INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2006; **52**:89–118
Published online 10 February 2006 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.1176

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, some pathologies found for simple tests solved by means of preconditioned full iterative schemes are presented. According to these results (Sections 4 and 5), the accuracy deterioration observed should be considered as a warning for the final application given to these solutions. Even though it is well known that full iterative solvers are not the best selection for comparison, they were chosen because they are widely used by the computational fluid dynamic (CFD) community for a diversity of complex fluid dynamics applications.

FEM simulated solutions are compared with analytical solutions or measured data for problems that have been considered as 'benchmarks' in the CFD literature. For this purpose, the study of the solution obtained via parallelized iterative methods that have been extensively used (e.g. conjugate gradients (CG), GMRes global iteration and its variants, 'overlapping' and 'non-overlapping' *additive Schwarz* domain decomposition schemes) in CFD computations and those obtained with the new interface strip preconditioner (*J. Comput. Meth. Sci. Engng* 2003; *Int. J. Numer. Meth. Engng* 2005; **62**(13):1873–1894) for the Schur complement method is carried out. 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 schemes applied to a internal/external viscous compressible/incompressible flows around bodies of complex shapes.

Therefore, the target of this work is to show how the reliability of CFD codes is affected by the solver selection and why domain decomposition methods should be viewed not only as a more efficient strategy, but also to guarantee the solution quality. Copyright © 2006 John Wiley & Sons, Ltd.

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

*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, http://www.cimec.org.ar
‡E-mail: nnigro@intec.unl.edu.ar
§E-mail: mstorti@intec.unl.edu.ar

Contract/grant sponsor: Consejo Nacional de Investigaciones Científicas y Técnicas; contract/grant number: 198/Germen-CFD
Contract/grant sponsor: Agencia Nacional de Promoción Científica y Tecnológica; contract/grant numbers: ANPCyT-PID-99/74 FLAGS, ANPCyT-FONCyT-PICT-6973 PROA, ANPCyT-PICT-12-14573/03
Contract/grant sponsor: Universidad Nacional del Litoral; contract/grant number: CAI+D-UNL-PIP-02552-2000

Received 3 May 2005
Revised 9 December 2005
Accepted 12 December 2005

1. INTRODUCTION

Over the last decades a lot of effort has been concentrated in making computational fluid dynamics (CFD) a reliable and efficient design tool for engineers and scientists. Both features are strongly linked with the final purpose of giving a good solution for real problems.

To obtain a realistic solution it is very important to adopt mathematical models according to the physical problem to solve and also to choose a proper discretization scheme and an (efficient and accurate) solver for the algebraic system of equations. As a result of the significant advancement in computational capabilities over the last few years, an important amount of scientific work has been focused on obtaining efficient techniques to solve large systems of equations resulting from complex applications from the real world. In this sense, multigrid and domain decomposition methods (DDM) give an answer to these demands and appear to be the 'next generation' solvers replacing the full iterative solvers that are currently widely used by the CFD community.

However, in spite of being less efficient, full iterative solvers are specially chosen for its inherent simplicity and also for the extensively used software libraries containing such solvers. Nevertheless, not only the efficiency is at stakes but also the accuracy of the solution is in danger. There is a huge amount of research work currently being carried out with full iterative solvers for many different applications. Commonly, these works are oriented to solve real world problems for industrial and environmental purposes where the fluid mechanics phenomena is so complex that verification is possible only by experimental measurements. In general, this last resource is not always available which leaves the simulation results unchecked.

Relatively simple test cases are deeply evaluated in order to show the accuracy, the robustness and the problem-dependent features. According to the results obtained, the scientific contribution should be viewed as relieving the accuracy loosing that, in general, full iterative solvers may deliver.

The main objective of DDM is the efficient solution on parallel architectures of problems arising in computational mechanics defined on irregular geometries and discretized on very fine meshes. One of the existing methods used to solve large-scale problems is the (*non-overlapping*) sub-structuring method known in the literature as Schur complement method. The implementation of sub-structuring technique on parallel architectures is convenient and well established in different areas of computational mechanics. However, the speedup and the efficient memory administration of this method poses a major challenge. The Schur complement method was widely used in structural mechanics to solve large-scale systems with limited memory computers for more than two decades (see References [1–8]). Besides, several ways of preconditioning the Schur complement matrix (associated with the interface problem) can be found in these articles an the references cited therein.

Another family of DDM, the *overlapping* Schwarz domain decomposition schemes, have also been extensively used in computational mechanics. A good introduction and application of these methods is presented by Smith and coworkers [5]. In the CFD area, Rachowicz [9] applied successfully the GMRes solver with a domain decomposition Schwarz-type preconditioner in the solution of hypersonic high Reynolds number flows with strong shock-boundary layer interaction.

In Reference [10] the mathematical formulation and the sequential implementation of a new preconditioner for Schur complement DDM (i.e. the interface strip preconditioner (ISP)) based on solving a problem in a narrow strip around the interface between sub-domains is introduced. Also, convergence properties of the iterative solution step were contrasted with several preconditioners for the discretization of the Poisson and the scalar convection–diffusion equations. In Reference [11] the implementation details in parallel environments were presented. Besides, this study shows that the ISP preconditioner is much less memory and time consuming than classical preconditioners such as Neumann–Neumann [4, 6] and Block Jacobi preconditioner in the context of the solution of non-symmetric system of equations arising from the discretization of hydrological problems.

The accuracy and the efficiency of ISP preconditioner (see References [10, 11]) based on domain decomposition coupled with a classical finite element formulation is compared to full iterative solvers and other domain decomposition techniques applied to the resolution of partial differential equation describing different kind of flows. Moreover, this study shows that the computational effort and/or resources could be insufficient to solve some problems with classical global iterative methods, like Krylov subspace methods (e.g. conjugate gradient (CG) and generalized minimum residual and its variants). For this purpose, the CPU consumed time, the required core memory and the accuracy in the solution of several problems in CFD is analysed. Furthermore, this new solver is introduced 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

The solution of both compressible and incompressible Navier–Stokes equations with the SUPG (‘*streamline upwind/Petrov Galerkin*’) and SUPG-PSPG (SUPG-‘*pressure stabilized/Petrov Galerkin*’) methods proposed by Brooks *et al.* [12] and Tezduyar *et al.* [13], respectively, is shown in this paper.

2.1. Incompressible Navier–Stokes equations

The incompressible Navier–Stokes equations present two important difficulties for solution with finite elements. First, the character of the equation becomes highly advective dominant when the Reynolds number increases. In addition, the incompressibility condition does not represent an evolution equation but a constraint on the equations. This is a drawback as only some combination of interpolation spaces for velocity and pressure can be used with the Galerkin formulation, namely those ones that satisfy the so-called Ladyzhenskaya–Brezzi–Babuška 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 is discretized in time with the standard trapezoidal rule (backward Euler and Crank–Nicolson schemes are allowed to be used). The resulting non-linear system of equation is solved iteratively at every time step.

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 $\Omega \in \mathbb{R}^{n_{\rm sd}}$ and $(0, t_+]$ be the spatial and temporal domains, respectively, where $n_{\rm sd}$ is the number of space dimensions, and let Γ be the boundary of Ω.

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

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

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

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 [14], i.e. $\mu^* = \mu + \mu_{SGS}$. $\mathbf{I}$ represents the identity tensor and ϵ the strain rate tensor. The initial and boundary conditions are

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

where Γ_g and Γ_h are the Dirichlet and Neumann boundaries, respectively. When the flow velocity is very small (i.e. the fluid is very viscous) or the geometric dimensions are very small, that is when Reynolds number is very small, the inertial term in (1) plays a minor role and the flow is dominated by the viscous and the pressure gradient terms. This is the so-called ‘Stokes flow’.

2.1.1. Spatial discretization. The spatial discretization has 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.* [13] 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, $e = 1, \ldots, n_{\text{el}}$, and let $\mathscr{E}$ be the set of these elements, and H^{1h} the finite-dimensional space defined by

$$H^{1h} = \{\phi^h | \phi^h \in C^0(\overline{\Omega}), \phi^h|_{\Omega^e} \in P^1 \forall \Omega^e \in \mathscr{E}\} \tag{4}$$

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

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

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

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

where the stabilization parameters in Equation (6) are defined as

$$\begin{aligned}
\boldsymbol{\delta}^h &= \tau_{\text{SUPG}} (\mathbf{u}^h \cdot \nabla) \mathbf{w}^h \\
\epsilon^h &= \tau_{\text{PSPG}} \frac{1}{\rho} \nabla q^h \\
\tau_{\text{PSPG}} = \tau_{\text{SUPG}} &= \frac{h_{\text{elem}}}{2 \|\mathbf{u}^h\|} z(Re_u)
\end{aligned} \tag{7}$$

Note that the SUPG and the PSPG terms are defined on different functional spaces. These stabilizations terms act, at the linear system level, adding nonzero values on the diagonal entries associated with the pressure equations. The Reynolds number Re_u based on the element parameters is

$$Re_u = \frac{\|\mathbf{u}^h\| h_{\text{elem}}}{2\nu} \tag{8}$$

the element size h_{elem} is computed as

$$h_{\text{elem}} = 2 \left(\sum_{a=1}^{nn} |\mathbf{s} \cdot \nabla N_a| \right)^{-1} \tag{9}$$

N_a being the shape function associated with the node a, nn the number of nodes in the element, and $\mathbf{s}$ a unit vector on the streamline direction. The function $z(Re)$ is defined as

$$z(Re) = \begin{cases} Re/3 & 0 \leqslant Re < 3 \\ 1 & 3 \leqslant Re \end{cases} \tag{10}$$

2.1.2. Disaggregated scheme. Fractional step methods for the incompressible Navier–Stokes equations have been popular over 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 [15] the study of computed pressure stability of schemes that use a pressure Poisson equation was presented. These results are used in this paper.

The results to be presented refer to second-order algorithm based on the implicit $(\theta = 1)$ 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.

The time discretization of problem (1) written in a compact matrix form is

$$\mathbf{M}\frac{1}{\Delta t}(\mathbf{U}^{n+1} - \mathbf{U}^{n}) + \mathbf{K}(\mathbf{U}^{n+\theta})\mathbf{U}^{n+\theta} + \mathbf{G}\mathbf{P}^{n+1} = \mathbf{F}^{n+\theta} \tag{11}$$

$$\mathbf{D}\mathbf{U}^{n+1} = 0 \tag{12}$$

where $\mathbf{M}$ is the mass matrix, $\mathbf{U}$ is the vector of velocity unknowns, $\mathbf{K}$ is the stiffness matrix, $\mathbf{G}$ is the matrix form of the gradient operator, $\mathbf{P}$ is the vector of nodal pressures, $\mathbf{D}$ is the matrix form of the divergence operator and $\mathbf{F}$ is the vector of source terms. Superscripts n and $n+1$ denote variables at time $t = n\Delta t$ and $t = (n+1)\Delta t$, respectively.

The fractional step scheme applied to the fully discrete problem (11) and (12) is exactly equivalent to

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

$$\mathbf{M}\frac{1}{\Delta t}(\mathbf{U}^{n+1} - \hat{\mathbf{U}}^{n+1}) + \mathbf{G}(\mathbf{P}^{n+1} - \gamma\mathbf{P}^{n}) = 0 \tag{14}$$

$$\mathbf{D}\mathbf{U}^{n+1} = 0 \tag{15}$$

where $\hat{\mathbf{U}}^{n+1}$ is an auxiliary variable and γ is a numerical parameter, whose values of interest are between 0 and 1. The essential approximation $\mathbf{K}(\mathbf{U}^{n+\theta})\mathbf{U}^{n+\theta} \sim \mathbf{K}(\hat{\mathbf{U}}^{n+\theta})\hat{\mathbf{U}}^{n+\theta}$ is made, where $\hat{\mathbf{U}}^{n+\theta} = \theta\hat{\mathbf{U}}^{n+1} + (1-\theta)\mathbf{U}^{n}$. Writing $\mathbf{U}^{n+1}$ in terms of $\hat{\mathbf{U}}^{n+1}$ using (14) and inserting the result in (15), the equations to be solved are

$$\mathbf{M}\frac{1}{\Delta t}(\hat{\mathbf{U}}^{n+1} - \mathbf{U}^{n}) + \mathbf{K}(\hat{\mathbf{U}}^{n+\theta})\hat{\mathbf{U}}^{n+\theta} + \gamma\mathbf{G}\mathbf{P}^{n} = \mathbf{F}^{n+\theta} \tag{16}$$

$$\Delta t\mathbf{D}\mathbf{M}^{-1}\mathbf{G}(\mathbf{P}^{n+1} - \gamma\mathbf{P}^{n}) = \mathbf{D}\hat{\mathbf{U}}^{n+1} \tag{17}$$

$$\mathbf{M}\frac{1}{\Delta t}(\mathbf{U}^{n+1} - \hat{\mathbf{U}}^{n+1}) + \mathbf{G}(\mathbf{P}^{n+1} - \gamma\mathbf{P}^{n}) = 0 \tag{18}$$

the order of this equations is made according to the sequence of solution, i.e. first for $\hat{\mathbf{U}}^{n+1}$, then $\mathbf{P}^{n+1}$ and finally $\mathbf{U}^{n+1}$. The operator $\mathbf{DM}^{-1}\mathbf{G}$ in (17) can be approximated by the Laplace operator if the matrix $\mathbf{M}$ 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 govern the dynamics of compressible and viscous fluid flow may be written in a compact intrinsic (vector) form as (Einstein summation convention is assumed, $i,j=1,2,3$):

$$\frac{\partial \mathbf{U}}{\partial t}+\frac{\partial(\mathscr{F}^{\mathrm{a}})_i}{\partial x_i}=\frac{\partial(\mathscr{F}^{\mathrm{d}})_i}{\partial x_i}+\mathscr{G} \quad \text{in } \Omega\times(0,t_+] \tag{19}$$

where Ω is the model domain with boundary Γ. $\mathbf{U}=(\rho,\rho\mathbf{u},\rho E)^t$ is the unknown state vector expressed in conservative variables, E represents the specific total energy, $\mathscr{F}^{\mathrm{a}}$ accounts for the (vector) advective fluxes, $\mathscr{F}^{\mathrm{d}}$ for the (vector) diffusive fluxes and $\mathscr{G}$ is used for the external source terms (i.e. $\mathscr{G}=(0,\rho\mathbf{f_e},W_f+q_H)$, $W_f=\rho\mathbf{f_e}\cdot\mathbf{u}$ is the work done by the external forces $\mathbf{f_e}$ and $\mathbf{n}$ represents an outward unit normal vector at boundary). The advective and the diffusive fluxes are defined as

$$\mathscr{F}^{\mathrm{a}}=\begin{pmatrix}\rho u_i\\ \rho u_1u_i+\delta_{i1}p\\ \rho u_2u_i+\delta_{i2}p\\ \rho u_3u_i+\delta_{i3}p\\ \rho\mathscr{H}u_i\end{pmatrix},\quad \mathscr{F}^{\mathrm{d}}=\begin{pmatrix}0\\ \tau_{i1}\\ \tau_{i2}\\ \tau_{i3}\\ \tau_{ik}u_k-q_i\end{pmatrix} \tag{20}$$

where $\mathscr{H}$ is the total specific enthalpy defined in terms of the specific internal energy e and the specific kinetic energy as $\mathscr{H}=e+p/\rho+\frac{1}{2}|\mathbf{u}|^2$. In (20), δ_{ij} is the Kronecker isotropic tensor of rank 2 (also denoted as $\mathbf{I}$), τ_{ij} are the components of the Newtonian viscous stress tensor: $\tau=2\mu\epsilon(\mathbf{u})-2/3\mu(\nabla\cdot\mathbf{u})\mathbf{I}$. The strain rate tensor ϵ is $\epsilon(\mathbf{u})=\frac{1}{2}(\partial_j u_i+\partial_i u_j)$. 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 absolute temperature. The coefficients of viscosity and thermal conductivity can be modeled by the Sutherland formula as (i.e. the gas is considered in a *standard atmosphere*)

$$\mu=\mu_0\left(\frac{T}{T_0}\right)^{3/2}\left(\frac{T_0+110}{T+110}\right) \quad \text{and} \quad \kappa=\frac{\gamma_a R\mu}{(\gamma_a-1)Pr}$$

where μ_0 is the viscosity at the reference temperature T_0 and Pr is the Prandtl number (i.e. $Pr=v/\imath$, $\imath$ is the thermal diffusivity coefficient).

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, then $\rho e=p/(\gamma_a-1)+\frac{1}{2}\rho\|\mathbf{u}\|^2$ and $p=\rho RT$, where $R=(\gamma_a-1)C_v$ is the particular gas constant and $\gamma_a=C_p/C_v$ is the ratio of the specific heat at constant pressure relative to that at constant volume. Alternatively, Equation (19) can be written in the quasi-linear form:

$$\frac{\partial \mathbf{U}}{\partial t} + \mathbf{A}_i \frac{\partial \mathbf{U}}{\partial x_i} = \frac{\partial}{\partial x_i}\left(\mathbf{K}_{ij}\frac{\partial \mathbf{U}}{\partial x_j}\right) + \mathscr{G} \tag{21}$$

where the assumption that the flux vectors are only function of the state variables, i.e. $\mathscr{F}^{\mathrm{a}} = \mathscr{F}^{\mathrm{a}}(\mathbf{U})$ and $\mathscr{F}^{\mathrm{d}} = \mathscr{F}^{\mathrm{d}}(\mathbf{U})$ is made. Then, the divergence of the flux vector functions can be written as

$$\frac{\partial \mathscr{F}^{\mathrm{a}}}{\partial x_i} = \frac{\partial \mathscr{F}^{\mathrm{a}}}{\partial \mathbf{U}}\frac{\partial \mathbf{U}}{\partial x_i} = \mathbf{A}_i \frac{\partial \mathbf{U}}{\partial x_i} \quad \text{and} \quad \frac{\partial \mathscr{F}^{\mathrm{d}}}{\partial x_i} = \frac{\partial \mathscr{F}^{\mathrm{d}}}{\partial \mathbf{U}}\frac{\partial \mathbf{U}}{\partial x_i} = \mathbf{K}_{ij}\frac{\partial \mathbf{U}}{\partial x_i}$$

2.2.1. Variational formulation. In this section, the variational formulation of the compressible Navier–Stokes equations using SUPG finite elements method and the shock-capturing operator is presented. Consider a finite element discretization of the Ω into sub-domains $\Omega^e, e = 1, 2, \ldots, n_{\mathrm{el}}$. Based on this discretization, the finite element function spaces for the trial solutions and for the weighting functions, $\mathscr{V}^h$ and $\mathscr{S}^h$, respectively, can be defined. These function spaces are selected as subsets of $[\mathbf{H}^{1h}(\Omega)]^{n_{\mathrm{dof}}}$ when taking Dirichlet boundary conditions, where $\mathbf{H}^{1h}(\Omega)$ is the finite-dimensional Sobolev functional space over Ω, and $n_{\mathrm{dof}} = n_{\mathrm{sd}} + 2$ is the number of degrees of freedom dof's in the continuum problem.

The stabilized finite element formulation of the quasi-linear form of (19) is written as follows: find $\mathbf{U}^h \in \mathscr{S}^h$ such that $\forall \mathbf{W}^h \in \mathscr{V}^h$

$$\begin{aligned}
&\int_\Omega \mathbf{W}^h \cdot \left(\frac{\partial \mathbf{U}^h}{\partial t} + \frac{\partial \mathbf{F}_{\mathrm{a}}^h}{\partial x_i}\right) \mathrm{d}\Omega = \int_\Omega \mathbf{W}^h \cdot \left(\frac{\partial \mathbf{F}_{\mathrm{d}}^h}{\partial x_i} + \mathscr{G}\right) \mathrm{d}\Omega \\
&\int_\Omega \mathbf{W}^h \cdot \left(\frac{\partial \mathbf{U}^h}{\partial t} + \mathbf{A}_i^h \frac{\partial \mathbf{U}^h}{\partial x_i} - \mathscr{G}\right) \mathrm{d}\Omega + \int_\Omega \frac{\partial \mathbf{W}^h}{\partial x_i} \cdot \mathbf{K}_{ij}^h \frac{\partial \mathbf{U}^h}{\partial x_j} \mathrm{d}\Omega - \int_{\Gamma_h} \mathbf{W}^h \cdot H^h \, \mathrm{d}\Gamma \\
&\quad + \sum_{e=1}^{n_{\mathrm{el}}} \int_{\Omega^e} \boldsymbol{\tau}(\mathbf{A}_k^h)^{\mathrm{T}} \frac{\partial \mathbf{W}^h}{\partial x_k} \cdot \left\{\frac{\partial \mathbf{U}^h}{\partial t} + \mathbf{A}_i^h \frac{\partial \mathbf{U}^h}{\partial x_i} - \frac{\partial}{\partial x_i}\left(\mathbf{K}_{ij}^h \frac{\partial \mathbf{U}^h}{\partial x_j}\right) - \mathscr{G}\right\} \mathrm{d}\Omega \\
&\quad + \sum_{e=1}^{n_{\mathrm{el}}} \int_{\Omega^e} \delta_{\mathrm{shc}} \frac{\partial \mathbf{W}^h}{\partial x_i} \cdot \frac{\partial \mathbf{U}^h}{\partial x_i} \mathrm{d}\Omega = \mathbf{0}
\end{aligned} \tag{22}$$

where

$$\begin{aligned}
\mathscr{S}^h &= \{\mathbf{U}^h | \mathbf{U}^h \in [\mathbf{H}^{1h}(\Omega)]^{n_{\mathrm{dof}}}, \mathbf{U}^h|_{\Omega^e} \in [P^1(\Omega^e)]^{n_{\mathrm{dof}}}, \mathbf{U}^h = \mathbf{g} \text{ on } \Gamma_g\} \\
\mathscr{V}^h &= \{\mathbf{W}^h | \mathbf{W}^h \in [\mathbf{H}^{1h}(\Omega)]^{n_{\mathrm{dof}}}, \mathbf{W}^h|_{\Omega^e} \in [P^1(\Omega^e)]^{n_{\mathrm{dof}}}, \mathbf{W}^h = \mathbf{0} \text{ on } \partial\Omega_g\}
\end{aligned} \tag{23}$$

where matrices $\mathbf{A}_i$ and $\mathbf{K}_{ij}$ are defined in Section 2.2. Various options for calculating the stabilization parameters and defining the shock-capturing terms in the context of the SUPG formulation were introduced in Reference [13]. Some of these options are defined in this section. The first one is the standard SUPG intrinsic time tensor $\boldsymbol{\tau}$ introduced by Aliabadi in Reference [16]. In this case the matrix is defined as $\boldsymbol{\tau} = \max[\mathbf{0}, \boldsymbol{\tau}_{\mathrm{a}} - \boldsymbol{\tau}_{\mathrm{d}} - \boldsymbol{\tau}_\delta]$, with each $\boldsymbol{\tau}_x$

defined as follows (considering the advective and diffusive effects and avoiding the duplication of the shock-capturing operator and the SUPG operator):

$$\boldsymbol{\tau}_{\mathrm{a}} = \frac{h}{2(c+|\mathbf{u}|)}\mathbf{I}, \quad \boldsymbol{\tau}_{\mathrm{d}} = \frac{\sum_{j=1}^{n_{\mathrm{sd}}} \beta_j^2 \,\mathrm{diag}\,(\mathbf{K}_{jj})}{(c+|\mathbf{u}|)^2}\mathbf{I}, \quad \boldsymbol{\tau}_{\delta} = \frac{\delta_{\mathrm{shc}}}{(c+|\mathbf{u}|)^2}\mathbf{I} \tag{24}$$

where c is the acoustic speed, h is the element size computed as the element length in the direction of the streamline using for its definition the multi-lineal trial function N_a (as in Equation (9)) and $\beta = \nabla\|\mathbf{U}\|^2/\|\nabla\|\mathbf{U}\|^2\|$.

The design of the shock-capturing operator is also an open problem. Two versions are presented here: an isotropic operator and an anisotropic one, both proposed by Tezduyar *et al.* [17]. A unit vector oriented with the density gradient is defined as $\mathbf{j} = \nabla\rho^h/|\nabla\rho^h|$ and a characteristic length as $h_{JGN} = 2(\sum_{a=1}^{n_{\mathrm{en}}} |\mathbf{j}\cdot\nabla N_a|)^{-1}$, where N_a is the finite element shape function corresponding to the node a. The above-cited isotropic shock-capturing factor included in (22) is then defined as

$$\delta_{\mathrm{shc}} = \frac{h_{JGN}}{2} u_{\mathrm{char}} \left(\frac{|\nabla\rho^h| h_{JGN}}{\rho_{\mathrm{ref}}} \right)^{\beta^*} \tag{25}$$

where $u_{\mathrm{char}} = |\mathbf{u}| + c$ is the characteristic velocity. ρ_{ref} is the gaussian point interpolated density and β^* parameter may be taken as 1 or 2 according to the sharpness of the discontinuity to be captured as suggested in Reference [17]. However, only $\beta^* = 1$ was successfully used in this study.

The anisotropic version of the shock-capturing term in (22) is changed as follows:

$$\sum_{e=1}^{n_{\mathrm{el}}} \int_{\Omega^e} \frac{\partial \mathbf{W}^h}{\partial x_i} \mathbf{j}_i \delta_{\mathrm{shc}} \mathbf{j}_k \frac{\partial \mathbf{U}^h}{\partial x_k} \,\mathrm{d}\Omega \tag{26}$$

The anisotropic shock-capturing term showed good behaviour. Nevertheless, for some applications, both terms may be needed, the isotropic one weighted by a factor close to 0.2 or lower.

3. DOMAIN DECOMPOSITION METHODS AND THE PRECONDITIONING OF THE SCHUR COMPLEMENT MATRIX

The performance of iterative solvers depends strongly on the spectral distribution of the eigenvalues of the global matrix and its condition number. If the linear system is ill conditioned, it is necessary to perform some preconditioning on the original matrix. Moreover, it is required that the preconditioner does not degrade its performance with the increase of the problem size and the number of processors when it is used in parallel environments. Considerable work has been done over the last two decades to formulate scalable 'general purpose' preconditioners in order to outperform the convergence of iterative global solvers in terms of CPU time and memory requirements. A very simple case (but not often effective) is, for instance, the diagonal scaling of the original matrix. That is the so-called point-Jacobi preconditioner.

Domain decomposition preconditioning techniques are based on subdividing the original domain into several (*overlapping* and *non-overlapping*) sub-parts, also called sub-domains, which can be processed (almost) independently. These subproblems are coupled through the values of the dof's at the sub-domain interfaces. The coupling is removed at each solution step by the solution of internal problems constrained to some condition on the interfaces. Domain decomposition methods can be classified into several groups. In the 'overlapping DDM' group, the computational domain is divided into sub-domains whose intersections are not empty sets. On each solution step local Dirichlet-type problems are solved and the continuity across interfaces is guaranteed by the overlapping region. The family of *additive* and *multiplicative Schwarz Methods* are members of this group (see Reference [5]).

On the other hand, in the 'non-overlapping DDM' schemes group, it is possible to decompose the unknowns into two disjoint sets: one set with the unknowns belonging to the *interior* nodes of a sub-domain, and another set formed with the unknowns of the interfaces between sub-domains. By 'condensing' the internal dof's at an algebraic level, one can compute the Schur complement matrix that defines the interface problem. The Schur complement matrix is a discrete approximation to a Steklov–Poincaré operator which enforces transmission boundary conditions on the interface. Then, the non-overlapping DDMs based on the Schur complement matrix start by first determining the unknowns on the interfaces between sub-domains. After that, the sub-domain problems decouple and may be solved in parallel. Those methods are often called *iterative sub-structuring*. In order to further improve the efficiency of this method, a preconditioner has to be added so that the condition number of the Schur complement matrix is lowered. Iterative sub-structuring preconditioning methods have been studied by Bramble *et al.* [1, 2], De Roeck and Le Tallec [18], Le Tallec and Vidrascu [6], Mandel [4] and Farhat and Roux [3].

Among the most popular techniques are the Neumann–Neumann (NN) preconditioner [18] and its variants e.g. the balancing Neumann–Neumann (BNN) preconditioner [4]; and the family of the finite element tearing and interconnecting (FETI) preconditioners [3]. In the Neumann–Neumann scheme the application of inverses of the local Schur complement matrices corresponds on the continuous level to solving local Neumann problems on each sub-domains. Mandel added a coarse problem to the Neumann–Neumann original scheme in order to assure the scalability of this preconditioner. Instead of solving the Schur complement system problem, the FETI method uses a space of discontinuous functions across the interface between sub-domains. Then, the continuity of the solution is enforced by using a vector of Lagrange multipliers that leads to a saddle-point problem. It can be proved that both preconditioners reduce the condition number of the preconditioned operator to $O(1)$ (i.e. independent of h) in some special cases.

A key point about the Steklov–Poincaré operator is that its high-frequency eigenfunctions decay very strongly far from the interface, so that a preconditioning that represents correctly the high-frequency modes can be constructed if a problem on a narrow strip around the interface is solved. Keeping this in mind, a new preconditioner (i.e. the ISP) for Schur complement matrix is proposed in References [10, 11]. This technique, that has been called '*interface iterative/sub-domain direct*' with the added '*interface strip preconditioner*' (IISD+ ISP) has shown the ability to handle the flux splitting between sub-domains well, specially for problems where advective terms are high. A description of the implementation details (that could be made at an algebraic level) in parallel environments is given in a previous article [11]. In that work, CPU requirements, memory demands and convergence properties of the iterative solution step were contrasted with several preconditioners like Balancing Neumann–Neumann, non-overlapping additive Schwarz or Block Jacobi, global diagonal scaling preconditioner for the discretization of the Poisson equation, the scalar convection–diffusion equations and the system of conservation laws that governs the flow in open channels. The IISD + ISP solver is part of a parallel FEM production code [19] and it is used to solve large-scale multiphysics problems with unstructured 1D/2D/3D meshes.

4. NUMERICAL TEST IN SEQUENTIAL ENVIRONMENT

The hypersonic flow over a flat plate test at Mach = 5. In this section the hypersonic flow around a flat plate is analysed. This is a typical flow problem where nonlinearities become high so that any difficulty in the convergence of the linear system may influence the nonlinear convergence and finally make the solution blow up. This problem, deeply documented by Carter in Reference [20], shows a strong interaction between the boundary layer and the shock wave; and there is 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 challenging problem. Figure 1 shows the problem definition with a sketch of the physical structures present in the flow field and the boundary condition applied to it.

The compressible laminar Navier–Stokes equations are used to model this problem. A constant viscosity $\mu = 2.5 \times 10^{-5}\,\mathrm{kg/m\,s}$ is adopted and the Reynolds number based on the flat plate length and the free stream state is 10^4. The test case is an isothermal flow at Mach $M = 5$ at the inlet wall. The thermal conductivity coefficient is $\kappa = 3.47 \times 10^{-5}\,\mathrm{W/mK}$. The plate is located 0.02 m from the inflow wall. The characteristic length is the length of the plate, $L = 0.25\,\mathrm{m}$. The free stream Prandtl number based on this length is 0.72. The gas constant is $R = 287\,\mathrm{J/(kg\,K)}$ and the specific heat ratio of the gas is $\gamma_a = 1.4$. The temperature and pressure of the free stream are $T_\infty = 80\,\mathrm{K}$ and $p_\infty = 10^5\,\mathrm{Pa}$, respectively. The temperature of the flat plate surface is $T_{\mathrm{wall}} = 288\,\mathrm{K}$.

Experimental and theoretical data are available for the skin friction coefficient and the wall Stanton number (heat conduction problem). This problem was successfully solved using the

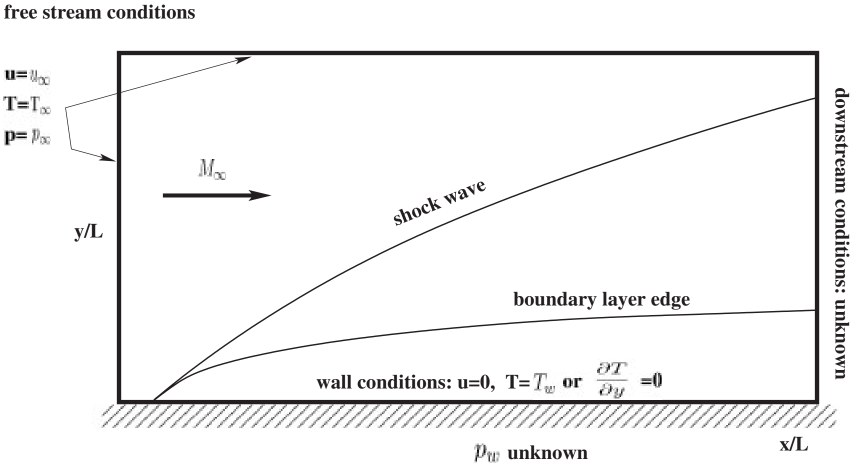

Figure 1. Problem definition.

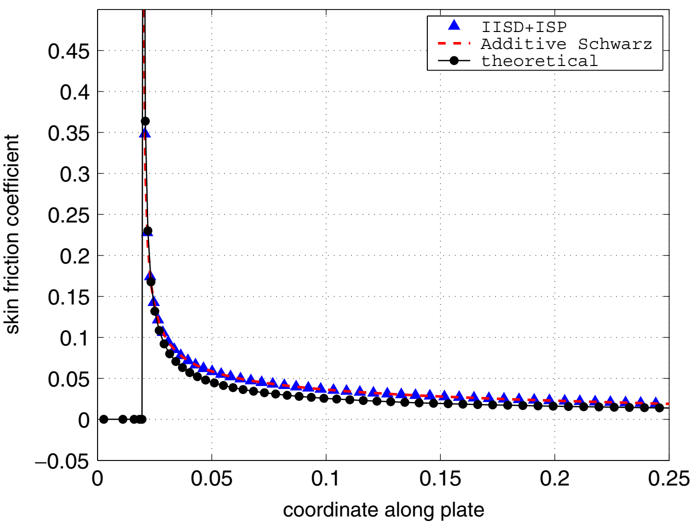

Figure 2. Skin friction coefficient.

IISD+ISP solver and the overlapping additive Schwarz preconditioner, but it was not possible to obtain a solution with a global GMRes with diagonal scaling preconditioner (i.e. GMRes over the whole matrix with point-Jacobi preconditioning) solver, using for the three cases a Krylov subspace dimension of 200. In the latter case, the solution presented poor resolution of the strong shock wave after some time steps and finally crashed. It should be remarked that up to $M = 2.5$ preconditioned global GMRes iteration works fine, giving results in agreement with experimental results and theoretical approaches. The number of sub-domains used for the IISD + ISP case is 4. For the Schwarz scheme, 4 sub-blocks (an ILU(0) solver is used on each block) and an overlapping of a single layer of nodes around the interface between the blocks are used.

This kind of example is for cases where the computational resources are limited to a single processor architecture and it is not possible to get a solution using the preconditioned global GMRes scheme. The mesh used was composed by 24 150 quadrangular elements and 24 462 nodes. In order to capture the high thermal and flow gradients, the normal spacing close to the flat plate was chosen about 4×10^{-6}. The time step adopted was $\Delta t = 0.005$. The initial state adopted is a stationary flow at Mach 2.5 at the inlet, previously obtained via the IISD + ISP method. Two Newton loops were used for the non-linear problem.

Figures 2 and 3 show the skin friction coefficient and the Stanton number against theoretical predictions based on analytical solutions of an approximate theory called *Eckert reference enthalpy method* [21]. These results show good behaviour of the numerical results relative to the analytical predictions.

The test was conducted in a PC Pentium IV-2.8 GHz (RAM DDR, 400 MHz). The CPU time per time steps (i.e. less than 3 min in average) and the residual convergence rate (roughly 150–170 iterations to converge 7 orders of magnitude) were comparable for both DDMs. In the IISD + ISP solver, the sub-domain problems are solved with a LU decomposition with *nested dissection* re-ordering. If complete LU factorization is used in the Schwarz method, the memory requirements and CPU time per time steps are increased.

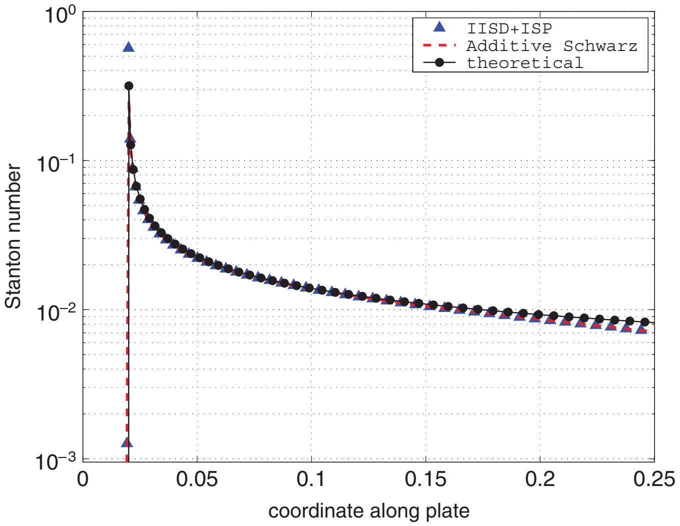

Figure 3. Stanton number.

5. NUMERICAL TESTS AND EXAMPLES IN PARALLEL ENVIRONMENT

In this section, the quality of numerical results for several problems considered as benchmarks in the CFD community is studied. The tests were carried out on a *Beowulf* cluster of PC's. The cluster at CIMEC laboratory has 16 (uniprocessor) nodes Pentium IV-2.8 GHz, 2 GB RAM (DDR, 400 MHz). Usually, the first node works as server. The nodes are connected through a switch Fast Ethernet (100 Mbit/s, latency $= O(100)$ μs, each node has a 3COM 3c509 (Vortex) Nic cards).

The tests presented in this work were solved using the PETSc-FEM code (see Reference [19]), a general purpose, parallel, multi-physics FEM program for CFD applications based on MPI and PETSc libraries (see References [22, 23], respectively). PETSc-FEM comprises both a library that allows the user to develop FEM (or FEM-like, i.e. unstructured 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). Mesh partitioning is performed by using METIS (see Reference [24]).

5.1. The Stokes flow in a long horizontal channel

Triggered by observed discrepancies between experimental results and computer simulations [25] using standard solvers [26] (i.e. GMRes with acceptable rates of convergence), this example shows the improvement of the solution of lubricated contacts by means of the ISP preconditioner. So far 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 at low Reynolds number ($Re < 1$) when a narrow gap is assumed

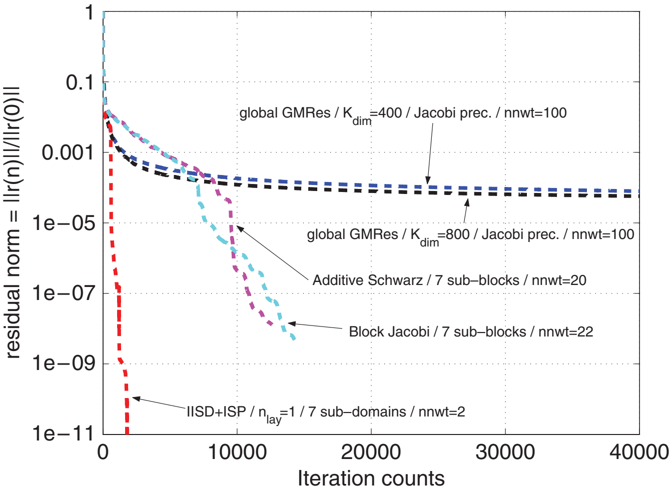

Figure 4. Residual history.

(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. An accurate description of the flow then requires the use of incompressible laminar Navier–Stokes model.

The channel is 8×10^{-5} m width and 9×10^{-2} m long. The dynamic viscosity used is $5.33\times 10^{-4}\,\mathrm{m}^2/\mathrm{s}$ and no body forces are considered. The Reynolds number based on the channel width is $Re=0.1$. The aspect ratio of the quadrilateral elements is 5 to assure non-stretched elements. This problem leads to an ill-conditioned matrix due to the high aspect ratio of channel dimensions and a big number of residual vectors (iterations) in Krylov methods it is necessary to converge to an accurate solution. The non-linear steady simulation with a maximum of 100 Newton loops is considered. The normalized residuals in the solution step of the linear system are shown in Figure 4 for all Newton iterations (hereafter nnwt is the number of iterations in the non-linear loop). In the case of global GMRes solver (point-Jacobi preconditioning is assumed hereafter for this method) two Krylov subspace dimensions are considered (i.e. 400 and 800). The test was conducted in 16 nodes and each sub-domain was sub-partitioned into 7 interior sub-domains (2000 dof's per interior sub-domains in average) in the case of IISD + ISP solver. The interface strip width used is $n_{\mathrm{lay}}=1$ (see Reference [11]). For the overlapping additive Schwarz and the Block Jacobi methods 7 sub-blocks per processor were chosen (ILU(0) decomposition is used on each block). As in previous tests, sub-blocks overlap (for the overlapping additive Schwarz) each other by a layer of nodes. In Stokes flow the convective terms are quite small. However, as these terms remain in the formulation they lead to a weak non-linear problems with non-symmetrical matrices. 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 LU factorization stage (solution of local problems) and the GMRes iteration (solution of inter-sub-domain problems). The CPU time was 0.33 min per Newton loop.

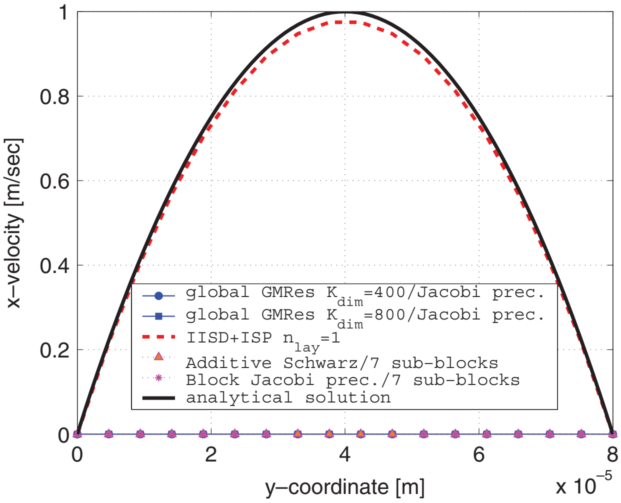

Figure 5. Velocity field in the channel height (nnwt = 1).

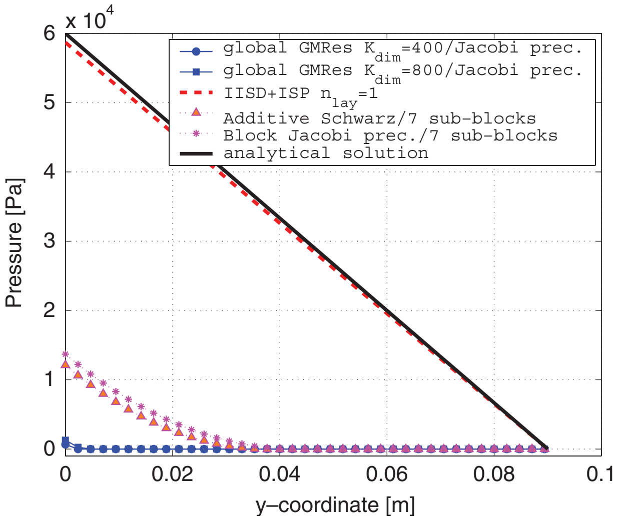

Figure 6. Pressure field along channel (nnwt = 1).

The memory used in the global GMRes stage was 126.9 Mb per processor for a Krylov subspace dimension (K_{dim}) of 800 and 63.2 Mb per processor for a K_{dim} of 400. The CPU time was 6.61 and 1.87 min per Newton iteration, respectively.

For the overlapping additive Schwarz preconditioning the consumed memory and the CPU time per Newton iteration were 107 Mb and 1.1 min, respectively. Block Jacobi scheme consumed 99.3 Mb and 0.97 min per nonlinear iteration.

In Figures 5–8 the numerical solution of the horizontal velocity and pressure fields are compared to the analytical one. Figures 5 and 6 correspond to the solution of both fields after

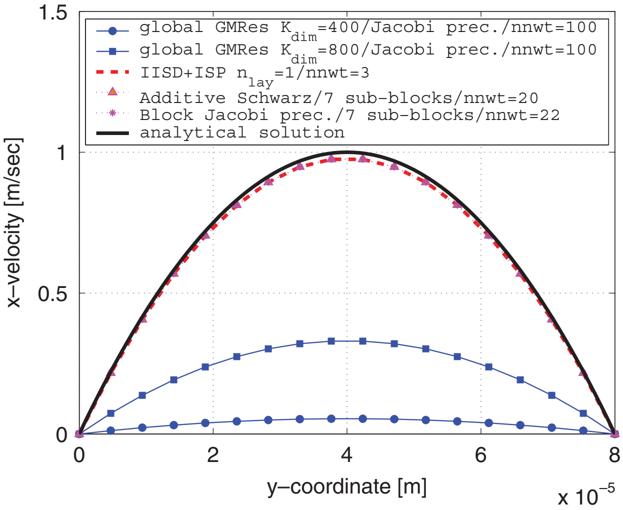

Figure 7. Velocity field in the channel height (nnwt = 100 for global GMRes, nnwt = 3 for IISD + ISP, nnwt = 20 for additive Schwarz, nnwt = 22 for Block Jacobi).

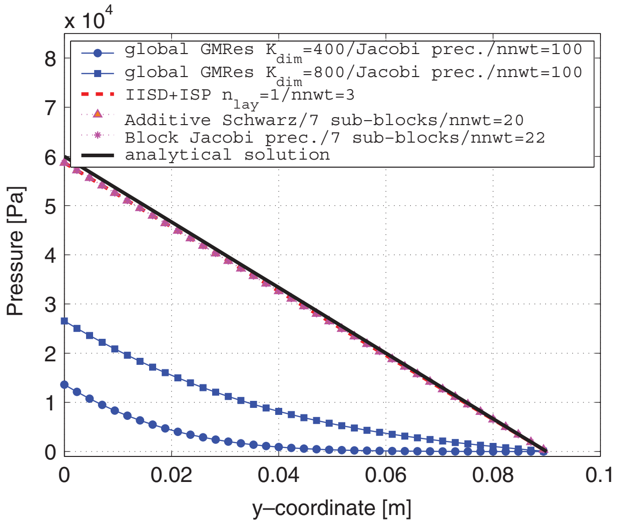

Figure 8. Pressure field along channel (nnwt = 100 for global GMRes, nnwt = 3 for IISD + ISP, nnwt = 20 for additive Schwarz, nnwt = 22 for Block Jacobi).

one loop in the Newton scheme and Figures 7 and 8 correspond to one hundred iterations in Newton's loop for the preconditioned global GMRes method (20 and 22 iterations for additive Schwarz and Block Jacobi schemes, respectively). 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. The same residual tolerance was obtained by additive Schwarz and Block Jacobi preconditioned on the 20th and 22nd loops, respectively. Clearly, in this case IISD + ISP outperforms other domain decomposition techniques not only in memory and CPU time demands, but also in the number of non-linear iterations to achieve a given tolerance.

Figures 5 and 7 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 inspired this work. It shows that for high aspect ratio geometries the global GMRes (and probably many full iterative solvers) suffers a strong convergence deterioration and even when using an unusually high size of Krylov subspace dimension the final solution is unacceptable.

5.2. The incompressible Navier–Stokes flow around an infinite cylinder

The unsteady viscous external flows past objects have been extensively studied (experimentally and numerically, see References [27–33]) because of their many practical applications. This example is directly related to a great number of problems. For example, airfoils have streamline shapes in order to increase the lift and reduce the aerodynamic drag exerted on the wings at the same time. On the other hand, the 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 the boundary layer vortex being shed alternatively from either side of the cylinder. This regular pattern of vortices in the wake is called the von-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 order to generate vortex shedding, an artificial perturbation may be imposed introducing for example 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.* [27]). The problem is 2D and the cylinder radius is 1 m. On the inlet boundary a uniform free stream velocity ($\|\mathbf{u}\| = 1$) is imposed. On the outlet section the pressure is equal to a reference value (zero in this test) and the velocity vector has no component in the y-direction. On the top and bottom walls a slip condition is adopted. These boundaries are located far enough to prevent any influence on the flow development (see Reference [29]). The mesh of 138600 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 (2000 dof's per local subdivision).

In Figure 9 the residual history for several time steps (one Newton iteration is considered) is plotted for the unsteady simulation of this flow. Clearly, the IISD + ISP solver reaches the lower residual tolerances (10^{-7} vs 10^{-3}) with roughly 70% fewer iterations compared to global GMRes iteration with different K_{dim}. The lower tolerances achieved with IISD + ISP are directly related to the accuracy in the solution and the reduction of the iteration number influences the overall simulation time.

In Figures 10–12 the viscous forces and moment evolution in time are shown. The solution obtained with both Global Iteration ($K_{\mathrm{dim}} = 400$) and IISD + ISP are in agreement with

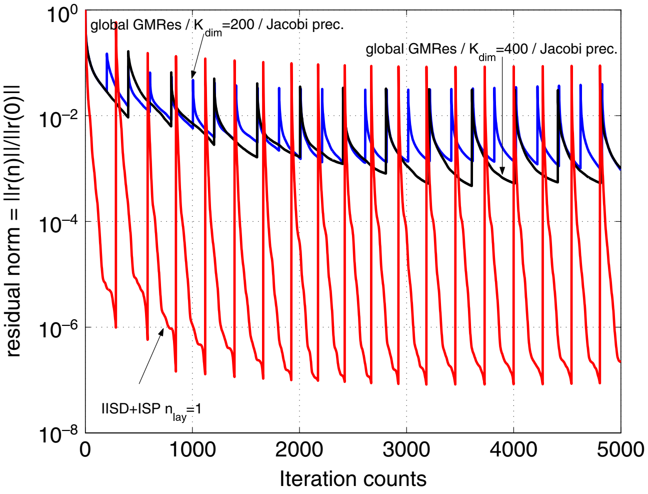

Figure 9. $Re = 100$. Residual history.

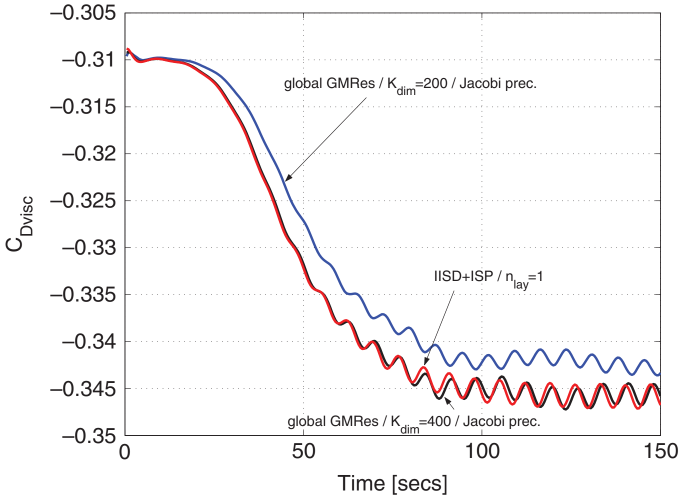

Figure 10. $Re = 100$. viscous x-force coefficient.

the experimental results reported by Braza *et al.*, and with the numerical results shown in References [27, 29, 30]. However, if Global iteration is stopped prematurely (i.e. $K_{\mathrm{dim}} = 200$, see Figure 9) the solution is not more accurate and divergences of 50% are observed (see Figure 12). 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. Please note that to go from 200 iterations to 400 iterations in GMRes scheme the computational resources (CPU and consumed memory) are considerably increased due to the storage requirements for the

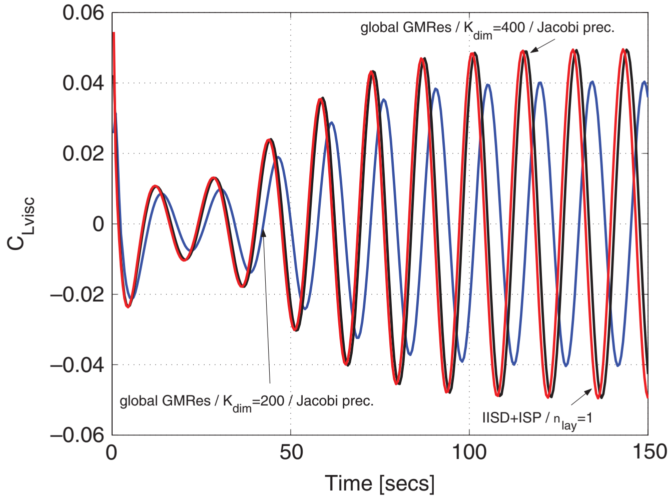

Figure 11. $Re = 100$. viscous y-force coefficient.

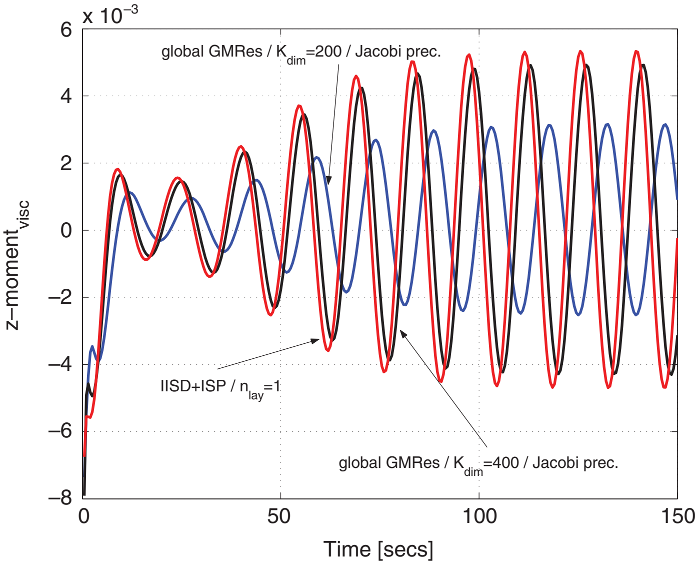

Figure 12. $Re = 100$. viscous z-moment coefficient.

Krylov subspace basis. The CPU consumed time and core memory requirements for each time step were: 28 s and 100 Mb for global GMRes with $K_{\text{dim}} = 200$, 107.5 s and 152 Mb for global GMRes with $K_{\text{dim}} = 400$ and 18.5 s and 98 Mb for IISD + ISP.

This well-known test case shows the need to use global GMRes with a high Krylov subspace dimension size to reach the accuracy of IISD + ISP solver. Usually, 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. In summary, global GMRes iteration makes the simulation more user dependent.

5.3. Incompressible Navier–Stokes flow using the fractional step scheme. The lid-driven cavity

A test for disaggregated methods was performed on a 2D unit cavity flow at $Re = 1000$. This test has been computed extensively in the past and it is well understood (see Reference [34] for a detailed description of this example). A structured mesh of 400×400 quadrilateral was used for calculations. Fourteen processors for global GMRes iteration and 14 sub-domains subdivided into local partitions in order to have 1500 dof's per subdivision for the IISD + ISP solver were used. Also, for additive Schwarz scheme method 14 sub-blocks per processor were used. $\Delta t = 0.02$ and $\gamma = 0.9$ (Equation (13)) were chosen.

In Figure 13 the residual history for the Poisson step for different solvers is shown.

In the predictor (advection–diffusion equation) and projection steps a few iterations are needed to achieve relative low tolerances for these schemes. Nevertheless, in the Poisson step, the mesh size used leads to a high condition number for CG iteration (i.e. $\propto 1/h^2$ without preconditioning), so it is necessary to increase the iteration number in order to avoid spurious oscillation in the solution. If global CG iteration is stopped at 1000–1200 iterations (Poisson step), where the residual history plot reaches a 'plateau', i.e. the residuals may be considered acceptable, large spurious oscillations appear in the solution when the steady state is reached. Moreover, it is necessary to overpass the 2400 iterations to avoid oscillation in all time steps. Although the required memory is not affected (please recall that in CG iteration, only the last two residual vectors are needed), but the CPU time grows linearly with the

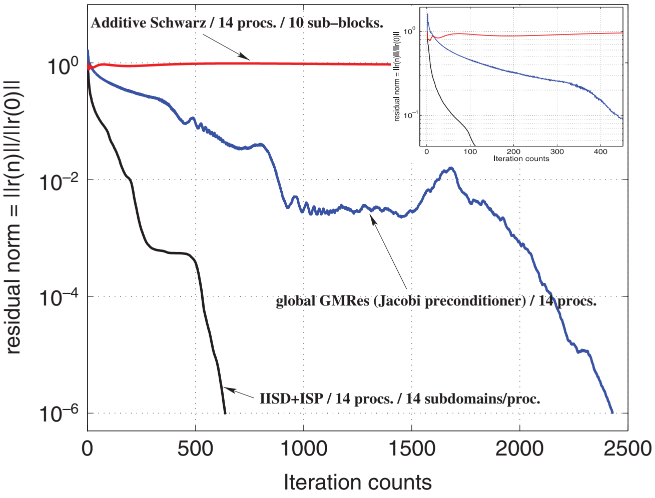

Figure 13. Residual history for Poisson step.

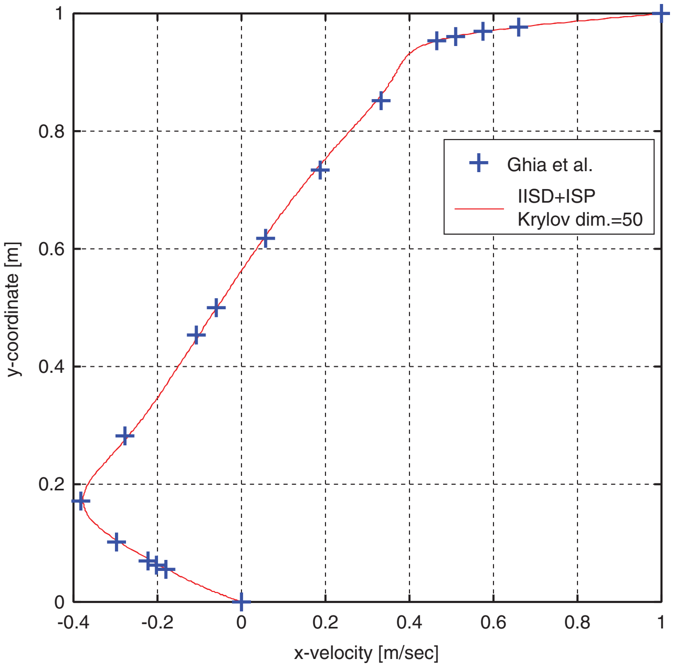

Figure 14. Time-converged solution for IISD + ISP solver ($Re = 1000$).

iterations. Using IISD + ISP the amount of memory to solve the interior direct problem and the preconditioning is chosen (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 14 shows the steady-converged solution stopping when the iteration reaches 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 s and for IISD + ISP (with one layer around the interface) solver was 2.13 s. Additive Schwarz scheme shows a poor convergence in residuals but it is not necessary to go beyond 200 iterations to have an acceptable solution. This last scheme uses 26 s per time step. The reader could refer to Reference [11] for a study of the performance of several preconditioners (including IISD+ISP and Neumann–Neumann preconditioners) applied to a Poisson problem.

Although the residuals for IISD+ISP simulation are higher than those of 1000 iterations of global CG (see Figure 13), the solution for DDM and preconditioning, (IISD+ISP) is accurate enough, being oscillatory for global iteration. This behaviour can be explained through a study of the error in residuals when iteration proceeds.

Let $b, u_k, u_0 \in \mathbb{R}^N$, A a non-singular matrix such that $u^* = A^{-1}b$. Here, b is the load vector and u_k the solution at iteration k. Since

$$r_k = b - Au_k = Au^* - Au_k = A(u_k - u^*) = -Ae_k \tag{27}$$

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

$$\|e_k\| = \|A^{-1}Ae_k\| \leqslant \|A^{-1}\| \|Ae_k\| = \|A^{-1}\| \|r_k\| \tag{28}$$

and

$$\|r_0\| \leqslant \|A\| \|e_0\| \tag{29}$$

thus,

$$\frac{\|e_k\|}{\|e_0\|} \leqslant \frac{\|A^{-1}\| \, \|r_k\|}{\|A\|^{-1}\|r_0\|} = \kappa(A) \, \frac{\|r_k\|}{\|r_0\|} \tag{30}$$

$\kappa(A)$ is the condition number of A and $\|.\|$ is any suitable norm. The division by $\|r_0\|$ and $\|e_0\|$ in Equation (30) normalizes the residuals. In Reference [10] it is shown 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 DDMs. Moreover, the IISD + ISP still reduces the last condition number. Though the error in residuals for IISD+ISP (due to earlier stopping) 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 distribution of the eigenvalues of the global matrix and its condition number κ (see Equation (30)). If the solution of the Poisson step is not accurate, the error is propagated to the other steps and oscillation may occur. This problem is stressed with the refinement (i.e. $h \to 0$).

The primary vortex centre was computed to be at $(x, y) = (0.531, 0.562)$ with the coordinate reference system placed at the bottom left corner of the cavity. IISD+ISP solver compares well (for the 50 iterations run) with the values reported by Ghia *et al.* The core memory used was 45 Mb/processor for Global iteration, 60 Mb/processor for IISD + ISP and 55 Mb/processor for the additive Schwarz preconditioner. Recalling that for CG iteration only the last two residuals are needed, then the memory is not increased with iterations.

This example allows to emphasize another interesting use of IISD+ISP solver. In fractional step-like flow solvers the Poisson step has normally the higher CPU time consumption and for ill-conditioned problems this step demands a lot of resources to achieve a good solution. It is difficult to know right from the start how large the Krylov subspace dimension size of CG 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 its usage too limited. With IISD+ISP it is possible to strongly reduce these requirements, which drastically improves the solution and makes the system solution less user dependent.

Some comments on the scalability of the IISD+ISP preconditioner. In Reference [11] the condition number for the preconditioned Poisson and advection–diffusion problems was calculated theoretically for the ISP. In this section the scalability of the ISP is studied by showing how the number of iterations grows with the global size of the problem while keeping the size of the problem in each processor constant. Stokes flow at $Re = 0.01$ with monolithic time integration (time step used is $\Delta t = 0.1$) is considered. The square cavity is divided into $(20\,n_{\mathrm{proc}}) \times (20\,n_{\mathrm{proc}})$ bilinear elements, with n_{proc} being the number of processors. Inside each processor, the problem is further subdivided into 4 sub-domains and the number of element layers around the interface is kept constant to $n_{\mathrm{lay}} = 1$. In Figure 15 the number of iterations to achieve a relative tolerance of 10^{-8} in the residuals in a given time step is

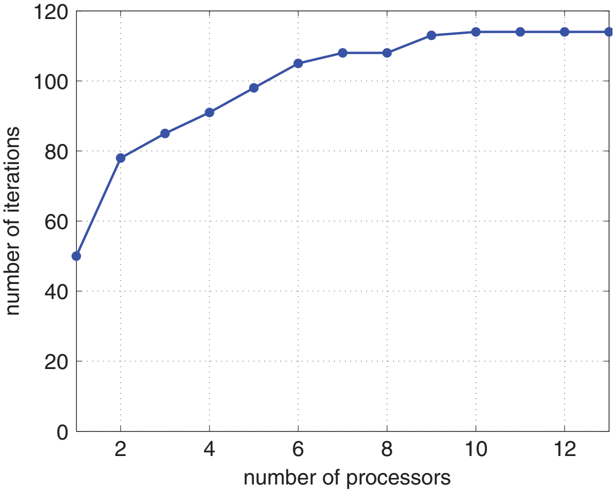

Figure 15. Scalability properties.

shown. It may be observed that the number of iterations saturates for an increasing problem size, ensuring the scalability of the preconditioner.

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

Current vehicle design needs a strong background in aerodynamics to improve flow control via mechanical devices. The complexity involved in the automobile design specially due to the great amount of accessories that define its geometry makes the validation tasks unaffordable. 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 [14, 35]). The aerodynamic forces on road vehicles are the result of complex interactions between flow separations and the dynamic behaviour of the released vortex wake. The results obtained with two solvers (i.e. IISD + ISP and Jacobi preconditioned global GMRes methods) are compared to the detailed flow patterns previously published by Ahmed and coworkers [36].

The body geometry is defined in Reference [36]. 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 streamwise (y), spanwise (x) and stream-normal (z) directions. The boundary conditions for this problem are: uniform flow at the inlet (given by the Reynolds number), slip condition on both sides, non-slip condition for the surface of the body, non-slip condition at the floor. Imposed pressure (zero in this case) is used at the outflow boundary condition.

5.4.1. Ahmed body: numerical results for very low Reynolds number. First, consider the steady Stokes flow (solved using incompressible Navier–Stokes model at $Re = 0.1$) around the Ahmed body. In this test a non-structured tetrahedral mesh is used in the whole flow domain. A three-layer structured mesh of wedge (prismatic) elements is built for capturing

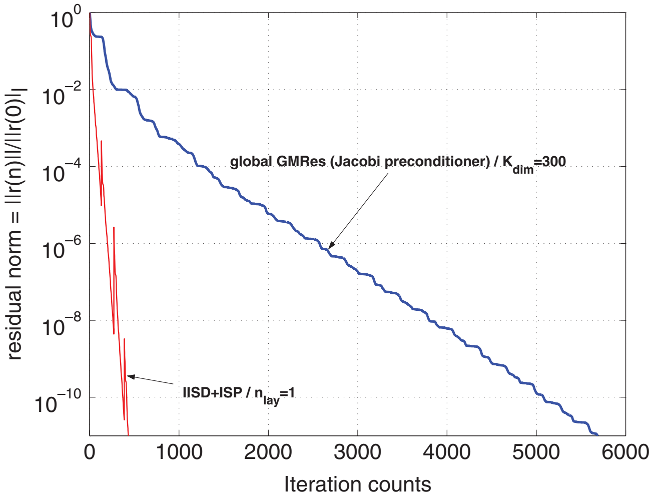

Figure 16. Stokes flow. Residual history (max. of 100 Newton iterations).

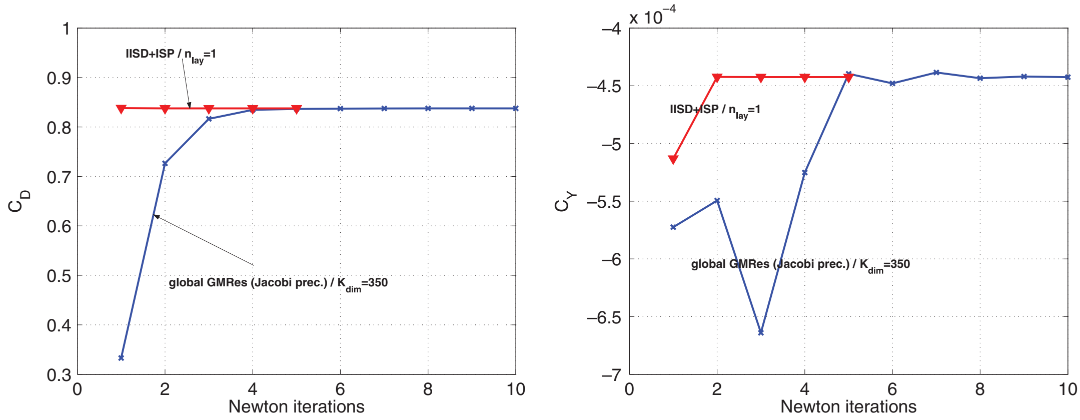

Figure 17. Stokes flow. Force and moment coefficients.

details at the boundary layers. The body surface mesh contains 90606 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+ISP solver 2000 dof's were considered for each local subdivision. The inlet (free stream) condition is $Re = 0.1$ with no transversal velocity. Pressure, $p = 0$ atm, is imposed at the outflow wall (located far enough from the body).

The residual history for several Newton loops are shown in Figure 16. The calculated forces and moments for both solvers are shown in Figures 17–19.

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

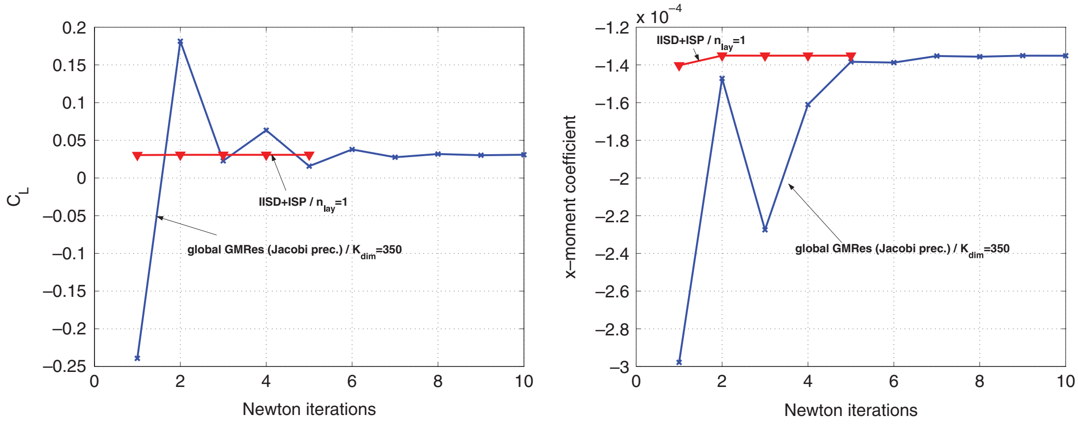

Figure 18. Stokes flow. Force and moment coefficients.

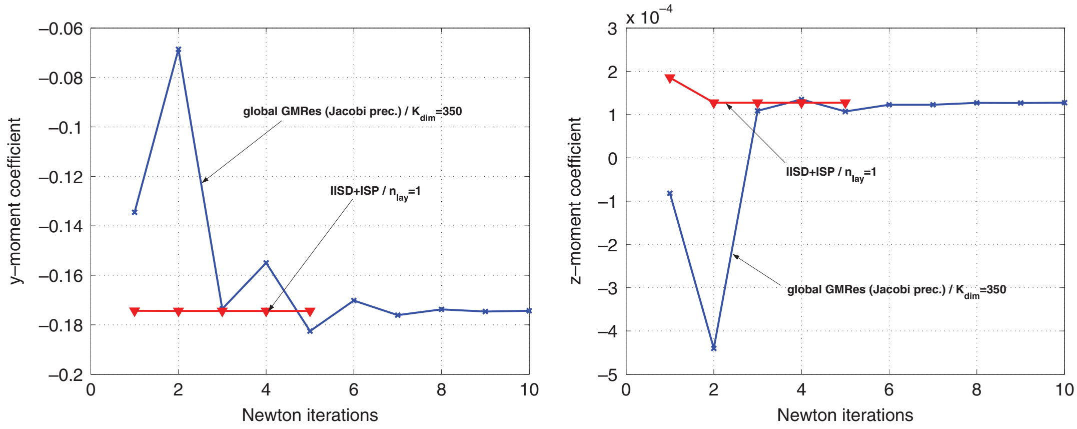

Figure 19. Stokes flow. Force and moment coefficients.

The CPU consumed time and core memory requirements (in average) for each Newton iteration were: 185.1 s and 443 Mb for global GMRes with $K_{\text{dim}} = 300$ and 64.18 s and 588 Mb for IISD + ISP.

5.4.2. Ahmed body: numerical results for high Reynolds number. In this section the unsteady incompressible Navier–Stokes simulation for the flow around the Ahmed body for $Re = 1000$ is shown. The same architecture, mesh and partitions of the previous example was used. The initial state used is the steady-converged solution of the previous example (IISD + ISP case). The Smagorinsky model (with the Smagorinsky parameter equal to 0.18) is used for the LES prediction of the turbulent effects.

The residual history for several time steps is shown in Figure 20.

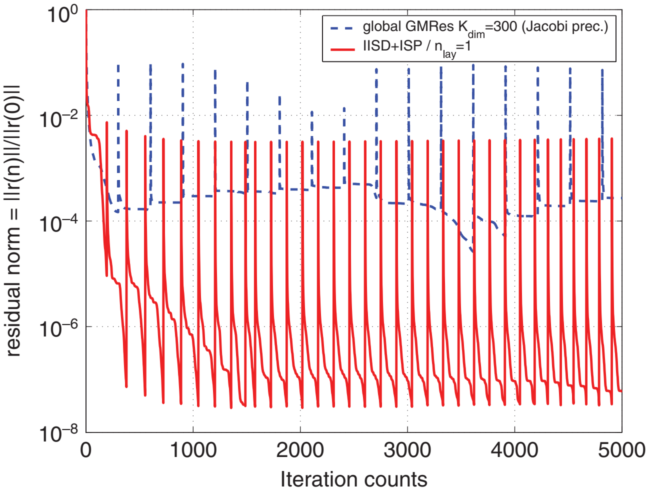

Figure 20. $Re = 1000$. Residual history (100 time steps, 10 s of simulation).

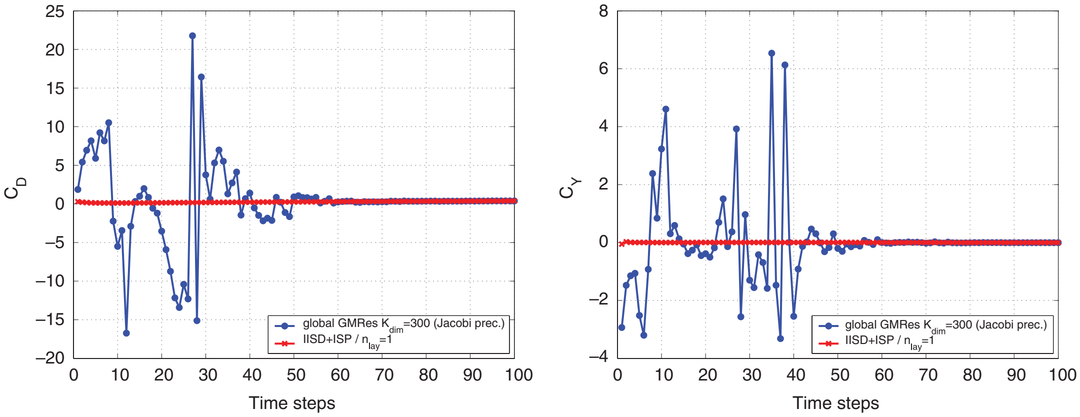

Figure 21. $Re = 1000$. Force and moment coefficients.

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

The scale of the force and moment coefficient plots is dominated by the bad approximation and oscillations in the solution obtained with the preconditioned global GMRes iterations. The calculated forces with IISD + ISP solver converge very fast to the values reported in the literature [36].

The CPU consumed time and core memory requirements (in average) for time step in this test were: 186 s and 460 Mb for global GMRes iteration with $K_{\text{dim}} = 300$ and 114.5 s and 630 Mb for the ISD + ISP preconditioner.

The 'friction lines' for the Navier–Stokes flow at $Re = 4.25 \times 10^6$ are shown in Figure 24 (IISD + ISP case).

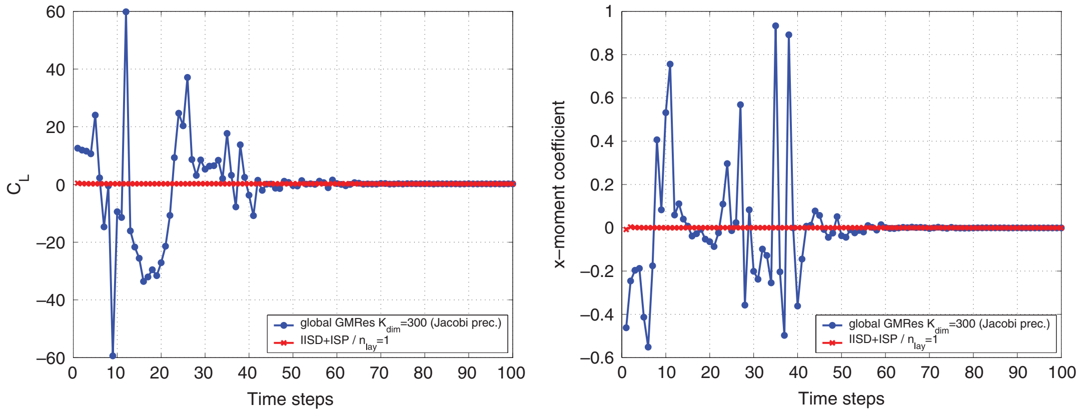

Figure 22. $Re = 1000$. Force and moment coefficients.

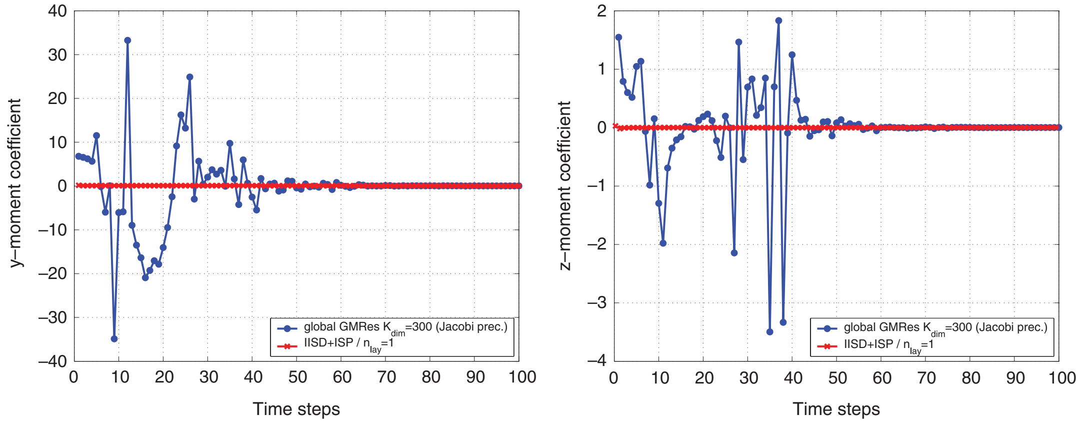

Figure 23. $Re = 1000$. Force and moment coefficients.

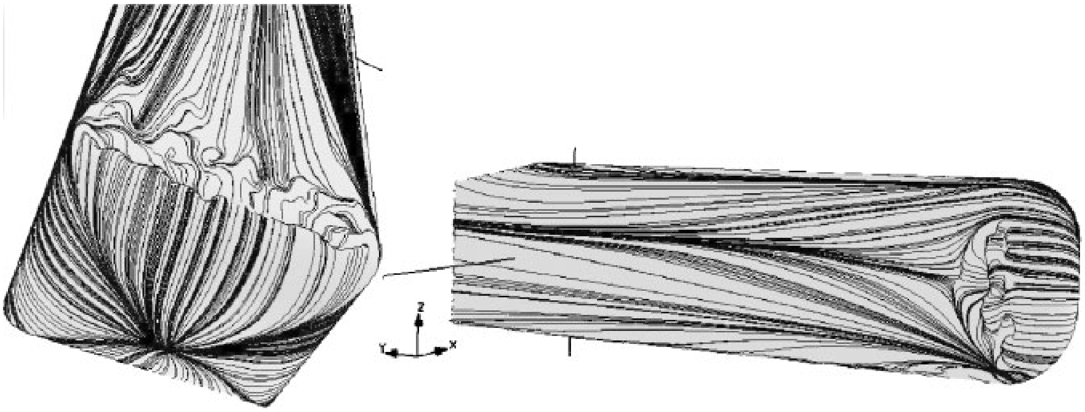

Figure 24. $Re = 4.25 \times 10^6$. Friction lines.

This 3D example shows that even when the final solution of global GMRes solver 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 highlighted in the Stokes flow example ($Re = 0.1$). For medium Reynolds numbers the global GMRes has a strong oscillatory behaviour until it reaches the good solution, making the scheme under external perturbations more unstable.

6. CONCLUSIONS

This paper emphasizes the quality and the efficiency of solver schemes for CFD problems. Both criteria should be evaluated together to analyse the performance of a simulation. Reasonable efficiency might not be very significant if the solution is not accurate enough for the final purpose. Several examples presented in this paper shed light on the pathologies that may appear when solving large-scale CFD problems by means of fully iterative solvers with limited computational resources.

Numerical experiments of several physical (real) problems were carried out to show its convergence properties, the computation time and memory requirements using both monolithic and disaggregated schemes. These tests showed that it is not always possible to obtain an acceptable solution for the problem using classical global Krylov methods. Moreover, for some problems Krylov dimension and Newton iterations need to be enlarged to obtain an accurate solution making its usage more user dependent.

Domain Decomposition techniques, especially the Schur Complement domain decomposition using the interface strip preconditioner are suitable in order to achieve accurate solutions efficiently. In all cases, the performance of IISD + ISP is decisive when it comes to assigning computational resources to solve a time step in the simulation of a problem. Also, the ISP 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 preconditioners such as Neumann–Neumann as References [10, 11] show. Moreover, it permits to decide how much memory to assign for preconditioning purposes.

The ISP 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 diffusion-dominated regions well. Furthermore, IISD + ISP is a good alternative to treat problems where domain discretization presents high refinement 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*, CAI+D-UNL-PIP-02552-2000 and ANPCyT-PICT 12-14573/03. Extensive use of freely distributed software such as *GNU/Linux* OS, MPI, PETSc, Metis, Octave, the CGAL geometrical library, OpenDX and many others is done in this work. The authors would like to thank Dr Gustavo Buscaglia for helpful discussions related to the lubrication channel problem.

REFERENCES

1. Bramble JH, Pasciak JE, Schatz AH. The construction of preconditioners for elliptic problems by substructuring, I. *Mathematics of Computation* 1986; **47**(175):103–134.
2. Bramble JH, Pasciak JE, Schatz AH. The construction of preconditioners for elliptic problems by substructuring, IV. *Mathematics of Computation* 1989; **53**(187):1–24.
3. Farhat C, Roux FX. A method of finite element tearing and interconnecting and its parallel solution algorithm. *International Journal for Numerical Methods in Engineering* 1991; **32**:1205–1227.
4. Mandel J. Balancing domain decomposition. *Communications in Applied Numerical Methods* 1993; **9**:233–241.
5. Smith B, Bjørstad P, Gropp W. *Domain Decomposition*: *Parallel Multilevel Methods for Elliptic Partial Differential Equations*. Cambridge University Press: Cambridge, MA, 1996.
6. Le Tallec P, Vidrascu M. Solving large scale structural problems on parallel computers using domain decomposition techniques. In *Parallel Solution Methods in Computational Mechanics*, *Chapter 2*, Papadrakakis M (ed.). Wiley: New York, NY, 1997; 49–85.
7. Meurant G. *Computer Solution of Large Linear Systems*. Studies in Mathematics and its Applications, vol. 28. North-Holland: Amsterdam, 1999.
8. Saad Y. *Iterative Methods for Sparse Linear Systems*. PWS Publishing Co.: MA, 2000. http://www-users.cs.umn.edu/saad/books.html
9. Rachowicz W. An anisotropic h-adaptive finite element method for compressible Navier–Stokes equations. *Computer Methods in Applied Mechanics and Engineering* 1997; **146**:231–252.
10. Storti MA, Dalcin L, Paz RR, Yommi A, Sonzogni V, Nigro N. An interface strip preconditioner for domain decomposition methods. *Journal of Computational Methods in Science and Engineering*, 2003, in press.
11. Paz RR, Storti MA. An interface strip preconditioner for domain decomposition methods: application to hydrology. *International Journal for Numerical Methods in Engineering* 2005; **62**(13):1873–1894.
12. 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.
13. 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.
14. Smagorinsky J. General circulation experiments with the primitive equations. *Monthly Weather Review* 1963; **91**(3):99–165.
15. Codina R. Pressure stability in fractional step finite element methods for incompressible flows. *Journal of Computational Physics* 2001; **170**:112–140.
16. Aliabadi SK. Parallel finite element computations in aerospace applications. *Ph.D. Thesis*, Department of Aerospace Engineering and Mechanics, University of Minnesota, 1994.
17. Tezduyar T, Senga M. Determination of the shock-capturing parameters in supg formulation of compressible flows. In *Computational Mechanics WCCM IV*, Springler (ed.). Tsinghua University Press: Beijing, China, 2004.
18. De Roeck YH, Le Tallec P. Analysis and test of a local domain decomposition preconditioner. In *Fourth International Symposium on Domain Decomposition Methods for Partial Differential Equations*, Glowinski, Kuznetsov, Meurant, Pèriaux, Widlund (eds). SIAM: Philadelphia, PA, 1991.
19. Storti MA, Nigro NM, Paz RR. PETSc-FEM: a general purpose, parallel, multi-physics FEM program. http://www.cimec.org.ar/petscfem
20. Carter JE. Numerical solutions of the Navier–Stokes equations for the supersonic laminar flow over two-dimensional compression corner. *Technical Report R-385*, National Aeronautics and Space Administration (NASA), 1972.
21. Grasso F, Leone G, Delery J. Validation procedure for the analysis of shock-wave/boundary-layer interaction problems. *AIAA Journal* 1994; **32**(9):1820–1827.
22. Gropp W, Lusk E, Skjellum A. *Using MPI*: *Portable Parallel Programming with the Message-Passing Interface* (2nd edn). 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. George K. METIS: family of multilevel partitioning algorithms (1996–2005), http://www-users.cs.umn.edu/karypis/metis
25. 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.
26. Stay MS, Barocas VH. Coupled lubrication and Stokes flow finite elements. *International Journal for Numerical Methods in Fluids* 2003; **42**(2):129–146.
27. 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.

28. Chorin AJ. Numerical study of slightly viscous flow. *Journal of Fluid Mechanics* 1973; **57**:785–796.
29. Shih R, Tezduyar TE. Numerical experiments with the location of the downstream boundary for flow past a cylinder. *Research Report UMSI 90/38*, University of Minnesota Supercomputer Institute, 1990.
30. 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. *Research Report UMSI 90/87*, University of Minnesota Supercomputer Institute, 1990.
31. Roshko A. On the drag and shedding frequency of two dimensional bluff bodies. *Technical Note 3169*, National Advisory Committee for Aeronautics (NACA), 1954.
32. Williamson CHK. Evolution of a single wake behind a pair of bluff bodies. *Journal of Fluid Mechanics* 1985; **159**:1–18.
33. Norberg C. Flow around a circular cylinder: aspects of fluctuating lift. *Journal of Fluids and Structures* 2001; **15**:459–469.
34. 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.
35. Krajnović S, Davidson L. Numerical study of the flow around the bus-shaped body. *Journal of Fluids Engineering* (ASME) 2003; **125**:500–509.
36. Ahmed SR, Ramm G, Faltin G. Some salient features of the times-averaged ground vehicle wake. *SAE Society of Automotive Engineering Inc* 1984; **1**(840300):1–31.