	12	12	12
$l_4 - l_2$	12	12	12	12	12
$l_4 - l_1$	13	13	13	13	14

computations. We note that both the cpu time and the relative speeding up scale with the number of processors. The number of V-cycles needed to compute the steady solution are reported in Table 2. Each V-cycle consists of four pre-smoothing and post-smoothing iterations. The number of V-cycle remains basically the same due to good communication among the processors. We note also that the communication time is negligible in almost all these cases. The updating of the solution after a single block relaxation appears to be important for a very fast and regular convergence of the multigrid. In fact the

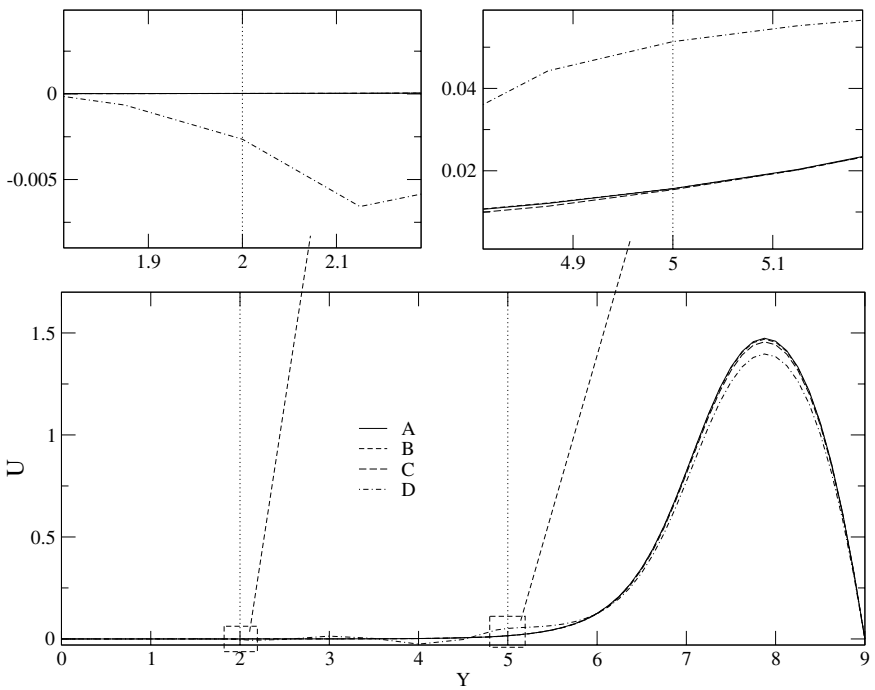

Fig. 10. U-component along the vertical centerline for the different cases A, B, C and D.

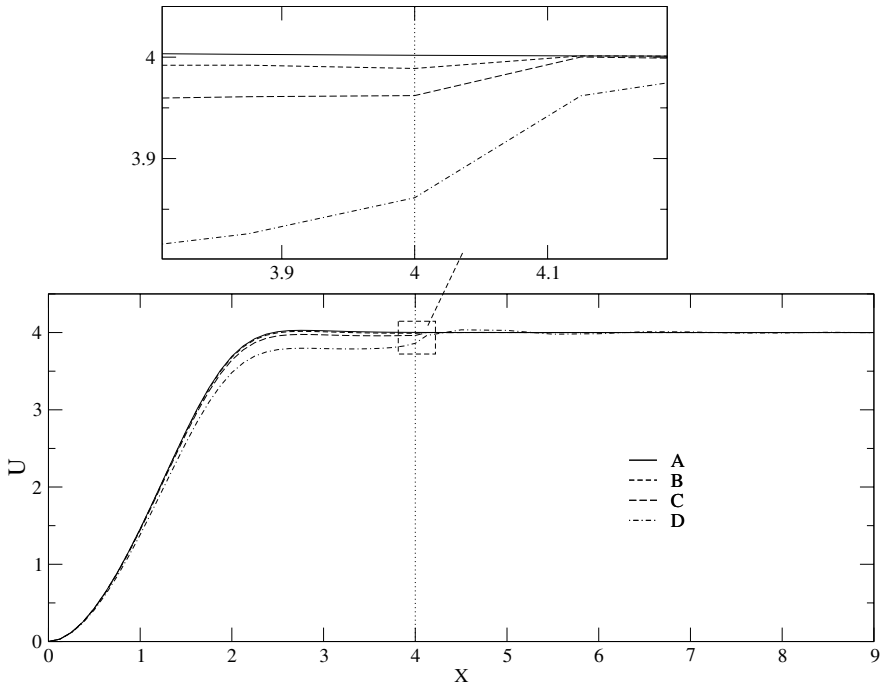

Fig. 11. U-component along the horizontal centerline for the different cases A, B, C and D.

situation is different in the opposite limiting case where the data exchange is performed at the end of a global relaxation. In this case both the cpu time and the relative speeding up do not scale with the number of processors and reach saturation.

In Figs. 10 and 11 the u-component of the extended velocity field for cases A, B, C and D is plotted as a function of the centerline coordinate y across the first branch of the L-shape channel and as a function of the centerline coordinate x across the second branch of the L-shape channel, respectively. The computations over the different configurations A, B and C cannot be distinguished. The solution for the case D shows a different pattern in the region Ω_4 due to the very coarse mesh.

5. Conclusion

In this paper, a multigrid approach to domain decomposition for solving Navier–Stokes equations has been presented and tested to various benchmark applications. Our computational results clearly indicate that the domain decomposition herein used in conjunction with the multigrid method, is a robust and reliable technique for solving the Navier–Stokes system.

The method developed in this paper leads to a fast and flexible algorithm to compute solutions in different domains over different multigrid levels efficiently. The results are very encouraging and the method has prompted us to investigate more open questions.

Acknowledgments

We acknowledge the reviewers for the suggestions that improve noticeably the article. The research of the last author has been supported in part by the National Science Foundation under grant DMS 0207327.

References

[1] R. Adams, Sobolev Spaces, Academic Press, New York, 1975.
[2] Y. Achdou, Y. Maday, O. Pironneau, Substructuring preconditioners for Q_1 mortar element method, Numer. Math. 71 (1995) 419–449.
[3] F. Ben Belgacem, Y. Maday, The mortar finite element method for three dimensional finite elements, Math. Model. Anal. Numer. 31 (1997) 289–302.
[4] F. Ben Belgacem, The mortar finite element method with Lagrange multipliers, Numer. Math. 84 (2) (1999) 173–197.
[5] F. Ben Belgacem, L.K. Chilton, P. Seshaiyer, The *hp*-mortar finite element method for mixed elasticity and Stokes problems, Comput. Math. Appl. 46 (2003) 35–55.
[6] C. Bernardi, Y. Maday, A.T. Patera, Domain decomposition by the mortar element method, in: H.G. Kaper, M. Garbey (Eds.), Asymptotic and Numerical Methods for Partial Differential Equations with Critical Parameters, NATO ASI, Kluwer Academic Publishers, 1993, pp. 269–286.
[7] C. Bernardi, Y. Maday, A.T. Patera, A new nonconforming approach to domain decomposition: the mortar element method, in: H. Brezis, J.-L. Lions (Eds.), Collège de France Seminar, Pitman, 1994, pp. 13-51.

[8] J.H. Bramble, Multigrid methods, Pitman Research Notes in Math, 294, Longman, London, 1993.
[9] D. Braess, W. Dahmen, C. Wieners, A multigrid algorithm for the mortar finite element method, SIAM J. Numer. Anal. 37 (1) (1999) 48–69.
[10] M. Casarin, O. Widlund, A hierarchical preconditioner for the mortar finite element method, ETNA 4 (1996) 75–88.
[11] C. Farhat, F.X. Roux, A method of finite element tearing and interconnecting and its parallel solution algorithm, Int. J. Numer. Meth. Engrg. 32 (6) (1991) 1205–1228.
[12] J. Gopalakrishnan, J.E. Pasciak, Multigrid for the mortar finite element method, SIAM J. Numer. Anal. 37 (3) (2000) 1029–1052.
[13] M. Gunzburger, L.H. Hou, Treating inhomogeneous essential boundary conditions in finite element methods and the calculation of boundary stresses, SIAM J. Numer. Anal. 29 (2) (1992) 390–424.
[14] M. Gunzburger, S. Manservisi, Analysis and approximation of the velocity tracking problem for Navier–Stokes equations with distributed control, SIAM J. Numer. Anal. 37 (5) (2000) 1481–1512.
[15] M. Gunzburger, L. Hou, T. Svobodny, Analysis and finite element approximations of optimal control problems for the stationary Navier–Stokes equations with Dirichlet control, Math. Model. Numer. Anal. 25 (1991) 711–748.
[16] M. Gunzburger, L. Hou, T. Svobodny, Boundary velocity control of incompressible flow with an application to viscous drag reduction, SIAM J. Control Optim. 30 (1992) 167–181.
[17] V. Girault, P. Raviart, The Finite Element Method for Navier–Stokes Equations: Theory and Algorithms, Springer, New York, 1986.
[18] M. Hanke, Benchmarking femlab 3.0a: laminar flows in 2D, Royal Inst. Uppsala Univ., Rep. 2004, 2004, p. 1.
[19] R.H.W. Hoppe, Y. Iliash, Y. Kuznetsov, Y. Vassilevski, B. Wohlmuth, Analysis and parallel implementation of adaptive mortar finite element methods, East–West J. Numer. Math. 6 (1998) 223–248.
[20] V. John, P. Knobloch, G. Matthies, L. Tobiska, Non-nested multi-level solvers for finite element discretisations of mixed problems, Computing 68 (2002) 313–341.
[21] V. John, A comparison of parallel solvers for the incompressible Navier–Stokes equations, Comput. Visual. Sci. 1 (4) (1999) 193–200.
[22] V. John, L. Tobiska, Numerical performance of smoothers in coupled multigrid methods for the parallel solution of the incompressible Navier–Stokes equations, Int. J. Numer. Meth. Fluids 33 (2000) 453–473.
[23] M.F. Paisley, N.M. Bhatti, Comparison of multigrid methods for neutral and stably stratified flows over two-dimensional obstacles, J. Comput. Phys. 142 (1998) 581–610.
[24] P.A. Raviart, J.M. Thomas, Primal hybrid finite element methods for 2nd order elliptic equations, Math. Comput. 31 (1977) 391–396.
[25] P. Seshaiyer, Stability and convergence of non-conforming *hp* finite element methods, Comput. Math. Appl. 46 (2003) 165–182.
[26] P. Seshaiyer, M. Suri, Uniform hp convergence results for the mortar finite element method, Math. Comput. 69 (2000) 521–546.
[27] P. Seshaiyer, M. Suri, *hp* Submeshing via non-conforming finite element methods, Comput. Methods Appl. Mech. Engrg. 189 (2000) 1011–1030.
[28] M. Schafer, S. Turek, The benchmark problem: flow around a cylinder, in: E.H. Hirschel (Ed.), Flow Simulation with High Performance Computers II, Notes on Numerical Fluid Mechanics, vol. 52, Vieweg, 1996, p. 547.
[29] H. Swann, On the use of Lagrange multipliers in domain decomposition for solving elliptic problems, Math. Comput. 60 (1993) 49–78.
[30] R. Temam, Navier–Stokes Equation, North-Holland, Amsterdam, 1979.
[31] S. Turek, Efficient solvers for incompressible flow problems: an algorithmic and computational approach, Lecture Notes in Computational Science and Engineering, vol. 6, Springer, 1999.
[32] S. Vanka, Block-implicit multigrid calculation of two-dimensional recirculation flows, Comput. Methods Appl. Mech. Engrg. 59 (1) (1986) 29–48.
[33] C. Wieners, B. Wohlmuth, The coupling of mixed and conforming finite element discretizations, in: J. Mandel et al. (Eds.), Proceedings of the 10th International Conference on Domain Decomposition, AMS, Contemp. Math., 1998, pp. 546–553.