Mathematics

Research article

Stabilized bi-grid projection methods in finite elements for the 2D incompressible Navier-Stokes equations

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

* **Correspondence:** Email: Jean-Paul.Chehab@u-picardie.fr; Tel: +33(0)322827591; Fax: +33(0)322827838.

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
Mathematics Subject Classification: 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. [27] for steady linear and nonlinear Dirichlet Problems in finite differences, but also for Steady Navier-Stokes Equations [9, 20].

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 coarsest (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 [16] 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 [14]. 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 [49]: 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 [46–48] and the references therein for Finite Elements discretizations, [13, 38] for Finite Differences and [6, 18] 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], [21] and [28]. 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:

$$\frac{d}{dt}(u, v) + (\nabla u, \nabla v) + (f(u), v) = 0, \forall v \in V, \tag{1}$$

$$v(0) = v_0, \tag{2}$$

where V is a proper Hilbert space.

Given two finite dimensional subspaces of V, W_H and V_h, with $dim(W_H) << dim(V_h)$, we define the bigrid-scheme as

---

Algorithm 1 Bi-grid Stabilized scheme for Reaction-Diffusion

1: u_h^0, u_H^0 given

2:

3: **for** $k = 0, 1, \cdots$ **do**

4:      **Step 1 (Coarse Space Implcit Scheme)**

5:      $(\dfrac{u_H^{k+1} - u_H^k}{\Delta t}, \psi_H) + (\nabla u_H^{k+1}, \nabla \psi_H) + (f(u_H^{k+1}), \psi_H) = 0, \ \forall \psi_H \in W_H$

6:      **Step 2 (Fine Space semi-implicit Scheme)**

7:      $(\dfrac{u_h^{k+1} - u_h^k}{\Delta t}, \phi_h) + \tau(\mathbf{u_h^{k+1}} - \mathbf{u_h^k}, \phi_\mathbf{h}) + (\nabla u_h^{k+1}, \nabla \phi_h) + (f(u_h^k), \phi_h) = \tau(\mathbf{u_H^{k+1}} - \mathbf{u_H^k}, \phi_\mathbf{h}), \forall \phi_h \in V_h$

8: **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 $\tau(\mathbf{u_h^{k+1}} - \mathbf{u_h^k}, \phi_\mathbf{h})$ and $\tau(\mathbf{u_H^{k+1}} - \mathbf{u_H^k}, \phi_\mathbf{h})$ are interpreted as a high mode filter and $\tau > 0$ is the stabilization parameter [1] and allow to compense 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++ [19].

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_h^0, u_H^0 given

2:

3: **for** $k = 0, 1, \cdots$ **do**

4:      **Step 1 (Coarse Space Implicit Scheme)**

5:      $(\dfrac{u_H^{k+1} - u_H^k}{\Delta t}, \psi_H) + (\nabla u_H^{k+1}, \nabla \psi_H) + (f(u_H^{k+1}), \psi_H) = 0, \ \forall \psi_H \in W_H$

6:      **Step 2 (Fine Space semi-implicit Scheme)**

7:      $(\dfrac{u_h^{k+1} - u_h^k}{\Delta t}, \phi_h) + (\nabla u_h^{k+1}, \nabla \phi_h) + (f(u_h^k), \phi_h) + (f'(u_H^{k+1})u_h^{k+1}, \phi_h) = 0, \forall \phi_h \in V_h$

8: **end for**

This approach was proposed by [21, 28, 30]. However, $f'(u_H^{k+1})$, the linearized of f at u_H^{k+1} 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 $\tau(u_h^{k+1} - u_h^k)$, obtaining as second step:

1: **Step 2 (Stabilized Fine Space semi-implicit Scheme)**

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

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)**

2: $(\dfrac{u_h^{k+1} - u_h^k}{\Delta t}, \phi_h) + \tau(\mathcal{P}(u_h^{k+1} - u_h^k), \phi_h) + (\nabla u_h^{k+1}, \nabla \phi_h) + (f(u_h^k), \phi_h) = 0, \forall \phi_h \in V_h$

---

where $\mathcal{P}(u_h^{k+1} - u_h^k)$ 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 $\tilde{u}_h$ 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 $\mathbb{P}_1$ 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 pre-conditioner for the stiffness matrices but also proceeds to a scale separation [10, 11, 33, 34, 49].

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

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

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

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 $\Phi(u_H) = z_h$, [33, 34].

The approach followed in [35] 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 $w_h = (Id - P_H)u_h$ 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 $(Id - P_H)V_h$.

Finally let us mention similar developments in finite differences [12, 13] and spectral methods [14, 16, 17, 29]

- A L^2-like filtering, used in [1] and to which we will concentrate: we define first the prolongation operator $\mathcal{P} : V_H \rightarrow V_h$ by

$$(u_H - \mathcal{P}(u_H), \phi_h) = 0, \forall\phi_h \in V_h. \tag{5}$$

Then, setting $\tilde{u}_h = \mathcal{P}(u_H)$, 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 simplifications

---

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

2: $(\dfrac{u_h^{k+1} - u_h^k}{\Delta t}, \phi_h) + \tau(z_h^{k+1} - z_h^k, \phi_h) + (\nabla u_h^{k+1}, \nabla\phi_h) + (f(u_h^k), \phi_h) = 0, \forall\phi_h \in V_h$

---

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

Algorithm 3 Bi-grid Stabilized scheme for Reaction-Diffusion

1: u_h^0, u_H^0 given

2:

3: **for** $k = 0, 1, \cdots$ **do**

4:      **Step 1 (Coarse Space Implicit Scheme)**

5:      $(\dfrac{u_H^{k+1} - u_H^k}{\Delta t}, \psi_H) + (\nabla u_H^{k+1}, \nabla \psi_H) + (f(u_H^{k+1}), \psi_H) = 0, \ \forall \psi_H \in W_H$

6:      **Step 2 (Fine Space semi-implicit Scheme)**

7:      $(\dfrac{u_h^{k+1} - u_h^k}{\Delta t}, \phi_h) + \tau(\mathbf{u_h^{k+1}} - \mathbf{u_h^k}, \phi_\mathbf{h}) + (\nabla u_h^{k+1}, \nabla \phi_h) + (f(u_h^k), \phi_h) = \tau(\mathbf{u_H^{k+1}} - \mathbf{u_H^k}, \phi_\mathbf{h}), \forall \phi_h \in V_h$

8: **end for**

We now consider incompressible Navier-Stokes equations. FEM Bi-grid methods for incompressible NSE as proposed, e.g., in [2, 3, 21, 28, 30], 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_h^0, u_H^0 given

2:

3: **for** $k = 0, 1, \cdots$ **do**

4:      **Step 1 (Coarse Space Implicit Scheme)**

5:      $(\dfrac{u_H^{k+1} - u_H^k}{\Delta t}, \psi_H) + \dfrac{1}{Re}(\nabla u_H^{k+1}, \nabla \psi_H) + ((u_H^{k+1}.\nabla)u_H^{k+1}, \psi_H) - (div(\psi_H), p_H^{k+1}) = (f, \psi_H), \forall \psi_H \in X_H$

6:      $(div(u_H^{k+1}), q_H) = 0, \forall q_H \in Y_H$

7:      **Step 2 (Fine Space semi-implicit Scheme)**

8:      $(\dfrac{u_h^{k+1} - u_h^k}{\Delta t}, \phi_h) + \dfrac{1}{Re}(\nabla u_h^{k+1}, \nabla \phi_h) + (u_H^{k+1}.\nabla u_h^{k+1}, \phi_h) - (div(\psi_h), p_h^{k+1}) = (f, \psi_h), \forall \psi_h \in X_h$

9:      $(div(u_h^{k+1}), q_h) = 0, \forall q_h \in Y_h$

10: **end for**

We have used here the simple linearization of the nonlinear term $(u_H^{k+1}.\nabla u_h^{k+1}, \phi_h)$ proposed in [21], but other linearizations 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 Ω is described by the unsteady Navier-Stokes equation

$$\begin{cases} (\dfrac{\partial u}{\partial t}, v) + (u \cdot \nabla u, v) - \nu(\nabla u, \nabla v) - (p, \operatorname{div} v) = (f, v), \ \forall v \in X = (H_0^1(\Omega))^d \\ (\operatorname{div} u, q) = 0, \ \forall q \in Y = L^2(\Omega) \end{cases}$$

where Ω is a domain of $\mathbb{R}^d, d = 2, 3$ of lipschizian bound $\partial\Omega, \overrightarrow{n}$ 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

$$X_h = \{v \in C^0(\bar{\Omega})^2 | v_{|\kappa} \in \mathbb{P}_2^2, \ \forall \kappa \in T_h, \ v = 0 \text{ on } \partial\Omega\}, \tag{6}$$

$$Y_h = \{q \in C^0(\bar{\Omega}) | q_{|\kappa} \in \mathbb{P}_1, \ \forall \kappa \in T_h\} \cap L_0^2(\Omega). \tag{7}$$

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]):

$$\inf_{q \in Y_h} \sup_{u \in X_h} \frac{\int_\Omega q(div \ u)dx}{\|q\|_{L^2(\Omega)} \|\nabla u\|_{L^2(\Omega)}} \geq \beta^*, \tag{8}$$

where β^* 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; Δt 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. [15, 24–26, 40, 42, 43]). We start by presenting the implicit reference scheme [40]:

Algorithm 5 Reference Scheme

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

2:      **Find** u_h^* **in** X_h $(\dfrac{u_h^* - u_h^k}{\Delta t}, \psi_h) + \nu(\nabla u_h^*, \nabla \psi_h) + ((u_h^* \cdot \nabla)u_h^*, \psi_h) = (f, \psi_h), \forall \psi_h \in X_h$

3:      **Find** p_h^{k+1} **in** Y_h $(div \, \nabla p_h^{k+1}, \chi_h) = (\dfrac{div \, u_h^*}{\Delta t}, \chi_h), \forall \chi_h \in Y_h$

4:      **Find** u_h^{k+1} **in** X_h $(u_h^{k+1} - u_h^* + \Delta t \nabla p_h^{k+1}, \psi_h) = 0, \forall \psi_h \in X_h$

5: **end for**

This will be our reference scheme, used on the coarse FEM Space. It enjoys of unconditional stability properties, see Temam [40], chapter III, section 7.3 and [41].

Remark 2.1. *We could use also an incremental projection method introduced by Goda [22], see also [24]. Goda has proven that adding a previous value of the gradient of the pressure (∇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:      $(\dfrac{u_h^* - u_h^k}{\Delta t}, \psi_h) + \nu(\nabla u_h^*, \nabla \psi_h) + ((u_h^* \cdot \nabla)u_h^*, \psi_h) + (\nabla p_h^k, \psi_h) = (f, \psi_h), \forall \psi_h \in X_h$

4:      **Find** p_h^{k+1} **in** Y_h $\alpha(div \, \nabla p_h^{k+1} - div \, \nabla p_h^k, \chi_h) = (\dfrac{div \, u_h^*}{\Delta t}, \chi_h), \forall \chi_h \in Y_h$

5:      **Find** u_h^{k+1} **in** X_h $(u_h^{k+1} - u_h^* + \alpha \Delta t (\nabla p_h^{k+1} - \nabla p_h^k), \psi_h) = 0, \forall \psi_h \in X_h$

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 [15, 40, 42, 43]. 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. In all our bi-grid schemes we solve the nonlinear problem using the usual Picard fixed point method on the coarse space:

> Initilisation: set $u_H^{*,0} = u_H^k$,
> For $m \geq 0$ up to convergence
> Find $u_H^{*,m+1}$ solution of
> $(u_H^{*,m+1}, \phi_H) + \Delta t(\nabla u_H^{*,m+1}, \nabla \phi_H) = (u_H^{*,m}, \phi_H) + \Delta t((u_H^{*,m}.\nabla)u_H^{*,m}, \phi_H), \ \forall \phi_H \in X_H$
> End
> Set $u_H^* = u_H^{*,m+1}$

Only few iterations are needed at each time step (4 at the beginning and only 1 close to the steady state) making this simple fixed point method efficient and well adapted in this context.

In the same way we solve the nonlinear problem in the reference scheme on a fine grid. 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^k for the next iteration.

Algorithm 7 Two grids Algo1

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

2:     **Find** u_H^* **in** X_H $(\dfrac{u_H^* - u_H^k}{\Delta t}, \psi_H) + \nu(\nabla u_H^*, \nabla \psi_H) + ((u_H^* \cdot \nabla)u_H^*, \psi_H) = (f, \psi_H), \forall \psi_H \in X_H$

3:     **Find** u_h^* **in** X_h

$$(1 + \tau \Delta t)(\dfrac{u_h^* - u_h^k}{\Delta t}, \phi_h) + \nu(\nabla u_h^*, \nabla \phi_h) + ((u_h^k \cdot \nabla)u_h^k, \phi_h) = \tau(u_H^* - u_H^k, \phi_h) + (f, \phi_h), \forall \phi_h \in X_h$$

4:     **Find** p_h^{k+1} **in** Y_h $(div \, \nabla p_h^{k+1}, \chi_h) = (\dfrac{div \, u_h^*}{\Delta t}, \chi_h), \forall \chi_h \in Y_h$

5:     **Find** u_h^{k+1} **in** X_h $(u_h^{k+1} - u_h^* + \Delta t \nabla p_h^{k+1}, \phi_h) = 0, \forall \phi_h \in X_h$

6:     **Solve in** X_H $(u_H^{k+1} - u_h^{k+1}, \psi_H) = 0, \forall \psi_H \in X_H$

7: **end for**

We propose in the second algorithm to replace $(u_h^k.\nabla)u_h^k$ by $(u_H^* \cdot \nabla)u_h^*$. We obtain:

Algorithm 8 Variant of the Two grids Algo1 : Two grids Algo2

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

2:     **Find** u_H^* **in** X_H $(\dfrac{u_H^* - u_H^k}{\Delta t}, \psi_H) + \nu(\nabla u_H^*, \nabla \psi_H) + ((u_H^* \cdot \nabla)u_H^*, \psi_H) = (f, \psi_H), \forall \psi_H \in X_H$

3:     **Find** u_h^* **in** X_h

$$(1 + \tau \Delta t)(\dfrac{u_h^* - u_h^k}{\Delta t}, \phi_h) + \nu(\nabla u_h^*, \nabla \phi_h) + ((u_H^* \cdot \nabla)u_h^*, \phi_h) = \tau(u_H^* - u_H^k, \phi_h) + (f, \phi_h), \forall \phi_h \in X_h$$

4:     **Find** p_h^{k+1} **in** Y_h $(div \, \nabla p_h^{k+1}, \chi_h) = (\dfrac{div \, u_h^*}{\Delta t}, \chi_h), \forall \chi_h \in Y_h$

5:     **Find** u_h^{k+1} **in** X_h $(u_h^{k+1} - u_h^* + \Delta t \nabla p_h^{k+1}, \phi_h) = 0, \forall \phi_h \in X_h$

6:     **Solve in** X_H $(u_H^{k+1} - u_h^{k+1}, \psi_H) = 0, \forall \psi_H \in X_H$

7: **end for**

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

Algorithm 9 Two grids Algo3

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

2:     **Find** u_H^* **in** X_H $(\dfrac{u_H^* - u_H^k}{\Delta t}, \psi_H) + \nu(\nabla u_H^*, \nabla \psi_H) + ((u_H^* \cdot \nabla)u_H^*, \psi_H) = (f, \psi_H), \forall \psi_H \in X_H$

3:     **Find** p_H^{k+1} **in** Y_H $(div \, \nabla p_H^{k+1}, \chi_H) = (\dfrac{div \, u_H^*}{\Delta t}, \chi_H), \forall \chi_H \in Y_H$

4:     **Find** u_H^{k+1} **in** X_H $(u_H^{k+1} - u_H^* + \Delta t \nabla p_H^{k+1}, \psi_H) = 0, \forall \psi_H \in X_H$

5:     **Find** u_h^* **in** X_h

$$(1 + \tau \Delta t)(\dfrac{u_h^* - u_h^k}{\Delta t}, \phi_h) + \nu(\nabla u_h^*, \nabla \phi_h) + ((u_h^* \cdot \nabla)u_h^*, \phi_h) = \tau(u_H^* - u_H^k, \phi_h) + (f, \phi_h), \forall \phi_h \in X_h$$

6:     **Find** p_h^{k+1} **in** Y_h $(div \, \nabla p_h^{k+1}, \chi_h) = (\dfrac{div \, u_h^*}{\Delta t}, \chi_h), \forall \chi_h \in Y_h$

7:     **Find** u_h^{k+1} **in** X_h $(u_h^{k+1} - u_h^* + \Delta t \nabla p_h^{k+1}, \phi_h) = 0, \forall \phi_h \in X_h$

8: **end for**

Algorithm 10 Variant of the Two grids Algo3: Two grids Algo4

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

2:     **Find** u_H^* **in** X_H $(\dfrac{u_H^* - u_H^k}{\Delta t}, \psi_H) + \nu(\nabla u_H^*, \nabla \psi_H) + ((u_H^* \cdot \nabla)u_H^*, \psi_H) = (f, \psi_H), \forall \psi_H \in X_H$

3:     **Find** p_H^{k+1} **in** Y_H $(div \, \nabla p_H^{k+1}, \chi_H) = (\dfrac{div \, u_H^*}{\Delta t}, \chi_H), \forall \chi_H \in Y_H$

4:     **Find** u_H^{k+1} **in** X_H $(u_H^{k+1} - u_H^* + \Delta t \nabla p_H^{k+1}, \psi_H) = 0, \forall \psi_H \in X_H$

5:     **Find** u_h^* **in** X_h

$$(1 + \tau \Delta t)(\dfrac{u_h^* - u_h^k}{\Delta t}, \phi_h) + \nu(\nabla u_h^*, \nabla \phi_h) + ((u_H^* \cdot \nabla)u_h^*, \phi_h) = \tau(u_H^* - u_H^k, \phi_h) + (f, \phi_h), \forall \phi_h \in X_h$$

6:     **Find** p_h^{k+1} **in** Y_h $(div \, \nabla p_h^{k+1}, \chi_h) = (\dfrac{div \, u_h^*}{\Delta t}, \chi_h), \forall \chi_h \in Y_h$

7:     **Find** u_h^{k+1} **in** X_h $(u_h^{k+1} - u_h^* + \Delta t \nabla p_h^{k+1}, \phi_h) = 0, \forall \phi_h \in X_h$

8: **end for**

Remark 2.2. *We have considered above only first order time marching schemes. However, the stabilization technique can be applied/adapted to more accurate methods such as BDF. For instance, when looking to the classical Gear-Scheme. We refer to [45], [44] and to [24] where an incremental projection method has been introduced. We can propose the following one grid algorithm, considered as our reference second-order in time scheme:*

Algorithm 11 Non-Incremental Second-order Reference Scheme

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

2:      **Find** u_h^* **in** X_h

3:      $(\dfrac{3u_h^* - 4u_h^k + u_h^{k-1}}{2\Delta t}, \psi_h) + \nu(\nabla u_h^*, \nabla \psi_h) + ((u_h^* \cdot \nabla)u_h^*, \psi_h) + (\nabla p_h^k, \psi_h) = (f, \psi_h), \forall \psi_h \in X_h$

4:      **Find** p_h^{k+1} **in** Y_h $(div \, \nabla p_h^{k+1}, \chi_h) = (\dfrac{div \, 3u_h^*}{2\Delta t}, \chi_h), \forall \chi_h \in Y_h$

5:      **Find** u_h^{k+1} **in** X_h $(u_h^{k+1} - u_h^* + \dfrac{2}{3}\Delta t \nabla p_h^{k+1}, \psi_h) = 0, \forall \psi_h \in X_h.$

6: **end for**

Since the time scheme used on the equation in u_h^* is a second accurate Gear's one, it is natural to associate a second order stabilization, namely related to the discretization of the second derivative in time. Hence, when one grid only is used, the damping term to add will be

$$\tau(u_h^* - 2u_h^k + u_h^{k-1}, \phi_h) \simeq \tau \Delta t^2(\frac{\partial^2 u}{\partial t^2}, \phi_h),$$

as proposed in [39] for the stabilization of semi-explicit schemes applied to reaction diffusion problems, when using one-grid FEM. Now, we can apply this stabilization procedure to the high mode components, as presented below, and we can derive the following new bi-grids algorithms, considering several options:

Algorithm 12 Two grids Second-order Algo1Order2

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

2:      **Find** u_H^* **in** X_H $(\dfrac{3u_H^* - 4u_H^k + u_H^{k-1}}{2\Delta t}, \psi_H) + \nu(\nabla u_H^*, \nabla \psi_H) + ((u_H^* \cdot \nabla)u_H^*, \psi_H)s = (f, \psi_H), \forall \psi_H \in X_H$

3:      **Find** u_h^* **in** X_h

$$(\frac{3u_h^* - 4u_h^k + u_h^{k-1}}{2\Delta t}, \phi_h) + \nu(\nabla u_h^*, \nabla \phi_h) + ((u_h^k \cdot \nabla)u_h^k, \phi_h) + \tau(u_h^* - 2u_h^k + u_h^{k-1}, \phi_h)$$

$$= \tau(u_H^* - 2u_H^k + u_H^{k-1}, \phi_h) + (f, \phi_h), \forall \phi_h \in X_h$$

4:      **Find** p_h^{k+1} **in** Y_h $(div \, \nabla p_h^{k+1}, \chi_h) = (\dfrac{div \, 3u_h^*}{2\Delta t}, \chi_h), \forall \chi_h \in Y_h$

5:      **Find** u_h^{k+1} **in** X_h $(u_h^{k+1} - u_h^* + \dfrac{2}{3}\Delta t \nabla p_h^{k+1}, \phi_h) = 0, \forall \phi_h \in X_h$

6:      **Solve in** X_H $(u_H^{k+1} - u_h^{k+1}, \psi_H) = 0, \forall \psi_H \in X_H$

7: **end for**

Algorithm 13 Variant of the Two grids Second-order "Algo1Order2" algorithm: "Algo2Order2"

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

2:     **Find** u_H^* **in** X_H $(\dfrac{3u_H^* - 4u_H^k + u_H^{k-1}}{2\Delta t}, \psi_H) + \nu(\nabla u_H^*, \nabla \psi_H) + ((u_H^* \cdot \nabla)u_H^*, \psi_H) = (f, \psi_H), \forall \psi_H \in X_H$

3:     **Find** u_h^* **in** X_h

$$(\frac{3u_h^* - 4u_h^k + u_h^{k-1}}{2\Delta t}, \phi_h) + \nu(\nabla u_h^*, \nabla \phi_h) + ((u_H^* \cdot \nabla)u_h^*, \phi_h) + \tau(u_h^* - 2u_h^k + u_h^{k-1}, \phi_h)$$

$$= \tau(u_H^* - 2u_H^k + u_H^{k-1}, \phi_h) + (f, \phi_h), \forall \phi_h \in X_h$$

4:     **Find** p_h^{k+1} **in** Y_h $(div \, \nabla p_h^{k+1}, \chi_h) = (\dfrac{div \, 3u_h^*}{2\Delta t}, \chi_h), \forall \chi_h \in Y_h$

5:     **Find** u_h^{k+1} **in** X_h $(u_h^{k+1} - u_h^* + \dfrac{2}{3}\Delta t \nabla p_h^{k+1}, \phi_h) = 0, \forall \phi_h \in X_h$

6:     **Solve in** X_H $(u_H^{k+1} - u_h^{k+1}, \psi_H) = 0, \forall \psi_H \in X_H$

7: **end for**

Algorithm 14 Two grids Second-order "Algo3Order2"

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

2:     **Find** u_H^* **in** X_H $(\dfrac{3u_H^* - 4u_H^k + u_H^{k-1}}{2\Delta t}, \psi_H) + \nu(\nabla u_H^*, \nabla \psi_H) + ((u_H^* \cdot \nabla)u_H^*, \psi_H) = (f, \psi_H), \forall \psi_H \in X_H$

3:     **Find** p_H^{k+1} **in** Y_H $(div \, \nabla p_H^{k+1}, \chi_H) = (\dfrac{div \, 3u_H^*}{2\Delta t}, \chi_H), \forall \chi_H \in Y_H$

4:     **Find** u_H^{k+1} **in** X_H $(u_H^{k+1} - u_H^* + \dfrac{2}{3}\Delta t \nabla p_H^{k+1}, \phi_H) = 0, \forall \phi_H \in X_H$

5:     **Find** u_h^* **in** X_h

$$(\frac{3u_h^* - 4u_h^k + u_h^{k-1}}{2\Delta t}, \phi_h) + \nu(\nabla u_h^*, \nabla \phi_h) + ((u_h^k \cdot \nabla)u_h^k, \phi_h) + \tau(u_h^* - 2u_h^k + u_h^{k-1}, \phi_h)$$

$$= \tau(u_H^* - 2u_H^k + u_H^{k-1}, \phi_h) + (f, \phi_h), \forall \phi_h \in X_h$$

6:     **Find** p_h^{k+1} **in** Y_h $(div \, \nabla p_h^{k+1}, \chi_h) = (\dfrac{div \, 3u_h^*}{2\Delta t}, \chi_h), \forall \chi_h \in Y_h$

7:     **Find** u_h^{k+1} **in** X_h $(u_h^{k+1} - u_h^* + \dfrac{2}{3}\Delta t \nabla p_h^{k+1}, \phi_h) = 0, \forall \phi_h \in X_h$

8:     **Solve in** X_H $(u_H^{k+1} - u_h^{k+1}, \psi_H) = 0, \forall \psi_H \in X_H$

9: **end for**

Algorithm 15 Variant of the Two grids Second-order "Algo3Order2" algorithm: "Algo4Order2"

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

2:     **Find** u_H^* **in** X_H $(\dfrac{3u_H^* - 4u_H^k + u_H^{k-1}}{2\Delta t}, \psi_H) + \nu(\nabla u_H^*, \nabla \psi_H) + ((u_H^* \cdot \nabla) u_H^*, \psi_H) = (f, \psi_H), \forall \psi_H \in X_H$

3:     **Find** p_H^{k+1} **in** Y_H $(div \, \nabla p_H^{k+1}, \chi_H) = (\dfrac{div \, 3u_H^*}{2\Delta t}, \chi_H), \forall \chi_H \in Y_H$

4:     **Find** u_H^{k+1} **in** X_H $(u_H^{k+1} - u_H^* + \dfrac{2}{3}\Delta t \nabla p_H^{k+1}, \phi_H) = 0, \forall \phi_H \in X_H$

5:     **Find** u_h^* **in** X_h

$$(\frac{3u_h^* - 4u_h^k + u_h^{k-1}}{2\Delta t}, \phi_h) + \nu(\nabla u_h^*, \nabla \phi_h) + ((u_H^* \cdot \nabla)u_h^*, \phi_h) + \tau(u_h^* - 2u_h^k + u_h^{k-1}, \phi_h)$$

$$= \tau(u_H^* - 2u_H^k + u_H^{k-1}, \phi_h) + (f, \phi_h), \forall \phi_h \in X_h$$

6:     **Find** p_h^{k+1} **in** Y_h $(div \, \nabla p_h^{k+1}, \chi_h) = (\dfrac{div \, 3u_h^*}{2\Delta t}, \chi_h), \forall \chi_h \in Y_h$

7:     **Find** u_h^{k+1} **in** X_h $(u_h^{k+1} - u_h^* + \dfrac{2}{3}\Delta t \nabla p_h^{k+1}, \phi_h) = 0, \forall \phi_h \in X_h$

8:     **Solve in** X_H $(u_H^{k+1} - u_h^{k+1}, \psi_H) = 0, \forall \psi_H \in X_H$

9: **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 $\mathbb{P}_2/\mathbb{P}_1$ 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 preceding 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 Figure 5).

There is a relationship between $R = \dfrac{dim(X_H)}{dim(X_h)}$ (or $R = \dfrac{dim(Y_H)}{dim(Y_h)}$) and the gain in CPU time: the smaller R is, the faster but the less accurate the scheme is. We here concentrate on a factor 2 between the mesh sizes of X_H and that of X_h, say $H \simeq 2h$, then $R \simeq 0.25$. This simple situation is to be first considered before looking to optimal ratio R. It gives a simple framework for exploring the multilevel case in which the bi-grid scheme is recursively applied using a fine FEM space and a sequence of coarse FEM spaces. We mention the optimal relations between H and h, such as $h = O(H^2)$ proposed in [3] for NSE (with a comparable CPU ratio reduction as for our methods taking $H \simeq 2h$) and $H = O(h^{1/3})$ in [35] for Reaction-Diffusion equations.

3.1.1. Computation of steady states

These flow patterns in Figures 1, 2, 3, 4 fit with earlier results of Bruneau et al. [9], Ghia et al. [20], Goyon [23], Pascal [37] and Vanka [45]. Also, the numerical values and the localization of the extrema of the vorticity and of the stream function are good agreement with the ones given in these references.
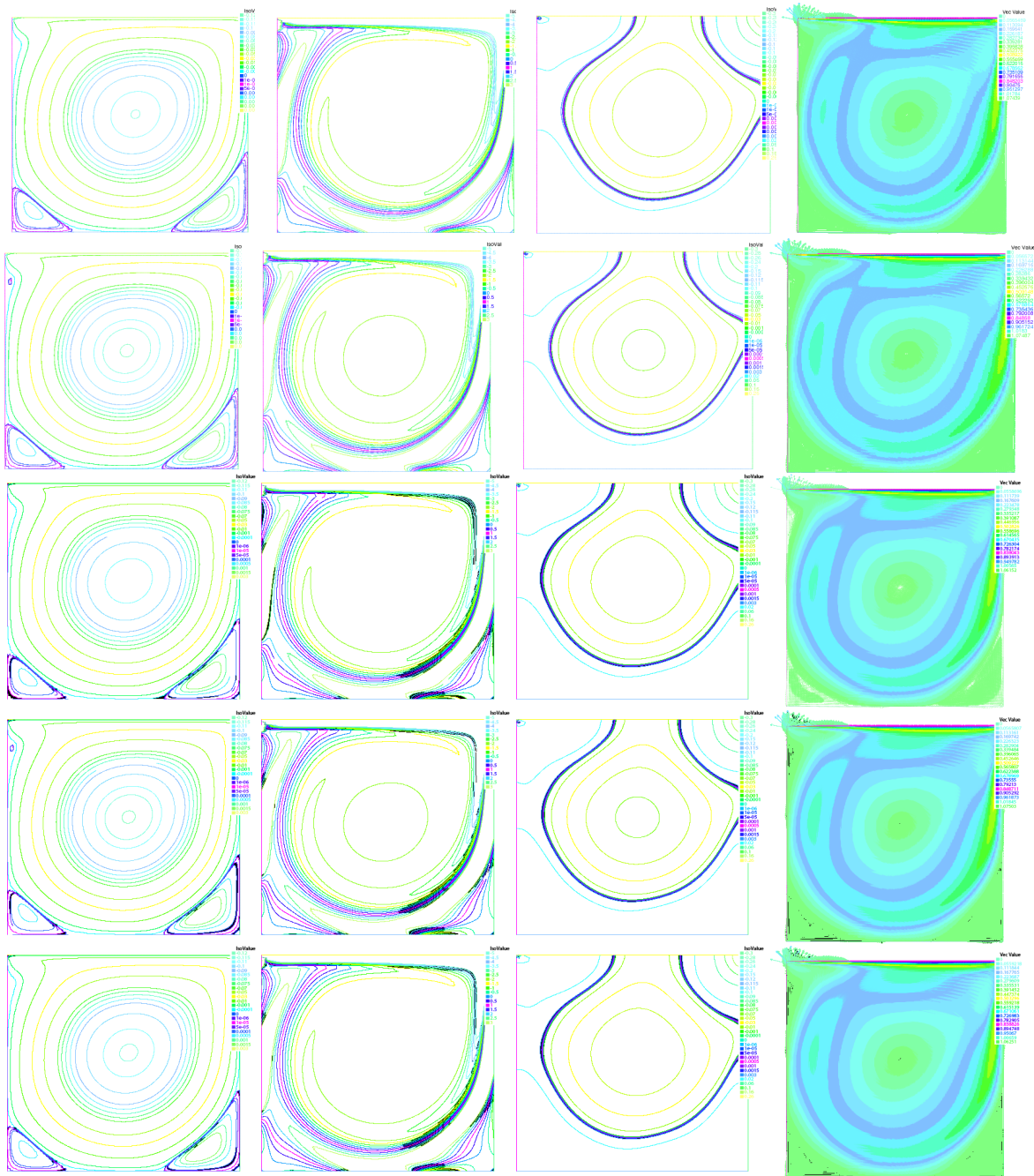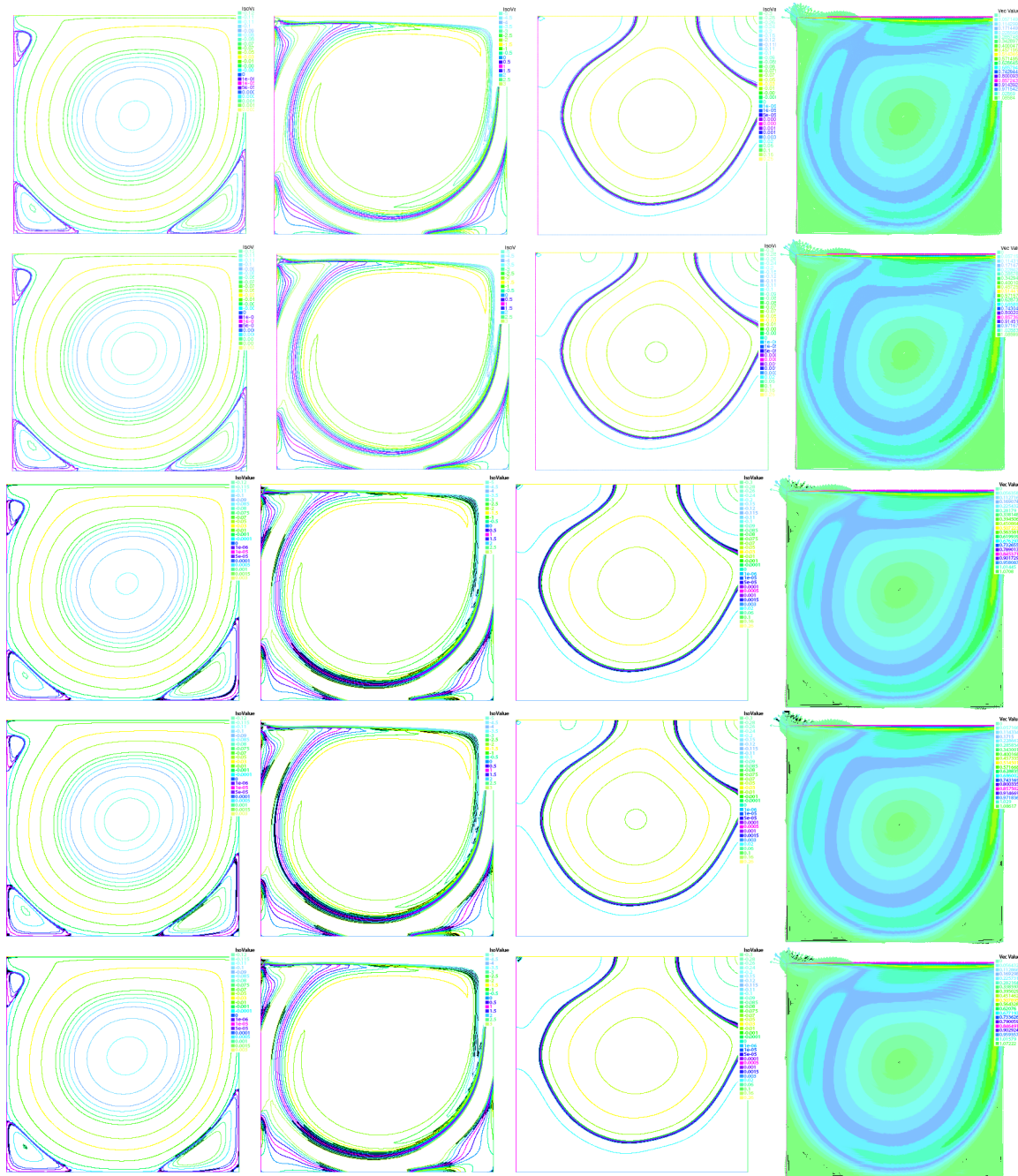

Figure 1. From top to bottom: Algorithms 5, 7, 8, 9 and 10, and from left to right : flow, vorticity, pressure and velocity for $\Delta t = 10^{-3}, Re = 2000, \tau = 0.5, T = 70, dim(X_H) = 6561, dim(Y_H) = 1681, dim(X_h) = 25921, dim(Y_h) = 6561$.
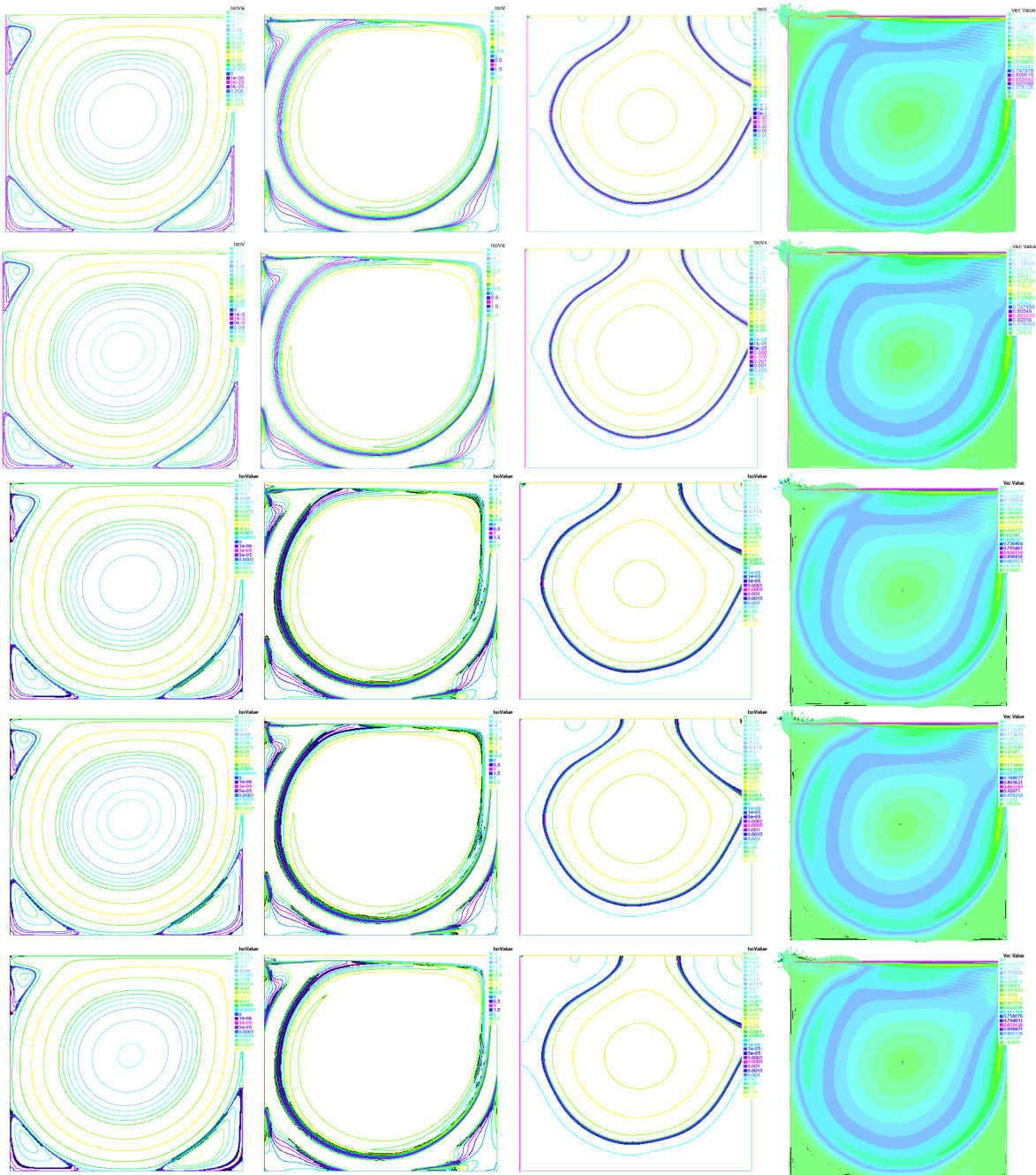
Figure 2. From top to bottom: Algorithms 5, 7, 8, 9 and 10, and from left to right : flow, vorticity, pressure and velocity for $\Delta t = 10^{-3}, Re = 3200, \tau = 0.5, T = 130, dim(X_H) = 6561, dim(Y_H) = 1681, dim(X_h) = 25921, dim(Y_h) = 6561$.

Figure 3. From top to bottom: Algorithms 5, 7, 8, 9 and 10, and from left to right : flow, vorticity, pressure and velocity for $\Delta t = 10^{-3}, Re = 5000, \tau = 0.5, T = 148, dim(X_H) = 6561, dim(Y_H) = 1681, dim(X_h) = 25921, dim(Y_h) = 6561$.
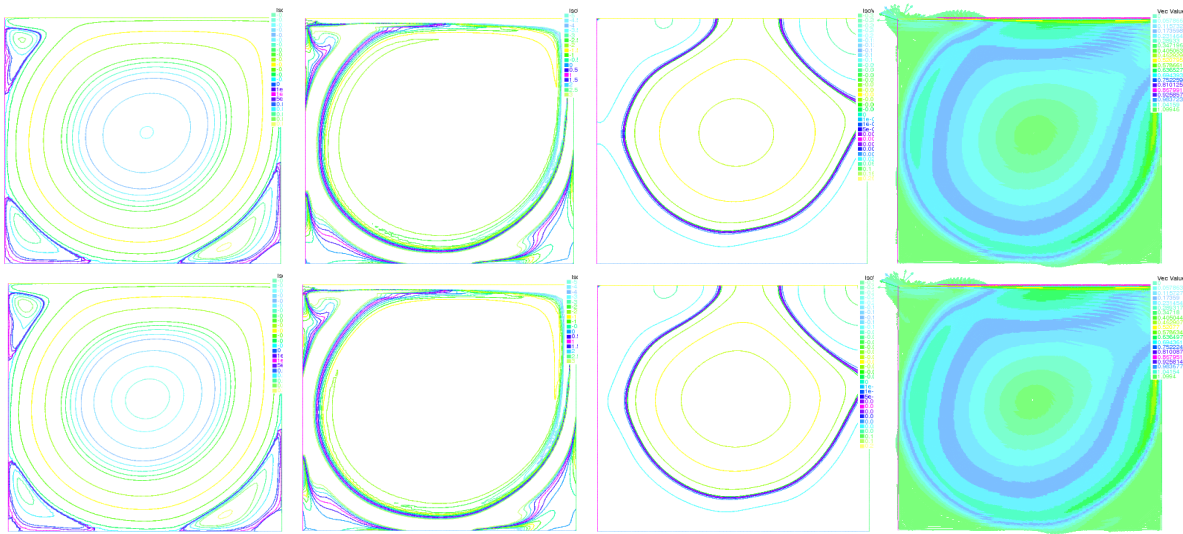
Figure 4. From top to bottom: Algorithms 5 and 7 for $Re = 7500, T = 195$ respectively and from left to right : flow, vorticity, pressure and velocity for $\Delta t = 5 \times 10^{-4}, \tau = 0.5, dim(X_H) = 6561, dim(Y_H) = 1681, dim(X_h) = 25921, dim(Y_h) = 6561.$
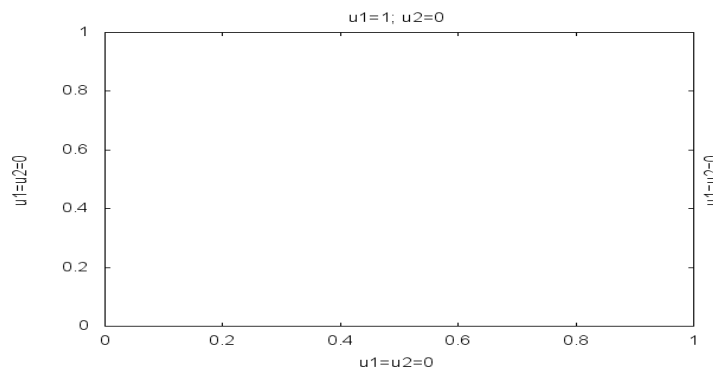

Figure 5. Boundary conditions

3.1.2. CPU time reduction

In Figures 6–7 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 time step in the neighborhood of the steady state: it means that only a linear system has then to be 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.

We denote by t_{1G} and t_{2G} respectively the computational time (CPU) of the resolution of the reference scheme on a single grid and the resolution by the bi-grid method. We compute the number of nonlinear iterations as a function of time and compute the L^2 norm of $\dfrac{\partial u}{\partial t}$; we use the

approximation $\dfrac{\partial u}{\partial t}|_{t=t_k} \simeq \dfrac{u^{k+1} - u^k}{\Delta t}$. We identify the time t^* from which the reference scheme (Algorithm 5) uses only one linear iteration at each time step and we define θ_s as the associated threshold value of $\|\dfrac{\partial u}{\partial t}\|_{L^2(\Omega)}$. 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 $\|\dfrac{\partial u}{\partial t}\| > \theta_s$ we apply the bi-grid scheme and as soon as $\|\dfrac{\partial u}{\partial t}\|_{L^2(\Omega)} \leq \theta_s$ we apply the reference scheme on the fine space V_h. We report in Table 1 (resp. Table 2) the CPU times for the four bi-grid algorithms (resp. on-grid scheme with Algorithm 7). We notice that the ratio $\dfrac{t_{2G}}{t_{1G}}$ is now equal to 0.7 instead of 0.9 when no threshold strategy was applied on the values of $\dfrac{\partial u}{\partial t}$.
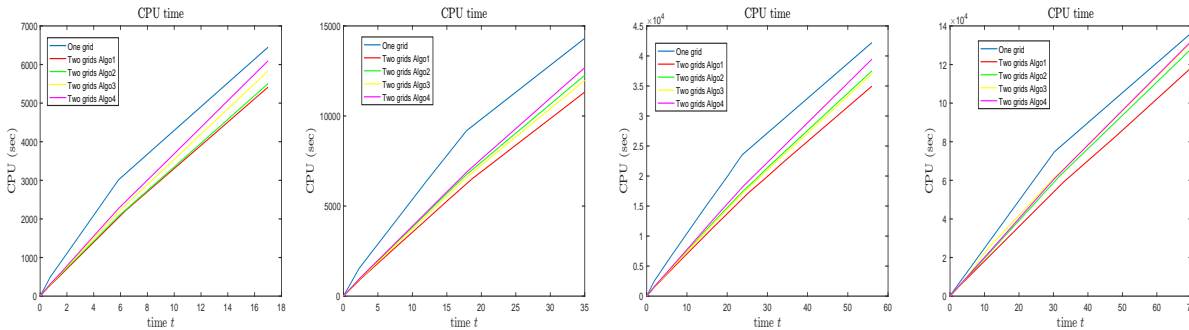

Figure 6. From left to right: CPU time of the scheme on a fine grid (Algorithm 5) and that of the scales separation methods for $Re = 100, 400, 1000, 2000$ respectively.
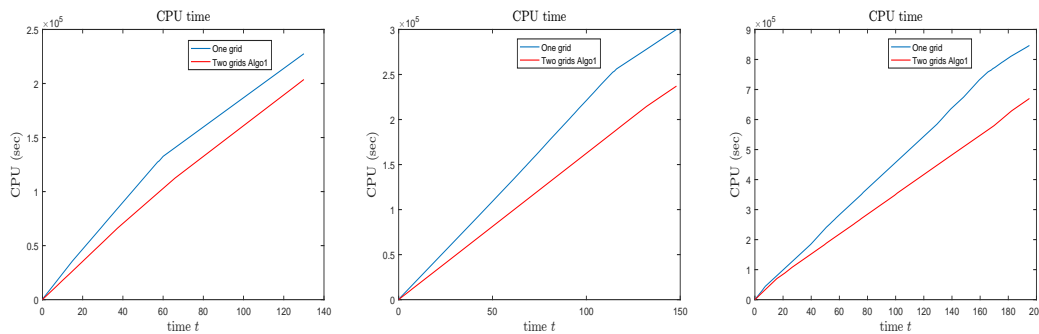

Figure 7. From left to right: CPU time of the scheme on a fine grid (Algorithm 5) and that of the scales separation method (Algorithm 7) for $Re = 3200, 5000, 7500$ respectively.

As for the first order algorithms, the second order bi-grid schemes are faster in computation time than the (second order) reference scheme (Algorithm 11). A gain in CPU is obtained see Figure 8 and Tables 3, 4, 5, 6 for $Re = 100, 400, 1000$ and 2000.

The relative performances of semi-implicit reference one grid and of the bi-grid schemes in stability time are the same as for the first order stabilized bi-grid algorithms: the high mode stabilization allows to take larger time steps Δt.

Table 1. Non incremental Navier-Stokes $Re = 2000$, $\Delta t = 10^{-3}$ and $\mathbb{P}_2/\mathbb{P}_1$.

80×40 points	Algorithm 7	Algorithm 8	Algorithm 9	Algorithm 10
t_{1G} at $T = 70$ (in sec.)	136366	136366	136366	136366
t_{2G} at $T = 70$ (in sec.)	118398	129836	131503	132104
$\dfrac{t_{2G}}{t_{1G}}$	0.868	0.952	0.964	0.969
θ_s	0.00521845	0.00521845	0.00521845	0.00521845
t_{1G} (in sec.)	74803.8	74803.8	74803.8	74803.8
t_{2G} (in sec.)	54363.8	60544.3	61401.9	60842.7
$\dfrac{t_{2G}}{t_{1G}}$	0.727	0.809	0.821	0.813

Table 2. Non incremental Navier-Stokes for the Algorithm 7 $Re = 2000, 3200, 5000, 7500$ and $\mathbb{P}_2/\mathbb{P}_1$

80×40 points	$Re = 2000$ $\Delta t = 10^{-3}$	$Re = 3200$ $\Delta t = 10^{-3}$	$Re = 5000$ $\Delta t = 10^{-3}$	$Re = 7500$ $\Delta t = 5 \times 10^{-4}$
Final time	$T = 70$	$T = 130$	$T = 148$	$T = 195$
t_{1G} at T(in sec.)	136366	227543	299998	846647
t_{2G} at T (in sec.)	118398	203692	237105	670244
$\dfrac{t_{2G}}{t_{1G}}$	0.868	0.8952	0.7904	0.791
θ_s	0.00521845	0.00502797	0.00321512	0.00462578
t_{1G} (in sec.)	74803.8	132604	256175	758398
t_{2G} (in sec.)	54363.8	102922	188600	565283
$\dfrac{t_{2G}}{t_{1G}}$	0.727	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 X_h) and which consists in treating the nonlinear term $(u^k \cdot \nabla)u^k$ explicitly. Particularly we compare the maximum time steps Δt that can be taken for each of the schemes and compare the respective final CPU time on the time interval $[0, T]$. We concentrate on moderate and High Reynolds numbers $Re = 2000, 3200, 5000$ and $Re = 7500$. We give hereafter in Tables 7, 8, 9 and 10 the results comparing the stability of the bi-grid Algorithms 7, 8, 9, 10 and the following semi-implicit scheme:
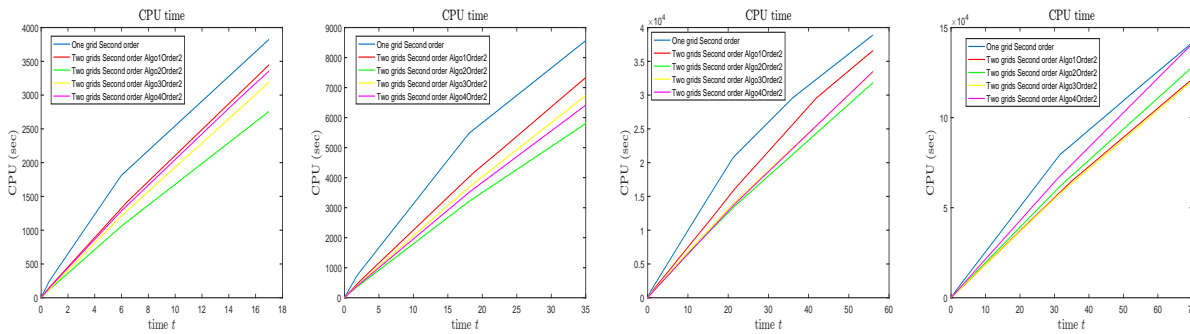
Figure 8. From left to right: CPU time of the scheme on a fine grid (Algorithm 11) and that of the scales separation method (Algorithms 12, 13, 14, 15) for $Re = 100, 400, 1000, 2000$ respectively.

Table 3. Non incremental second order Navier-Stokes $Re = 100$, $\Delta t = 10^{-2}$ and $\mathbb{P}_2/\mathbb{P}_1$.

80×40 points	Algorithm 12	Algorithm 13	Algorithm 14	Algorithm 15
t_{1G} at $T = 17$ (in sec.)	3825.49	3825.49	3825.49	3825.49
t_{2G} at $T = 17$ (in sec.)	3448.5	2758.86	3189.22	3357.44
$\dfrac{t_{2G}}{t_{1G}}$	0.901	0.721	0.834	0.878
θ_s	0.000538847	0.000538847	0.000538847	0.000538847
t_{1G} (in sec.)	1812.3	1812.3	1812.3	1812.3
t_{2G} (in sec.)	1327.1	1060.33	1211.73	1286.01
$\dfrac{t_{2G}}{t_{1G}}$	0.732	0.585	0.669	0.710

Table 4. Non incremental second order Navier-Stokes $Re = 400$, $\Delta t = 10^{-2}$ and $\mathbb{P}_2/\mathbb{P}_1$.

80×40 points	Algorithm 12	Algorithm 13	Algorithm 14	Algorithm 15
t_{1G} at $T = 35$ (in sec.)	8565.76	8565.76	8565.76	8565.76
t_{2G} at $T = 35$ (in sec.)	7334.26	5804.16	6731.14	6416.9
$\dfrac{t_{2G}}{t_{1G}}$	0.856	0.678	0.786	0.749
θ_s	0.000442771	0.000442771	0.000442771	0.000442771
t_{1G} (in sec.)	5479	5479	5479	5479
t_{2G} (in sec.)	4022.09	3209.97	3685.41	3508.57
$\dfrac{t_{2G}}{t_{1G}}$	0.734	0.586	0.673	0.640

Table 5. Non incremental second order Navier-Stokes $Re = 1000$, $\Delta t = 3 \times 10^{-3}$ and $\mathbb{P}_2/\mathbb{P}_1$.

80×40 points	Algorithm 12	Algorithm 13	Algorithm 14	Algorithm 15
t_{1G} at $T = 56$ (in sec.)	38900.4	38900.4	38900.4	38900.4
t_{2G} at $T = 56$ (in sec.)	36594.2	31834.6	33335.9	33493.8
$\dfrac{t_{2G}}{t_{1G}}$	0.941	0.818	0.857	0.861
θ_s	0.00147126	0.00147126	0.00147126	0.00147126
t_{1G} (in sec.)	20786.7	20786.7	20786.7	20786.7
t_{2G} (in sec.)	15856.7	13376.5	13917.3	13661
$\dfrac{t_{2G}}{t_{1G}}$	0.763	0.644	0.670	0.657

Table 6. Non incremental second order Navier-Stokes $Re = 2000$, $\Delta t = 10^{-3}$ and $\mathbb{P}_2/\mathbb{P}_1$.

80×40 points	Algorithm 12	Algorithm 13	Algorithm 14	Algorithm 15
t_{1G} at $T = 70$ (in sec.)	141635	141635	141635	141635
t_{2G} at $T = 70$ (in sec.)	121453	128157	120371	141042
$\dfrac{t_{2G}}{t_{1G}}$	0.858	0.905	0.850	0.996
θ_s	0.00514984	0.00514984	0.00514984	0.00514984
t_{1G} (in sec.)	79614.3	79614.3	79614.3	79614.3
t_{2G} (in sec.)	58606.9	61953.7	57919.8	67550.1
$\dfrac{t_{2G}}{t_{1G}}$	0.736	0.778	0.728	0.848

Algorithm 16 Non-incremental semi-implicit scheme

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

2:     **Find u_h^* in X_h** $(\dfrac{u_h^* - u_h^k}{\Delta t}, \psi_h) + \nu(\nabla u_h^*, \nabla \psi_h) + ((u_h^k \cdot \nabla)u_h^k, \psi_h) = (f, \psi_h), \forall \psi_h \in X_h$

3:     **Find p_h^{k+1} in Y_h** $(div \, \nabla p_h^{k+1}, \chi_h) = (\dfrac{div \, u_h^*}{\Delta t}, \chi_h), \forall \chi_h \in Y_h$

4:     **Find u_h^{k+1} in X_h** $(u_h^{k+1} - u_h^* + \Delta t \nabla p_h^{k+1}, \psi_h) = 0, \forall \psi_h \in X_h$

5: **end for**

We observe in Figure 9 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 τ. 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 Δt, we present in Table 12 a comparison between the maximum time step allowing the stability of the semi-implicit

Table 7. $Re = 2000, dim(X_H) = 6561, dim(Y_H) = 1681, dim(X_h) = 25921, dim(Y_h) = 6561, \mathbb{P}_2/\mathbb{P}_1$ elements.

Scheme	τ	Δt	Stability	Δt	Stability	τ	Δt	Stability
Algorithm 7	0.5	0.001	yes	0.003	no	140	0.005	yes
Algorithm 8	0.5	0.001	yes	0.003	yes	0.5	0.005	yes
Algorithm 9	0.5	0.001	yes	0.003	no	200	0.005	yes
Algorithm 10	0.5	0.001	yes	0.003	yes	0.5	0.005	yes
Algorithm 16		0.001	yes	0.007	no		0.005	no

Table 8. $Re = 3200, dim(X_H) = 6561, dim(Y_H) = 1681, dim(X_h) = 25921, dim(Y_h) = 6561, \mathbb{P}_2/\mathbb{P}_1$ elements.

Scheme	τ	Δt	Stability	Δt	Stability	τ	Δt	Stability
Algorithm 7	0.5	0.001	yes	0.003	no	200	0.005	yes
Algorithm 8	0.5	0.001	yes	0.003	yes	0.5	0.005	yes
Algorithm 9	0.5	0.001	yes	0.003	no	300	0.005	yes
Algorithm 10	0.5	0.001	yes	0.003	yes	0.5	0.005	yes
Algorithm 16		0.001	yes	0.003	no		0.005	no

Table 9. $Re = 5000, dim(X_H) = 6561, dim(Y_H) = 1681, dim(X_h) = 25921, dim(Y_h) = 6561, \mathbb{P}_2/\mathbb{P}_1$ elements.

Scheme	τ	Δt	Stability	Δt	Stability	τ	Δt	Stability
Algorithm 7	0.5	0.001	yes	0.003	no	300	0.005	yes
Algorithm 8	0.5	0.001	yes	0.003	yes	0.5	0.005	yes
Algorithm 9	0.5	0.001	yes	0.003	no	500	0.005	yes
Algorithm 10	0.5	0.001	yes	0.003	yes	0.5	0.005	yes
Algorithm 16		0.001	yes	0.003	no		0.005	no

Table 10. $Re = 7500, dim(X_H) = 6561, dim(Y_H) = 1681, dim(X_h) = 25921, dim(Y_h) = 6561, \mathbb{P}_2/\mathbb{P}_1$ elements.

Scheme	τ	Δt	Stability	Δt	Stability	τ	Δt	Stability
Algorithm 7	0.5	0.001	yes	0.003	no	400	0.005	yes
Algorithm 8	0.5	0.001	yes	0.003	yes	0.5	0.005	yes
Algorithm 9	0.5	0.001	yes	0.003	no	900	0.005	yes
Algorithm 10	0.5	0.001	yes	0.003	yes	0.5	0.005	yes
Algorithm 16		0.001	yes	0.003	no		0.005	no

scheme and our Algorithm 7 for the minimum value of τ necessary for stabilization.
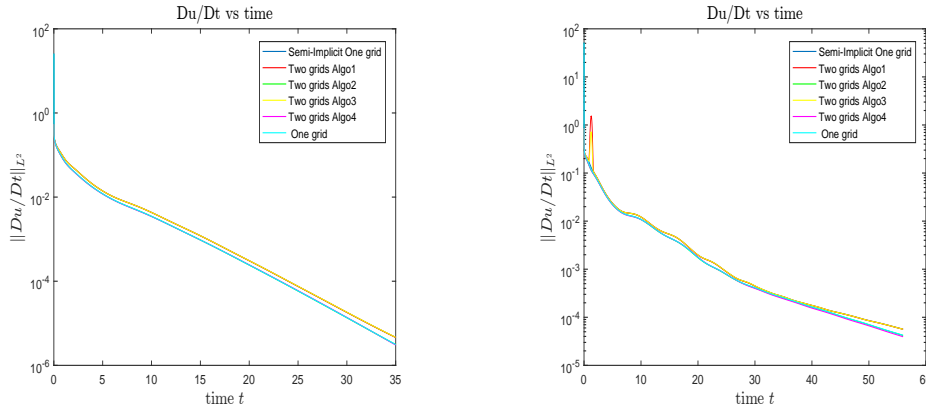

Figure 9. $\|\frac{\partial u}{\partial t}\|_{L^2(\Omega)}$ vs time: left for $\Delta t = 10^{-2}, Re = 400, T = 35, dim(X_H) = 6561, dim(Y_H) = 1681, dim(X_h) = 25921, dim(Y_h) = 6561$ and right for $\Delta t = 5 \times 10^{-3}, Re = 1000, T = 56$ and $dim(X_H) = 6561, dim(Y_H) = 1681, dim(X_h) = 25921, dim(Y_h) = 6561$.

Table 11. Non incremental Navier-Stokes Δt maximum and stability.

	Re=400		Re=1000	
Algorithm 16	$\Delta t = 10^{-2}$	$\Delta t = 10^{-2}$	$\Delta t = 5 \times 10^{-3}$	$\Delta t = 5 \times 10^{-3}$
Algorithm 7	$\Delta t = 0.1$	$\Delta t = 0.5$	$\Delta t = 0.1$	$\Delta t = 0.5$
	$\tau = 30$	$\tau = 30$	$\tau = 100$	$\tau = 100$
$\dfrac{Dt_{Algo7}}{Dt_{Algo16}}$	10	50	20	100

Table 12. Non incremental Navier-Stokes Δt maximum and stability.

	Re=2000		Re=3200		Re=5000	Re=7500
Algorithm 16	$\Delta t = 10^{-3}$	$\Delta t = 10^{-3}$	$\Delta t = 10^{-3}$	$\Delta t = 10^{-3}$	$\Delta t = 10^{-3}$	$\Delta t = 5 \times 10^{-4}$
Algorithm 7	$\Delta t = 0.01$	$\Delta t = 0.1$	$\Delta t = 0.01$	$\Delta t = 0.1$	$\Delta t = 0.01$	$\Delta t = 0.01$
	$\tau = 200$	$\tau = 300$	$\tau = 240$	$\tau = 400$	$\tau = 340$	$\tau = 500$
$\dfrac{Dt_{Algo7}}{Dt_{Algo16}}$	10	100	10	100	10	20

For small Reynolds number ($Re = 100$), we do not obtain an enhanced stability (larger Δt) as compared to the semi-implicit scheme. The latter is stable even for large values of the time step Δt. When considering larger values of Re, say $Re = 400$ to $Re = 7500$, the bi-grid scheme appears as 10 times to

100 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

$$(u_1, u_2) = (-2x^2 y(1-y)(1-2y)(1-x)^2 \exp^{\sin t}, 2y^2 x(1-x)(1-2x)(1-y)^2 \exp^{\sin t}),$$
$$p = (x-0.5)(y-0.5),$$

and with the corresponding term f. We can observe in Figures 10 and 11 that the solutions coincide with the reference one at final time $T = 56$ and that the L^2 norm of the error of the bi-grid schemes is of the order of Δt on all interval time.
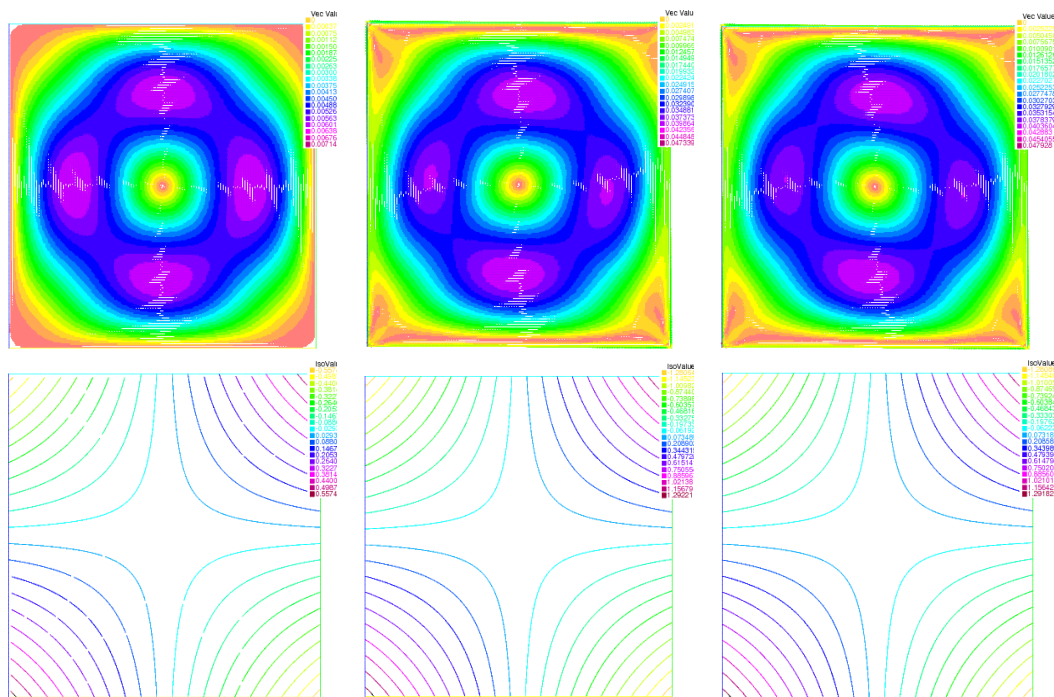
Figure 10. From top to bottom : velocity and pressure, and from left to right : the exact solution and Algorithms 5 and 7 solutions for $\Delta t = 5 \times 10^{-3}, Re = 1000, T = 56, dim(X_H) = 6561, dim(Y_H) = 1681, dim(X_h) = 25921, dim(Y_h) = 6561$.

4. Concluding remarks and further developments

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
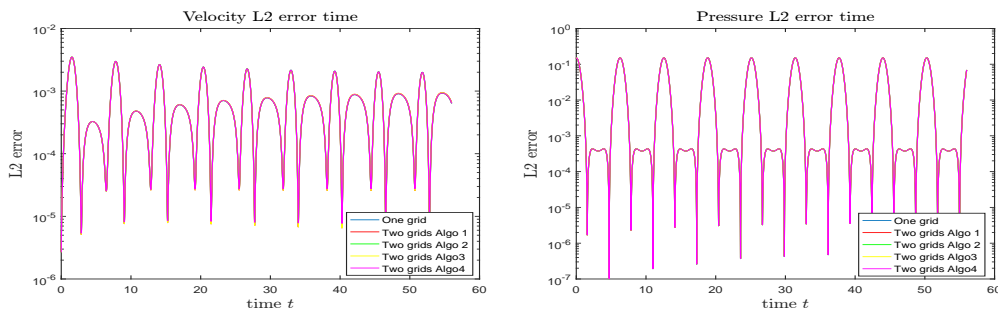
Figure 11. From left to right : the L^2−error for the velocity and the pressure between the exact solution and the solution of algorithms 5, 7, 8, 9 and 10, for $\Delta t = 5 \times 10^{-3}, Re = 1000, T = 56, dim(X_H) = 6561, dim(Y_H) = 1681, dim(X_h) = 25921, dim(Y_h) = 6561$.

as compared to semi-implicit projection scheme (larger time steps can be used). We choose to stabilize only the velocity: the stabilization we used on the high mode components is simple, directly derived from the technique introduced for reaction-diffusion equation in [1] and does not deteriorate the consistency.

The numerical results we obtained on the benchmark driven cavity, corresponding to laminar regimes, 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.

As further developments, the following topics could be considered

- Stabilization on the pressure. Of course other type of stabilization techniques can be considered, e.g. such as those applied to the pressure and related also to artificial compressibility approach [40]; we refer to the pressure stabilization method introduced by Brezzi and Pitkäranta [8] for the Stokes problem and to Olshanski and al. [36] for the Navier-Stokes equations. The mass conservation (divergence free condition) is modified as

$$\epsilon \frac{\partial p}{\partial t} + \nabla u = 0,$$

where $\epsilon > 0$ is a small positive real number. High mode stabilization on p can be obtained by solving the In such a context a high-mode stabilization on the pressure can be obtained as follows: First of all, solution of the pressure in Y_H:

$$(\nabla p_H^{k+1}, \nabla \psi_H) = -(\nabla u_H^*, \psi_H), \forall \psi_H \in Y_H,$$
$$(u_H^{k+1}, \phi_H) = (u_H^*, \phi_H) - \Delta t(\nabla p_H^{k+1}, \phi_H), \forall \phi_H \in X_H,$$

Then, we make the correction on the fine space Y_h

$$\tau(p_h^{k+1}, \psi_h) + (\nabla p_h^{k+1}, \nabla \psi_h) = \tau(p_h^k, \psi_h) + \tau(p_H^{k+1} - p_H^k, \psi_h) - (\nabla u_h^*, \psi_h), \forall \psi_h \in Y_h,$$
$$(u_h^{k+1}, \psi_H) = (u_h^*, \phi_h) - \Delta t(\nabla p_h^{k+1}, \phi_h), \forall \psi_h \in X_h.$$

- Adaptation to models of turbulence. Laminar regimes are interesting at first sight to validate the approach up to high Reynolds numbers ($Re = 7500$), even not in turbulence regime; this was the scope of our paper. An issue that can be considered is to apply the bi-grid approach to turbulence modeling, using a modified viscosity term. Smagorinsky's turbulence model can be considered. Numerical simulations, with different choices of the turbulent viscosity constant, can be compared to some other studies that consider this benchmark problem. In such cases the turbulence is modeled by introducing a turbulent viscosity v_T, e.g., as follows:

$$u_t + (u.\nabla)u - \nabla \left((v + v_T)\nabla u\right) + \nabla p = f \quad x \in \Omega, t \geq 0, \tag{9}$$

$$\nabla.u = 0 \qquad x \in \Omega, \tag{10}$$

$$u = 0 \qquad x \in \partial\Omega, \tag{11}$$

$$u(0, x) = u_0 \qquad x \in \Omega. \tag{12}$$

  Here $v > 0$ is a constant viscosity, $v_T = \beta\delta|u - \bar{u}|$, where $\bar{u}$ is a local average of u, $\beta > 0$, $\delta > 0$. This model of Smagorinsky's type is considered, e.g., in [32]. In a more recent work [31], Layton has considered

$$v + v_T = (C_s\delta)^2|\nabla u|.$$

  We refer also to [16].

- 3D flows. We have considered the two-dimensional case. The application of the bi-grid approach should be applied as an issue to the 3D case. A larger gain in CPU time can be expected since for $H \simeq 2h$ we have $R = \dfrac{dim(X_H)}{dim(X_h)} \simeq 0.25$ in 2D while $R \simeq 0.125$ in 3D. Notice that when looking to NSE equations, different behavior of the small eddies (high frequency components) can occur in space dimension 2 and space dimension 3, see [16] and the references therein.

Acknowledgment

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*, accepted to be published in Computational and Applied Mathematics, 2018.

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., **114** (2009), 189–231.

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., **42** (2008), 141–174.

4. I. Babuška, *The finite element method with Lagrangian multipliers*, Numer. Math., **20** (1973), 179–192.

5. M. Bercovier and M. Engelman, *A finite element for the numerical solution of viscous incompressible flows*, J. Comput. Phys., **30** (1979), 181–201.

6. A. Bousquet, M. Marion, M. Petcu, et al. *Multilevel finite volume methods and boundary conditions for geophysical flows*, Comput. 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., **8** (1974), 129–151.

8. F. Brezzi, J. Pitkaranta, *On the Stabilization Finite Elements Approximation of the Stokes Equation*, In: Efficient Solution of Elliptic Problems, Proceedings of a GAMM-Seminar, Kiel, (1984), 11–19.

9. C. H. Bruneau and C. Jouron, *An efficient scheme for solving steady incompressible Navier-Stokes equations*, J. Comput. Phys, **89** (1990), 389–413.

10. 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), 403–442.

11. C. Calgaro, A. Debussche, J. Laminie, *On a multilevel approach for the two-dimensional Navier-Stokes equations with finite elements*, Int. J. Numer. Meth. Fl., **27** (1998), 241–258.

12. C. Calgaro, J.-P. Chehab, J. Laminie, et al. *Schémas multiniveaux pour les équations d'ondes*, (French) [Multilevel schemes for waves equations], ESAIM Proc., **27** (2009), 180–208.

13. J.-P. Chehab, B. Costa, *Time explicit schemes and spatial finite differences splittings*, J. Sci. Comput., **20** (2004), 159–189.

14. B. Costa. L. Dettori, D. Gottlieb, et al. *Time Marching Multilevel Techniques for Evolutionary Dissipative Problems*, SIAM J. Sci. comput., **23** (2001), 46–65.

15. A. J. Chorin, *Numerical solution of the Navier-Stokes equations*. Math. Comput., **22** (1968), 745–762.

16. T. Dubois, F. Jauberteau, R. Temam, *Dynamic multilevel methods and the numerical simulation of homogeneous and non homogeneous turbulence*, Cambridge Academic Press.

17. T. Dubois, F. Jauberteau, R. Temam, et al. *Multilevel schemes for the shallow water equations* J. Comput. Phys., **207** (2005), 660–694.

18. S. Faure, J. Laminie, R. Temam, *Finite Volume Discretization and Multilevel Methods in Flow Problems*, J. Sci. Comput., **25** (2005), 231–261.

19. *FreeFem++* . Available from:
http://www.freefem.org.

20. 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** (1982), 387–411.

21. V. Girault and J.-L. Lions, *Two-grid finite-element schemes for the transient Navier-Stokes equations*, ESAIM: Mathematical Modelling and Numerical Analysis, **35** (2001), 945–980.

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

23. O. Goyon, *High-Reynolds number solutions of Navier-Stokes equations using incremental unknowns*. Comput. Method. Appl. M., **130** (1996), 319–335.

24. J. L. Guermond, P. Minev and J. Shen, *An overview of projection methods for incompressible flows*, Comput. Method. Appl. M., **195** (2006), 6011–6045.

25. J. L. Guermond and J. Shen, *Velocity-correction projection methods for incompressible flows*, SIAM J. Numer. Anal., **41** (2003), 112–134.

26. J. L. Guermond and J. Shen, *Quelques résultats nouveaux sur les méthodes de projection*. Comptes Rendus de l'Académie des Sciences-Series I-Mathematics, **333** (2001), 1111–1116.

27. W. Hackbusch, *Multi-grid methods and applications*, Berlin, Springer, 1985.

28. Y. He and K. M. Liu, *Multi-level spectral Galerkin method for the NavierStokes equations, II: time discretization*, Adv. Comput. Math., **25** (2006), 403–433.

29. 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. Comput. Phys., **270** (2014), 506–531.

30. W. Layton, *A two-level discretization method for the Navier-Stokes equations*. Comput. Math. Appl., **26** (1993), 33–38.

31. W. Layton, *Energy Dissipation in the Smagorinsky Model of Turbulence*, Appl. Math. Lett., **59** (2016), 56–59.

32. W. Layton, R. Lewandowski, *Analysis of an Eddy Viscosity Model for Large Eddy Simulation of Turbulent Flows*, J. math. Fluid Mech., **4** (2002), 374–399.

33. M. Marion and R. Temam, *Nonlinear Galerkin Methods*, SIAM J. Numer. Anal., **26** (1989), 1139–1157.

34. M. Marion and R. Temam, *Nonlinear Galerkin Methods: The Finite elements case*, Numer. Math., **57** (1990), 205–226.

35. M. Marion and J. Xu, *Error estimates on a new nonlinear Galerkin method based on two-grid finite elements*, SIAM J. Numer. Anal., **32** (1995), 1170–1184.

36. M. Olshanskii, G. Lube, T. Heister, et al. *Grad-div stabilization and subgrid pressure models for the incompressible Navier-Stokes equations*, Comput. Method. Appl. M., **198** (2009), 3975–3988.

37. 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.

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

39. J. Shen, X. Yang, *Numerical Approximations of Allen-Cahn and Cahn-Hilliard Equations*. Discrete Cont. Dyn-A, **28** (2010), 1669–1691.

40. R. Temam, *Navier-Stokes equations*, Revised version, North-Holland, Amsterdam, 1984.

41. R. Temam, *Navier-Stokes equations, Theory and numerical analysis*, North-Holland, Amsterdam, 1977.

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

43. R. Temam, *Une méthode d'approximation de la solution des équations de Navier-Stokes* , B. Soc. Math. Fr., **98** (1968), 115–152.

44. L. J. P. Timmermans, P. D. Minev, F. N. Van De Vosse, *An approximate projection scheme for incompressible flow using spectral elements* , Int. J. Numer. Meth. Fl., **22** (1996), 673–688.

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

46. J. Xu, *Some Two-Grid Finite Element Methods*, Tech. Report, P.S.U, 1992.

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

48. J. Xu, *Two-grid discretization techniques for linear and nonlinear PDE*, SIAM J. Numer. Anal., **33** (1996), 1759–1777.

49. H. Yserentant, *On the multi-level splitting of finite element spaces*, Numer. Math, **49** (1986), 379–412.