

On the efficiency and quality of numerical solutions in CFD problems using the Interface Strip Preconditioner for domain decomposition methods.

Rodrigo R. Paz*, Norberto M. Nigro and Mario A. Storti

*Centro Internacional de Métodos Computacionales en Ingeniería (CIMEC)
CONICET - INTEC - U.N.L.
Güemes 3450, (3000) Santa Fe, Argentina*

SUMMARY

In this paper, the efficiency and the quality of the solutions obtained with a new parallelizable preconditioner for Domain Decomposition Methods (DDM) by means of Finite Element discretization of partial differential equations arising in the context of CFD problems is studied. We compare the convergence to the analytical solution or measured data for problems that have been considered as 'benchmarks' in the computational fluid dynamic literature. For this purpose, we study the solution obtained via parallelized iterative methods that have been extensively used (e.g. CG and GMRES global iteration and its variants) in CFD computations and those obtained with the Interface Strip (IS) preconditioner for the Schur Complement method. The idea is to present the new solver as an alternative to obtain more accurate and faster solutions in the context of monolithic and non-monolithic CFD schemes. For this purpose, the confined internal flows and external viscous compressible/incompressible flow around (complex-shape) bodies are considered.

Copyright © 2005 John Wiley & Sons, Ltd.

KEY WORDS: computational fluid dynamics, iterative solutions, parallel computing, domain decomposition

1. INTRODUCTION

In Reference [1], Storti *et al.* introduced the mathematical formulation and the sequential implementation of a new preconditioner for Schur complement Domain Decomposition

*Correspondence to: Rodrigo R. Paz, Centro Internacional de Métodos Computacionales en Ingeniería (CIMEC). Güemes 3450, (3000) Santa Fe, Argentina.
E-mail: rodrigop@@intec.unl.edu.ar

Contract/grant sponsor: Consejo Nacional de Investigaciones Científicas y Técnicas, Agencia Nacional de Promoción Científica y Tecnológica and Universidad Nacional del Litoral; contract/grant number: PIP 198/*Germen-CFD*, ANPCyT-PID-99/74 *FLAGS*, ANPCyT-FONCyT-PICT-6973 *PROA* and CAI+D-UNL-PIP-02552-2000

Methods (i.e. the IS Preconditioner) based on solving a problem in a narrow strip around the interface between sub-domains. Also, convergence properties of the iterative solution step were contrasted with several preconditioners [3, 6, 5] for the discretization of the Poisson and the scalar convection-diffusion equations. In Reference [2] the implementation details in parallel environments, were presented. Besides, in this work it is showed that the IS preconditioner is much less memory and time-consuming than classical optimal preconditioners such as Neumann-Neumann [3, 4], in the context of the solution of non-symmetric linearization of equations arised from the discretization of the governing equations of hydrological problems. The performance of this preconditioner was assessed with an analytical study of Schur complement matrix eigenvalues.

The aim of this article is to study the efficiency of this new preconditioner and the quality of the solution obtained when classical Finite Element formulation is applied to a partial differential equations that describe different kind of flows. Also, we will show that the computational effort and/or resources could be insufficient if we try to solve some problems (i.e. the linear system that results at each time simulation step) with classical global iterative methods, like Krylov Subspace methods (e.g. Conjugate Gradient and Generalized Minimum RESidual and its variants). For this purpose, we study the CPU consumed time, the required core memory and the accuracy in the solution of several problems in Computational Fluid Dynamics.

Moreover, we present this new solver as an alternative to obtain more accurate and faster solutions in the context of monolithic and non-monolithic CFD schemes.

2. THE NAVIER-STOKES EQUATIONS

We focus this work in the solution of both compressible and incompressible Navier-Stokes equations with the SUPG and SUPG-PSPG methods proposed by Brooks *et al.* [30] and Tezduyar *et al.* [19], respectively.

2.1. Incompressible Navier-Stokes equations.

The incompressible Navier-Stokes equations present two important difficulties for the solution with finite elements. First, the character of the equation becomes more and more advective dominant when the Reynolds number increases. In addition the incompressibility condition represents not an evolution equation but, rather, a constraint on the equations. This has the drawback that only some combination of interpolation spaces for velocity and pressure can be used, namely those ones that satisfy the so called Brezzi-Babūška condition. In the formulation of Tezduyar *et al.* advection is stabilized with the well known SUPG stabilization term, and a similar stabilization term called PSPG is included in order to stabilize incompressibility. In this way, it is possible to use stable equal order interpolations. Once these equations. are discretized, the resulting system of ODE's are discretized in time with the standard trapezoidal rule (backward and forward Euler and Crank-Nicolshon schemes are allowed to use). At every time step, the resulting non-linear system of equation is solved iteratively.

Viscous flow is well represented by Navier-Stokes equations. The incompressible version of this model includes the mass and momentum balances that can be written in the following form. Let $\Omega \in \mathbb{R}^{n_{sd}}$ and $(0, T)$ be the spatial and temporal domains respectively, where n_{sd} is

the number of space dimensions, and let Γ be the boundary of Ω .

$$\begin{aligned} \nabla \cdot \mathbf{u} &= 0 & \text{in } \Omega \times (0, T) \\ \rho \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) - \nabla \cdot \boldsymbol{\sigma} &= \mathbf{0} & \text{in } \Omega \times (0, T), \end{aligned} \quad (1)$$

with ρ and \mathbf{u} the density and velocity of the fluid and $\boldsymbol{\sigma}$ the stress tensor, given by

$$\begin{aligned} \boldsymbol{\sigma} &= -p\mathbf{I} + 2\mu^* \boldsymbol{\epsilon}(\mathbf{u}) \\ \boldsymbol{\epsilon}(\mathbf{u}) &= 1/2(\nabla \mathbf{u} + (\nabla \mathbf{u})^t) \end{aligned} \quad (2)$$

where p is the pressure and μ^* is the effective dynamic viscosity defined as sum of the dynamic (molecular) viscosity and the algebraic eddy viscosity of the LES model proposed by Smagorinsky [16], i.e. $\mu^* = \mu + \mu_{SGS}$. \mathbf{I} represents the identity tensor and $\boldsymbol{\epsilon}$ the strain rate tensor. The initial and boundary conditions are

$$\begin{aligned} \Gamma &= \Gamma_g \cup \Gamma_h \\ \Gamma_g \cap \Gamma_h &= \emptyset \\ \mathbf{u} &= \mathbf{g} & \text{at } \Gamma_g \\ \mathbf{n} \cdot \boldsymbol{\sigma} &= \mathbf{h} & \text{at } \Gamma_h, \\ \mathbf{u}(t=0) &= u_0 & \forall x \in \Omega \\ p(t=0) &= p_0 & \forall x \in \Omega \end{aligned} \quad (3)$$

where Γ_g and Γ_h are the Dirichlet and Neumann boundaries, respectively. When the flow velocity is very small, the fluid is very viscous; or the geometric dimensions are very small (i.e. when Reynolds number is very small), the inertial term in (1) play a minor role and can be neglected. The flow is then dominated by the viscous and pressure.

2.1.1. Spatial discretization The spatial discretization is equal order for pressure and velocity and is stabilized through the addition of two operators. Advection at high Reynolds numbers is stabilized with the well known SUPG operator, while the PSPG operator proposed by Tezduyar *et al.* [19] stabilizes the incompressibility condition, which is responsible of the checkerboard pressure modes.

The computational domain Ω is divided in n_{el} finite elements Ω_e , $e = 1, \dots, n_{el}$, and let \mathcal{E} be the set of these elements, and H^{1h} the finite dimensional space defined by

$$H^{1h} = \left\{ \phi^h \mid \phi^h \in C^0(\overline{\Omega}), \phi^h|_{\Omega^e} \in P^1, \forall \Omega^e \in \mathcal{E} \right\}, \quad (4)$$

with P^1 representing polynomials of first order. The functional spaces for weight and interpolation are defined as

$$\begin{aligned} S_{\mathbf{u}}^h &= \{ \mathbf{u}^h \mid \mathbf{u}^h \in (H^{1h})^{n_{sd}}, \mathbf{u}^h \doteq \mathbf{g}^h \text{ on } \Gamma_g \} \\ V_{\mathbf{u}}^h &= \{ \mathbf{w}^h \mid \mathbf{w}^h \in (H^{1h})^{n_{sd}}, \mathbf{w}^h \doteq \mathbf{0} \text{ on } \Gamma_g \} \\ S_p^h &= \{ q \mid q \in H^{1h} \}. \end{aligned} \quad (5)$$

The SUPG-PSPG is written as follows: Find $\mathbf{u}^h \in S_u^h$ and $p^h \in S_p^h$ such that

$$\begin{aligned}
& \int_{\Omega} \mathbf{w}^h \cdot \rho \left(\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \nabla \mathbf{u}^h \right) + \int_{\Omega} \epsilon(\mathbf{w}^h) : \sigma^h d\Omega + \\
& + \underbrace{\sum_{e=1}^{nel} \int_{\Omega} \delta^h \cdot \left[\rho \left(\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \nabla \mathbf{u}^h \right) - \nabla \cdot \sigma^h \right]}_{(SUPG \text{ term})} + \\
& + \underbrace{\sum_{e=1}^{nel} \int_{\Omega} \epsilon^h \cdot \left[\rho \left(\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \nabla \mathbf{u}^h \right) - \nabla \cdot \sigma^h \right]}_{(PSPG \text{ term})} + \\
& + \int_{\Omega} q^h \nabla \cdot \mathbf{u}^h d\Omega = \int_{\Gamma_h} \mathbf{w}^h \cdot \mathbf{h}^h d\Gamma \quad \forall \mathbf{w}^h \in V_u^h, \forall q^h \in V_p^h
\end{aligned} \tag{6}$$

The stabilization parameters and the resultant system of equations are defined in Reference [19]

2.1.2. Disaggregated Scheme Fractional step methods for the incompressible Navier-Stokes equations have enjoyed popularity in the last two decades. The reason for this relies on the computational efficiency of these methods, basically because of the uncoupling of the pressure from the velocity components. In Reference [17] the study of computed pressure stability of schemes that use a pressure Poisson equation was presented. We use these results in this paper.

The results to be presented refer to second-order algorithm based on the Crank-Nicholson discretization for the viscous and convective terms and a second-order pressure splitting, leaving the pressure gradient at a given time level in the first step and computing its increment in the second one (see Reference [17]).

Let us introduce the discretization in time in the compact matrix form of the problem (1).

$$\mathbf{M} \Delta t \mathbf{U}^n + \mathbf{K}(\mathbf{U}^{n+\theta}) \mathbf{U}^{n+\theta} + \mathbf{G} \mathbf{P}^{n+1} = \mathbf{F}^{n+\theta} \tag{7}$$

$$\mathbf{D} \mathbf{U}^{n+1} = 0 \tag{8}$$

The fractional step schemes we will consider applied to the fully discrete problem (7) and (8) is exactly equivalent to

$$\mathbf{M} \frac{1}{\Delta t} (\hat{\mathbf{U}}^{n+1} - \mathbf{U}^n) + \mathbf{K}(\mathbf{U}^{n+\theta}) \mathbf{U}^{n+\theta} + \gamma \mathbf{G} \mathbf{P}^n = \mathbf{F}^{n+\theta} \tag{9}$$

$$\mathbf{M} \frac{1}{\Delta t} (\mathbf{U}^{n+1} - \hat{\mathbf{U}}^{n+1}) + \mathbf{G} (\mathbf{P}^{n+1} - \gamma \mathbf{P}^n) = 0, \tag{10}$$

$$\mathbf{D} \mathbf{U}^{n+1} = 0 \tag{11}$$

where $\hat{\mathbf{U}}^{n+1}$ is an auxiliary variable and γ is a numerical parameter, whose values of interest are 0 and 1. We make the essential approximation

$$\mathbf{K}(\mathbf{U}^{n+\theta}) \mathbf{U}^{n+\theta} \sim \mathbf{K}(\hat{\mathbf{U}}^{n+\theta}) \hat{\mathbf{U}}^{n+\theta} \tag{12}$$

where $\hat{\mathbf{U}}^{n+\theta} = \theta\hat{\mathbf{U}}^{n+1} + (1-\theta)\mathbf{U}^n$. If we write \mathbf{U}^{n+1} in terms of $\hat{\mathbf{U}}^{n+1}$ using (10) and inserting the result in (11), the equations to be solved are

$$\mathbf{M}\frac{1}{\Delta t}(\hat{\mathbf{U}}^{n+1} - \mathbf{U}^n) + \mathbf{K}(\hat{\mathbf{U}}^{n+\theta})\hat{\mathbf{U}}^{n+\theta} + \gamma\mathbf{G}\mathbf{P}^n = \mathbf{F}^{n+\theta}, \quad (13)$$

$$\Delta t\mathbf{D}\mathbf{M}^{-1}\mathbf{G}(\mathbf{P}^{n+1} - \gamma\mathbf{P}^n) = \mathbf{D}\hat{\mathbf{U}}^{n+1}, \quad (14)$$

$$\mathbf{M}\frac{1}{\Delta t}(\mathbf{U}^{n+1} - \hat{\mathbf{U}}^{n+1}) + \mathbf{G}(\mathbf{P}^{n+1} - \gamma\mathbf{P}^n) = 0, \quad (15)$$

which have been ordered according to the sequence of solution, for $\hat{\mathbf{U}}^{n+1}$, \mathbf{P}^{n+1} and \mathbf{U}^{n+1} . We can approximate the operator $\mathbf{D}\mathbf{M}^{-1}\mathbf{G}$ in (14) to the Laplace operator if \mathbf{M} is approximated by a diagonal matrix.

2.2. The Compressible Navier-Stokes equations.

The differential form of the conservation equations of mass, momentum and total energy that governs the dynamics of compressible and viscous fluid flow may be written in a compact vector form as (Einstein summation convention is assumed):

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial(\mathbf{F}_a)_i}{\partial x_i} = \frac{\partial(\mathbf{F}_d)_i}{\partial x_i} + \mathbf{G} \quad (16)$$

$$\mathbf{U} = (\rho \quad \rho \mathbf{u} \quad \rho e)$$

where \mathbf{U} is the unknown state vector expressed in conservative variables as above, e represents the specific total energy, \mathbf{F}_a accounts for the advective fluxes, \mathbf{F}_d for the diffusive fluxes and \mathbf{G} is used for the external source terms (i.e $\mathbf{G} = (0, \rho \mathbf{f}_e, W_f + q_H)$, $W_f = \rho \mathbf{f}_e \cdot \mathbf{u}$ is the work done by the external forces \mathbf{f}_e and \mathbf{n} represents the boundary unit normal vector).

The above mentioned advective and diffusive fluxes are defined as:

$$\mathbf{F}_a = \begin{pmatrix} \rho u_i \\ \rho u_1 u_i + \delta_{i1} p \\ \rho u_2 u_i + \delta_{i2} p \\ \rho u_3 u_i + \delta_{i3} p \\ (\rho e + p) u_i \end{pmatrix} \quad \mathbf{F}_d = \begin{pmatrix} 0 \\ T_{i1} \\ T_{i2} \\ T_{i3} \\ T_{ik} u_k - q_i \end{pmatrix} \quad (17)$$

Here δ_{ij} is the Kronecker operator or the components of the identity tensor, q_i are the components of the heat flux vector, and T_{ij} are the components of the Newtonian viscous stress tensor: $\mathbf{T} = 2\mu\epsilon(\mathbf{u}) - 2/3\mu(\nabla \cdot \mathbf{u})\mathbf{I}$. The strain rate tensor ϵ is equal to the equation (2). q_i is the heat flux defined according to the Fourier law assumptions as: $q_i = -\kappa\nabla T$ with κ the thermal conductivity and T the temperature.

The coefficients of viscosity and thermal conductivity are assumed be given by the Sutherland formula,

$$\begin{aligned}\mu &= \mu_0 \left(\frac{T}{T_0} \right)^{3/2} \frac{T_0 + 110}{T + 110} \\ \kappa &= \frac{\gamma R \mu}{(\gamma - 1) Pr}\end{aligned}\tag{18}$$

where μ_0 is the viscosity at the reference temperature T_0 and Pr is the Prandtl number.

The physical model is closed by the definition of the constitutive law for the specific internal energy in terms of the thermodynamic state and some state equation for the thermodynamic variables, normally an ideal gas law is adopted:

$$\begin{aligned}\rho e &= \frac{p}{\gamma - 1} + 1/2 \rho \|\mathbf{u}\|^2 \\ p &= \rho RT\end{aligned}\tag{19}$$

where $R = (\gamma - 1)C_v$ is the particular gas constant and $\gamma = \frac{C_p}{C_v}$ is the ratio of the specific heat at constant pressure relative to that at constant volume.

2.2.1. Finite element formulation for fix domains The stabilized finite element formulation of the quasi-linear form of (16) is written as follows: find $\mathbf{U}^h \in \mathcal{S}^h$ such that $\forall \mathbf{W}^h \in \mathcal{V}^h$

$$\begin{aligned}& \int_{\Omega} \mathbf{W}^h \cdot \left(\frac{\partial \mathbf{U}^h}{\partial t} + \mathbf{A}_i^h \frac{\partial \mathbf{U}^h}{\partial x_i} - \mathbf{G} \right) d\Omega + \int_{\Omega} \frac{\partial \mathbf{W}^h}{\partial x_i} \cdot \mathbf{K}_{ij}^h \frac{\partial \mathbf{U}^h}{\partial x_j} d\Omega - \int_{\Gamma_h} \mathbf{W}^h \cdot H^h d\Gamma + \\ & + \sum_{e=1}^{n_{el}} \int_{\Omega^e} (\tau)^T (\mathbf{A}_k^h)^T \frac{\partial \mathbf{W}^h}{\partial x_k} \cdot \left\{ \frac{\partial \mathbf{U}^h}{\partial t} + \mathbf{A}_i^h \frac{\partial \mathbf{U}^h}{\partial x_i} - \frac{\partial}{\partial x_i} \left(\mathbf{K}_{ij}^h \frac{\partial \mathbf{U}^h}{\partial x_j} \right) - \mathbf{G} \right\} d\Omega + \\ & + \sum_{e=1}^{n_{el}} \int_{\Omega^e} \delta_{shc} \frac{\partial \mathbf{W}^h}{\partial x_i} \cdot \frac{\partial \mathbf{U}^h}{\partial x_i} d\Omega = \mathbf{0}\end{aligned}\tag{20}$$

where

$$\begin{aligned}\mathcal{S}^h &= \{ \mathbf{U}^h | \mathbf{U}^h \in [\mathbf{H}^{1h}(\Omega)]^{ndf}, \mathbf{U}^h|_{\Omega^e} \in [P^1(\Omega^e)]^{ndf}, \mathbf{U}^h = \mathbf{g} \text{ on } \Gamma_g \} \\ \mathcal{V}^h &= \{ \mathbf{W}^h | \mathbf{W}^h \in [\mathbf{H}^{1h}(\Omega)]^{ndf}, \mathbf{W}^h|_{\Omega^e} \in [P^1(\Omega^e)]^{ndf}, \mathbf{W}^h = \mathbf{0} \text{ on } \partial\Omega_g \}\end{aligned}\tag{21}$$

where matrices \mathbf{A}_i and \mathbf{K}_{ij} are defined in Reference [20] and in Reference [21].

The first three terms inside the first two integrals in the variational formulation of (20) constitute the Galerkin formulation of the problem, the third integral accounts for the Neumann boundary conditions. The first series of element level integrals in (20) are the SUPG stabilization terms added to prevent spatial oscillations in the advection-dominated range. The second series of element level integrals in (20) are the shock capturing terms added to assure the stability at high Mach and Reynolds number flows, specially to suppress spurious overshoot and undershoot effects in the vicinity of discontinuities.

The stabilization parameters are defined in Reference [20]

3. SCHUR COMPLEMENT DOMAIN DECOMPOSITION METHOD

We consider solving in each time step a linearized form of systems (1) and (16) (i.e. $\mathbf{A}u = f$) resulting from finite element discretization as described in the previous sections. Let Ω denote the computational domain, and $\{\Omega^i\}_{i=1}^n$ its decomposition into n non-overlapping sub-domains. We shall re-order the state variables vector u and the forces vector f as $u = (u_L, u_I)^T$ and $f = (f_L, f_I)^T$, numbering the global nodes such that the coefficient matrices of the state variables (velocity, pressure and eventually the energy) assume block-ordered structure

$$\mathbf{A} = \begin{bmatrix} A_{LL} & A_{LI} \\ A_{IL} & A_{II} \end{bmatrix}, \quad (22)$$

where $A_{LL} = \text{diag}[A_{11}, A_{22}, \dots, A_{nn}]$ is a block-diagonal with each block A_{ii} , $i = 1, 2, \dots, n$ being the matrix corresponding to the unknowns belonging to the *interior* vertices of sub-domain Ω_i . A_{LI} and A_{IL} represents connections between sub-domains to interfaces.

A_{II} corresponds to the discretization of the differential operator restricted to the interfaces and represents the coupling between local interface points.

The numerical solution of $\mathbf{A}u = f$ is equivalent to solving

$$\mathbf{S}u_I = g \quad \text{on interfaces } \Gamma, \quad (23)$$

$$\mathbf{A}_{LL}u_L = f_L - \mathbf{A}_{LI}u_I \quad \text{in } \Omega_i \quad (24)$$

where

$$\mathbf{S} = \mathbf{A}_{II} - \sum_{i=1}^n \mathbf{A}_{IL} \mathbf{A}_{LL}^{-1} \mathbf{A}_{LI}, \quad (25)$$

and

$$g = f_I - \sum_{i=1}^n \mathbf{A}_{IL} \mathbf{A}_{LL}^{-1} f_L, \quad (26)$$

where \mathbf{S} is the well-known Schur complement matrix.

The Schur domain decomposition method starts by first determining u_I on the interfaces between sub-domains by solving (23). Upon obtaining u_I , the sub-domain problems (24) decouple and may be solved in parallel. The main computational cost for the iterative solution of (23) depends on the number of iteration, i.e. the condition number, to achieve convergence to a given accuracy criterion.

In order to further improve the efficiency of iterative methods, a preconditioner has to be added so that the condition number of the Schur complement matrix is lowered. In Reference [1], Storti *et al.* studied the spectrum of the Schur complement matrix in order to develop a suitable preconditioners. This is one of the most important issues in preconditioning techniques.

4. THE INTERFACE STRIP PRECONDITIONER FOR DOMAIN DECOMPOSITION METHODS

Consider a sub-domain interface with a strip of two element layers ($n_{\text{lay}} = 2$), as shown in figure (1).

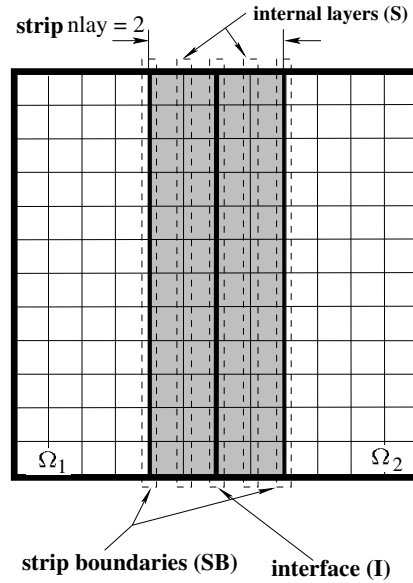


Figure 1. Strip Interface problem.

The preconditioning consists in, given a vector f_I defined on the nodes at the interface (I in the figure) to compute an approximate solution v_I given by

$$\begin{bmatrix} A_{II} & A_{IS} & A_{I,SB} \\ A_{SI} & A_{SS} & A_{S,SB} \\ A_{SB,I} & A_{SB,S} & A_{SB,SB} \end{bmatrix} \begin{bmatrix} v_I \\ v_S \\ v_{SB} \end{bmatrix} = \begin{bmatrix} f_I \\ 0 \\ 0 \end{bmatrix}, \quad (27)$$

with ‘Dirichlet boundary conditions’ at the strip boundary $v_{SB} = 0$, so that it reduces to

$$\begin{bmatrix} A_{II} & A_{IS} \\ A_{SI} & A_{SS} \end{bmatrix} \begin{bmatrix} v_I \\ v_S \end{bmatrix} = \begin{bmatrix} f_I \\ 0 \end{bmatrix}, \quad (28)$$

Once this equation is solved, v_I is the value of the proposed preconditioner applied to f_I , i.e.

$$v_I = \mathcal{P}_{IS}^{-1} f_I \quad (29)$$

A direct solution of this interface problem is not easily parallelizable. This approach would involve transferring all the interface matrix to a single processor and solving the problem there. So that, the possibility is to partition the strip problem among processors, much in the same way as the global (i.e. the non-overlapping Domain Decomposition method) problem is, and solving the strip problem by an iterative method. The idea of an iterative method is also suggested by the fact that the preconditioning matrix (i.e. the matrix obtained by assembling on the strip domain with Dirichlet boundary conditions at the strip boundary) is highly diagonal dominant for narrow strips. Care must be taken to avoid nesting a *non-stationary* method like CG or GMRES inside another outer non-stationary method [9]. We recall that in a *stationary* method the solution x at the iteration k depends, only, on the solution at the previous step (i.e., $x_k = f(x_{k-1})$), then we can find the guess x_k after k

successive applications of the same operator to the initial value x_0 . The problem here is that a non-stationary method executed a finite number of times *is not* a linear operator, unless the inner iterative method is iterated enough and then approaches the inverse of the preconditioner. In this respect, relaxed Richardson iteration is suitable.

For the Richardson interface problem, a fixed predetermined number m of Richardson iterations are performed. If m is too low, then the preconditioner has no effect, and if it is too large the efficiency of the preconditioner tends to saturate, while the cost is roughly proportional to m , so in general there is an optimal value for m . We have found that adjusting m so that Richardson iteration converges one order of magnitude (relative to the initial residual) is fine for most problems. Note that the number of iterations may depend on the intrinsic conditioning of the interface problem and also on the strip width. For small strip widths ($n_{\text{lay}} < 5$) m was chosen in the range $5 \leq m \leq 10$.

A subsequent possibility is preconditioning the Interface Strip preconditioner problem itself with block Jacobi.

In general, in parallel implementation, each processor may have several sub-domains. In this way, the memory and time computation requirements (i.e. the cost to factorize smaller matrices) are reduced. If the number of dof's in the interfaces grows toward the number of total dof's the method results in a fully iterative method.

Even if the preconditioner has been described through figures in terms of finite element structured meshes, the implementation is purely algebraic (in contrast to previous approaches, like, notably, the wire-basket ones, see References [7] and [8]) based on the graph connectivity of the matrix. The preconditioner has been implemented in a FEM production code [10] and it is used to solve large scale problems with unstructured 1D/2D/3D meshes.

5. NUMERICAL TEST IN SEQUENTIAL ENVIRONMENT

The hypersonic flow over a flat plate test.

In this section we analyze the hypersonic flow around a flat plate, a typical flow problem where the nonlinearities are so high that any difficulty in the convergence of the linear system may influence the nonlinear convergence and finally make the solution to blow up. This problem, deeply documented by James Carter [11], has from the physical point of view a strong interaction between the boundary layer and the shock wave and also a discontinuity introduced at the flat plate leading edge where the flow has to stagnate from a very high free stream velocity. Both are sources of numerical drawbacks making this test a very challenge problem. Figure (2) shows the problem definition with a sketch of the physical structures present in the flow field and the boundary condition applied to it.

For solving this problem a compressible laminar Navier-Stokes flow solver was used with a constant viscosity for a Reynolds number of 10^4 based on the flat plate length and the free stream state. The Mach number was chosen as 5. This problem was successfully solved using IISD+ISP solver but it was not possible to obtain a solution with a global GMRES solver, using in both cases a Krylov subspace dimension of 200. In this last case after some time steps the solution began to present a poor resolution of the strong shock wave and finally the solution crashed.

Figures (3,4) show the skin friction coefficient and the Stanton number against theoretical

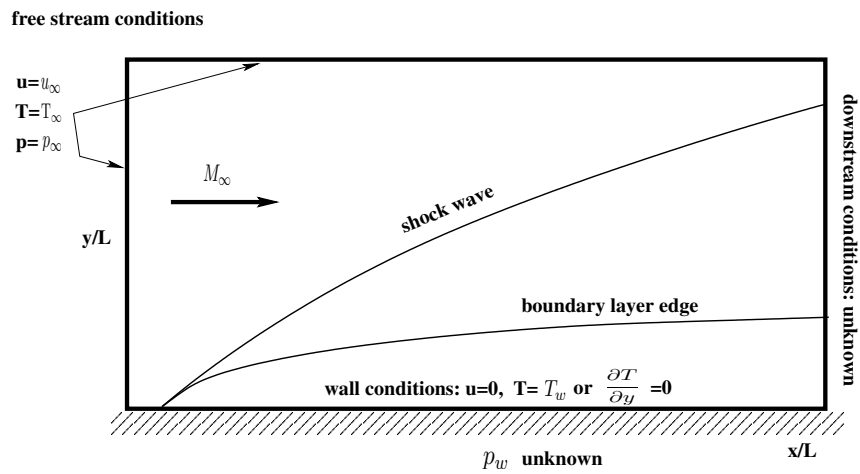


Figure 2. Problem definition.

predictions based on analytical solutions of an approximate theory. These results show an acceptable behavior of the numerical results relative to the analytical predictions.

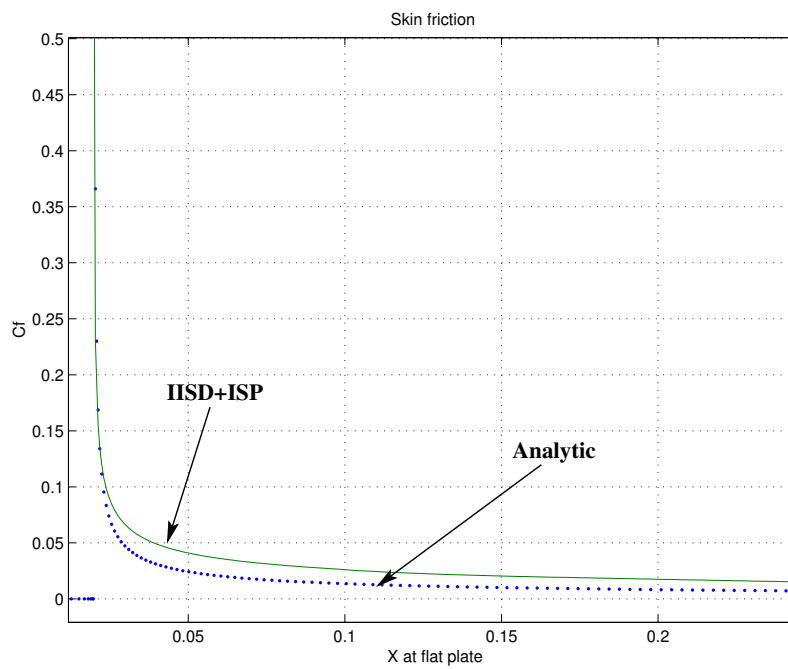


Figure 3. Skin friction coefficient.

The mesh used was composed by 24150 quadrangular elements and 24462 nodes. In order

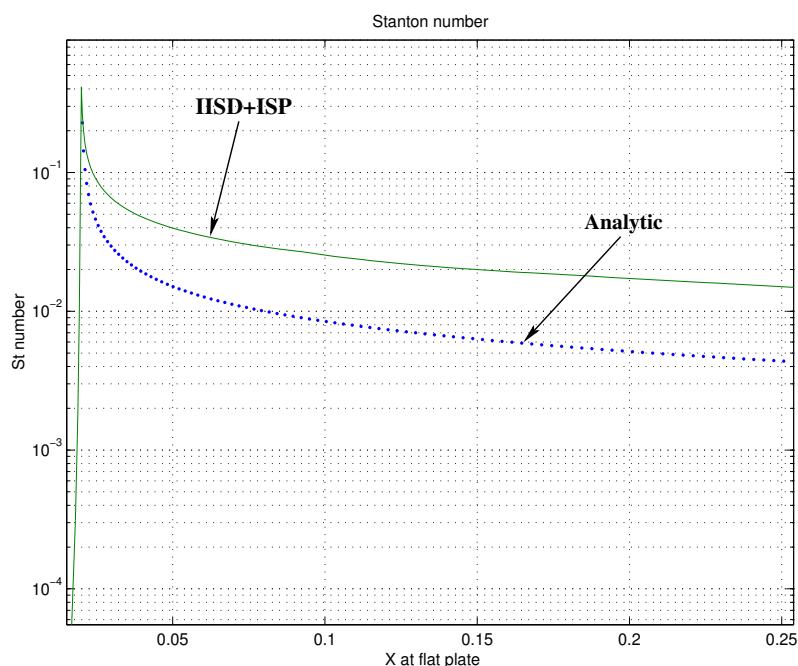


Figure 4. Stanton number.

to capture the high thermal and flow gradients the normal spacing close to the flat plate was chosen about 4×10^{-6} .

This other kind of examples is for cases where it is not possible to get a solution using single processor architectures (as in the case of Global GMRES iteration). The physical solution contains some features that are hard to be solved. Also in such a case IISD+ISP does not show difficulties to get an accurate solution.

6. NUMERICAL TESTS AND EXAMPLES IN PARALLEL ENVIRONMENT

In this section, we study the quality of numerical results for several problems considered as benchmarks in CFD community. The tests were carried out on a *Beowulf* cluster of PC's. The cluster at CIMEC laboratory has twenty (uniprocessor) nodes; where 10 nodes are Pentium IV - 2.4 GHz, 1 GB RAM (DDR, 333 MHz), 7 nodes Pentium IV - 1.7 GHz, 512 MB RAM (RIMM, 400/800 MHz) and 2 nodes Pentium IV 1.7 GHz, 256 MB RAM (RIMM, 400/800 MHz). Usually, the first node works as server. The nodes are connected through a switch Fast Ethernet (100 Mbit/sec, latency= $O(100)$ μ secs).

We develop the code PETSc-FEM (see Reference [10]) that is a general purpose, parallel, multi-physics FEM program for CFD applications based on MPI and PETSc libraries (see [22] and [23], respectively). PETSc-FEM comprises both a library that allows the user to develop FEM (or FEM-like, i.e. non-structured mesh oriented) programs, and a suite

of application programs (e.g. compressible/incompressible Navier-Stokes, multi-phase flow, compressible Euler equations, shallow water model, general advective-diffusive systems, coupled surface/subsurface water flow over multi-aquifer systems, linear elasticity and Laplace equation). These involves two parts:

- The element routines does the computation of the state vector, the residual and/or the stiffness matrix, etc.
- The main algorithm which describes the general strategies of resolution (i.e. linear/non-linear, steady/unsteady, etc.). In this, the application writer calls a set of routines that read the mesh, assemble MPI/PETSc vectors and matrices, check convergence, etc.

Mesh partitioning is performed by using METIS and the library takes in charge of passing the appropriate elements to each processor, to find out the vectors, to assemble the residual and matrices, and to fix the boundary conditions, for all the processors. It is written in the C++ language with an Object Oriented Programming philosophy, but always keeping in mind the scope of efficiency.

6.1. The Stokes flow in a long horizontal channel.

Initiated by observed discrepancies between the results of experiments and the results of computer simulations [12] using standard solvers [13] (i.e. GMRES with acceptable rates of convergence), this example shows the improvement of the solutions of lubricated contacts by means of the IS preconditioner. Until now in tribology the lubricant flow in the narrow gap between two contacting elements has been described using the Reynolds equation. This equation follows from the Navier-Stokes equations when a narrow gap is assumed (i.e. when $e = H/L \ll 1$ if H is the gap width and L a characteristic length scale). Nominally the assumption $e \ll 1$ will generally hold. However, technical surfaces are never perfectly smooth and at least locally in the contact, due to surface roughness, it may be violated. Accurate description of the flow then requires the use of Stokes or Navier-Stokes equations. Because the viscosity will generally be large, convective terms are likely to be negligible. Therefore we restrict ourselves to the Stokes equations.

The channel is $8 \cdot 10^{-5}$ m width and $9 \cdot 10^{-2}$ m long. The dynamic viscosity used is $5.33 \cdot 10^{-4}$ m²/sec and no body forces are considered. The aspect ratio of the quadrilateral elements is 5 to assure non-stretched elements. This problem conducts to a ill-conditioned matrix due to the high aspect ratio of channel dimensions and it is necessary a big number of residual vectors (iterations) in Krylov methods to converge to an accurate solution. We consider the nonlinear steady simulation with a maximum of 100 Newton loops. The normalized residuals in the solution step of the linear system are shown in figure (5) for all nonlinear iterations. In the case of global GMRES solver two Krylov subspace dimension are considered (i.e. 400 y 800). The test was conducted in 16 nodes and each sub-domain was sub-partitioned into 7 interior sub-domains (2,000 dof's per interior sub-domains in average) in the case of IISD+ISP solver. The interface strip width used is $n_{lay} = 1$. Although the Stokes problem leads to a symmetrical matrix, it is not positive definite. For this reason GMRES iteration is accomplished. The core memory demanded for the IISD+ISP solver was 48.9 Mb per processor at each Newton iteration, including the factorization stage (solution of local problems) and the GMRES iteration. The CPU consuming time was 0.33 minutes per Newton iteration.

The memory used in the global GMRES stage was 126.9 Mb per processor for a Krylov

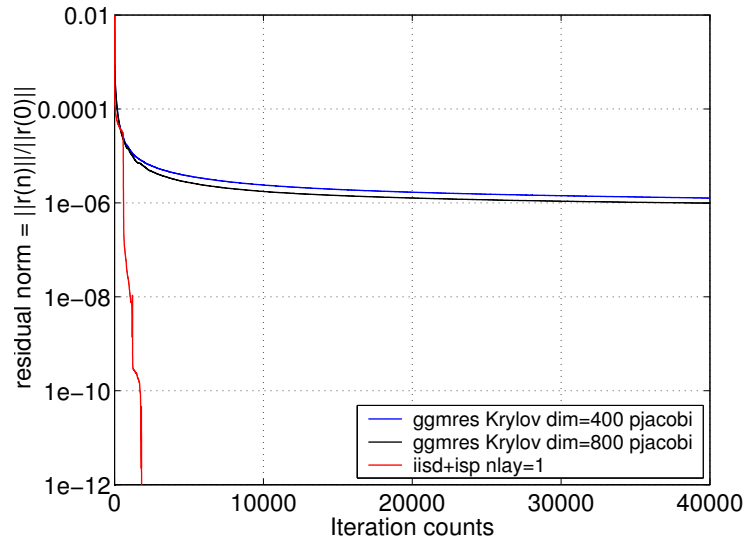


Figure 5. Iteration counts.

subspace dimension (KSP_{dim}) of 800 and 63.2 Mb per processor for a KSP_{dim} of 400. The CPU time was 6.61 minutes and 1.87 minutes per Newton interior, respectively.

In figures (6), (7), (8) and (9) the numerical solution of the horizontal velocity and pressure fields are compared with the analytical one. Figures (6) and (7) correspond to the solution of both fields after one loop in the Newton scheme and figures (8) and (9) correspond to ten iterations in Newton's loop. For the IISD+ISP solver the residual of the Newton loop after three iterations was 10^{-14} and we consider that there is no need to go further in this loop to converge to the solution.

Figures (6) and (8) show a slight loss in momentum due to the coarse discretization in the transversal direction in order to maintain the aspect ratio of the elements.

This example was the first evidence that inspire this work. It shows that for high aspect ratio geometries the global GMRES suffers of a strong convergence deterioration and even though using an unusually high size of Krylov subspace dimension the final solution is unacceptable.

6.2. The Navier-Stokes flow around an infinite cylinder.

The unsteady viscous external flows past objects have been studied (experimentally and numerically [24, 14, 25, 26, 27, 28, 29]) extensively because of their many practical applications. This example is directly related to a great number of problems. For example, airfoils are made into streamline shapes in order to increase the lifts, and at the same time, reducing the aerodynamic drag exerted on the wings. On the other hand, flow past a blunt body, such as a circular cylinder (e.g. the wind forces acting on the tensors of a hanging bridge), usually experiences boundary layer separation and very strong flow oscillations in the wake region behind the body. In certain Reynolds number range, a periodic flow motion will develop in the wake as a result of boundary layer vortex being shed alternatively from either side of

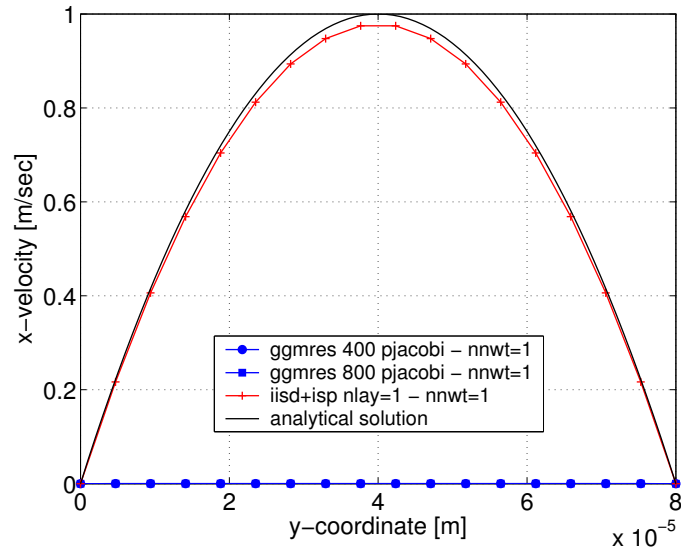


Figure 6. Velocity field in the channel height (nnwt=1).

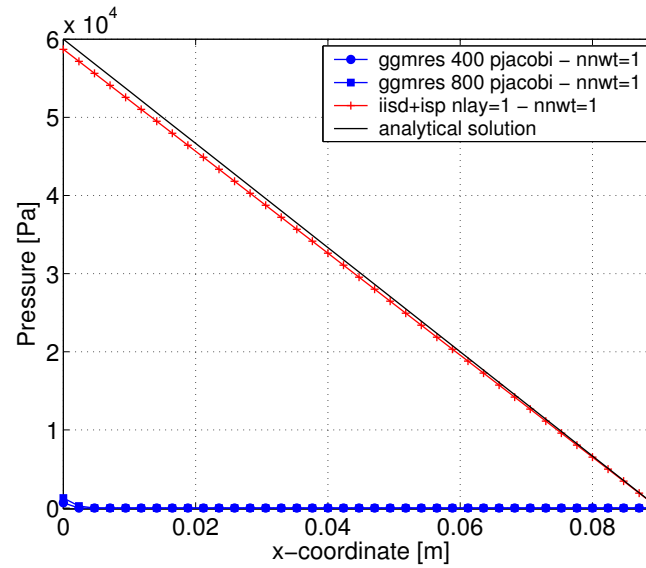


Figure 7. Pressure field along channel (nnwt=1).

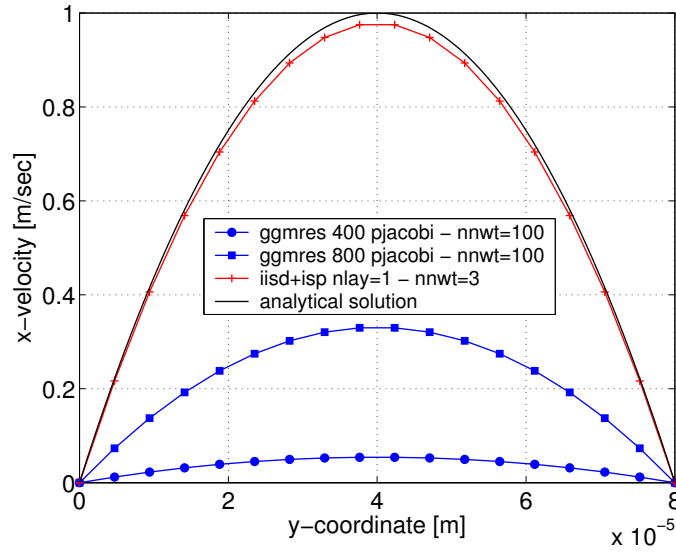


Figure 8. Velocity field in the channel height (nnwt=100 for gmres, nnwt=3 for IISD+ISP).

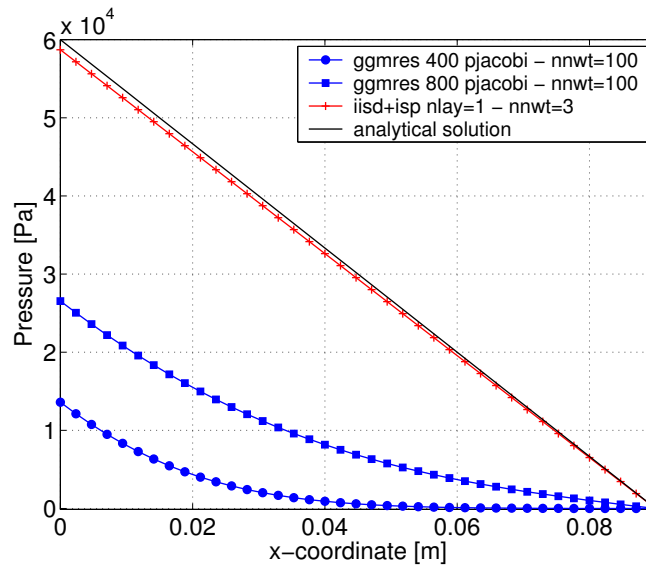


Figure 9. Pressure field along channel (nnwt=100 for gmres, nnwt=3 for IISD+ISP).

the cylinder. This regular pattern of vortices in the wake is called a Kármán vortex street. It creates an oscillating flow at a discrete frequency that is correlated to the Reynolds number of the flow. The periodic nature of the vortex shedding phenomenon can sometimes lead to

unwanted structural vibrations, especially when the shedding frequency matches one of the resonant frequencies of the structure.

In the case of numerical simulation of the unsteady flow over a circular cylinder the destabilizing effects are not absent but the simulation time needed to make them appreciable could be outside. As the geometry of the flow and the initial boundary conditions are symmetric, the Navier-Stokes equations obviously lead to a symmetric solution even for $Re \sim 100$. The round-off errors are, of course, disturbances which could eventually generate vortex shedding. Nevertheless, numerical simulations for Reynolds numbers of 100 and 200 and even up to 1000 have shown a symmetric pattern in the flow. In order to generate vortex shedding an artificial perturbation may be imposed, for example introducing a rotation of the cylinder for a short time. The perturbations introduced correspond to a clockwise rotation of the cylinder followed by a counterclockwise rotation (it is of the same nature as the perturbation used by Braza *et al* [24]). The mesh of 138,600 quadrilaterals (with homogeneous refinement near the cylinder wall) was partitioned into 15 sub-domains (processors) and sub-partitioned into 14 interior (local) sub-domains in average (2,000 dof's per local subdivision).

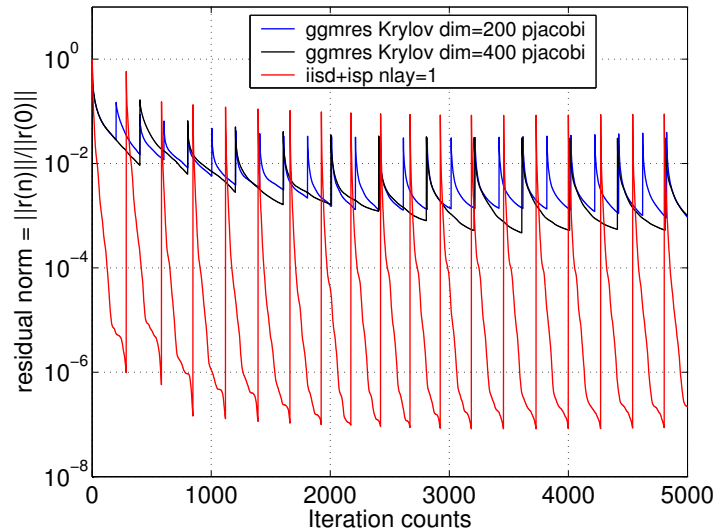
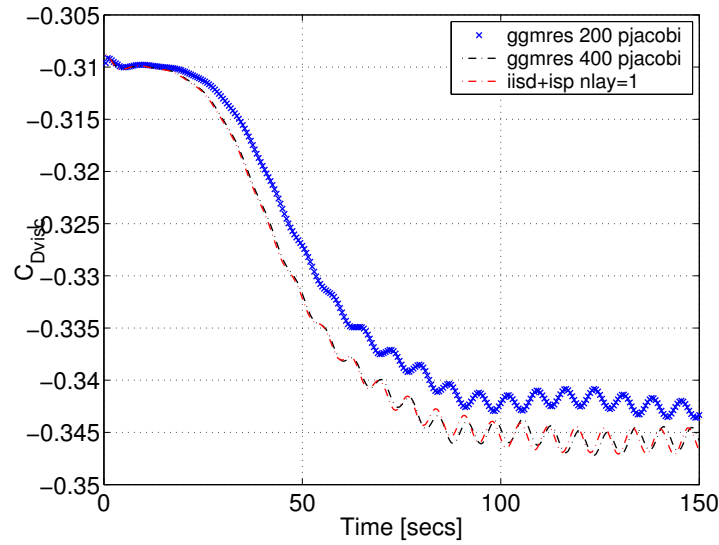
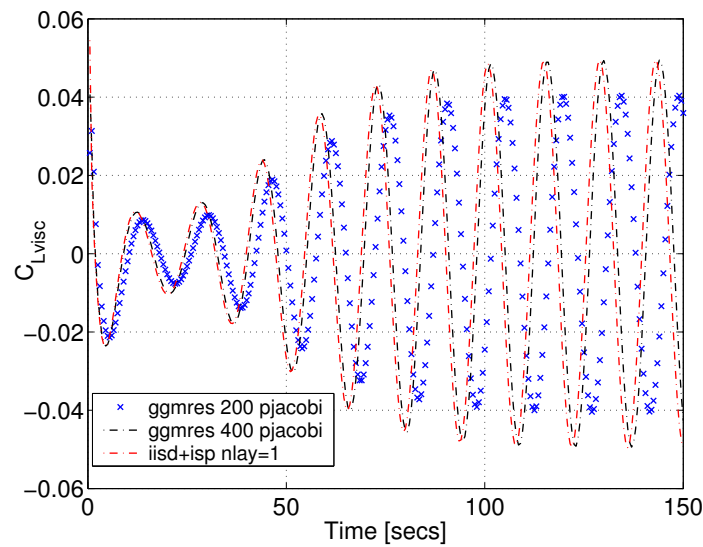


Figure 10. $Re = 100$. Iteration counts.

In figure (10) we see the iteration number for several time steps for the unsteady simulation of this flow. We consider only one Newton iteration for time step. Clearly, the IISD+ISP solver reaches the lower residual tolerances (10^{-7} vs. 10^{-3}) with 70 % less iterations (roughly) compared to global GMRES iteration with different KSP_{dim} . The lower tolerances achieved with IISD are directly related to the accuracy in the solution and the reduction of the iteration number has influence in the overall simulation time.

In figures (11), (12) and (13) we see the viscous forces and moment evolution in time. The solution obtained with both Global Iteration ($KSP_{dim} = 400$) and IISD+ISP are in agreement with the experimental results reported by Braza *et al.*, and with the numerical results shown by Shih *et al.* and Behr *et al.* (References [24, 25, 26]). However, if we stop Global

Figure 11. $Re = 100$. viscous x -force coefficient.Figure 12. $Re = 100$. viscous y -force coefficient.

iteration prematurely (i.e. $KSP_{dim} = 200$, see figure (10)) the solution is no more accurate and divergences of 50% are observed (see figure (13)). Although the residuals are lowered between two and three orders of magnitude the error in the solution of the linear system is not accurate enough. We recall that to pass from 200 iterations to 400 iterations in GMRES scheme the computational resources (CPU and consumed memory) are considerably increased due to the

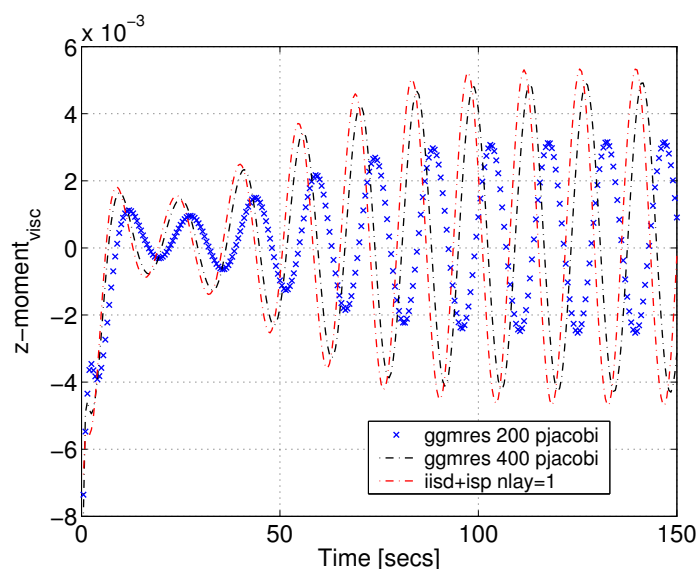


Figure 13. $Re = 100$. viscous z -moment coefficient.

Krylov base orthonormalization against all previous residual vectors and the resolution of a minimization problem. The CPU consumed time and core memory requirements for each time step were: 28 secs and 100 Mbytes for Global GMRES with $KSP_{dim} = 200$, 107.5 secs and 152 Mbytes for Global GMRES with $KSP_{dim} = 400$ and 18.5 secs and 98 Mbytes for IISD+ISP.

This well known test case shows the necessity of global GMRES to be used with a commonly high Krylov subspace dimension size to reach the accuracy of IISD+ISP solver. Commonly the user is pushed to adopt a small value of Krylov subspace dimension in order to reduce memory and cpu time consumed. The sensitivity of the results obtained using global GMRES to the Krylov subspace dimension size is high, and with no a priori knowledge of this dimension the uncertainties in the results tend to be high. Therefore, IISD+ISP makes the simulation less user-dependent.

6.3. Navier-Stokes flow using the Fractional Step Scheme. The Lid Driven Cavity

A test of the disaggregated method was performed on two-dimensional unit cavity flow at $Re = 1000$. This is a test that has been computed extensively in the past and it is well understood (see Reference [18]). A structured mesh of 400×400 quadrilateral was used for calculations. We used 14 processor for Global GMRES iteration and 14 sub-domains subdivided into local partitions in order to have 1,500 dof's per subdivision for the IISD+ISP solver. The time step and γ used in equations (13-15) were $\Delta t = 0.05$ (see Reference [17]) and $\gamma = 0.9$.

In figure (14) we can see the iteration count for the Poisson step for both solvers.

In the predictor (advection-diffusion equation) and projection steps a few iterations are needed to achieve relative low tolerances for both schemes. Nevertheless, in the Poisson step, the mesh size used leads to a high condition number for CG iteration (i.e. $\propto 1/h^2$), so we need to go further in iterations to achieve a converged solution of this step in order to

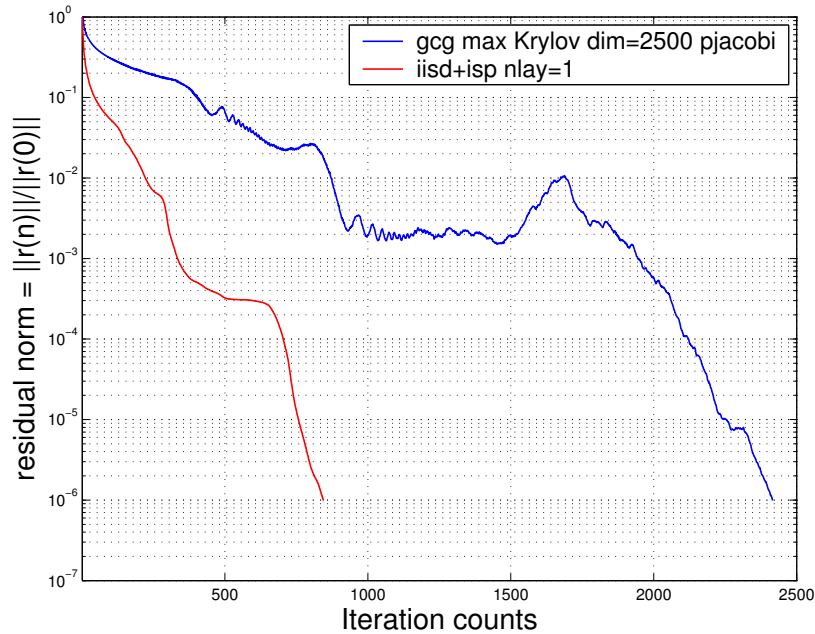


Figure 14. Iteration counts for both solvers: Poisson Step

avoid spurious oscillation in the solution. If we stop the global CG iteration around 1000-1200 iterations (Poisson step), where convergence history reaches a ‘plateau’ and the residuals may be considered acceptable, spurious oscillations with high amplitude appear in the solution at times where the steady state is accomplished). Moreover, we need to overpass the 2,400 iterations to avoid oscillation in all time steps. Although the required memory is not affected (we recall that in CG iteration, we need only the last two residual vectors), but the CPU time grows linearly with the iterations. Using IISD+ISP we can choose the amount of memory to solve the interior direct problem and the preconditioning (1500 dof’s where considered for each interior subdivision). The residuals reach low tolerances with few iterations and the CPU time required for each time step is diminished (although the local and interface problems are solved). Moreover, it is not necessary to iterate over the 50 iterations to obtain accurate solutions. Figure (15) shows the steady-converged solution when we stop the iteration in 50 using IISD+ISP. The total CPU time consumed in average for each time step (i.e. predictor, pressure and projection steps) for Global iteration (GMRES in predictor step, CG in Poisson and projection step) was 23.61 seconds and for IISD+ISP (with one layer around the interface) solver was 2.13 secs. Although we stop the IISD+ISP simulation with residuals being higher than residuals for 1000 iterations of global CG (see figure (14), i.e. the pressure step), the solution for domain decomposition method and preconditioning, IISD+ISP, is accurate enough and oscillating for global iteration. This behavior can be explained if we study the error in residuals when iteration proceeds.

Let $b, u_k, u_0 \in \mathbb{R}^N$, A non-singular such that $u^* = A^{-1}f$. Here f is the load vector and u_k

the solution at iteration k . Since

$$r_k = b - Au_k = Au^* - Au_k = A(u_k - u^*) = -Ae_k, \quad (30)$$

where e is the error in the solution u at the iteration k . Then,

$$\|e_k\| = \|A^{-1}Ae_k\| \leq \|A^{-1}\| \|Ae_k\| = \|A^{-1}\| \|r_k\|, \quad (31)$$

and

$$\|r_0\| \leq \|A\| \|e_0\|, \quad (32)$$

thus,

$$\frac{\|e_k\|}{\|e_0\|} \leq \frac{\|A^{-1}\| \|r_k\|}{\|A\|^{-1} \|r_0\|} = \kappa(A) \frac{\|r_k\|}{\|r_0\|}. \quad (33)$$

$\kappa(A)$ is the condition number of A and $\|\cdot\|$ is any suitable norm. The division by $\|r_0\|$ and $\|e_0\|$ in equation (33) normalizes the residuals. Storti *et al.* showed that the condition number is $\kappa(A) \propto O(1/h^2)$ for Global iteration and $\kappa(A) \propto O(1/h)$ for Schur complement domain decomposition methods. Moreover, the IISD+ISP still reduces the last condition number. Though the error in residuals for IISD+ISP (due to stop iteration earlier) is higher than the error in residuals for CG at high iteration count, the factor that determines the error in the solution (x_k) is the condition number κ (see equation 33). If the solution in the Poisson step is not accurate, the error is propagated to the other steps and oscillation may occur. We remark that this problem is stressed with the refinement (i.e. $h \rightarrow 0$).

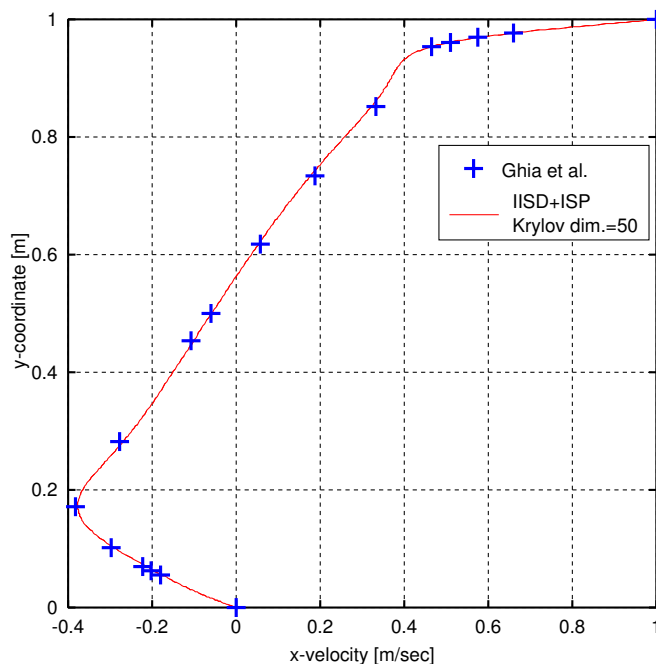


Figure 15. Time-converged solution for IISD+ISP solver ($Re = 1000$).

The primary vortex center was computed to be at $(x, y) = (0.531, 0.562)$ when the coordinate reference is at the bottom left corner of the cavity. IISD+ISP solver compares well (for the 50 iteration) with the values reported by Ghia *et al.* The core memory used was 45 Mbytes/processor for Global iteration and 60 Mbytes/processor for IISD+ISP (direct problems in sub-domain's subdivisions and interface problem). We recall that for CG iteration only we need the last two residuals, then the memory is not increased with iterations.

This example allows to emphasize other interesting usage of IISD+ISP solver. In fractional step like flow solvers the Poisson step has normally the major CPU time consume and for ill-conditioned problems this step demands a lot of resources for getting a good solution. It is difficult to know at the beginning how large the Krylov subspace dimension size of conjugate gradient method has to be and this example shows its strong influence on the final solution. Moreover, with an unusually high Krylov subspace dimension of 1000-1500 the solution of lid driven square cavity is already unacceptable making so dangerous its usage. With IISD+ISP it is possible to strongly reduce these requirements improving drastically the solution and mainly making the system solution less user-dependent.

6.4. The wind flow around a 3D immersed body. The AHMED model.

Current vehicle design needs a strong background in aerodynamics to improve flow control by means of active or passive control devices. The complexity involved in the automobile design specially due to the great amount of accessories and devices that form its geometry makes the validation tasks unfordable. The Ahmed model is a simple geometric body that retains the main flow features, specially the vortex wake flow where most part of the drag is concentrated and it is a good candidate to be used as a benchmark test. The flow regime of interest for car designers is fully turbulent. So, a large eddy simulation (LES) turbulence model is employed (see References [31, 16]). We compare the results obtained with several solvers and preconditioners with the detailed flow patterns previously published by Ahmed and coworkers [15].

The Ahmed reference model is a generic car type blunt body shape which is simple enough for accurate flow simulation but retains some important practical relevant features of automobile bodies. The body geometry is defined in [15]. The flow domain chosen is one in which the body of length L is suspended to 0.05 m to the ground in a domain of $10L \times 2L \times 1.5L$ in the stream-wise (y), span-wise (x) and stream-normal (z) directions. The boundary conditions for the problem are uniform flow at the inlet, slip at both sides, at rest for the surface of the body, non-slip at the floor with velocity equal to inlet and imposed pressure is used at the outflow boundary condition. The aerodynamics forces on road vehicles are the result of complex interactions between flow separations and the dynamic behavior of the released vortex wake.

6.4.1. Ahmed body: Numerical results for 3D Stokes flow. First, we consider the steady Stokes flow around the Ahmed body in order to compare the performance of the solvers in a more complicated problem than the flow presented in section §6.1. In this test, the computational domain is more complex and the flow is fully three dimensional. Nevertheless, the gap between the model and the floor acts like a elongated channel and we expect the same effect as in section §6.1). In this test a non-structured tetrahedral mesh is used in the whole flow domain. An 3-layer structured mesh of wedge (prismatic) elements is constructed for simulations of

boundary layers. This set of elements improves the resolution of the boundary layer around the blunt body. The body surface mesh contains 90,606 nodes and the boundary layer mesh has 180,600 elements. The tetrahedral mesh has 1,322,876 elements. The tests were carried out on 15 processors. For IISD solver 2,000 dof's were considered for each local subdivision.

The iteration counts for several Newton loops are shown in figure (16).

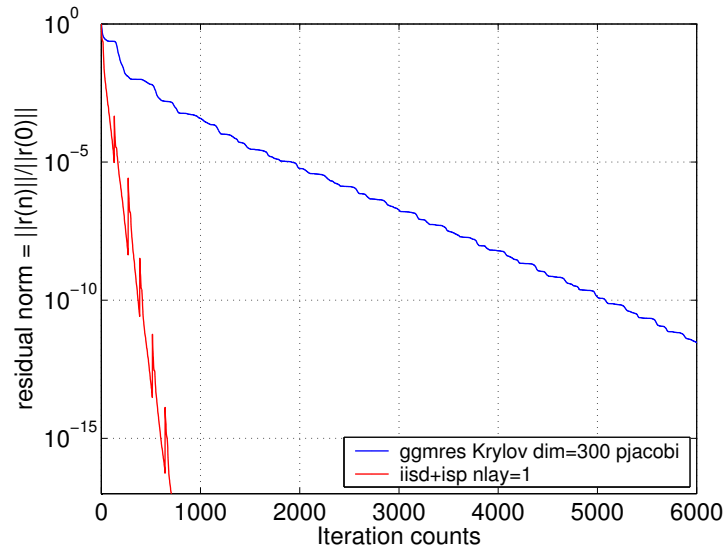


Figure 16. Stokes Flow. Iteration Counts (max. of 100 Newton iterations).

The calculated forces and moments for both solvers are shown in figures (17) to (19)

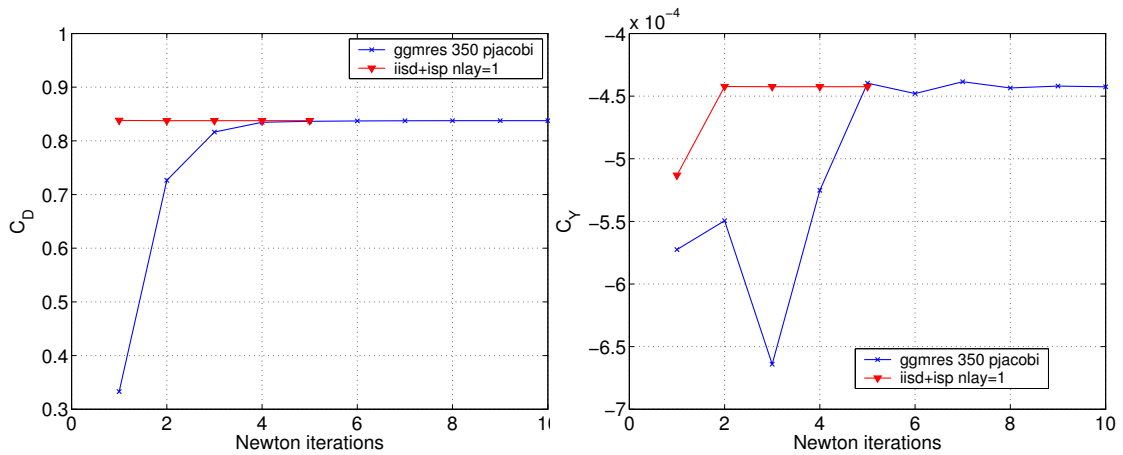


Figure 17. Stokes Flow. Force and moment coefficients.

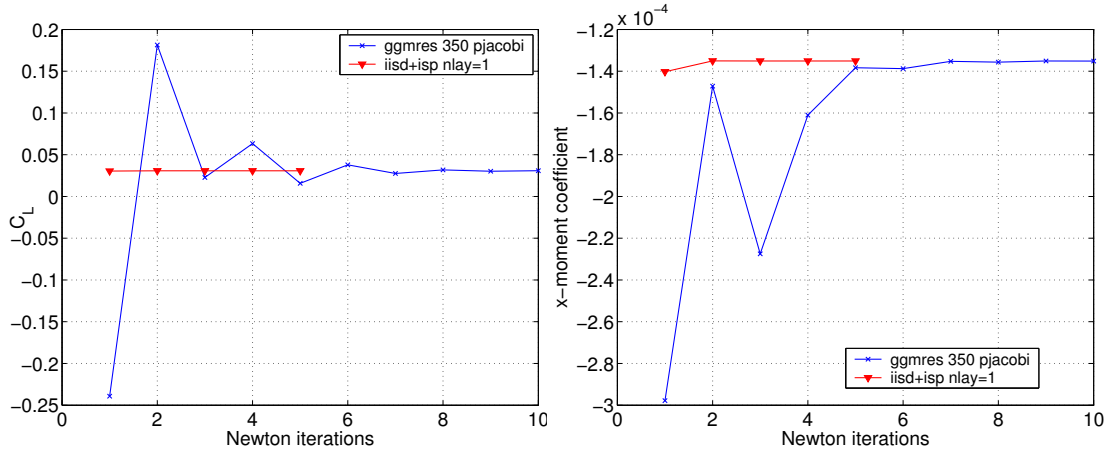


Figure 18. Stokes Flow. Force and moment coefficients

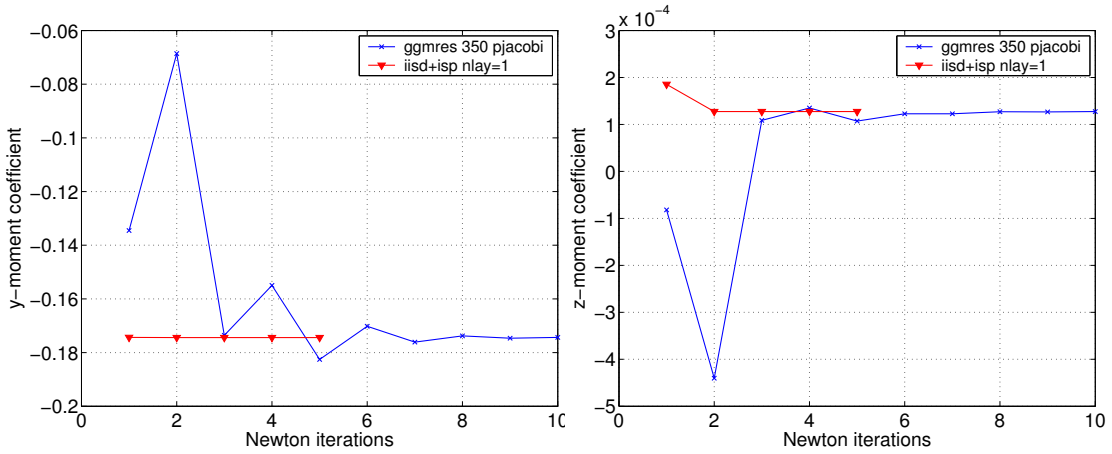


Figure 19. Stokes Flow. Force and moment coefficients

Clearly, IISD+ISP converges to the steady solution in one Newton iteration whereas Global iteration needs more iteration counts to achieve convergence. This behavior is directly related to the CPU time needed to obtain converged solution in a simulation.

The CPU consumed time and core memory requirements (in average) for each Newton iteration were: 185.1 secs and 443 Mbytes for Global GMRES with $KSP_{dim} = 300$ and 64.18 secs and 588 Mbytes for IISD+ISP.

6.4.2. Ahmed body: Numerical results for full 3D Navier-Stokes flow using LES In this section unsteady simulation for the flow around the Ahmed body are shown for $Re = 1000$ and $Re = 10^5$. The same architecture, mesh and partitions of the previous example was used. The

initial state used for the $Re = 1000$ simulation is the steady converged solution for the Stokes problem (IISD+ISP case). In the same manner the converged solution to the steady regime for $Re = 1000$ is used as initial condition for the $Re = 10^5$ test.

The iteration count for several time steps at $Re = 1000$ is shown in figure (20)

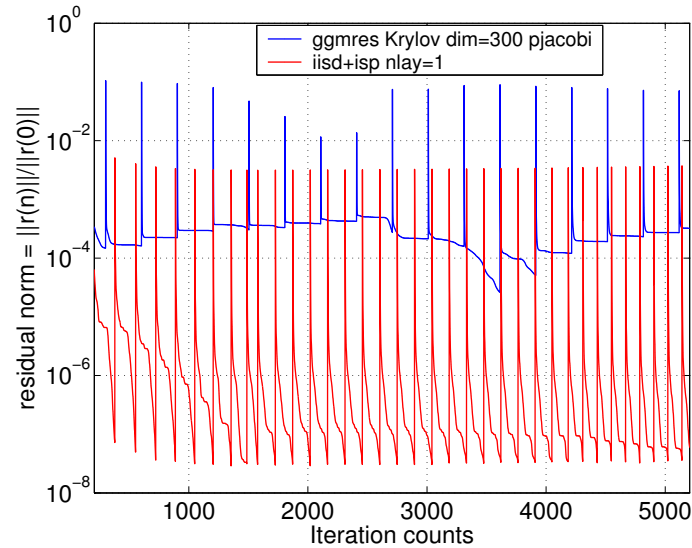


Figure 20. $Re = 1000$. Iteration Counts (100 time steps, 10 seconds simulation).

The calculated force and moment coefficients for both solvers ($Re = 1000$.) are shown in figures (21) to (23)

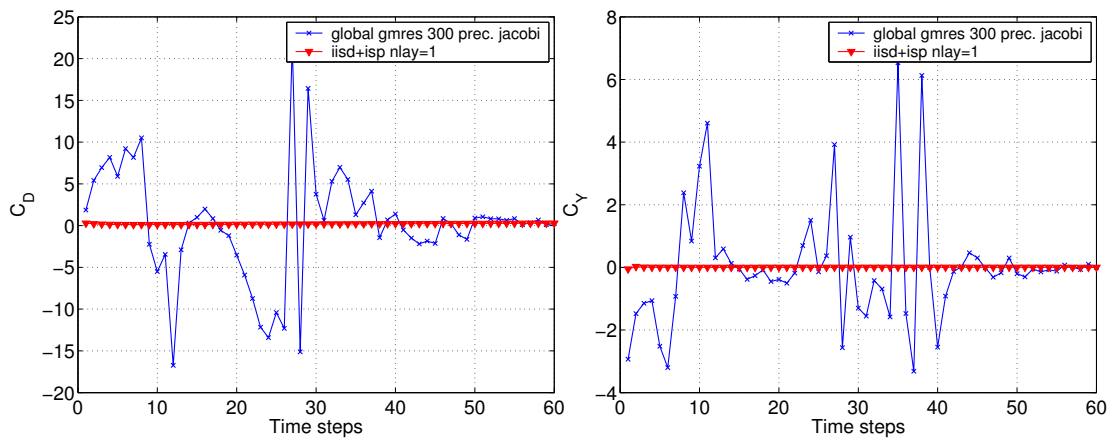
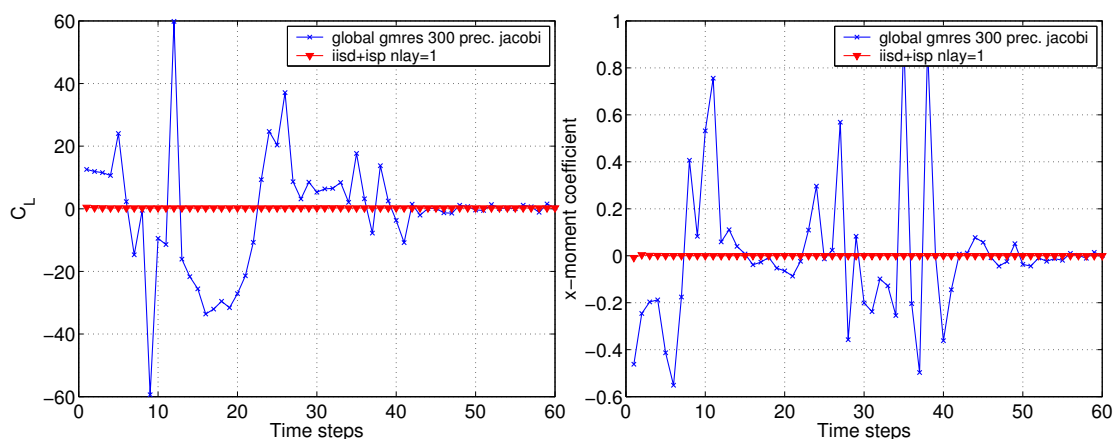
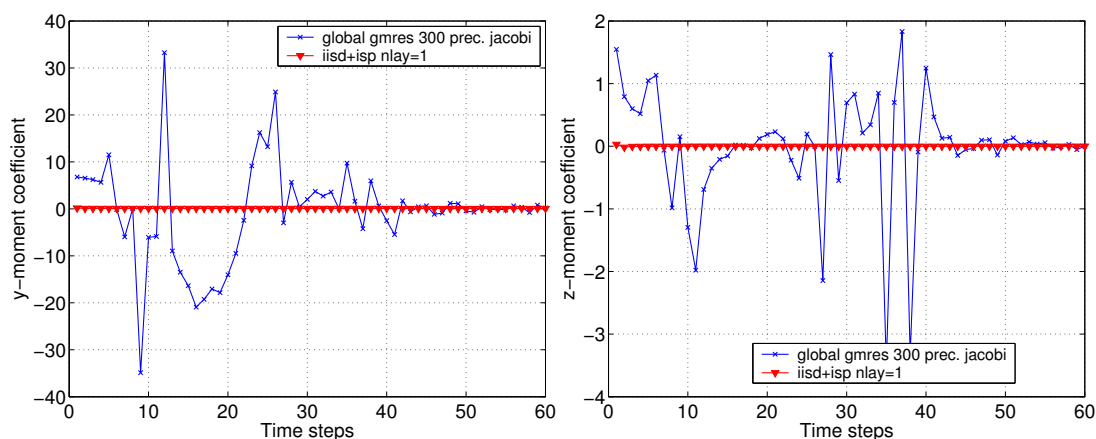


Figure 21. $Re = 1000$. Force and moment coefficients.

Figure 22. $Re = 1000$. Force and moment coefficients.Figure 23. $Re = 1000$. Force and moment coefficients

The force and moment coefficients scale in the figures is dominated by the bad approximation and oscillations in the solution obtained with Global Iterations. The calculated forces with IISD solver converge very fast to the values reported in the literature.

The CPU consumed time and core memory requirements (in average) for time step in the $Re = 1000$ test were: 186 secs and 460 Mbytes for Global GMRES with $KSP_{dim} = 300$ and 114.5 secs and 630 Mbytes for IISD+ISP.

This 3D example shows that even though the final solution of global GMRES seems to be similar to IISD+ISP the former needs more time steps or more Newton iterations to reach the final solution. This fact is strongly highlight for the Stokes problem. For medium Reynolds numbers the global GMRES has a strong oscillatory behavior until reach the good solution making more unstable the scheme under external perturbations. Therefore, here it is possible to argue the robustness as an indicator of the better behavior of IISD+ISP versus global

GMRES.

7. CONCLUSIONS

Numerical experiments of several physical (real) problems have been carried out to show its convergence properties, the computation time and memory requirements using both monolithic and disaggregated schemes. Through those tests, we have shown that not always is possible to obtain an acceptable solution for the problem using classical global Krylov methods.

We have shown that Domain Decomposition techniques, especially the Schur Complement Domain Decomposition using the Interface Strip Preconditioner, are suitable in order to achieve accurate solutions.

In all cases, the performance shown by IISD is decisive at the point of assigning computational resources to solve a time step in the simulation of a (real world) problem.

Also, The IS preconditioner is easy to construct as it does not require any special calculation (it can be assembled with a subset of sub-domain matrix coefficients). It is much less memory-consuming than classical optimal preconditioners such as Neumann-Neumann in primal methods (or Dirichlet in FETI methods) as it has been showed in References [1] and [2]. Moreover, it permits to decide how much memory to assign for preconditioning purposes.

The IS preconditioner is well suited for flows with high Reynolds numbers where the contribution of advective terms are predominant in the governing equations, while it is capable to handle reasonably well diffusion-dominated regions.

Furthermore, IISD+ISP is a good alternative to treat problems where domain discretization presents high refination gradients.

ACKNOWLEDGEMENTS

This work has received financial support from *Consejo Nacional de Investigaciones Científicas y Técnicas* (CONICET, Argentina), *Agencia Nacional de Promoción Científica y Tecnológica* (ANPCyT) and *Universidad Nacional del Litoral* (UNL) through grants CONICET PIP 198/ *Germen-CFD*, ANPCyT-PID-99/74 *FLAGS*, ANPCyT-FONCyT-PICT-6973 *PROA* and CAI+D-UNL-PIP-02552-2000. We made extensive use of freely distributed software such as *GNU/Linux* OS, MPI, PETSc, Metis, Octave, the CGAL geometrical library, OpenDX and many others. The authors would like to thank Dr. Gustavo Buscaglia for helpful discussions related to the lubrication channel problem.

REFERENCES

1. Storti MA, Dalcin L, Paz RR, Yommi A, Sonzogni V, Nigro N. An Interface Strip Preconditioner for Domain Decomposition Methods. to appear in *Journal of Computational Methods in Science and Engineering*. Cambridge International Science Publishing. 2003.
2. Paz RR, Storti MA. An Interface Strip Preconditioner for Domain Decomposition Methods: Application to Hydrology. accepted in *International Journal for Numerical Methods in Engineering*. John Wiley & Sons. 2004. In press.
3. Mandel J. Balancing domain decomposition. *Communications in Applied Numerical Methods*. 1993; **9**:233-241.
4. Le Tallec P, Vidrascu M. *Solving Large Scale Structural Problems on Parallel Computers using Domain Decomposition Techniques*. Parallel Solution Methods in Computational Mechanics. Papadrakakis M. (ed). John Wiley & Sons Ltd. 1997; chapter 2:49-85.

5. Meurant G. *Computer Solution of Large Linear Systems*. Studies in Mathematics and Its Applications. North-Holland. vol. 28; 1999.
6. Saad Y. *Iterative Methods for Sparse Linear Systems*. 2000; PWS Publishing Co.
<http://www-users.cs.umn.edu/~saad/books.html> .
7. Bramble JH, Pasciak JE, Schatz AH. The construction of preconditioners for elliptic problems by substructuring, I. *Mathematics of Computation*. 1986; **47(175)**:103-134.
8. Bramble JH, Pasciak JE, Schatz AH. The construction of preconditioners for elliptic problems by substructuring, IV. *Mathematics of Computation*. 1989; **53(187)**:1-24.
9. Kelley CT. *Iterative Methods for Linear and Nonlinear Equations (Frontiers in Applied Mathematics, Vol. 16)*. SIAM. 1995.
10. Storti MA, Nigro NM, Paz RR. PETSc-FEM: A General Purpose, Parallel, Multi-Physics FEM Program.
<http://www.cimec.org.ar/petscfem>
11. Carter JE. Numerical solutions of the Navier-Stokes equations for the supersonic laminar flow over two-dimensional compression corner. *National Aeronautics and Space Administration (NASA) 1972; Technical Report R-385*.
12. Koo J, Kleinstreuer C. Liquid flow in micro-channels: experimental observations and computational analysis of microfluidics effects. *Journal of Micromechanics and Microengineering*. 2003; **13**:568-579.
13. Stay MS, Barocas VH. Coupled lubrication and Stokes flow finite elements. *International Journal for Numerical Methods in Fluids* 2003; **42(2)**:129-146.
14. Chorin AJ. Numerical study of slightly viscous flow. *Journal of Fluid Mechanics*. 1973; **57**:785-796.
15. Ahmed SR, Ramm G, Faltin G. Some salient features of the times-averaged ground vehicle wake. *SAE Society of Automotive Eng., Inc.* 1984; **1(840300)**:1-31.
16. Smagorinsky J. General circulation experiments with the primitive equations. *Monthly Weather Review* 1963; **91(3)**:99-165.
17. Codina R. Pressure stability in fractional step finite element methods for incompressible flows. *Journal of Computational Physics* 2001; **170**:112-140.
18. Ghia U, Ghia KN, Shin CT. High-Re Solutions for Incompressible Flow Using the Navier-Stokes Equations and a Multigrid Method. *Journal of Computational Physics* 1982; **48**:387-411.
19. Tezduyar T, Mittal S, Ray S, Shih R. Incompressible flow computations with stabilized bilinear and linear equal order interpolation velocity pressure elements. *Computer Methods in Applied Mechanics and Engineering*. 1992; **95(95)**:221-242.
20. Hughes T, Tezduyar T. Finite element methods for first-order hyperbolic systems with particular emphasis on the compressible Euler equations. *Computer Methods in Applied Mechanics and Engineering* 1984; **45**:217-284.
21. Nigro N, Storti M, Idelsohn S. GMRES physics based preconditioner for all Reynolds and Mach number. Numerical examples. *International Journal for Numerical Methods in Fluids* 1997; **25**:1-25.
22. Gropp W, Lusk E, Skjellum A. Using MPI: Portable Parallel Programming with the Message-Passing Interface. 2nd edition. The MIT Press, London, England. 1994.
23. Balay S, Gropp WD, Curfman McInnes L, Smith BF. PETSc 2.2.0 user's manual. Argonne National Laboratory, 2004. <http://www-unix.mcs.anl.gov/petsc>
24. Braza M, Chassaing P, Ha Minh H. Numerical study and physical analysis of the pressure and velocity fields in the near wake of a circular cylinder. *Journal of Fluid Mechanics*. 1986; **164**:79-130.
25. Shih R, Tezduyar TE. Numerical experiments with the location of the downstream boundary for flow past a cylinder. *University of Minnesota Supercomputer Institute Research Report* 1990; **UMSI 90/38**.
26. Behr M, Liou J, Shih R, Tezduyar TE. Vorticity-Stream function formulation of unsteady incompressible flow past a cylinder: sensitivity of the computed flow field to the location of the downstream boundary. *University of Minnesota Supercomputer Institute Research Report* 1990; **UMSI 90/87**.
27. Roshko A. On the drag and shedding frequency of two dimensional bluff bodies. *National Advisory Committee for Aeronautics (NACA)* 1954; **Technical Note 3169**.
28. Williamson CHK. Evolution of a single wake behind a pair of bluff bodies. *Journal of Fluid Mechanics* 1985; **159**:1-18.
29. Norberg C. Flow around a circular cylinder: Aspects of fluctuating lift. *Journal of Fluids and Structures* 2001; **15**:459-469.
30. Brooks AN, Hughes TJR. Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering* 1982; **32**:199-259.
31. Krajnović S, Davidson L. Numerical study of the flow around the Bus-Shaped body. *ASME: Journal of Fluids Engineering* 2003; **125**:500-509.