Computing and Visualization in Science manuscript No.
(will be inserted by the editor)

A fast preconditioner for the incompressible Navier Stokes Equations

Alain Gauthier, Fausto Saleri, and Alessandro Veneziani[1⋆]

MOX (Modeling and Scientific Computing), Department of Mathematics "F. Brioschi", Politecnico di Milano Piazza L. da Vinci 32, I-20133 MILAN, Italy

Received: date / Revised version: date

Abstract. The *pressure matrix method* is a well known scheme for the solution of the incompressible Navier-Stokes equations by splitting the computation of the velocity and the pressure fields (see, e.g., [13]). However, the set-up of effective preconditioners for the pressure matrix is mandatory in order to have an acceptable computational cost. Different strategies can be pursued (see, e.g., [4], [18]). Inexact block *LU* factorizations of the matrix obtained after the discretization and linearization of the problem, originally proposed as fractional step solvers, provide also a strategy for building effective preconditioners of the pressure matrix (see [19]). In this paper, we present numerical results about a new preconditioner, based on an inexact factorization. The new preconditioner applies to the case of the generalized Stokes problem and to the Navier-Stokes one, as well. In the former case, it improves the performances of the well known Cahouet-Chabard preconditioner (see [2]). In the latter one, numerical results presented here show an almost optimal behavior (with respect to the space discretization) and suggest that the new preconditioner is well suited also for "flexible" or "inexact" strategies, in which the systems for the preconditioner are solved inaccurately.

Key words Incompressible Navier-Stokes Equations, Fractional Step Schemes, Preconditioned Pressure Matrix Method

1 Introduction

Numerical solution of the unsteady Navier-Stokes equations for incompressible flows in real applications requires the solution of linear systems of large dimensions. Many numerical methods in this field lead to systems which are neither definite nor well conditioned. The set-up of efficient methods for solving such systems is therefore mandatory. A possible successful approach is provided by the pressure matrix method (PMM in the sequel - see e.g. [13]). This method (see [11]) can be regarded at an algebraic level as a suitable block *LU* factorization of the discretized and linearized problem, which implies the splitting of the velocity and pressure computation. The solution of the pressure matrix system at each time step is however very expensive. A possible way of reducing the computational cost is to resort to *inexact factorization methods* (see [9], [11], [12]). These schemes compute velocity and pressure separately: (i) by computing an *auxiliary* (or *intermediate*) velocity; (ii) by solving an approximate problem for the pressure; (iii) by correcting the velocity (*end-of-step velocity*). The three steps stem from an approximation of the block *LU*-factorization underlying the PMM. In [12] we have introduced a general class of algebraic splitting methods that can be considered in this framework. We recall, in particular, the *Algebraic Chorin-Temam* (ACT) scheme, which is an algebraic formulation of the well-known Chorin-Temam projection method ([9]) and the *Yosida* (YOS) one ([11]). The computational cost is reduced, even if the inexactness of the factorization introduces a splitting error. A relevant point, in this respect, is to ensure that the splitting error does not affect significantly the global accuracy of the solution (see [11], [12], [15]).

Another way for reducing the computational cost relies on building effective preconditioners of the pressure matrix in the PMM. In this context many approaches have been investigated (see [2], [7], [16], [4], [18] and the references quoted there). Our starting point is that inexact factorizations can be considered not only for building approximate Navier-Stokes solvers, but also for the set-up of preconditioners of the pressure matrix. In [19] it is proven, for example, that the ACT and YOS inexact factorizations amount to use the so called "compatible Laplacian discretization" as a preconditioner of the pressure matrix. Such preconditioner leads to accurate

Send offprint requests to: A. Veneziani

⋆ This work has been supported by Cofin2000 MIUR Project. The authors wish to thank Alessandro Gilardi (MOX, PoliMi), Luca Formaggia (MOX, PoliMi), Jean Frederic Gerbeau (INRIA, Paris) and Miguel Fernandés (IMA, EPF Lausanne) for the development of the C++ 3D code.

results in time with a small number of iterations, even if it is not optimal with respect to the space discretization.

In this paper, we introduce a new preconditioner based on a modified ACT method, featuring a fourth step corresponding to a final "pressure correction" (see [15]). Numerical results show that the new preconditioner improves the results of the well known Cahouet-Chabard one (see [2] and [1]) in the case of the generalized Stokes problem. On the other hand, in the case of the Navier-Stokes problem, it still features an almost optimal behavior. Moreover, it seems also to be well suited for inexact or "flexible" strategies, ensuring the convergence in a small number of iterations even if it is solved inaccurately.

The outline of the paper is the following. In Sect. 2 we introduce some basic notations about the incompressible Navier-Stokes equations and their numerical solutions. The "pressure-correction" factorization and the set-up of the corresponding preconditioner are illustrated in Sect. 3. Numerical results are presented in Sect. 4, both for 2D and 3D computations.

2 Definition of the problem and its discretization

Consider an open and bounded domain $\Omega \subset \mathbb{R}^d$ for $d = 2, 3$ with boundary $\partial\Omega$ for a time $t \geq 0$. The Navier-Stokes equations for an incompressible flow in terms of the velocity, $\mathbf{u} = \mathbf{u}(\mathbf{x}, t)$, and the pressure, $p = p(\mathbf{x}, t)$, read:

$$\begin{cases} \dfrac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} - \nu \triangle \mathbf{u} + \nabla p = \mathbf{f}, \\ \nabla \cdot \mathbf{u} = 0, \end{cases} \tag{1}$$

for any $(\mathbf{x}, t) \in \Omega \times (0, T]$, with $T > 0$. This system must be completed with the initial condition $\mathbf{u}(\mathbf{x}, 0) = \mathbf{u}^0(\mathbf{x})$ (where $\mathbf{u}^0(\mathbf{x})$ is a given function) and suitable boundary conditions on $\partial\Omega$. For example, we split $\partial\Omega$ in two parts, Γ_D and Γ_N, such that:

$$\Gamma_D \cup \Gamma_N = \partial\Omega, \quad \Gamma_D \cap \Gamma_N = \emptyset$$

and

$$\mathbf{u}\,|_{\Gamma_D} = \mathbf{g}, \quad -p\mathbf{n} + \nu \nabla \mathbf{u} \cdot \mathbf{n}\,|_{\Gamma_N} = \mathbf{s}, \tag{2}$$

where $\mathbf{g}$ and $\mathbf{s}$ are given functions defined on Γ_D and Γ_N respectively, and $\mathbf{n}$ is the normal outward unit vector to $\partial\Omega$.

In order to have a quantitative evaluation of the flow field, a numerical approximation has to be carried out. This aim is achieved discretizing the problem with respect to time and space variables.

Concerning the *space discretization*, we will basically refer to the *Galerkin method* and, in particular, to the *finite element method* (FEM) (see, e.g., [13]). In fact, we choose functional spaces for the approximate velocity and pressure fields which satisfy the *inf-sup* or *BB condition*. In the sequel, we will denote by $N_{\mathbf{u}}$ and by N_p the number of degrees of freedom for the velocity and the pressure, respectively.

Table 1. Coefficients of the BDF schemes and of the extrapolation for the linearization of the convective term, up to the order 3 of accuracy.

p	α_0	α_1	α_2	α_3	β_1	β_2	β_3
1	1	-1	-	-	1	-	-
2	3/2	-2	1/2	-	2	-1	-
3	11/6	-3	3/2	$-1/3$	3	-3	1

For what concerns the *time discretization*, we will refer to classic backward difference methods (BDF - see e.g. [10]). We consider a decomposition of the time interval into N subintervals (t^n, t^{n+1}) with $t^n = n\Delta t$, where $\Delta t = T/N$ is the uniform positive time step and collocates the equation in the instants $t^n = n\Delta t$. The time derivative will be therefore approximated as:

$$\frac{\partial \mathbf{u}}{\partial t}\,|_{t=t^n} \approx \frac{1}{\Delta t} \sum_{i=0}^{p} \alpha_i \mathbf{u}^{n-i},$$

where the coefficients α_i up to the third order of accuracy are given in Tab. 1.

The discretization with respect both to time and space variables leads to a non linear (typically large) algebraic system:

$$\mathcal{F}(\mathbf{w}^{n+1}) = \mathbf{0},$$

where $\mathbf{w}^{n+1} = [\mathbf{u}_h^{n+1}, p_h^{n+1}]^T$ denotes the vector of the nodal values of the discrete velocity and pressure. Different approaches can be pursued for the solution of this system. A first possibility is the classical Newton method. Another strategy is based on a semi-implicit linearization of the convective term. In the first case, at each time step we have to carry out an inner loop whose generic k^{th} iteration reads:

$$\mathcal{A}_k \left(\mathbf{w}^{n+1,k+1} - \mathbf{w}^{n+1,k} \right) = -\mathcal{F}(\mathbf{w}^{n+1,k}),$$

where $\mathcal{A}_k$ denotes the Jacobian matrix associated to $\mathcal{F}(\mathbf{w})$ evaluated in $\mathbf{w}^{n+1,k}$. In the latter case, we set:

$$\left(\mathbf{u}^{n+1} \cdot \nabla\right) \mathbf{u}^{n+1} \approx \left(\mathbf{u}^* \cdot \nabla\right) \mathbf{u}^{n+1}, \quad \text{with} \quad \mathbf{u}^* = \sum_{i=1}^{p} \beta_i \mathbf{u}^{n+1-i},$$

where the coefficients β_i up to the third order of accuracy are given in Tab. 1.

Both approaches lead to solve linear systems of the form:

$$\mathcal{A}\mathbf{v} = \mathbf{b}, \tag{3}$$

where the vector $\mathbf{b}$ is given by the forcing term, the boundary conditions and terms coming from the time discretization and, possibly, the Newton iterations. Matrix $\mathcal{A}$ reads:

$$\mathcal{A} = \begin{bmatrix} \mathrm{C} & \mathrm{D}^T \\ \mathrm{D} & 0 \end{bmatrix}. \tag{4}$$

Here D^T denotes the discrete gradient operator, D the discrete divergence operator and C collects contributions

from the time derivative, advection and diffusion operators. More specifically, we denote:

$$\mathrm{C} = \frac{\alpha_0}{\Delta t}\mathrm{M} + \mathrm{K},$$

where M is the velocity mass matrix and K corresponds to the discretization of the Laplace operator and of the convective term. In the case of the Stokes problem, the algebraic system obtained after the discretization is linear (it means that Newton method converges after one iteration), K corresponds just to the laplacian of the velocity and it is therefore symmetric and positive definite (s.p.d.). In this case system (3) is symmetric and however it is not definite. In any case, it typically features large dimensions and bad conditioning properties. The PMM is a way for reducing the computational effort, facing systems of lower dimension. In fact, it is based on a block Gaussian elimination. More precisely, it is equivalent to compute the following "exact" LU-block factorization:

$$\mathcal{A} = \begin{bmatrix} \mathrm{C} & 0 \\ \mathrm{D} & -\mathrm{D}\mathrm{C}^{-1}\mathrm{D}^T \end{bmatrix} \begin{bmatrix} \mathrm{I}_{N_\mathbf{u}} & \mathrm{C}^{-1}\mathrm{D}^T \\ 0 & \mathrm{I}_{N_p} \end{bmatrix}, \tag{5}$$

where $\mathrm{I}_{N_\mathbf{u}}$ and I_{N_p} denote the $N_\mathbf{u} \times N_\mathbf{u}$ and $N_p \times N_p$ identity matrices, respectively. Then, system (3) is solved by computing the two block triangular systems obtained, that amounts to compute a first intermediate velocity field, then the pressure and the end-of-step velocity. However, the scheme is expensive due to the presence of the matrix $\Sigma = \mathrm{D}\mathrm{C}^{-1}\mathrm{D}^T$ for the pressure computation. Indeed, such matrix can be faced only with an iterative approach, since it cannot be computed explicitly. On the other hand, in an iterative frame the residual computation is extremely costly due to the presence of the inverse of C (see [13]). Moreover, the condition number of Σ increases when Δt decreases. Therefore, the set-up of good preconditioners for Σ is mandatory.

3 The new preconditioner

3.1 Approximate Algebraic Factorizations

The computational cost of PMM can be reduced resorting to approximate, or *inexact* formulation of the block factorization (5) (see [12], [9], [11]) where C^{-1} is replaced by matrices easier to be computed. In [15], we propose the following inexact factorization:

$$\widehat{\mathcal{A}} = \begin{bmatrix} \mathrm{C} & 0 \\ \mathrm{D} & -\mathrm{D}\mathrm{H}\mathrm{D}^T \end{bmatrix} \begin{bmatrix} \mathrm{I}_{N_\mathbf{u}} & \mathrm{H}\mathrm{D}^T\mathrm{Z} \\ 0 & \mathrm{Z} \end{bmatrix} = \begin{bmatrix} \mathrm{C} & \mathrm{C}\mathrm{H}\mathrm{D}^T\mathrm{Z} \\ \mathrm{D} & 0 \end{bmatrix}, \tag{6}$$

where

$$H = \frac{\Delta t}{\alpha_0}\mathrm{M}^{-1}$$

is an approximation of C^{-1} based on the first-term truncation of the Neumann expansion:

$$\mathrm{C}^{-1} = \frac{\Delta t}{\alpha_0}\left(\mathrm{I}_{N_\mathbf{u}} + \frac{\Delta t}{\alpha}\mathrm{M}^{-1}K\right)^{-1}\mathrm{M}^{-1} = \frac{\Delta t}{\alpha_0}\sum_{i=0}^{\infty}\left(-\Delta t\mathrm{M}^{-1}K\right)^i\mathrm{M}^{-1}.$$

Approximate factorization (6) provides effective solvers for the original system, whenever $\widehat{\mathcal{A}}$ replaces $\mathcal{A}$. In particular, if $\mathrm{Z} = \mathrm{I}_{N_p}$, the resulting scheme is the ACT method (see [9]). In this case, the intermediate and the end-of-step pressure fields actually coincide. In the general case, for $\mathrm{Z} \neq \mathrm{I}_{N_p}$, the final pressure field is a correction of the intermediate one. For this reason, the corresponding factorization method is called *Algebraic Chorin Temam - Pressure Corrected* (ACT-PC). The features of ACT-PC are investigated in [15]. The preconditioner we are going to introduce is based on the matrix $\widehat{\mathcal{A}}$ of (6) used as a preconditioner for $\mathcal{A}$.

3.2 The pressure corrected preconditioner

Let us compute explicitly the inverse of $\widehat{\mathcal{A}}$:

$$\widehat{\mathcal{A}}^{-1} = \begin{bmatrix} \mathrm{I}_{N_\mathbf{u}} & -\mathrm{H}\mathrm{D}^T \\ 0 & \mathrm{Z}^{-1} \end{bmatrix} \begin{bmatrix} \mathrm{C}^{-1} & 0 \\ \mathrm{S}^{-1}\mathrm{D}\mathrm{C}^{-1} & -\mathrm{S}^{-1} \end{bmatrix} = \begin{bmatrix} \mathrm{C}^{-1} - \mathrm{H}\mathrm{D}^T\mathrm{S}^{-1}\mathrm{D}\mathrm{C}^{-1} & \mathrm{H}\mathrm{D}^T\mathrm{S}^{-1} \\ \mathrm{Z}^{-1}\mathrm{S}^{-1}\mathrm{D}\mathrm{C}^{-1} & -(\mathrm{S}\mathrm{Z})^{-1} \end{bmatrix},$$

where

$$\mathrm{S} = \mathrm{D}\mathrm{H}\mathrm{D}^T.$$

is the "compatible Laplacian discretization" (see [2]).

When using the matrix $\widehat{\mathcal{A}}$ as preconditioner for the system (3), we have:

$$\widehat{\mathcal{A}}^{-1}\mathcal{A} = \begin{bmatrix} \mathrm{I}_{N_\mathbf{u}} & \mathrm{C}^{-1}\mathrm{D}^T - \mathrm{H}\mathrm{D}^T\mathrm{S}^{-1}\Sigma \\ 0 & \mathrm{Z}^{-1}\mathrm{S}^{-1}\Sigma \end{bmatrix}. \tag{7}$$

From (7) it is evident that the convergence analysis of an iterative scheme where $\mathcal{A}$ is preconditioned by $\widehat{\mathcal{A}}$ actually reduces to consider SZ as a preconditioner for the pressure matrix Σ. Block LU inexact factorization can be, in fact, regarded as a way for the set-up of PMM preconditioners ([19]). In the present case, we would like to find out Z in such a way that the condition number of $(\mathrm{S}\mathrm{Z})^{-1}\Sigma$ is minimal. This specific choice seems to be quite difficult to be pursued and will be investigated elsewhere. Here, we follow a different approach. Consider again the original inexact factorization and select Z in order to minimize (in some sense) the splitting error matrix $\mathcal{A} - \widehat{\mathcal{A}}$. Since:

$$\mathcal{A} - \widehat{\mathcal{A}} = \begin{bmatrix} 0 & \mathrm{D}^T - \mathrm{C}\mathrm{H}\mathrm{D}^T\mathrm{Z} \\ 0 & 0 \end{bmatrix},$$

the splitting error matrix vanishes if:

$$\mathrm{C}\mathrm{H}\mathrm{D}^T\mathrm{Z} = \mathrm{D}^T. \tag{8}$$

Matrix equation (8) represents an overdetermined problem for Z. In order to obtain a solution, multiply both the sides of (8) by the matrix DH, yielding:

$$\mathrm{B}\mathrm{Z} = \mathrm{S}$$

where

$$\mathrm{B} = \mathrm{D}\mathrm{H}\mathrm{C}\mathrm{H}\mathrm{D}^T. \tag{9}$$

This means that the matrix equation (8) is solved up to matrix X on the right hand side such that DHX = 0. If the inf-sup condition is fulfilled, matrix B is non singular, so we finally get:

$$\mathrm{Z} = \mathrm{B}^{-1}\mathrm{S}, \tag{10}$$

and therefore the new preconditioner reads:

$$\mathrm{P}_{new} = \mathrm{S}\mathrm{B}^{-1}\mathrm{S}. \tag{11}$$

A rigorous analysis of P_{new} is in order, but it is beyond the scope of the present work. It is however worthwhile pointing out some interesting features. The first one refers to the computational cost associated to P_{new}. In any iterative method we have to compute the preconditioned residual. In our case this amounts to solve:

$$\mathrm{S}\mathrm{B}^{-1}\mathrm{S}\mathbf{q} = \mathbf{f} - \Sigma\mathbf{p},$$

where $\mathbf{q}, \mathbf{f}$ and $\mathbf{p}$ are generic vectors with the dimension of a (discrete) pressure field. In practice, this computation reduces to:

$$\begin{cases} \mathrm{S}\mathbf{y} = \mathbf{f} - \Sigma\mathbf{p} \\ \mathrm{S}\mathbf{q} = B\mathbf{y}. \end{cases} \tag{12}$$

A system in S is, therefore, solved twice for each computation of the preconditioned residual. This could be computationally expensive, in particular if an iterative method is adopted. However, systems for S can be faced also by means of direct methods (see [19]). Indeed, since H is a s.p.d. matrix, we can consider the QR factorization:

$$\mathrm{H}^{1/2}\mathrm{D}^T = \mathrm{Q}\mathrm{R}$$

where Q is an orthogonal square $(N_{\mathbf{u}} \times N_{\mathbf{u}})$ matrix and, if the inf-sup condition holds, R is a triangular full-rank $N_{\mathbf{u}} \times N_p$ matrix such that:

$$\mathrm{R} = \begin{bmatrix} \mathrm{R}_0 \\ \mathbf{0} \end{bmatrix}.$$

Here R_0 is a non-singular $N_p \times N_p$ triangular matrix. In practice, effective algorithms for the QR factorization of large sparse matrices are available (see [8] and the QR27 library; or in Matlab6, this factorization is obtained with the command `qr`). Thanks to this factorization, we have:

$$\mathrm{S} = \mathrm{R}_0^T\mathrm{R}_0.$$

In this way, the computational cost of the iterated solution of systems for S can be strongly reduced. Observe that the cost of the QR factorization in unsteady computations is limited at the beginning of the time loop, since matrix S is constant (at least in fixed computational domains). An alternative approach that we will investigate in Sect. 4.4 for the effective solution of (12) is based on the use of iterative solvers in a "flexible" sense (see [14]).

Another aspect we would like to point out is a relation between the new preconditioner and the well known Cahouet-Chabard one for the generalized Stokes problem. Since $\mathrm{C} = \mathrm{H}^{-1} + \mathrm{K}$, from (11) it folows:

$$\begin{aligned} &\mathrm{S}\mathrm{B}^{-1}\mathrm{S} = \mathrm{S}\left(\mathrm{D}\mathrm{H}(\mathrm{H}^{-1} + \mathrm{K})\mathrm{H}\mathrm{D}^T\right)^{-1}\mathrm{S} = \\ &\mathrm{S}\left(\mathrm{S} + \mathrm{D}\mathrm{H}\mathrm{K}\mathrm{H}\mathrm{D}^T\right)^{-1}\mathrm{S}. \end{aligned} \tag{13}$$

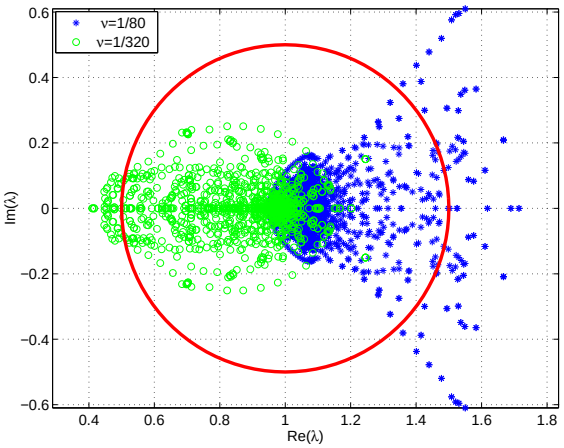

Fig. 1. Eigenvalues of the preconditioned matrix $\mathrm{P}_{new}^{-1}\Sigma$ for a square domain with $h = 1/32$, $\nu = 1/80$ and $\nu = 1/320$. We consider the Navier-Stokes problem with a semi-implicit treatment of the convective term (Test case: Kim and Moin solution at the first time step). The red line is the circle centred in (1,0) with radius 0.5.

In the Stokes problem, K is the discretization of the velocity Laplace operator. Now, suppose that:

$$\mathrm{K}\mathrm{H}\mathrm{D}^T = \nu\mathrm{D}^T\mathrm{M}_p^{-1}\mathrm{D}\mathrm{H}\mathrm{D}^T. \tag{14}$$

This assumption, in the context of Finite Difference space discretization, has been advocated in [7] as a *compatibility condition* holding for special set of velocity boundary conditions in rectangular domains. In the general case, it does not hold, so we exploit it as a further "degree of inexactness" in the set-up of the preconditioner. In this way, we obtain from (13):

$$\begin{aligned} \mathrm{S}\mathrm{B}^{-1}\mathrm{S} &= \mathrm{S}\left(\mathrm{S} + \nu\mathrm{D}\mathrm{H}\mathrm{D}^T\mathrm{M}_p^{-1}\mathrm{D}\mathrm{H}\mathrm{D}^T\right)^{-1}\mathrm{S} \\ &= \mathrm{S}\left(\mathrm{S} + \nu\mathrm{S}\mathrm{M}_p^{-1}\mathrm{S}\right)^{-1}\mathrm{S} = \left(\mathrm{S}^{-1} + \nu\mathrm{M}_p^{-1}\right)^{-1}. \end{aligned} \tag{15}$$

The matrix on the right hand side represents the algebraic version of the Cahouet-Chabard preconditioner, where S plays the role of the velocity Laplacian matrix. The Cahouet-Chabard preconditioner can be considered therefore an approximation of the present one. If the mass matrix M_p is lumped, the preconditioned residual computation of the Cahouet-Chabard preconditioner is actually cheaper than the one of P_{new}. However, the numerical results of next Section show that P_{new} in the case of the generalized Stokes problem requires a number of iterations significantly lower than the iterations required by the Cahouet-Chabard. Moreover, our preconditioner is naturally suited to account also for the non symmetric case, being based on a block factorization of the more general Navier-Stokes problem.

4 Numerical Results

We present numerical results[1] concerning the performances of P_{new}. We adopted a FEM discretization $\widetilde{\mathbb{P}^2}$-$\mathbb{P}^1$, where

[1] The 2D code has been implemented by the authors in Matlab. The 3D code has been implemented by the authors

$\widetilde{\mathbb{P}^2}$ elements are a modification of the usual $\mathbb{P}^2$ one which allows a correct velocity mass lumping (see [3]) coupled with BDF time discretization schemes up to the order 3.

In the 2D case, we solved the pressure matrix system preconditioned by P_{new} both with GMRes and BiCGStab schemes. We actually found similar results. In 3D we used only the GMRes method.

We also used both an implicit treatment of the convective term (Newton method) and a semi-implicit linearization. The linear systems were solved with a stopping criterion based on the residual (normalized with the right hand side) and a tolerance of 10^{-10} for 2D computations and 10^{-6} for 3D cases.

We considered the following test cases: (i) 2D Kim and Moin test case (see [6]) both for the Stokes and Navier-Stokes case; (ii) 2D Timmermans test case (see [17]) for the Navier-Stokes problem; (iii) 3D Lid driven cavity both for the Stokes and Navier-Stokes problems.

For what concerns the aims of the present work, the performances of the preconditioner were satisfactory in both the cases, and much better than the Cahouet Chabard preconditioner.

4.1 2D Kim and Moin test case

We start solving the *Stokes problem* on a unit square Ω with the following boundary conditions:

$$u_1 = -\cos(2\pi x)\sin(2\pi y)e^{-8\pi^2\nu t},$$
$$u_2 = \sin(2\pi x)\cos(2\pi y)e^{-8\pi^2\nu t}, \quad x, y \in \partial\Omega, \tag{16}$$

where we set $\nu = 0.01$ and the final time $T_f = 1$. The forcing term $\mathbf{f}$ is null. In Tab. 2 we compare the results of the Cahouet-Chabard preconditioner and the new one in terms of mean number of iterations (in the brackets the max number of iterations is reported) for different values of h and Δt. The new preconditioner performs better and seems to be less sensitive with respect to the mesh size.

When solving the *Navier-Stokes problem*, (16) for $x, y \in \Omega$ represents the exact solution for the velocity field, while the exact pressure reads (Kim and Moin solution [6]):

$$p = -\frac{1}{4}\left(\cos(4\pi x) + \cos(4\pi y)\right)e^{-16\pi^2\nu t}. \tag{17}$$

For this problem, we compare the performances of the two preconditioners in Tab. 3. As expected, the presence of the nonlinear convective term strongly affects the performances of the Cahouet-Chabard preconditioner, while P_{new} features good performances.

4.1.1 Eigenvalues of the preconditioned matrix

For this test case, we consider the behavior of the eigenvalues $\{\lambda\}$ of the preconditioned matrix $\mathrm{P}_{new}^{-1}\Sigma$ in the case of the 2D Navier-Stokes problem, for different values of the mesh size h and of the viscosity ν. More precisely, we build the matrices at the first time step, with a semi-implicit treatment of the non-linear term (for a first order of accuracy in time). We carried out the computation with the command `eig` in Matlab for different values of the mesh size h and the viscosity ν. The results are summarized in Tab. 4, where we report the extrema of the real part $\Re(\lambda)$ of the eigenvalues and the largest imaginary part $\Im(\lambda)$. Moreover, in the last column we indicate the number n of the eigenvalues featuring a distance from the point (1,0) lower than 0.5, over the total number of eigenvalues N. In Fig. 1 we illustrate the whole set of eigenvalues in the complex plane for $h = 1/32$ and two values of viscosity.

The table shows that the spectrum of the preconditioned matrix is clustered around the real value 1. It can be deduced from the third column that in any case the largest part of the eigenvalues is indeed at a distance lower than 0.5 from (1,0). In fact, we verified that in all cases considered in the table the eigenvalues were at a distance lower than 1 from (1,0) with the exception of the case $\nu = 1/40$ for $h = 1/32$. Another interesting point is that the $\max(\Im(\lambda))$ of the eigenvalues does

Table 2. Mean (Max) number of iterations required for the convergence with the Cahouet-Chabard and P_{new} for the generalized Stokes problem (Sect. 4.1).

	$\Delta t = 0.1$		
h	1/16	1/32	1/64
P_{CC}	8.7 (9)	10.5 (12)	11.2 (14)
P_{new}	5.2 (6)	7.9 (8)	8.1 (9)
	$\Delta t = 0.05$		
h	1/16	1/32	1/64
P_{CC}	7.15 (8)	9.6 (11)	11.7 (13)
P_{new}	4.5 (5)	6.15 (7)	7.25 (8)
	$\Delta t = 0.025$		
h	1/16	1/32	1/64
P_{CC}	6.085 (7)	8.175 (10)	11.475 (13)
P_{new}	3.7 (4)	5.175 (6)	6.775 (8)

Table 3. Mean (Max) number of iterations required for the convergence with the Cahouet-Chabard and the new preconditioner for the Navier-Stokes problem (Kim and Moin test case Sect. 4.1).

	$\Delta t = 0.1$		
h	1/16	1/32	1/64
P_{CC}	17.4 (20)	24.3 (30)	30.5 (36)
P_{new}	7.4 (9)	9 (11)	10.9 (13)
	$\Delta t = 0.05$		
h	1/16	1/32	1/64
P_{CC}	13.3 (18)	18.2 (22)	24.35 (32)
P_{new}	5.5 (7)	7.45 (9)	9.55 (11)
	$\Delta t = 0.025$		
h	1/16	1/32	1/64
P_{CC}	9.1 (12)	13.425 (18)	19.375 (23)
P_{new}	4.1 (5)	6.075 (7)	8.25 (10)

in C++ in the framework of the LifeV Project (joining MOX-PoliMi, INRIA-Paris and EPF Lausanne). The linear algebra package adopted in this framework is Aztec by R.S. Tuminaro, J.N. Shadid, M.Heroux, Sandia Nat. Lab.

Table 4. Real and imaginary parts of the eigenvalues of the preconditioned matrix for different values of h and ν. The last column denotes the number n of eigenvalues which are at a distance from (1,0) lower than 0.5 over the total number of eigenvalues N.

	$h = 1/8$			
ν	$\min(\Re(\lambda))$	$\max(\Re(\lambda))$	$\max(\Im(\lambda))$	n/N
1/40	0.9019	1.1569	0.1041	80/80
1/80	0.7310	1.0172	0.0789	80/80
1/160	0.5781	0.9963	0.0564	80/80
1/320	0.4693	0.9889	0.0565	79/80
1/640	0.4040	0.9852	0.0540	74/80
1/1280	0.3679	0.9835	0.0573	72/80
	$h = 1/16$			
1/40	0.9595	1.4924	0.3424	276/288
1/80	0.7690	1.2512	0.2903	288/288
1/160	0.5668	1.0854	0.1747	288/288
1/320	0.3524	1.0023	0.1323	244/288
1/640	0.2300	0.9931	0.1430	215/288
1/1280	0.1685	0.9895	0.1404	204/288
	$h = 1/32$			
1/40	0.9644	2.0411	0.5342	896/1088
1/80	0.9135	1.7143	0.6098	969/1088
1/160	0.6408	1.4456	0.4837	1088/1088
1/320	0.4100	1.2453	0.2510	1038/1088
1/640	0.2276	1.0794	0.1726	826/1088
1/1280	0.1078	1.0006	0.1596	728/1088

not increase when the viscosity decreases, as it happens for other preconditioners present in literature (see [5]). On the contrary, it decreases and it seems to assume a constant value for very small viscosities. This suggests that the preconditioner captures correctly the complex part of the spectrum induced by the convective term. Moreover, both the smallest and the greatest real parts of the eigenvalues shift to 0 when the viscosity decreases. However, since $\min(\Re(\lambda))$ decreases more rapidly than $\max(\Re(\lambda))$, the spectrum spreads over the real axis when the viscosity decreases (see Fig. 1), in the sense that $\max(\Re(\lambda)) - \min(\Re(\lambda))$ increases when ν decreases, even if this happens quite slowly (with respect to ν).

The conclusions that can be drawn from these computations, limited to quite coarse grids, is that the new preconditioner should require a number of iterations moderatley increasing when h decreases and almost constant with ν. This conjecture is confirmed by the numerical results of the present Section.

4.2 2D Timmermans test case

For the *Navier-Stokes equations*, we consider the test case proposed in [17], in which the right hand side is modified in order to ensure that the exact solution is:

$$u_1 = \sin(x + 5t)\sin(y + 5t),$$
$$u_2 = \cos(x + 5t)\cos(y + 5t),$$
$$p = \sin(x + y + 5t).$$

Let us consider firstly the results obtained with a semi-implicit BDF time discretization of order 1. In Fig. 2

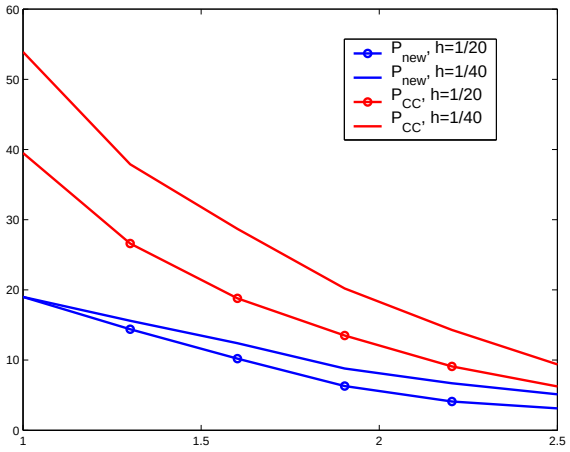

Fig. 2. Mean number of iterations required by the Cahouet-Chabard and the new preconditioners for different values of the mesh size h. On the abscissa $-\log_{10}(\Delta t)$. We used a time discretization scheme BDF of order 1 and set Re= 100.

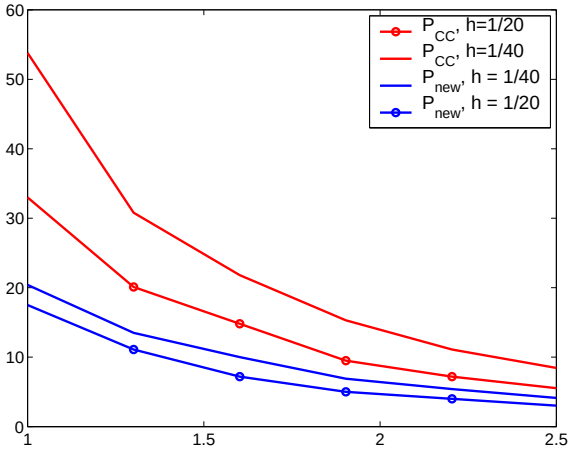

Fig. 3. Mean number of iterations required by the Cahouet-Chabard and the new preconditioners for different values of the mesh size h. On the abscissa $-\log_{10}(\Delta t)$. We used a time discretization scheme BDF of order 3 and set Re= 100

the mean number of iterations required by the Cahouet-Chabard and P_{new} for different values of the mesh size h are illustrated as a function of the time step. The efficiency of the new preconditioner is obviously better in comparison with the Cahouet-Chabard one. Similar results are obtained when we used a BDF time discretization scheme of order 3 (see Fig. 3).

We have also compared the effect of different values of the Reynolds number (Re) on the performances of the two preconditioners. The results are illustrated in Fig. 4. The new preconditioner is weakly affected by the Reynolds number, at least for a time step sufficiently small.

4.3 Lid Driven Cavity test case

We considered the domain $[0 \times 1]^3$, where we solved both the Stokes and the Navier-Stokes problems for the lid driven cavity test case.

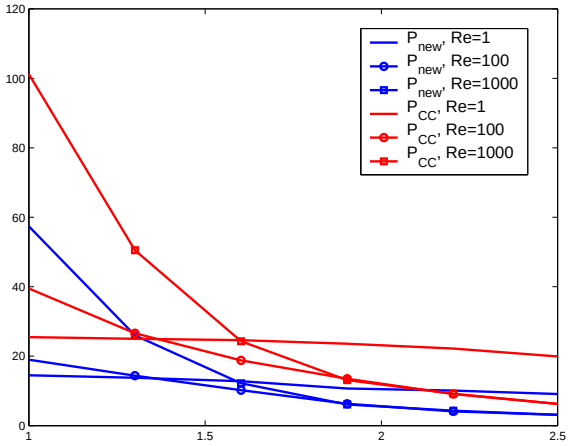

Fig. 4. Performances of the Cahouet-Chabard and the new preconditioners for different values of the Reynolds number. On the abscissa, $-\log_{10}(\Delta t)$ ($h = 1/20$).

Table 5. Mean (Max) number of iterations required for the convergence with the Cahouet-Chabard and P_{new} for the 3D Stokes problem.

	h			
	1/4	1/5	1/10	time step
P_{CC}	8 (21)	8.2 (21)	7.7 (22)	h
P_{CC}	7.1 (19)	6.9 (19)	4.2 (18)	h^2
P_{new}	5.4 (14)	5.8 (15)	7.1 (20)	h
P_{new}	4 (10)	4.4 (11)	2.6 (11)	h^2

In Tab. 5 we show the results obtained by the Cahouet-Chabard preconditioner and the new one for different values of the mesh size and the time step, in a 3D computation. We considered three meshes with 384 ($h = 1/4$), 750 ($h = 1/5$) and 6000 ($h = 1/10$) volumes respectively and solved the Stokes problem, considering a time step equal to h and h^2 respectively. Also in this case, P_{new} performs clearly better than the Cahouet-Chabard preconditioner.

For the Navier-Stokes problem, we considered both the Cahouet-Chabard and P_{new} for different values of h and of the Reynolds number. In Fig. 5 we illustrate the velocity field for $h = 1/5$ and Re=40.

Fig. 6 and 7 confirm that P_{new} performs better than the Cahouet-Chabard preconditioner also for 3D computations. In particular, Fig. 6 shows that both the preconditioners are robust with respect to the Reynolds number. More specifically, for large values of the time step the number of iterations required in the case $Re = 160$ is, in fact, lower than for $Re = 40$.

Fig. 7 confirms that P_{new} is robust also with respect to the mesh size.

4.4 "Flexible" strategy

In Sect. 3 we pointed out that systems (12) can be solved effectively through a QR factorization. Another cheap strategy relies on the use of iterative methods but in a "flexible" sense (see [14]). This means that we solve the

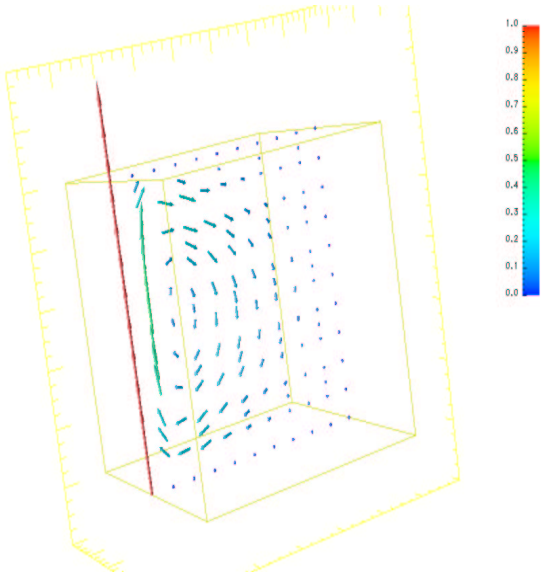

Fig. 5. Velocity field for the lid driven cavity, Re=40, $h = 0.2$.

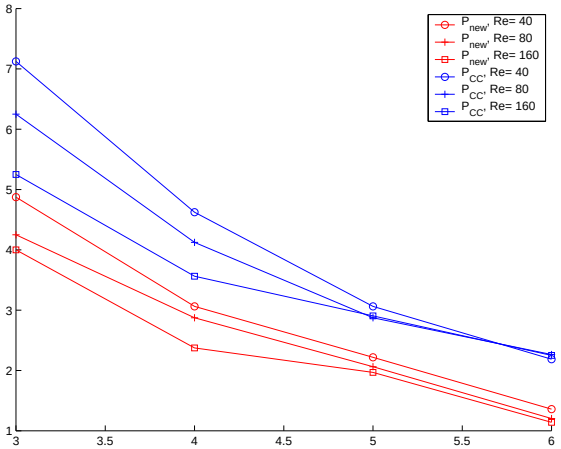

Fig. 6. Mean number of iterations required by the Cahouet-Chabard preconditioner and by P_{new} for $h = 0.2$ and different values of Re. On the abscissa we report $-\log_2(\Delta t)$.

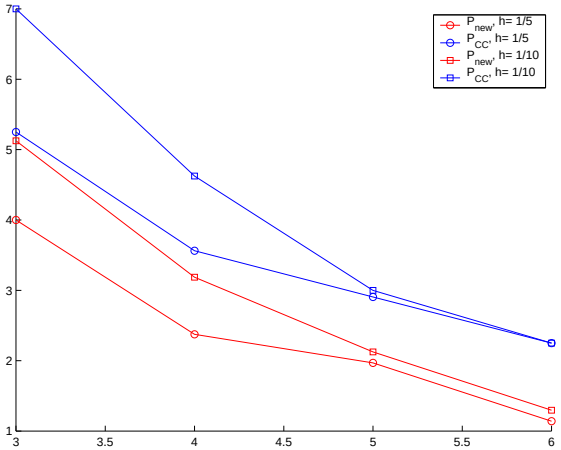

Fig. 7. Mean number of iterations required by the Cahouet-Chabard preconditioner and by P_{new} for $Re = 160$ and different values of h. On the abscissa we report $-\log_2(\Delta t)$.

Table 6. Mean number of iterations required for the convergence with the Cahouet-Chabard and P_{new} when the tolerance of the inner loops is increased.

	h			
	1/8	1/16	1/32	inner toll
P_{CC}	17.4	20.7	22.3	1.e(-14)
P_{CC}	17.8	20.7	22.3	1.e(-7)
P_{CC}	17.8	20.7	22.7	1.e(-5)
P_{CC}	20.7	24.6	83.9	1.e(-3)
P_{new}	8.3	11	14.2	1.e(-14)
P_{new}	8.3	11	14.7	1.e(-7)
P_{new}	8.3	11	15	1.e(-5)
P_{new}	9.4	12.6	37.9	1.e(-3)

preconditioned system iteratively, but with a low accuracy and, in particular, with a tolerance for the preconditioner systems (inner loops) greater than the tolerance for the pressure matrix systems (outer loops). A robust preconditioner is such that the inaccurate solution of the preconditioner systems does not affect the convergence. In this case, the CPU time is strongly reduced.

In Tab. 6, we compare the number of iterations required for the convergence, when the tolerance for the solution of the preconditioner system is increased, both for the Cahouet-Chabard and the new preconditioner. The results refer to the solution of the Timmermans test case, with a semi-implicit Euler time discretization and Re= 1. We have fixed $\Delta t = 0.1$ and used different values of h.

Each entry of the table corresponds to the mean number of iterations required for the convergence of the Navier-Stokes system. Both the preconditioners are robust with respect to the accuracy of the preconditioner system. Actually, a modification of the tolerance from 10^{-14} to 10^{-7} and to 10^{-5} does not modify significantly the number of iterations required for the convergence for every value of h. The number of external iterations required by both the preconditioners is really affected for h small when the tolerance is furtherly increased (10^{-3}).

Conclusions

The preconditioner (11) presented here provides promising results for the effective solution of the unsteady Navier-Stokes problem. QR factorization or a flexible strategy, on the other hand, ensure the computational effectiveness of P_{new}. A thorough convergence analysis, with particular emphasis on the dependence of the performances on the Reynolds number needs therefore to be carried out.

References

1. J. Bramble and J. E. Pasciack, *Iterative techniques for time dependent stokes problems*, Math. Comp., (1997).
2. J. Cahouet and J.-P. Chabard, *Some fast 3d finite element solvers for the generalized Stokes problem*, Int. Jou. Num. Meth. Fluids, 8 (1988), pp. 869–895.
3. G. Cohen, P. Joly, J. Roberts, and N. Tordjman, *Higher order triangular finite elements with mass lumping for the wave equation*, SIAM J. Numer. Anal., 38 (2001), pp. 2047–2078.
4. H. C. Elman, *Preconditioners for saddle point problems arising in computational fluid dynamics*, Tech. Report UMCP-CSD:CS-TR4311, Univ. Maryland, College Park, 2001.
5. H. C. Elman, D. J. Silvester, and A. J. Wathen, *Performance and analysis of saddle poin preconditioners for the discrete steady-state Navier-stokes equations.*, Tech. Report UMCP-CSD:CS-TR4164, Univ. Maryland, College Park, 2000.
6. J. Kim and P. Moin, *Application of a fractional step method to incompressible Navier-Stokes equations*, J. Comp. Phys., 59 (1985), pp. 308–323.
7. G. M. Kobelkow and M. Olshanskii, *Effective preconditioning of uzawa type schemes for a generalized stokes problem*, Num. Mat., 86 (2000), pp. 443–470.
8. P. Matstoms, *Sparse QR Factorization with Applications to Linear Least Squares Problems*, PhD thesis, Dept. of Mathematics, Linkoping University, Sweden, 1994.
9. B. Perot, *An analysis of the fractional step method*, J. Comp. Phys., 108 (1993), pp. 51–58.
10. A. Quarteroni, R. Sacco, and F. Saleri, *Numerical Mathematics*, Springer-Verlag, NY, 2000.
11. A. Quarteroni, F. Saleri, and A. Veneziani, *Analysis of the Yosida method for the incompressible Navier-Stokes equations*, J. Math. Pures Appl., 78 (1999), pp. 473–503.
12. ———, *Factorization methods for the numerical approximation of Navier-Stokes equations*, Comput. Methods Appl. Mech. Engrg., 188 (2000), pp. 505–526.
13. A. Quarteroni and A. Valli, *Numerical approximation of partial differential equations*, no. 23 in Springer Series in Computational Mathematics, Springer-Verlag, Berlin, 1994.
14. Y. Saad, *Iterative Methods for Sparse Linear Systems*, PWS Publishing Company, Boston, 1996.
15. F. Saleri and A. Veneziani, *Pressure-correction methods for the incompressible Navier-Stokes equations.* in preparation.
16. D. Silvester, H. Elman, D. Kay, and A. Wathen, *Efficient preconditioning of the linearized Navier-stokes equations for incompressible flow*, J. Comp. Appl. Math., 128 (2001), pp. 261–279.
17. L. Timmermans, P. Minev, and F. V. de Vosse, *An approximate projection scheme for incompressible flow using spectral methods*, Int. Jou. Num. Meth. Fluids, 22 (1996), pp. 673–688.
18. S. Turek, *Efficient Solvers for Incompressible Flow Problems*, Springer-Verlag, Berlin, 1999.
19. A. Veneziani, *Block factorized solvers and preconditioners for the incompressible Navier-Stokes equations.* submitted to Num. Meth. Part. Diff. Eqns.