Stabilized bi-grid projection methods in Finite Elements for the 2D incompressible Navier-Stokes

August 29, 2018

HYAM ABBOUD$^1$, CLARA AL KOSSEIFI$^{2,3}$ AND JEAN-PAUL CHEHAB$^2$

$^1$Département de mathématiques, Faculté des Sciences II, Université Libanaise, Fanar, Liban
$^2$Laboratoire Amiénois de Mathématiques Fondamentales et Appliquées (LAMFA), UMR CNRS 7352
Université de Picardie Jules Verne, 33 rue Saint Leu, 80039 Amiens France
$^3$Laboratoire de Physique Appliquée (LPA), Faculté des Sciences II, Université Libanaise, Fanar, Liban

Abstract

We introduce a family of bi-grid schemes in finite elements for solving 2D incompressible Navier-Stokes equations in velocity and pressure ($u, p$). The new schemes are based on projection methods and use two pairs of FEM spaces, a sparse and a fine one. The main computational effort is done on the coarsest velocity space with an implicit and unconditionally time scheme while its correction on the finer velocity space is realized with a simple stabilized semi-implicit scheme whose the lack of stability is compensated by a high mode stabilization procedure; the pressure is updated using the free divergence property. The new schemes are tested on the lid driven cavity up to $Re = 7500$. An enhanced stability is observed as respect to classical semi-implicit methods and an important gain of CPU time is obtained as compared to implicit projection schemes.

Keywords: Navier-Stokes equation, bi-grid method, stabilization, Chorin-Temam projection, separation of the scales

AMS Classification[2010]: 35N57, 65L07, 65M60, 65N55

1 Introduction

Multigrid methods have been widely developed since more than 40 years and were proposed as fast solvers to the numerical solution of elliptic problems, see e.g. [26] for steady linear and nonlinear Dirichlet Problems in finite differences, but also for Steady Navier-Stokes Equations \cite{8} \cite{19}.

The method is first defined for two levels of discretization (bi-grid case). The two key ingredients are both the separation of the high and the low mode components of the error provided by the use of coarse and fine grids (or spaces) $V_H$ and $V_h$ respectively, and also the concentration of the main computational effort on the coarser (lower dimensional) subspace $V_H \subset V_h$;
high modes can be represented in $V_h$ while only low modes can be captured in $V_H$. This leads to a drastic save in CPU computation time. The correction to the fine space components belonging to $V_h$ is usually realized by a simple and fast numerical scheme and is associated to a high mode smoothing. Then the scheme is recursively applied on a set of nested grids (or spaces).

When considering nonlinear dissipative equations, we can distinguish two approaches for bi-grid methods:

In the first one, the low and the high mode components are explicitly handled: thanks to the parabolic regularization, it is expected that the low mode components which carry the main part of the energy of the (regular) solution have a different dynamics from the high mode components, that can be seen as a correction, see [15] and the references therein. Also, a way to speed up the numerical integration is then to apply different schemes to these two sets of components, concentrating the effort on the computation of the low modes, that belong in $V_H$, see [13]. When dealing with spectral methods, the separation in frequency is natural but it is not the case when Finite Differences or FEM methods are used for the spatial discretization. To separate the modes, a hierarchical basis approach is used [43]: using a proper interpolation (or projection) operator between $V_H$ and $V_h$, one builds a transfer operator which defines a pre-conditioner for the stiffness matrices but it allows also to express the solution in terms of main part, associated to the low mode components, and of a fluctuant part, of lower magnitude, and associated to high modes components; we refer the reader to [40] [41] [42] and the references therein for Finite Elements discretizations, [12] [34] for Finite Differences and [6] [17] in Finite Volumes. These schemes showed to be efficient, however they necessitate to build and manipulate hierarchical bases.

In the second approach, the mode separation is not used explicitly and the methods consist, at each time step, in first computing the coarse approximation $u_H$ to the fine solution $u_h$ by an unconditionally implicit stable scheme and then to update the fine space approximation by using a linearized scheme at an extrapolated value $\tilde{u}_h$ of $u_H$ in $V_h$. These schemes allow to reduce the computational time with an optimal error as respected to the classical scheme when choosing accurately the mesh size of $V_h$ as respect to the one of $V_H$. It is to be underlined that $\tilde{u}_h$ represents the mean part of the solution while $z_h = u_h - \tilde{u}_h \in V_h$ represents the fluctuant part (carrying the high mode components of the solution $u_h \in V_h$) but is not simulated in the schemes. Such methods were developed and applied to the solution of time dependent incompressible Navier-Stokes equations [2], [3], [20] and [27]. Since a linearization is used on $V_h$ the matrix to solve at each time iteration changes and it can be costly in transient regime.

The numerical methods we propose here are somehow in between and are inspired from the approach developed in [1] for Allen-Cahn Equation: as previously described, the use of two levels of discretization allows to concentrate the effort on the coarsest, yet lower dimensional, space and at the same time to decompose the solution into its mean and fluctuant part $u_h = \tilde{u}_h + z_h$. The $z_h$ are not explicitly simulated, however, they are used implicitly for a high mode stabilization as follows: consider the time integrations of the reaction-diffusion equation, in its variational
form:
\[
\frac{d}{dt}(u,v) + (\nabla u, \nabla v) + (f(u), v) = 0, \forall v \in V, \\
v(0) = v_0,
\]
where \(V\) is a proper Hilbert space.

Given two finite dimensional subspaces of \(V\), \(W_H\) and \(V_h\), with \(\text{dim}(W_H) \ll \text{dim}(V_h)\), we define the bigrid-scheme as

**Algorithm 1 Bi-grid Stabilized scheme for Reaction-Diffusion**

1: \(u_0^h, u_0^H\) given
2:
3: \textbf{for} \(k = 0, 1, \cdots \) \textbf{do}
4: \textbf{Step 1 (Coarse Space Implicit Scheme)}
5: \[
\left( \frac{u_{H}^{k+1} - u_{H}^{k}}{\Delta t}, \psi_{H} \right) + (\nabla u_{H}^{k+1}, \nabla \psi_{H}) + (f(u_{H}^{k+1}), \psi_{H}) = 0, \forall \psi_{H} \in W_{H}
\]
6: \textbf{Step 2 (Fine Space semi-implicit Scheme)}
7: \[
\left( \frac{u_{h}^{k+1} - u_{h}^{k}}{\Delta t}, \phi_{h} \right) + \tau(u_{h}^{k+1} - u_{h}^{k}, \phi_{h}) + (\nabla u_{h}^{k+1}, \nabla \phi_{h}) + (f(u_{h}^{k}), \phi_{h}) =
\tau(u_{H}^{k+1} - u_{H}^{k}, \phi_{h}), \forall \phi_{h} \in V_{h}
\]
8: \textbf{end for}

It is not necessary to have \(W_H \subset V_h\), an inf-sup like compatibility condition has to be satisfied for defining uniquely the prolongation \(\tilde{u}\), \([1]\). The competition between the terms \(\tau(u_{h}^{k+1} - u_{h}^{k}, \phi_{h})\) and \(\tau(u_{H}^{k+1} - u_{H}^{k}, \phi_{h})\) are interpreted as a high mode filter and \(\tau > 0\) is the stabilization parameter \([1]\) and allow to compensate the explicit treatment of the nonlinear term. Furthermore, this stabilization has few influence on the global dynamics, \([1]\).

The aim of the present work is to adapt this two-grid scheme to Navier-Stokes equations. To this end, we consider a projection method that splits the time resolution into two steps: firstly a parabolic equation on the velocity and secondly an elliptic equation on the pressure. The bi-grid stabilization will be applied to the first step.

The paper is organized as follows: in Section 2 we first present briefly the principle of the bi-grid framework (reference scheme, separation of the modes, high mode stabilization), then, after recalling the definition and some properties of the projection scheme (reference scheme), we propose different bi-grid projection schemes. Section 3 is dedicated to the numerical results. We consider the classical benchmark lid-driven cavity problem. When computing the steady states, the results we obtain agree with those of the literature, also an important gain of CPU time is obtained as respect to classical methods, for a comparable precision. The article ends in Section 4 with concluding remarks. All the computation have been realized using FreeFem++ \([18]\).
2 Derivation of the Bi-grid Projection Schemes

We here build the bi-grid high mode stabilized projection schemes. For that purpose, we first recall the different approaches of the bi-grid schemes in finite elements and then describe the stabilization procedure, for reaction diffusion problems. Then, we present briefly the projection schemes in finite elements we will start from to introduce the new Bi-grid Projection Schemes.

2.1 Principle of the bi-grid approach

As stated in the introduction, when considering nonlinear dissipative equations, a well known way to obtain a gain in CPU time is to consider two levels of discretization and to concentrate the main computational effort on the lower dimensional approximation space $W_H$, while the higher accurate approximation to the solution on the fine space $V_h$ is updated by using a simple semi-implicit (yet linear) scheme. The general pattern of a bi-grid scheme for the reaction diffusion equation (1) writes as

Algorithm 2 Bi-grid Scheme for Reaction-Diffusion

1: $u^0_h, u^0_H$ given

2: for $k = 0, 1, \cdots$ do

3: \textbf{Step 1 (Coarse Space Implicit Scheme)}

4: $(\frac{u^{k+1}_H - u^k_H}{\Delta t}, \psi_H) + (\nabla u^{k+1}_H, \nabla \psi_H) + (f(u^{k+1}_H), \psi_H) = 0$, $\forall \psi_H \in W_H$

5: \textbf{Step 2 (Fine Space semi-implicit Scheme)}

6: $(\frac{u^{k+1}_h - u^k_h}{\Delta t}, \phi_h) + (\nabla u^{k+1}_h, \nabla \phi_h) + (f(u^k_h), \phi_h) + (f'(u^k_H)u^{k+1}_h, \phi_h) = 0, \forall \phi_h \in V_h$

8: end for

This approach was proposed by [20, 27, 29]. However, $f'(u^{k+1}_H)$, the linearized of $f$ at $u^{k+1}_H$ must be computed at each time step, changing the matrix to solve at each iteration. A way to avoid this drawback is to consider a classical semi-implicit scheme and to compensate the lack of stability by adding, e.g., a first order damping term $\tau (u^{k+1}_h - u^k_h)$, obtaining as second step:

1: \textbf{Step 2 (Stabilized Fine Space semi-implicit Scheme)}

2: $(\frac{u^{k+1}_h - u^k_h}{\Delta t}, \phi_h) + \tau (u^{k+1}_h - u^k_h, \phi_h) + (\nabla u^{k+1}_h, \nabla \phi_h) + (f(u^k_h), \phi_h) = 0, \forall \phi_h \in V_h$

This stabilization procedure allows to obtain unconditionally stable time scheme for large values of $\tau > 0$, that can be tuned. The resulting scheme is fast, however, it can slow down drastically the dynamics, particularly convergence to steady states occurs in longer times. This is due to the fact that the damping acts on all the mode components, including the low ones which are associated to the mean part of the function and carry the main part of the $L^2$ energy, when considering a Fourier-like interpretation. A way to overcome this drawback is to apply the stabilization to the only high mode components $z_h$ of $u_h$ which correspond to a fluctuent part of $u_h$; the stability of a scheme is indeed related to its capability to contain the propagation of...
the high frequencies. Hence the scheme becomes

1. **Step 2 (High Mode Stabilized Fine Space semi-implicit Scheme)**

\[
\frac{u_h^{k+1} - u_h^k}{\Delta t} + \tau (P(u_h^{k+1} - u_h^k), \phi_h) + (\nabla u_h^{k+1}, \nabla \phi_h) + (f(u_h^k), \phi_h) = 0, \forall \phi_h \in V_h
\]

where \(P(u_h^{k+1} - u_h^k)\) capture the high mode components of \(u_h^{k+1} - u_h^k\).

At this point we can distinguish two main ways to decompose \(u_h\) as a sum of its mean part \(\bar{u}_h\) and its fluctuent part \(z_h\) in finite elements:

- **Hierarchical basis in Finite Elements:** the fine space FEM approximation space \(V_h\) is decomposed as \(V_h = V_H \bigoplus W_h \subset H^1\), where \(V_H \subset V_h\) is the coarse FEM space that can capture only low modes, and \(W_h\) is the complementary space, generated by the basis functions of \(V_h\) that do not belong in \(V_H\), and that capture high mode components. When using \(P_1\) finite elements, the components of \(W_h\) of a function \(u_h \in V_h\) are built as proper local interpolation error from \(u_H \in V_H\). The functions of approximation \(V_h\) can be uniquely decomposed as

\[
u_h = u_H + z_h,
\]

with \(u_H \in V_H\) and \(z_h \in W_h\). It has been showed that the linear change of variable \(S : u_h \rightarrow (u_H, z_h)\) provides an efficient preconditioner for the stiffness matrices but also proceeds to a scale separation \([9, 10, 30, 31, 43]\).

The spatial discretization reads then to a differential system satisfied by \(u_H\) and \(z_h\), namely

\[
\frac{d}{dt}(u_H + z_h, \phi_H) + (\nabla (u_H + z_h), \nabla \phi_H) + (f(u_H + z_h), \phi_H) = 0, \forall \phi_H \in V_H, \quad (3)
\]

\[
\frac{d}{dt}(u_H + z_h, \psi_h) + (\nabla (u_H + z_h), \nabla \psi_h) + (f(u_H + z_h), \psi_h) = 0, \forall \psi_h \in V_h. \quad (4)
\]

New time marching methods are obtained applying two different schemes for \(u_h\) and for \(z_h\): particularly, the separation of the scales allows to use a simple and fast scheme for the \(z_h\) components and to save computational time, with a good accuracy. Approximations and simplifications can be applied to each equation, particularly using an approached law linking high and low mode components of type \(\Phi(u_H) = z_h\), \([30, 31]\).

The approach followed in \([32]\) is technically different but is based on the same principle: Denoting by \(P_H\) the \(L^2\) orthogonal projection from \(V_h\) on \(V_H\), the decomposition \(u_h = u_H + w_h\) with \(u_H = P_H u_h\) and \(w_h = (I - P_H) u_h\) is used, making \(u_H\) and \(w_h\) orthogonal. They proposed and analyzed schemes of Nonlinear Galerkin type, says in which an asymptotic approach law \(\Phi(u_H) = w_h\) is implemented together with a time marching scheme on \(u_H\). A drawback is that one must build a basis for \((I - P_H) V_h\).

Finally let us mention similar developments in finite differences \([11, 12]\) and spectral methods \([13, 15, 16, 28]\).
• A $L^2$-like filtering, used in [I] and to which we will concentrate: we define first the pro-
longation operator $P : V_H \rightarrow V_h$ by
\[
(u_H - P(u_H), \phi_h) = 0, \forall \phi_h \in V_h.
\] (5)

Then, setting $\tilde{u}_h = P(u_H)$, we write
\[
u_h = \tilde{u}_h + (u_h - \tilde{u}_h) = \tilde{u}_h + z_h.
\]

We now apply the high mode stabilization to the $z_h$ components and get, after usual simplifica-
tions

\begin{algorithm}
\begin{enumerate}
  \item \textbf{Step 2 (High Mode Stabilized Fine Space semi-implicit Scheme)}
  \item $\left( \frac{u_h^{k+1} - u_h^k}{\Delta t}, \phi_h \right) + \tau (z_h^{k+1} - z_h^k, \phi_h) + (\nabla u_h^{k+1}, \nabla \phi_h) + (f(u_h^k), \phi_h) = 0, \forall \phi_h \in V_h$
\end{enumerate}
\end{algorithm}

and using the identity
\[
(z_h^{k+1} - z_h^k, \phi_h) = (u_h^{k+1} - u_h^k, \phi_h) - (\tilde{u}_h^{k+1} - \tilde{u}_h^k, \phi_h) = (u_h^{k+1} - u_h^k, \phi_h) - (u_H^{k+1} - u_H^k, \phi_h), \forall \phi_h \in V_h
\]
we can write the high mode stabilized bi-grid scheme as

\begin{algorithm}
\begin{enumerate}
  \item $u_0^h, u_0^H$ given
  \item \textbf{for} $k = 0, 1, \cdots$ \textbf{do}
  \item \textbf{Step 1 (Coarse Space Implicit Scheme)}
  \item $\left( \frac{u_H^{k+1} - u_H^k}{\Delta t}, \psi_H \right) + (\nabla u_H^{k+1}, \nabla \psi_H) + (f(u_H^k), \psi_H) = 0, \forall \psi_H \in W_H$
  \item \textbf{Step 2 (Fine Space semi-implicit Scheme)}
  \item $\left( \frac{u_h^{k+1} - u_h^k}{\Delta t}, \phi_h \right) + \tau (u_h^{k+1} - u_h^k, \phi_h) + (\nabla u_h^{k+1}, \nabla \phi_h) + (f(u_h^k), \phi_h) = \tau (u_H^{k+1} - u_H^k, \phi_h), \forall \phi_h \in V_h$
\end{enumerate}
\end{algorithm}

We now consider incompressible Navier-Stokes equations. FEM Bi-grid methods for incompressible NSE as proposed, e.g., in [2,3,20,27,29], are built using the pattern of scheme 2 and written as following. Given two pairs of FEM spaces $(X_H, Y_H)$ and $(X_h, Y_h)$ satisfying the discrete inf-sup condition, with $X_H \subset X_h$ and $Y_H \subset Y_h$, one defines the bi-grid iterations as
Algorithm 4 Bi-grid Scheme for Navier-Stokes Equation

1: \( u_0^h, u_0^H \) given
2:
3: for \( k = 0, 1, \ldots \) do
4:   \textbf{Step 1 (Coarse Space Implicit Scheme)}
5:   \( \frac{(u_{k+1}^H - u_k^H, \psi_H)}{\Delta t} + \frac{1}{Re} (\nabla u_{k+1}^H, \nabla \psi_H) + ((u_{k+1}^H, \nabla) u_{k+1}^H, \psi_H) - (div(\psi_H), p_{k+1}^H) = (f, \psi_H), \forall \psi_H \in X_H \)
6:   \( (div(u_{k+1}^H), q_H) = 0, \forall q_H \in Y_H \)
7:   \textbf{Step 2 (Fine Space semi-implicit Scheme)}
8:   \( \frac{(u_{k+1}^h - u_k^h, \phi_h)}{\Delta t} + \frac{1}{Re} (\nabla u_{k+1}^h, \nabla \phi_h) + (u_{k+1}^h \cdot \nabla u_{k+1}^h, \phi_h) - (div(\phi_h), p_{k+1}^h) = (f, \phi_h), \forall \phi_h \in X_h \)
9:   \( (div(u_{k+1}^h), q_h) = 0, \forall q_h \in Y_h \)
10: end for

We have used here the simple linearization of the nonlinear term \((u_{k+1}^H, \nabla u_{k+1}^h, \phi_h)\) proposed in [20], but other linearisations can be considered. At each step the method needs to solve first a mixed nonlinear FEM problem on the coarse pair FEM spaces \((X_H, Y_H)\), then a linear mixed FEM problem on the fine FEM spaces \((X_h, Y_h)\); in this last step, the underlying matrix changes at each iteration. To avoid this drawback and to apply the stabilized bi-grid method, as described in Algorithm 3, we will use a projection method. In such a case the computation of the velocity is decoupled from the one of the pressure, the intermediary velocity satisfies a nonlinear convection-diffusion equation. The main idea is then to apply a stabilized bi-grid scheme to this equation to speed up the resolution. We present below several options of this approach.

2.2 Bi-grid Projection Schemes in Finite Elements

First of all, let us recall the framework of the Projection Schemes in Finite elements and then derive the new bi-grid stabilized methods.

The mixed variational formulation of the motion of a viscous and incompressible fluid in a domain \( \Omega \) is described by the unsteady Navier-Stokes equation

\[
\begin{align*}
\left\{ \begin{array}{l}
\left( \frac{\partial u}{\partial t}, v \right) + (u \cdot \nabla u, v) - \nu(\nabla u, \nabla v) - (p, \text{div} \ v) = (f, v), \ \forall v \in X = (H_0^1(\Omega))^d \\
\text{div} \ u = 0, \ \forall q \in Y = L^2(\Omega)
\end{array} \right.
\end{align*}
\]

where \( \Omega \) is a domain of \( \mathbb{R}^d, d = 2, 3 \) of lipschizian bound \( \partial \Omega, \nabla \) the normal outside unit and a time interval \([0, T], T > 0\).

Here is a choice of the approximation spaces \( X_h \) and \( Y_h \) for the velocity and the pressure

\[
X_h = \{ v \in C^0(\Omega)^2 | v|_\kappa \in \mathbb{P}_2, \ \forall \kappa \in T_h, \ v = 0 \text{ on } \partial \Omega \},
\]

(6)
\[ Y_h = \{ q \in C^0(\Omega) | q|_\kappa \in \mathbb{P}_1, \forall \kappa \in T_h \} \cap L^2_0(\Omega). \]  

(7)

The degrees of freedom for the velocity are the vertices of the triangulation and the midpoints of the edges of the triangles of the triangulation \( T_h \). The degrees of freedom for the pressure are the vertices of \( T_h \) assumed to be uniformly regular. It is well known that because of the constraint of incompressibility, the choice of \( X_h \) and \( Y_h \) is not arbitrary. They must satisfy a suitable compatibility condition, the "inf-sup" condition of Babuška-Brezzi (cf. [4, 7]):

\[
\inf_{q \in Y_h} \sup_{u \in X_h} \frac{\int_\Omega q(\text{div} u) \, dx}{\|q\|_{L^2(\Omega)} \|\nabla u\|_{L^2(\Omega)}} \geq \beta^*,
\]

(8)

where \( \beta^* \) is independent of \( h \). All the results presented in this paper are done with the Taylor-Hood finite element \( \mathbb{P}_2/\mathbb{P}_1 \).

Let us consider the semi-discretization in time and focus on marching schemes. Let \( u^k \simeq u(x, k\Delta t) \) be a sequence of functions; \( \Delta t \) is the time step. We compare our method to the projection method applied on the fine grid. It is a fractional step-by-step method of decoupling the computation of the velocity from that of the pressure, by first solving a convection-diffusion problem such that the resulting velocity is not necessarily zero divergence; then in a second step, the latter is projected onto a space of functions with zero divergence in order to satisfy the incompressibility condition (cf. [14, 23, 24, 25, 35, 37, 38]). We start by presenting the implicit reference scheme [36]:

**Algorithm 5 Reference Scheme**

1: \textbf{for} \( k = 0, 1, \cdots \) \textbf{do}
2: \textbf{Find} \( u^*_h \) in \( X_h \) \( \left( \frac{u^*_h - u^k_h}{\Delta t}, \psi_h \right) + \nu(\nabla u^*_h, \nabla \psi_h) + ((u^*_h \cdot \nabla) u^*_h, \psi_h) = (f, \psi_h), \forall \psi_h \in X_h \)
3: \textbf{Find} \( p^{k+1}_h \) in \( Y_h \) \( (\text{div} \nabla p^{k+1}_h, \chi_h) = \left( \frac{\text{div} u^*_h}{\Delta t}, \chi_h \right), \forall \chi_h \in Y_h \)
4: \textbf{Find} \( u^{k+1}_h \) in \( X_h \) \( (u^{k+1}_h - u^*_h + \Delta t \nabla p^{k+1}_h, \psi_h) = 0, \forall \psi_h \in X_h \)
5: \textbf{end for}

This will be our reference scheme, used on the coarse FEM Space. It enjoys of unconditional stability properties, see [36], chapter III, section 7.3.

**Remark 2.1** We could use also an incremental projection method introduced by Goda [21], see also [23]. Goda has proven that adding a previous value of the gradient of the pressure \( (\nabla p^k) \) in the first step of the projection method then rectify the value of the velocity in the second step will improve the accuracy, in other words, he proposed the following algorithm:
Algorithm 6 Incremental Reference

1: for $k = 0, 1, \cdots$ do
2:   Find $u_h^*$ in $X_h$
3:   $(u_h^* - u_h^k, \psi_h) + \nu(\nabla u_h^*, \nabla \psi_h) + ((u_h^* \cdot \nabla)u_h^*, \psi_h) + (\nabla p_h^k, \psi_h) = (f, \psi_h), \forall \psi_h \in X_h$
4:   Find $p_h^{k+1}$ in $Y_h$ $\alpha(div \nabla p_h^{k+1} - div \nabla p_h^k, \chi_h) = (\frac{div u_h^*, \chi_h}{\Delta t}), \forall \chi_h \in Y_h$
5:   Find $u_h^{k+1}$ in $X_h$ $(u_h^{k+1} - u_h^* + \Delta t(\nabla p_h^{k+1} - \nabla p_h^k), \psi_h) = 0, \forall \psi_h \in X_h$
6: end for

The results obtained are similar to the one produced by the non incremental scheme to which we focus for a sake of simplicity.

We can now derive the bi-grid schemes.

Our first approach consists in separating in scales the intermediate velocity $u_h^*$ which we introduce according to the projection method of Chorin-Temam [14] [35] [37] [38]. First we compute $u_h^*$ on the coarse space $X_H$ and then stabilize the high frequencies of $u_h^*$ on the fine space $X_h$. Based on the free divergence condition, we find the pressure $p_h$ whose mean is equal to zero. We end up finding $u_h$ and restricting it to the coarse grid to get $u_H^*$ for the next iteration.

Algorithm 7 Two grids Algo1

1: for $k = 0, 1, \cdots$ do
2:   Find $u_H^*$ in $X_H$ $(\frac{u_H^* - u_H^k}{\Delta t}, \psi_H) + \nu(\nabla u_H^*, \nabla \psi_H) + ((u_H^* \cdot \nabla)u_H^*, \psi_H) = (f, \psi_H), \forall \psi_H \in X_H$
3:   Find $u_h^*$ in $X_h$
4:   $(1+\tau\Delta t)(u_h^* - u_h^k, \phi_h) + \nu(\nabla u_h^*, \nabla \phi_h) + ((u_h^k \cdot \nabla)u_h^k, \phi_h) = \tau(u_H^* - u_H^k, \phi_h) + (f, \phi_h), \forall \phi_h \in X_h$
5:   Find $p_h^{k+1}$ in $Y_h$ $\alpha(div \nabla p_h^{k+1} - div \nabla p_h^k, \chi_h) = (\frac{div u_h^*, \chi_h}{\Delta t}), \forall \chi_h \in Y_h$
6:   Find $u_h^{k+1}$ in $X_h$ $(u_h^{k+1} - u_h^* + \Delta t(\nabla p_h^{k+1} - \nabla p_h^k), \psi_h) = 0, \forall \psi_h \in X_h$
7: Solve in $X_H$ $(u_H^{k+1} - u_H^k, \psi_H) = 0, \forall \psi_H \in X_H$
8: end for

We propose in the second algorithm to replace $(u_h^k \cdot \nabla)u_h^k$ by $(u_H^* \cdot \nabla)u_h^*$. We obtain:
Algorithm 8 Variant of the Two grids Algo1 : Two grids Algo2
\begin{enumerate}
\item for $k = 0, 1, \cdots$ do
\item Find $u^*_H$ in $X_H \left( \frac{u^*_H - u^k_H}{\Delta t}, \psi_H \right) + \nu (\nabla u^*_H, \nabla \psi_H) + (u^*_H \cdot \nabla) u^*_H, \psi_H) = (f, \psi_H), \forall \psi_H \in X_H$
\item Find $u^*_h$ in $X_h$
\begin{equation}
(1 + \tau \Delta t) \left( \frac{u^*_h - u^k_h}{\Delta t}, \phi_h \right) + \nu (\nabla u^*_h, \nabla \phi_h) + (u^*_h \cdot \nabla) u^*_h, \phi_h) = \tau (u^*_H - u^k_H, \phi_h) + (f, \phi_h), \forall \phi_h \in X_h
\end{equation}
\item Find $p^k_{H+1}$ in $Y_h \left( div \nabla p^k_{H+1}, \chi_h \right) = \left( \frac{div u^*_H}{\Delta t}, \chi_h \right), \forall \chi_h \in Y_h$
\item Find $u^k_{H+1}$ in $X_h \left( u^k_{H+1} - u^*_h + \Delta t \nabla p^k_{H+1}, \phi_h \right) = 0, \forall \phi_h \in X_h$
\item Solve in $X_H \left( u^k_{H+1} - u^*_h, \psi_H \right) = 0, \forall \psi_H \in X_H$
\item end for
\end{enumerate}

The second approach that we consider in the following is to find directly $u^{k+1}_H$ by the projection method applied on the coarse grid. By this technique, we do not need to restrict $u^{k+1}_h$ to $X_H$ to define $u^{k+1}_H$.

Algorithm 9 Two grids Algo3
\begin{enumerate}
\item for $k = 0, 1, \cdots$ do
\item Find $u^*_H$ in $X_H \left( \frac{u^*_H - u^k_H}{\Delta t}, \psi_H \right) + \nu (\nabla u^*_H, \nabla \psi_H) + (u^*_H \cdot \nabla) u^*_H, \psi_H) = (f, \psi_H), \forall \psi_H \in X_H$
\item Find $p^k_{H+1}$ in $Y_H \left( div \nabla p^k_{H+1}, \chi_H \right) = \left( \frac{div u^*_H}{\Delta t}, \chi_H \right), \forall \chi_H \in Y_H$
\item Find $u^k_{H+1}$ in $X_H \left( u^k_{H+1} - u^*_h + \Delta t \nabla p^k_{H+1}, \psi_h \right) = 0, \forall \psi_h \in X_h$
\item Find $u^*_h$ in $X_h$
\begin{equation}
(1 + \tau \Delta t) \left( \frac{u^*_h - u^k_h}{\Delta t}, \phi_h \right) + \nu (\nabla u^*_h, \nabla \phi_h) + (u^*_h \cdot \nabla) u^*_h, \phi_h) = \tau (u^*_H - u^k_H, \phi_h) + (f, \phi_h), \forall \phi_h \in X_h
\end{equation}
\item Find $p^k_{H+1}$ in $Y_h \left( div \nabla p^k_{H+1}, \chi_h \right) = \left( \frac{div u^*_H}{\Delta t}, \chi_h \right), \forall \chi_h \in Y_h$
\item Find $u^k_{H+1}$ in $X_h \left( u^k_{H+1} - u^*_h + \Delta t \nabla p^k_{H+1}, \phi_h \right) = 0, \forall \phi_h \in X_h$
\item end for
\end{enumerate}
Algorithm 10 Variant of the Two grids Algo3 : Two grids Algo4

1: for $k = 0, 1, \ldots$ do
2: \hspace{1em} Find $u^*_H$ in $X_H \left( \frac{u^*_H - u^k_H}{\Delta t}, \nabla \psi_H \right) + \nu (\nabla u^*_H, \nabla \psi_H) + (u^*_H \cdot \nabla) u^*_H, \psi_H = (f, \psi_H), \forall \psi_H \in X_H$
3: \hspace{1em} Find $p^{k+1}_H$ in $Y_H \left( \text{div} \nabla p^{k+1}_H, \chi_H \right) = \left( \frac{\text{div} u^*_H}{\Delta t}, \chi_H \right), \forall \chi_H \in Y_H$
4: \hspace{1em} Find $u^{k+1}_H$ in $X_H \left( u^{k+1}_H - u^*_H + \Delta t \nabla p^{k+1}_H, \psi_H \right) = 0, \forall \psi_H \in X_H$
5: \hspace{1em} Find $u^*_h$ in $X_h$
6: \hspace{1em} Find $p^{k+1}_h$ in $Y_h \left( \text{div} \nabla p^{k+1}_h, \chi_h \right) = \left( \frac{\text{div} u^*_h}{\Delta t}, \chi_h \right), \forall \chi_h \in Y_h$
7: \hspace{1em} Find $u^{k+1}_h$ in $X_h \left( u^{k+1}_h - u^*_h + \Delta t \nabla p^{k+1}_h, \phi_h \right) = 0, \forall \phi_h \in X_h$
8: end for

3 Numerical results

3.1 The Lid Driven Cavity

We simulate the Navier-Stokes equations in 2D by a stabilization technique. We use the unit square mesh $[0, 1] \times [0, 1]$ and vary the dimensions of the coarse and fine FEM spaces. We choose as spaces of approximation the well-known Taylor-Hood element $P_2/P_1$ for the velocity and the pressure respectively. Here are the results of the driven cavity flow obtained by adopting the variational formulation provided by the projection technique (Chorin-Temam) proposed by the two preceeding steps. In this case the velocity is imposed only in the upper boundary with $u = (1, 0)$ and zero Dirichlet conditions are imposed on the rest of the boundary (see Fig. 1), below

![Figure 1: Boundary conditions](image-url)

3.1.1 Computation of Steady States
Figure 2: From top to bottom: Algorithms \[5, 7, 8, 9\] and \[10\] and from left to right: flow, vorticity, pressure and velocity for \(\Delta t = 10^{-2}, Re = 400, \tau = 0.5, T = 35, \dim(X_H) = 6561, \dim(Y_H) = 1681, \dim(X_h) = 25921, \dim(Y_h) = 6561\).

These flow patterns in Figures \[2, 3, 4, 6, 7\] fit with earlier results of Bruneau et al. \[8\], Ghia et al. \[19\], Goyon \[22\], Pascal \[33\] and Vanka \[39\]. Also, the numerical values and the
Figure 3: From top to bottom: Algorithms 5, 7, 8, 9 and 10 and from left to right: flow, vorticity, pressure and velocity for $\Delta t = 5 \times 10^{-3}$, $Re = 1000$, $\tau = 0.5$, $T = 56$, $dim(X_H) = 6561$, $dim(Y_H) = 1681$, $dim(X_h) = 25921$, $dim(Y_h) = 6561$.

localization of the extrema of the vorticity and of the stream function are good agreement with the ones given in these references.
Figure 4: From top to bottom: Algorithms 5 and 7 for $Re = 2000$, $T = 70$ respectively and from left to right: flow, vorticity, pressure and velocity for $\Delta t = 10^{-3}$, $\tau = 0.5$, $\dim(X_H) = 6561$, $\dim(Y_H) = 1681$, $\dim(X_h) = 25921$, $\dim(Y_h) = 6561$.

Figure 5: From top to bottom: Algorithms 5 and 7 for $Re = 3200$, $T = 130$ respectively and from left to right: flow, vorticity, pressure and velocity for $\Delta t = 10^{-3}$, $\tau = 0.5$, $\dim(X_H) = 6561$, $\dim(Y_H) = 1681$, $\dim(X_h) = 25921$, $\dim(Y_h) = 6561$. 

14
Figure 6: From top to bottom: Algorithm 5 and 7 for $Re = 5000, T = 148$ respectively and from left to right: flow, vorticity, pressure and velocity for $\Delta t = 10^{-3}, \tau = 0.5, \dim(X_H) = 6561, \dim(Y_H) = 1681, \dim(X_h) = 25921, \dim(Y_h) = 6561$.

Figure 7: From top to bottom: Algorithms 5 and 7 for $Re = 7500, T = 195$ respectively and from left to right: flow, vorticity, pressure and velocity for $\Delta t = 5 \times 10^{-4}, \tau = 0.5, \dim(X_H) = 6561, \dim(Y_H) = 1681, \dim(X_h) = 25921, \dim(Y_h) = 6561$. 

15
3.1.2 CPU Time reduction

In Figures 8 - 10 we observe that the bi-grid schemes are faster in computation time than reference the implicit scheme (Algorithm 5) applied on the fine space $V_h$. However, the gain in CPU time is mainly obtained in the transient phase. Indeed, since a stationary solution is computed, the reference scheme will need only one nonlinear iteration at each the time step in the neighborhood of the steady state: it means that only a linear system has then to been solved at each iteration, exactly as for the semi-implicit scheme. Therefore, in a neighborhood of the steady state, an iteration of the bi-grid method needs an additional implicit iteration on the coarse grid and becomes then more expensive in CPU time.

![Figure 8](image1)

Figure 8: From left to right: CPU time of the scheme on a fine grid (Algorithm 5) and that of the scales separation methods for $Re = 100, 400, 1000$ respectively.

![Figure 9](image2)

Figure 9: From left to right: CPU time of the scheme on a fine grid (Algorithm 5) and that of the scales separation method (Algorithm 7) for $Re = 2000, 3200$ respectively.

We also compute the number of nonlinear iterations as a function of time and compute the $L^2$ norm of $\frac{\partial u}{\partial t}$; we use the approximation $\frac{\partial u}{\partial t}|_{t=t_k} \simeq \frac{u^{k+1} - u^k}{\Delta t}$. We identify the time from which the reference scheme (Algorithm 5) reduces to only one linear iteration at each time step and we define $\theta_s$ as the associated threshold value of $\| \frac{\partial u}{\partial t} \|_{L^2(\Omega)}$. We then can obtain an heuristic criteria to define a simple strategy to save more CPU time and we modify the bi-grid scheme as follows: as long as $\| \frac{\partial u}{\partial t} \| > \theta_s$ we apply the bi-grid scheme and as soon as $\| \frac{\partial u}{\partial t} \|_{L^2(\Omega)} \leq \theta_s$ we
Figure 10: From left to right: CPU time of the scheme on a fine grid (Algorithm 5) and that of the scales separation method (Algorithm 7) for $Re = 5000, 7500$ respectively.

apply the reference scheme on the fine space $V_h$. We represent in Tables 1, 2, 3, 4 the calculation times for the four algorithms. We notice that the gain ratio is now equal to 0.7 instead of 0.9 when no threshold strategy was applied on the values of $\frac{\partial u}{\partial t}$.

| 80 × 40 points | Algorithm 7 | Algorithm 8 | Algorithm 9 | Algorithm 10 |
|-----------------|-------------|-------------|-------------|-------------|
| $t_{1G}$ at $T = 17$ (in sec.) | 6448.39 | 6448.39 | 6448.39 | 6448.39 |
| CPU (in sec.) | 5414 | 5504.2 | 5833.06 | 6094.01 |
| $\frac{t_{2G}}{t_{1G}}$ | 0.8396 | 0.8536 | 0.9046 | 0.9450 |
| $\theta_s$ | 0.000528406 | 0.000528406 | 0.000528406 | 0.000528406 |
| $t_{1G}$ (in sec.) | 3015.92 | 3015.92 | 3015.92 | 3015.92 |
| CPU (in sec.) | 2036.78 | 2074.37 | 2190.44 | 2273.63 |
| $\frac{t_{2G}}{t_{1G}}$ | 0.675 | 0.687 | 0.726 | 0.784 |

Table 1: Non incremental Navier-Stokes $Re = 100, \Delta t = 10^{-2}$ and $\mathbb{P}_2/\mathbb{P}_1$.

| 80 × 40 points | Algorithm 7 | Algorithm 8 | Algorithm 9 | Algorithm 10 |
|-----------------|-------------|-------------|-------------|-------------|
| $t_{1G}$ at $T = 35$ (in sec.) | 14311.1 | 14311.1 | 14311.1 | 14311.1 |
| CPU (in sec.) | 11330.3 | 12259.1 | 11978 | 12680.4 |
| $\frac{t_{2G}}{t_{1G}}$ | 0.7917 | 0.8566 | 0.8369 | 0.8860 |
| $\theta_s$ | 0.000438706 | 0.000438706 | 0.000438706 | 0.000438706 |
| $t_{1G}$ (in sec.) | 9191.99 | 9191.99 | 9191.99 | 9191.99 |
| CPU (in sec.) | 6247.12 | 6756.45 | 6595.52 | 6886.03 |
| $\frac{t_{2G}}{t_{1G}}$ | 0.679 | 0.735 | 0.717 | 0.749 |

Table 2: Non incremental Navier-Stokes $Re = 400, \Delta t = 10^{-2}$ and $\mathbb{P}_2/\mathbb{P}_1$. 
Table 3: Non incremental Navier-Stokes \( \text{Re} = 1000, \Delta t = 5 \times 10^{-3} \) and \( P_2/P_1 \).

| 80 \times 40 points | Algorithm 7 | Algorithm 8 | Algorithm 9 | Algorithm 10 |
|---------------------|-------------|-------------|-------------|-------------|
| \( t_{1G} \) at \( T = 56 \) (in sec.) | 42230.3 | 42230.3 | 42230.3 | 42230.3 |
| CPU (in sec.)       | 34967.7 | 37511.5 | 37009.8 | 39466.6 |
| \( t_{2G} / t_{1G} \) | 0.8280 | 0.8882 | 0.8764 | 0.9346 |
| \( \theta_s \) (in sec.) | 0.000948096 | 0.000948096 | 0.000948096 | 0.000948096 |
| CPU (in sec.)       | 16212.7 | 17382 | 17121.7 | 18146.9 |
| \( t_{2G} / t_{1G} \) | 0.688 | 0.737 | 0.726 | 0.77 |

Table 4: Non incremental Navier-Stokes for \( \text{Re} = 2000, 3200, 5000, 7500 \) and \( P_2/P_1 \).

| 80 \times 40 points | \( \text{Re} = 2000 \) \( \Delta t = 10^{-3} \) | \( \text{Re} = 3200 \) \( \Delta t = 10^{-3} \) | \( \text{Re} = 5000 \) \( \Delta t = 10^{-3} \) | \( \text{Re} = 7500 \) \( \Delta t = 5 \times 10^{-4} \) |
|---------------------|------------------|------------------|------------------|------------------|
| Final time          | \( T = 70 \)    | \( T = 130 \)   | \( T = 148 \)   | \( T = 195 \)   |
| \( t_{1G} \) at \( T \) (in sec.) | 122826 | 227543 | 299998 | 846647 |
| CPU (in sec.)       | 108401 | 203692 | 237105 | 670244 |
| \( t_{2G} / t_{1G} \) | 0.8826 | 0.8952 | 0.7904 | 0.791 |
| \( \theta_s \) (in sec.) | 0.00521845 | 0.00502797 | 0.00321512 | 0.00462578 |
| CPU (in sec.)       | 51375.2 | 102922 | 188600 | 565283 |
| \( t_{2G} / t_{1G} \) | 0.751 | 0.776 | 0.736 | 0.745 |

3.1.3 Enhanced Stability: comparison with a semi-implicit scheme

As presented in the previous sections, the bi-grid methods are faster than the unconditionally stable reference scheme (Algorithm 5). To illustrate the stabilization properties of the bi-grid schemes, we make now a comparison with the semi-implicit scheme (applied on whole the fine space \( V_h \)) and which consists in treating the nonlinear term \( (u^k \cdot \nabla)u^k \) explicitly. Particularly we compare the maximum time steps \( \Delta t \) that can be taken for each of the schemes and compare the respective final CPU time on the time interval \( [0, T_f] \). We give hereafter in Tables 5, 6 and 7 the results comparing the stability of the bi-grid Algorithms 7, 8, 9, 10 and the following semi-implicit scheme:
Algorithm 11 Non-incremental semi-implicit scheme

1: \textbf{for} \( k = 0, 1, \cdots \) \textbf{do}
2: \hspace{1em} \textbf{Find} \( u^*_h \) in \( X_h \left( \frac{u^*_h - u^k_h}{\Delta t}, \psi_h \right) + \nu(\nabla u^*_h, \nabla \psi_h) + ((u^*_h \cdot \nabla) u^*_h, \psi_h) = (f, \psi_h), \forall \psi_h \in X_h \)
3: \hspace{1em} \textbf{Find} \( p^{k+1}_h \) in \( Y_h \left( \text{div} \nabla p^{k+1}_h, \chi_h \right) = \left( \text{div} u^*_h \Delta t, \chi_h \right), \forall \chi_h \in Y_h \)
4: \hspace{1em} \textbf{Find} \( u^{k+1}_h \) in \( X_h \left( u^{k+1}_h - u^*_h + \Delta t \nabla p^{k+1}_h, \psi_h \right) = 0, \forall \psi_h \in X_h \)
5: \textbf{end for}

| Scheme   | \( \tau \) | \( \Delta t \) | Stability | \( \tau \) | \( \Delta t \) | Stability | \( \tau \) | \( \Delta t \) | Stability |
|----------|------------|----------------|-----------|------------|----------------|-----------|------------|----------------|-----------|
| Algorithm 7 | 0.5       | 0.01           | yes       | 0.05       | yes            | 0.1       | yes        |
| Algorithm 8 | 0.5       | 0.01           | yes       | 0.05       | yes            | 0.1       | yes        |
| Algorithm 9 | 0.5       | 0.01           | yes       | 0.05       | yes            | 0.1       | yes        |
| Algorithm 10 | 0.5      | 0.01           | yes       | 0.05       | yes            | 0.1       | yes        |
| Algorithm 11 | 0.01     | yes            | yes       | 0.05       | yes            | 0.1       | yes        |

Table 5: \( Re = 100, \dim(X_H) = 6561, \dim(Y_H) = 1681, \dim(X_h) = 25921, \dim(Y_h) = 6561, P_2/P_1 \) elements.

| Scheme   | \( \tau \) | \( \Delta t \) | Stability | \( \tau \) | \( \Delta t \) | Stability | \( \tau \) | \( \Delta t \) | Stability |
|----------|------------|----------------|-----------|------------|----------------|-----------|------------|----------------|-----------|
| Algorithm 7 | 0.5       | 0.01           | yes       | 0.05       | no             | 0.1       | no         |
| Algorithm 8 | 0.5       | 0.01           | yes       | 0.05       | yes            | 0.1       | yes        |
| Algorithm 9 | 0.5       | 0.01           | yes       | 0.05       | yes            | 0.1       | yes        |
| Algorithm 10 | 0.5      | 0.01           | yes       | 0.05       | no             | 0.1       | no         |
| Algorithm 11 | 0.01     | yes            | yes       | 0.05       | yes            | 0.1       | yes        |

Table 6: \( Re = 400, \dim(X_H) = 6561, \dim(Y_H) = 1681, \dim(X_h) = 25921, \dim(Y_h) = 6561, P_2/P_1 \) elements.

We observe in Figure 11 that the dynamics of the convergence to the steady state of the bi-grid method is similar to the one of the implicit and semi-implicit schemes. We note also that the stability of the bi-grid scheme is guaranteed for large values of \( \tau \). For example, for \( Re = 1000 \) and \( \Delta t = 0.007 \), Algorithm 7 was unstable for \( \tau = 0.5 \) but taking a \( \tau = 30 \) we can have a stable scheme without deteriorating the history of the convergence to the steady state. In order to locate our gain at time step \( \Delta t \), we present in Table 8 a comparison between the maximum time step allowing the stability of the semi-implicit scheme and our Algorithm 7 for the minimum value of \( \tau \) necessary for stabilization.
Table 7: \( Re = 1000, \dim(X_H) = 6561, \dim(Y_H) = 1681, \dim(X_h) = 25921, \dim(Y_h) = 6561, \mathbb{P}_2/\mathbb{P}_1 \) elements.

| Scheme          | \( \tau \) | \( \Delta t \) | Stability | \( \Delta t \) | Stability | \( \tau \) | \( \Delta t \) | Stability |
|-----------------|------------|----------------|-----------|----------------|-----------|------------|----------------|-----------|
| Algorithm 7     | 0.5        | 0.005          | yes       | 0.007          | no        | 100        | 0.05           | yes       |
| Algorithm 8     | 0.5        | 0.005          | yes       | 0.007          | yes       | 0.5        | 0.05           | yes       |
| Algorithm 9     | 0.5        | 0.005          | yes       | 0.007          | no        | 150        | 0.05           | yes       |
| Algorithm 10    | 0.5        | 0.005          | yes       | 0.007          | yes       | 0.5        | 0.05           | yes       |
| Algorithm 11    | 0.005      | yes            | 0.007     | no             | 0.05      | no         |

For small Reynolds number (\( Re = 100 \)), we do not obtain an enhanced stability (larger \( \Delta t \)) as compared to the semi-implicit scheme. The latter is stable even for large values of the time step \( \Delta t \). When considering larger values of \( Re \), say \( Re = 400 \) and \( Re = 1000 \), the bi-grid scheme appears as 10 times to 50 times more stable than the semi-implicit one. As a consequence,
the gain in CPU time to compute the steady state increases: the stability of the semi-implicit scheme is limited by $\Delta t = 0.005$ while the bi-grid algorithm remains stable for $\tau = 100$ and $\Delta t = 0.1$ and 0.5.

### 3.2 Precision test: Comparison with an exact solution

In order to show the accuracy of the bi-grid schemes, we simulate the Navier-Stokes equations for a Bercovier-Engelman type solution [5], say

\[
(u_1, u_2) = (-2x^2y(1 - y)(1 - x)^2 \exp^{\sin t}, 2y^2 x(1 - x)(1 - y)^2 \exp^{\sin t})
\]

\[
p = (x - 0.5)(y - 0.5)
\]

and with the corresponding term $f$. We can observe in Figures 12 and 13 that the solutions coincide with the reference one at final time $T = 56$ and that the $L^2$ norm of the error bi-grid schemes is of the order of $\Delta t$ on all interval time.

![Figure 12: From top to bottom: velocity and pressure, and from left to right: the exact solution and Algorithms 5 and 7 solutions for $\Delta t = 5 \times 10^{-3}, Re = 1000, T = 56, \dim(X_H) = 6561, \dim(Y_H) = 1681, \dim(X_h) = 25921, \dim(Y_h) = 6561.](image)
Figure 13: From left to right: the $L^2$–error for the velocity and the pressure between the exact solution and the solution of algorithms 5, 7, 8, 9 and 10, for $\Delta t = 5 \times 10^{-3}$, $Re = 1000$, $T = 56$, $dim(X_H) = 6561$, $dim(Y_H) = 1681$, $dim(X_h) = 25921$, $dim(Y_h) = 6561$.

4 Concluding Remarks and perspectives

The new bi-grid projection schemes introduced in this work allow to reduce the CPU time when compared to fully implicit projection scheme, for a comparable precision. Their stability is enhanced as compared to semi-implicit projection scheme (larger time steps can be used). The stabilization we used on the high mode components is simple and does not deteriorate the consistency, the numerical results we obtained on the benchmark driven cavity agree with the ones of the literature, particularly the dynamics of the convergence to the steady state is close to the one of classical methods. This is a first and encouraging step before considering a strategy of multi-grid or multilevel adaptations of our methods, using more than 2 nested FEM spaces. We have used here FEM methods for the spatial discretization, but the approach is applicable to others discretizations techniques such as finite differences or spectral methods.

Acknowledgment

This project has been founded with support from the National Council for Scientific Research in Lebanon and the Lebanese University. We also thank the Fédération de Recherche ARC, CNRS FR 3399.

Conflict of interest

All authors declare no conflicts of interest in this paper.
References

[1] H. Abboud, C. Alkosseifi, J-P. Chehab. A stabilized bi-grid method for Allen-Cahn equation in Finite Elements. Arxiv.

[2] H. Abboud, V. Girault, T. Sayah. A second order accuracy for a full discretized time-dependent Navier-Stokes equations by a two-grid scheme. Numer. Math. DOI 10.1007/s00211-009-0251-5.

[3] H. Abboud and T. Sayah. A full discretization of a time-dependent two dimensional Navier-Stokes equations by a two-grid scheme. M2AN Math. Model. Numer. Anal., Vol 42, N 1(2008), pp. 141-174.

[4] I. Babuška. The finite element method with Lagrangian multipliers. Numer. Math. 20, 179-192, 1973.

[5] M. Bercovier and M. Engelman. A finite element for the numerical solution of viscous incompressible flows Journal of Computational Physics, 30(2):181?201, 1979.

[6] A. Bousquet, M. Marion, M. Petcu and R. Temam, Multilevel finite volume methods and boundary conditions for geophysical flows, Computers and Fluids, 74, 2013, 66-90

[7] F. Brezzi. On the existence, uniqueness and approximation of saddle-point problem arising from Lagrangian multipliers. RAIRO Anal. Numér. 129-151, 1974.

[8] C.H. Bruneau and C. Jouron. An efficient scheme for solving steady incompressible Navier-Stokes equations. J Comput Phys 89, 389-413, 1990.

[9] C.Calgaro, J. Laminie, R. Temam, Dynamical multilevel schemes for the solution of evolution equations by hierarchical finite element discretization Appl. Numer. Math. 23 (1997), no. 4, 403–442.

[10] C. Calgaro, A. Debussche, J. Laminie, On a multilevel approach for the two-dimensional Navier-Stokes equations with finite elements. Finite elements in fluids. Internat. J. Numer. Methods Fluids 27 (1998), no. 1–4, Special Issue, 241–258.

[11] C. Calgaro, J.-P. Chehab, J. Laminie, E. Zahrouni, Schémas multiniveaux pour les équations d’ondes, (French) [Multilevel schemes for waves equations],ESAIM Proc., 27, EDP Sci., (2009), p 180-208.

[12] J.-P. Chehab et B. Costa, Time explicit schemes and spatial finite differences splittings, Journal of Scientific Computing, 20, 2 (2004), pp 159-189.

[13] B. Costa. L. Dettori, D. Gottlieb and R. Temam, Time marching techniques for the nonlinear Galerkin method, SIAM J. SC. comp., 23, (2001), 1, 46-65.

[14] A.J. Chorin. Numerical solution of the Navier-Stokes equations. Math. Comput. 22, 745-762, 1968.
[15] Th. Dubois, F. Jauberteau, R. Temam, *Dynamic multilevel methods and the numerical simulation of homogeneous and non homogeneous turbulence*, Cambridge Academic Press.

[16] Th. Dubois, F. Jauberteau, R. Temam, J. Tribbia, *Multilevel schemes for the shallow water equations* J. Computational Physics, 207, 2, 660–694 (2005)

[17] S. Faure, J. Laminie, R. Temam, *Finite Volume Discretization and Multilevel Methods in Flow Problems*, Journal of Scientific Computing, Vol. 25, No. 1, 2005.

[18] FreeFem++ [http://www.freefem.org](http://www.freefem.org)

[19] U. Ghia, K.N. Ghia and C.T. Shin. *High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method*. J Comput Phys 48, 387-411, 1982.

[20] V. Girault and J.-L. Lions. *Two-grid finite-element schemes for the transient Navier-Stokes equations*. M2AN 35, 945-980 (2001).

[21] K. Goda. *A multistep technique with implicit difference schemes for calculating two- or three-dimensional cavity flows*. J Comput Phys 30, 76-95, (1979).

[22] O. Goyon. *High-Reynolds number solutions of Navier-Stokes equations using incremental unknowns*. Comput. Methods Appl. Mech. Engrg. 130, 319-335, 1996.

[23] J.L. Guermond, P. Minev and J. Shen. *An overview of projection methods for incompressible flows*. Comput. Methods Appl. Mech. Engrg. 195, 6011-6045, 2006.

[24] J.L. Guermond and J. Shen. *Velocity-correction projection methods for incompressible flows*. SIAM J. Numer. Anal., 41 (1): 112-134, 2003.

[25] J.L. Guermond and J. Shen. *Quelques résultats nouveaux sur les méthodes de projection*. CR Acad. Sci. Paris, Série I 333, 1111-1116, 2001.

[26] W. Hackbusch, *Multi-grid methods and application* 1985, Springer Berlin

[27] Y. He and K.M. Liu. *Multi-level spectral Galerkin method for the Navier-Stokes equations, II: time discretization* Advances in Computational Mathematics 25: 403-433, 2006.

[28] F. Jauberteau, R. Temam and J. Tribbia, *Multiscale/fractional step schemes for the numerical simulation of the rotating shallow water flows with complex periodic topography*. J. Computational Physics, 270, 2014, 506-531, [http://dx.doi.org/10.1016/j.cjcp2014.03.036](http://dx.doi.org/10.1016/j.cjcp2014.03.036)

[29] W. Layton. *A two-level discretization method for the Navier-Stokes equations*. Computers Math. Applic., 26, 2(1993), pp. 33-38.

[30] M. Marion and R. Temam, *Nonlinear Galerkin Methods*, SIAM Journal of Numerical Analysis, 26, 1989,1139-1157.

[31] M. Marion and R. Temam, *Nonlinear Galerkin Methods ; The Finite elements case*, Numerische Mathematik, 57, 1990 , 205-226.
[32] M. Marion and J. Xu, *Error estimates on a new nonlinear Galerkin method based on two-grid finite elements*. SIAM Journal on Numerical Analysis, Volume 32 Issue 4, Aug. 1995, Pages 1170-1184.

[33] F. Pascal. *Méthodes de Galerkin non linéaires en discrétisation par éléments finis et pseudo-spectrale. Application la mécanique des fluides*. Université de Paris-Sud Orsay, 1992.

[34] F. Pouit, *Etude de schémas numériques multiniveaux utilisant les inconnues incrémentales dans le cas des différences finies : application à la mécanique des fluides*. Thèse, Université Paris 11, 1998, in french.

[35] R. Temam. *Navier-Stokes equations*. North-Holland, Amsterdam (1984). Revised version.

[36] R. Temam. *Navier-Stokes equations, Theory and numerical analysis*. North-Holland, Amsterdam (1977).

[37] R. Temam. *Approximation d'équations aux dérivées partielles par des méthodes de décomposition*. Séminaire Bourbaki 381, 1969/1970.

[38] R. Temam. *Une méthode d'approximation de la solution des équations de Navier-Stokes*. Bull. Soc. Math. France 98, 115-152, (1968).

[39] S.P. Vanka. *Block-implicit multigrid solution of Navier-Stokes equations in primitive variables*. J. Comput. Phys. 65, 138-158, (1986).

[40] J. Xu. *Some Two-Grid Finite Element Methods*. Tech. Report, P.S.U, (1992).

[41] J. Xu. *A novel two-grid method of semilinear elliptic equations*. SIAM J. Sci. Comput., 15(1994), pp. 231-237.

[42] J. Xu. *Two-grid finite element discretization techniques for linear and nonlinear PDE*. SIAM J. Numer. Anal., 33(1996), pp. 1759-1777.

[43] H. Yserentant, *On multilevel splitting of finite element spaces*, Numer. Math, 49,(1986), pp 379-412.