



Lagrange–Galerkin methods for the incompressible Navier–Stokes equations: a review

Rodolfo Bermejo^{1*}, Laura Saavedra^{2,1}

¹Departamento de Matemática Aplicada a la Ingeniería Industrial ETSII ,
Universidad Politecnica de Madrid, Spain

²Departamento de Matemática Aplicada a la Ingeniería Aeroespacial ETSAA,
Universidad Politecnica de Madrid, Spain

*Email address for correspondence: rbermejo@etsii.upm.es

Communicated by Roberto Ferretti

Received on May 10, 2015. Accepted on November 9, 2015.

Abstract

We review in this paper the development of Lagrange-Galerkin (LG) methods to integrate the incompressible Navier-Stokes equations (NSEs) for engineering applications. These methods were introduced in the computational fluid dynamics community in the early eighties of the past century, and at that time they were considered good methods for both their theoretical stability properties and the way of dealing with the nonlinear terms of the equations; however, the numerical experience gained with the application of LG methods to different problems has identified drawbacks of them, such as the calculation of specific integrals that arise in their formulation and the calculation of the flow trajectories, which somehow have hampered the applicability of LG methods. In this paper, we focus on these issues and summarize the convergence results of LG methods; furthermore, we shall briefly introduce a new stabilized LG method suitable for high Reynolds numbers.

Keywords: Lagrange–Galerkin, finite elements, Navier–Stokes.

AMS subject classification: 65M12, 65M25, 65M60.

1. Introduction

LG methods, also known as Semi-Lagrangian (SL) methods in the Numerical Weather Prediction community, are efficient numerical techniques to integrate time dependent convection-diffusion problems, including the incompressible NSEs. The distinctive feature of both LG and SL methods is the way they deal with the material derivative. They discretize the material derivative backward in time along flow trajectories, but LG methods calculate the quantities at the feet of the trajectories by a Galerkin projection onto a suitable finite dimensional space, generally a finite element space, whereas SL methods use polynomial interpolation of order higher than one; so, LG can be applied with any type of mesh in contrast with SL

methods that are mostly applied with structured quadrilateral meshes. In this paper, we review the development of LG methods to integrate NSEs in an engineering context in which the use of unstructured meshes is almost mandatory. LG methods were introduced in [10] and [23]. The application of LG methods to integrate NSEs has some advantages, such as numerical stability and the way of dealing with the nonlinear terms. It is known that in the integration of NSEs by conventional implicit time marching schemes, the nonlinear terms yield an algebraic system of nonlinear equations that is solved by an iterative procedure, increasing thus the number of arithmetic operations to reach the solution and requiring a large space of memory. In contrast, backward integration of the material derivative along trajectories, which is a natural way of introducing upwinding in the space discretization of the equations, transforms the NSEs into a linear Stokes problem, so at each time step one has to solve an algebraic linear system of equations that is more manageable than the algebraic nonlinear system of equations produced by conventional implicit time marching schemes; furthermore, we must remark that upwinding along the trajectories is a numerical mechanism to stabilize the convective terms. A priori, these assets make LG methods look like efficient methods to integrate NSEs ; however, they have a drawback concerned with the calculation of the integrals of the form $\int_K \phi_j(X_h(x, t_{n+1}, t_n))\phi_i(x)dx$, which appear in the formulation of the numerical solution, here K is a generic element, ϕ_i is the i th global basis function of the finite element space and $X_h(x, t_{n+1}, t_n)$ is the foot of the characteristic associated with the point x . For stability and optimal convergence reasons these integrals have to be calculated with high accuracy, see [20] and [6], thus requiring the use of high order quadrature rules. Since each quadrature point has an associated foot of characteristic, this means that many systems of differential equations have to be solved backward in time by a numerical method incorporating a point searching algorithm to identify the element of the mesh where each foot is located. The location of points inside the elements of a mesh is a trivial task in structured meshes, for instance, in meshes composed of squares or hexahedra, but if the mesh is unstructured the location of points is not that simple; hence, LG methods may become less efficient than they look at first. To partially overcome these drawbacks, some variations of conventional LG method, such as the area-weighting method for quadrilateral structured meshes [20], exact integration [22] for straight side triangular meshes with linear elements, and the modified LG methods [5,6], have been proposed. We do not consider such variations in this paper.

We introduce some notation about the functional spaces we use in the paper. For $s \geq 0$ real and real $1 \leq p \leq \infty$, $W^{s,p}(\Omega)$ denotes the real Sobolev

LG methods for Navier-Stokes Equations

spaces defined on $\Omega \subset \mathbb{R}^d$ for scalar real-valued functions. $\|\cdot\|_{W^{s,p}(\Omega)}$ and $|\cdot|_{W^{s,p}(\Omega)}$ denote the norm and semi-norm, respectively, of $W^{s,p}(\Omega)$. When $s = 0$, $W^{0,p}(\Omega) := L^p(\Omega)$. For $p = 2$, the spaces $W^{s,2}(\Omega)$ are denoted by $H^s(\Omega)$, which are real Hilbert spaces with inner product $(\cdot, \cdot)_s$. For $s = 0$, $H^0(\Omega) := L^2(\Omega)$, the inner product in $L^2(\Omega)$ is denoted by (\cdot, \cdot) . $H_0^1(\Omega)$ is the space of functions of $H^1(\Omega)$ which vanish on the boundary $\partial\Omega$ in the sense of trace. H^{-1} denotes the dual of $H_0^1(\Omega)$. The corresponding spaces of real vector (and tensor)-valued functions, $v : \Omega \rightarrow \mathbb{R}^d$ are denoted by boldface letters; for instance, $\mathbf{W}^{s,p}(\Omega) := (W^{s,p}(\Omega))^d := \{v : \Omega \rightarrow \mathbb{R}^d : v_i \in W^{s,p}(\Omega), 1 \leq i \leq d\}$. Let X be a real Banach space $(X, \|\cdot\|_X)$, if $v : (0, T) \rightarrow X$ is a strongly measurable function with values in X , we denote by $L^p(X)$, $H^s(X)$ and $C(X)$ the spaces $L^p(0, T; X)$, $H^s(0, T; X)$ and $C([0, T]; X)$ respectively. $C^{r,1}(\bar{\Omega})$, $r \geq 0$, is the space of functions defined in the closure of Ω , r times differentiable and with the r th derivative being Lipschitz continuous.

The layout of the paper is as follows. In Section 2, we introduce the semidiscrete Lagrange formulation of the incompressible NSEs to motivate the formulation of LG methods. In Section 3, a detailed description of first and second order in time conventional LG methods is presented, paying attention to important issues such as the efficient implementation of the methods and the calculation of the feet of the characteristic curves. Section 4 is devoted to the presentation of the error analysis, whereas in Section 5 we illustrate the performance of conventional LG-BDF methods in some three dimensional benchmark problems. In Section 6, we present the implementation of LG methods in the framework of projection methods. Finally, some new developments of LG methods for high Reynolds numbers are introduced in Section 7.

Throughout this paper, C will denote a generic positive constant which is independent of h and Δt . C will have different values at different places of appearance.

2. Semidiscrete Lagrangian formulations of Navier-Stokes equations

Let $\Omega \subset \mathbb{R}^d$ ($d = 2$ or 3) be a bounded domain with Lipschitz boundary Γ and let $[0, T]$ denote a time interval. We consider the following Cauchy problem:

Find the functions $v : \Omega \times (0, T] \rightarrow \mathbb{R}^d$ and $p : \Omega \times (0, T] \rightarrow \mathbb{R}$ that

satisfy

$$(1) \quad \begin{cases} \frac{\partial v}{\partial t} + (v \cdot \nabla)v - \nu \Delta v + \nabla p = f, & \text{in } \Omega \times (0, T), \\ \operatorname{div} v = 0, & \text{in } \Omega \times (0, T), \end{cases}$$

and the following initial and boundary conditions

$$(2) \quad v(\cdot, 0) = v^0(\cdot) \text{ in } \Omega,$$

$$(3) \quad v = 0 \text{ on } \Gamma \times (0, T).$$

Physically, system (1) describes the unsteady flow of a constant density Newtonian fluid with constant kinematic viscosity $\nu > 0$. v , p , and $f : \bar{\Omega} \times (0, T) \rightarrow \mathbb{R}^d$ denote the flow velocity, the pressure and the density of body forces per unit of mass, respectively. A divergence free velocity $v^0 : \Omega \rightarrow \mathbb{R}^d$ is prescribed at the initial instant $t = 0$. Recalling the expression of the material derivative, $\frac{Dv}{Dt} := \frac{\partial v}{\partial t} + v \cdot \nabla v$, we calculate a numerical solution using the following weak formulation:

Given $f \in L^2(\mathbf{H}^{-1})$ and $v^0 \in \mathbf{H}$, find $v \in L^2(\mathbf{H}_0^1(\Omega)) \cap L^\infty(\mathbf{L}^2(\Omega))$ and $p \in L^2(L_0^2(\Omega))$, such that for all $u \in \mathbf{H}_0^1(\Omega)$ and $q \in L_0^2(\Omega)$:

$$(4) \quad \begin{cases} \left(\frac{Dv}{Dt}, u \right) + \nu(\nabla v, \nabla u) - (p, \operatorname{div} u) = (f, u), \\ (\operatorname{div} v, q) = 0, \end{cases}$$

where $\mathbf{H} := \{v \in \mathbf{L}^2(\Omega) : \operatorname{div} v = 0 \text{ and } \mathbf{n} \cdot v|_\Gamma = 0\}$, \mathbf{H}^{-1} is the dual space of $\mathbf{H}_0^1(\Omega)$, $L_0^2(\Omega) := \{q \in L^2(\Omega) : \int_\Omega q dx = 0\}$, and \mathbf{n} is the unit outward normal vector.

In this paper, we focus on the approximation of equations (1)-(3) based on the backward in time discretization along the characteristic curves $X(x, s; t)$ of the operator $\frac{D}{Dt} = \frac{d}{dt} + v \cdot \nabla$, which are solution of the initial value problem

$$(5) \quad \frac{dX(x, s; t)}{dt} = v(X(x, s; t), t), \quad X(x, s; s) = x.$$

$t \rightarrow X(x, s; t)$ can be viewed as the trajectory of a fluid particle that at time s is at the point x . It is well known that if $v \in C^0(\mathbf{C}^{0,1}(\bar{\Omega}))$, the solution $X(x, s; t)$ is unique and can be represented by the integral form

$$(6) \quad X(x, s; t) = x + \int_s^t v(X(x, s; \tau), \tau) d\tau.$$

LG methods for Navier-Stokes Equations

The mapping $x \rightarrow X(x, s; t)$ has the group property; i.e., let t_1 and $t_2 \in [0, T]$, then $X(x, s; t_2) = X(\cdot, t_1; t_2) \circ X(x, s; t_1)$. Hereafter, unless otherwise stated, we adopt the notation $X^{k,l}(x) := X(x, t_l; t_k)$, k and l being positive integers. The following results are well known.

Lemma 2.1. *Assume that $v \in C^0(\mathbf{C}^{0,1}(\bar{\Omega}))$ and $s - \tau$ is sufficiently small, then $x \rightarrow X(x, s; t)$ is a quasi-isometric homeomorphism of Ω onto Ω and its Jacobian determinant $J = 1$ a.e. in Ω . Moreover,*

$$K_1 |x - y| \leq |X(x, s; t) - X(y, s; t)| \leq K_2 |x - y|,$$

where $K_2 = \exp(|s - t| \cdot \|\nabla \mathbf{u}\|_{L^\infty(\mathbf{L}^\infty(D)^d)})$ and $K_1 = (1 - |s - t| \cdot \|\nabla \mathbf{u}\|_{L^\infty(\mathbf{L}^\infty(D)^d)})^{-1}$, and $|a - b|$ denotes the Euclidean distance between the points a and $b \in \mathbb{R}^d$.

For a proof of this lemma see [27]. In the following lemma we collect some facts concerning the solution to (5) which are standard in the theory of ODE systems.

Lemma 2.2. *Assume that $v \in C^0(\mathbf{C}^{k-1,1}(\bar{\Omega}))$, $k \geq 1$. Then for any integer n , $0 \leq n \leq N - 1$, the unique solution $t \rightarrow X(x, t_{n+1}; t)$, ($t \in [t_n, t_{n+1}] \subset [0, T]$) to (5) is such that $X(x, t_{n+1}; t) \in C^{0,1}(\mathbf{C}^{k-1,1}(\bar{\Omega}))$. Furthermore, let the multi-index $\alpha \in \mathbb{N}^d$, then for all α , $1 \leq |\alpha| \leq k$, it follows that $\partial_{x_j}^{|\alpha|} X_i(x, t_{n+1}; t) \in C([0, T]; \mathbf{L}^\infty(\Omega) \times [0, T])$, $1 \leq i, j \leq d$.*

To motivate the introduction of LG methods, we shall derive the Lagrangian formulation of the NSEs. Let $s, t \in [0, T]$ and $x = X(y, s; t)$, $x, y \in \Omega$, then it follows that $\frac{dX}{dt} = v(x, t)$. We assume that $y \rightarrow X(y, s; t)$ is a diffeomorphism with Jacobian matrix $\mathbf{F}(y, s; t) := \left(\frac{\partial X(y, s; t)}{\partial y} \right)$ the determinant of which is denoted by $J(y, s; t)$. Setting $c(x, t) = c(X(y, s; t), t) = \bar{c}(y, t)$, where c is either a scalar- or a vector-valued function, we can easily compute

$$\begin{aligned} \frac{\partial \bar{c}}{\partial t} &= \frac{\partial c}{\partial t} + v \cdot \nabla c = \frac{Dc}{Dt}, \quad \nabla c = \mathbf{F}^{-T} \nabla \bar{c}, \\ \nu \Delta c &= \nu \operatorname{div}(\nabla c) = \frac{\nu}{J} \operatorname{div}(J \mathbf{F}^{-1} \mathbf{I} \mathbf{F}^{-T} \nabla \bar{c}), \end{aligned}$$

where \mathbf{I} is the unit matrix and \mathbf{F}^{-T} is the transpose of \mathbf{F}^{-1} . Let $I_n :=$

$[t_n, t_{n+1}]$, then the NSEs can be recast in $\Omega \times I_n$ as

$$(7) \quad \begin{cases} \frac{\partial \bar{v}}{\partial t} + \mathbf{F}^{-T} \nabla \bar{p} - \frac{\nu}{J} \operatorname{div}(J \mathbf{F}^{-1} \mathbf{I} \mathbf{F}^{-T} \nabla \bar{v}) - \bar{f} = 0, \\ \frac{1}{J} \operatorname{div}(J \mathbf{F}^{-1} \bar{v}) = 0, \\ \bar{v}|_{\Gamma} = 0, \\ \bar{v}(y, t_n) \text{ known.} \end{cases}$$

Letting

$$\bar{G}(y, t) = -\mathbf{F}^{-T} \nabla \bar{p} + \frac{\nu}{J} \operatorname{div}(J \mathbf{F}^{-1} \mathbf{I} \mathbf{F}^{-T} \nabla \bar{v}) + \bar{f},$$

and taking as initial time $s = t_{n+1}$, the application of the Backward Differentiation Formula of order 1 (BDF1) to discretize in time (7) yields the semidiscrete system

$$\begin{cases} \frac{\bar{v}(y, t_{n+1}) - \bar{v}(y, t_n)}{\Delta t} = \bar{G}(y, t_{n+1}) + O(\Delta t), \\ \frac{1}{J} \operatorname{div}(J \mathbf{F}^{-1} \bar{v}(y, t_{n+1})) = 0, \\ \bar{v}(y, t_{n+1})|_{\Gamma} = 0. \end{cases}$$

Noting that for $t = t_{n+1}$, $\bar{v}(y, t_{n+1}) = v(x, t_{n+1})$, $\mathbf{F}(y, t_{n+1}; t_{n+1}) = \mathbf{I}$, $J(y, t_{n+1}; t_{n+1}) = 1$, and

$$\bar{G}(y, t_{n+1}) = -\nabla p(x, t_{n+1}) + \nu \Delta v(x, t_{n+1}) + f(x, t_{n+1}),$$

and recalling the notation $X^{n,n+1}(x) = X(x, t_{n+1}; t_n)$ so that $\bar{v}(y, t_n) = v(X^{n,n+1}(x), t_n) = v(\cdot, t_n) \circ X^{n,n+1}$, we obtain the semidiscrete Lagrangian scheme

$$(8) \quad \begin{cases} \frac{v(x, t_{n+1}) - v(\cdot, t_n) \circ X^{n,n+1}}{\Delta t} + \nabla p(x, t_{n+1}) - \nu \Delta v(x, t_{n+1}) = f(x, t_{n+1}), \\ \operatorname{div} v(x, t_{n+1}) = 0, \\ v(x, t_{n+1})|_{\Gamma} = 0. \end{cases}$$

The semidiscrete Lagrangian scheme ((8)), which was proposed by Pironneau [23] in combination with finite elements, yields an approximation to the weak solution (v, p) of order $O(\Delta t)$.

LG methods for Navier-Stokes Equations

Applying the trapezoidal rule to integrate (7) yields the second order semidiscrete scheme

$$\left\{ \begin{array}{l} \frac{v(x, t_{n+1}) - v(\cdot, t_n) \circ X^{n, n+1}}{\Delta t} + \frac{1}{2} (p(x, t_{n+1}) + \mathbf{F}^{-T} \nabla_{X^{n, n+1}} p(\cdot, t_n) \circ X^{n, n+1}) \\ = \frac{1}{2} \nu \Delta v(x, t_{n+1}) + \frac{\nu}{2J} \operatorname{div} (J \mathbf{F}^{-1} \mathbf{I} \mathbf{F}^{-T} \nabla_{X^{n, n+1}} v(\cdot, t_n) \circ X^{n, n+1}(x)) \\ + \frac{1}{2} (f(x, t_{n+1}) + f(\cdot, t_n) \circ X^{n, n+1}), \\ \operatorname{div} v(x, t_{n+1}) = 0, \\ v(x, t_{n+1})|_{\partial D} = 0, \end{array} \right.$$

here, $J := J(x, t_{n+1}; t_n)$ and $\mathbf{F}^{-T} = \mathbf{F}^{-T}(x, t_{n+1}; t_n)$. A variant of this scheme combined with finite elements has been studied in [21]. We can also derive a second order scheme by applying the Backward Differentiation Formula of order 2 (BDF2) for the time discretization of (7); thus, setting

$$d_t g(y, t_{n+1}) := \frac{3g(y, t_{n+1}) - 4g(y, t_n) + g(y, t_{n-1})}{2\Delta t},$$

it follows the second order semidiscrete Lagrangian scheme:

$$\left\{ \begin{array}{l} d_t \bar{v}(y, t_{n+1}) = \bar{G}(y, t_{n+1}), \\ \frac{1}{J} \operatorname{div} (J \mathbf{F}^{-1} \bar{v}(y, t_{n+1})) = 0, \\ \bar{v}(y, t_{n+1})|_{\partial D} = 0, \end{array} \right.$$

or equivalently

$$(9) \quad \left\{ \begin{array}{l} \mathcal{D}v(x, t_{n+1}) + \nabla p(x, t_{n+1}) = \nu \Delta v(x, t_{n+1}) + f(x, t_{n+1}), \\ \operatorname{div} v(x, t_{n+1}) = 0, \\ v(x, t_{n+1})|_{\partial D} = 0, \end{array} \right.$$

where

$$\mathcal{D}v(x, t_{n+1}) := \frac{3v(x, t_{n+1}) - 4v(\cdot, t_n) \circ X^{n, n+1} + v(\cdot, t_{n-1}) \circ X^{n-1, n+1}}{2\Delta t}.$$

This scheme has been used in [7,9,19], and recently in [4,5] and [11].

3. First and second order in time LG methods

Hereafter, we shall focus on the finite element formulation of the semi-discrete schemes ((8)) and ((9)), they will be denoted by LG-BDF1 and LG-BDF2 methods, respectively. Let $\bar{\Omega}_h = \bigcup_{j=1}^{NE} T_j$ be a quasi-uniform triangulation of the region $\bar{\Omega}$, here T_j denotes a simplex of dimension d the diameter of which is h_j , $NE \in \mathbb{N}$ ($NE > 1$) is the number of elements in $\bar{\Omega}_h$, and $h = \max_j h_j$. In relation with $\bar{\Omega}_h$, we consider the reference element $\hat{T} := \{\hat{x} \in \mathbb{R}^d : 0 \leq x_i \leq 1, 1 - \sum_{i=1}^d x_i \geq 0\}$, such that for each T_j there exists an invertible mapping $F_j : \hat{T} \rightarrow T_j$ of class C^1 , and the H^1 -conforming finite element spaces $\mathbf{V}_h \subset H^1(\Omega)$, $\mathbf{X}_h = \mathbf{V}_h \cap \mathbf{H}_0^1(\Omega)$ and $M_h \subset L_0^2(\Omega)$; we assume that the pair of finite element spaces (\mathbf{X}_h, M_h) is inf-sup stable and the following approximation properties hold: for $v \in \mathbf{H}^{s_1+1}(\Omega) \cap \mathbf{H}_0^1(\Omega)$, $p \in H^{s_1+1}(\Omega)$, $0 \leq s_1 \leq m_1$ and $0 \leq s \leq m$,

$$(10a) \quad \inf_{v_h \in \mathbf{X}_h} \left(\|v - v_h\|_{\mathbf{L}^2(\Omega)} + h \|v - v_h\|_{\mathbf{H}^1(\Omega)} \right) \leq Ch^{s_1+1} \|v\|_{\mathbf{H}^{s_1+1}(\Omega)}$$

and

$$(10b) \quad \inf_{q_h \in M_h} \left(\|p - q_h\|_{L^2(\Omega)} + h \|p - q_h\|_{H^1(\Omega)} \right) \leq Ch^{s_1+1} \|p\|_{H^{s_1+1}(\Omega)},$$

where m and m_1 denote the degree of the polynomials of \mathbf{X}_h and M_h respectively. We also assume that the following inverse properties hold in \mathbf{X}_h : for $0 \leq k \leq m \leq 1$ and $1 \leq p \leq q \leq \infty$,

$$(10c) \quad \|v_h\|_{\mathbf{W}^{m,q}(\Omega)} \leq Ch^{d/q-d/p+k-m} \|v_h\|_{\mathbf{W}^{k,p}(\Omega)}$$

and

(10d)

$$\|v_h\|_{\mathbf{L}^\infty(\Omega)} \leq D(h) \|v_h\|_{\mathbf{H}^1(\Omega)}; \quad D(h) := \begin{cases} C(1 + |\log h|^{1/2}) & \text{if } d = 2, \\ Ch^{-1/2} & \text{if } d = 3. \end{cases}$$

The formulation of LG-BDF1 and LG-BDF2 methods is as follows.

LG-BDF1: Given $v_h^0 \in \mathbf{X}_h$, for $n = 1, \dots, N-1$ find $(v_h^{n+1}, p_h^{n+1}) \in (\mathbf{X}_h \times M_h)$, such that for any $u_h \in \mathbf{X}_h$ and $q_h \in M_h$:

$$(11) \quad \begin{cases} (v_h^{n+1}, u_h) + \Delta t \nu (\nabla v_h^{n+1}, \nabla u_h) - \Delta t (p_h^{n+1}, \operatorname{div} u_h) \\ = (v_h^n \circ X_h^{n,n+1}, u_h) + \Delta t (f^{n+1}, u_h), \\ (\operatorname{div} v_h^{n+1}, q_h) = 0. \end{cases}$$

LG methods for Navier-Stokes Equations

LG-BDF2: Given $v_h^0, v_h^1 \in \mathbf{X}_h$, for $n = 1, \dots, N-1$ find $(v_h^{n+1}, p_h^{n+1}) \in (\mathbf{X}_h \times M_h)$, such that for any $u_h \in \mathbf{X}_h$ and $q_h \in M_h$:

$$(12) \quad \begin{cases} \left(\frac{3v_h^{n+1}}{2}, u_h \right) + \Delta t \nu (\nabla v_h^{n+1}, \nabla u_h) - \Delta t (p_h^{n+1}, \operatorname{div} u_h) \\ = (2v_h^n \circ X_h^{n,n+1}, u_h) - (\frac{1}{2}v_h^{n-1} \circ X_h^{n-1,n+1}, u_h) + \Delta t (f^{n+1}, u_h), \\ (\operatorname{div} v_h^{n+1}, q_h) = 0. \end{cases}$$

Since

$$(v_h^{n-l} \circ X_h^{n-l,n+1}, u_h) = \sum_j \int_{T_j} v_h^{n-l}(X_h^{n-l,n+1}(x)) \cdot u_h(x) dx, l = 0, 1,$$

then the calculation of $\int_{T_j} v_h^{n-l}(X_h^{n-l,n+1}(x)) \cdot u_h(x) dx$ and $X_h^{n-l,n+1}(x)$ are key issues of LG methods. On the other hand, we notice that v_h^0 in (11), and v_h^0 and v_h^1 in (12) are needed to perform the methods. To this respect, v_h^0 is calculated as the elliptic projection of v^0 onto \mathbf{X}_h , that is,

$$(13) \quad (\nabla v_h^0, \nabla u_h) = (\nabla v^0, \nabla u_h) \quad \text{for all } u_h \in \mathbf{X}_h.$$

As for the calculation of (v_h^1, p_h^1) , we define in $[0, \Delta t]$ the uniform partition $0 = t_0^* < t_1^* < \dots < t_{m_0}^* = \Delta t$ of step $\Delta t^* = \frac{\Delta t}{m_0}$, such that $\frac{\Delta t^2}{m_0} = C\Delta t^3$, where C is a constant of moderate size; then we calculate the sequence $\{\bar{v}_h^k, \bar{p}_h^k\}_{k=1}^{m_0}$ by (11) and set $p_h^1 = \bar{p}_h^{m_0}$, $v_h^1 = \bar{v}_h^{m_0}$. If $v^0 = 0$ in Ω , then we calculate (v_h^1, p_h^1) by solving time dependent Stokes problems with BDF1 as time marching scheme.

For any $x \in \Omega$, the points $X_h^{n-l,n+1}(x)$ ($l = 0, 1$) are numerical solutions at time instants t_{n-l} of the initial value problem

$$(14) \quad \begin{cases} \frac{dX_h(x, t_{n+1}; t)}{dt} = v_h(X_h(x, t_{n+1}; t), t), & t_{n-l} \leq t < t_{n+1}, \\ X_h(x, t_{n+1}; t_{n+1}) = x, \end{cases}$$

where $v_h(\cdot, t)$ is usually calculated by some extrapolation/interpolation formula of the values v_h^n and v_h^{n-1} . Noting that for all n , v_h^n and v_h^{n-1} are in $\mathbf{W}^{1,\infty}(D)$, then there is a unique solution $X_h(x, t_{n+1}; t)$ to (14).

3.1. Calculation of $\int_{T_j} v_h^{n-l}(X_h^{n-l,n+1}(x)) \cdot u_h(x) dx$, $l = 0, 1$

The evaluation of the element integrals is usually done numerically by applying a quadrature rule of high order so as to maintain both the stability and the accuracy that the method would possess if the integrals were calculated exactly. Thus, noting that for $x \in T_j$, $X_h^{n-l,n+1}(x)$ is in some T_i of the mesh, and letting $u_h(x)$ be the p th basis function of T_j , i.e., $\varphi_p^{(j)}$, and ne be the number of velocity nodes per element, we can set

$$(15) \quad \int_{T_j} v_h^{n-l}(X_h^{n-l,n+1}(x)) \cdot u_h(x) dx = \sum_{k=1}^{ne} V_{k(i)}^{n-l} \int_{T_j} \varphi_k^{(i)}(X_h^{n-l,n+1}(x)) \varphi_p^{(j)}(x) dx,$$

where $k(i)$ denotes the global number of the node of the mesh $\bar{\Omega}_h$ that is the k th node of T_i , and $\{\varphi_k^{(i)}\}_{k=1}^{ne}$ is the set of local basis functions for the element T_i . Now, assuming that $X_h^{n-l,n+1}(x) \in T_i$

$$\int_{T_j} \varphi_k^{(i)}(X_h^{n-l,n+1}(x)) \varphi_p^{(j)}(x) dx = \int_{\hat{T}} \hat{\varphi}_k(\hat{z}) \hat{\varphi}_p(\hat{x}) \left| \frac{\partial F_j}{\partial \hat{x}} \right| d\hat{x},$$

where $\hat{z} := F_i^{-1} \circ X_h^{n-l,n+1}(x)$ and $\{\hat{\varphi}_i\}_{i=1}^{ne}$ is the set of basis functions for the reference element \hat{T} . Finally, we approximate the integrals over \hat{T} by high order quadrature rules as

$$(16) \quad \int_{\hat{T}} \hat{\varphi}_k(\hat{z}) \hat{\varphi}_p(\hat{x}) \left| \frac{\partial F_j}{\partial \hat{x}} \right| d\hat{x} \simeq \text{meas}(T_j) \sum_{g=1}^{nqp} \varpi_g \hat{\varphi}_k(\hat{z}_g) \hat{\varphi}_p(\hat{x}_g) \left| \frac{\partial F_j(\hat{x}_g)}{\partial \hat{x}} \right|,$$

where nqp denotes the number of weights, ϖ_g , and points, \hat{x}_g , of the quadrature rule. An algorithmic presentation of the procedure is as follows.

For $j = 1, 2 \dots NE$ (NE is the number of the elements in $\bar{\Omega}_h$)

For $l = 0, 1$

For $k = 1, 2 \dots nqp$

(a) Calculate $x_k = F_j(\hat{x}_k)$ and then calculate $X_h^{n-l,n+1}(x_k)$ by solving (14) with initial condition x_k .

(b) Find the element T_i containing the point $X_h^{n-l,n+1}(x_k)$ and calculate $\hat{z}_k = F_i^{-1}(X_h^{n-l,n+1}(x_k))$.

(c) Calculate

$$v_h^{n-l}(X_h^{n-l,n+1}(x_k)) = \sum_{m=1}^{ne} V_{m(i)}^{n-l} \hat{\varphi}_m(\hat{z}_k).$$

LG methods for Navier-Stokes Equations

For $p = 1, \dots, ne$

Calculate

$$\text{meas}(T_j) \sum_{k=1}^{nqp} \varpi_k v_h^{n-l}(X^{n,n+1}(x_k)) \widehat{\varphi}_p(\widehat{x}_k) \left| \frac{\partial F_j(\widehat{x}_k)}{\partial \widehat{x}} \right|.$$

Assemble these values into a right hand side column vector.

End

3.2. Calculation of the approximate departure points $X_h^{n-l,n+1}(x)$

The first thing we should notice is that for all t_n , $n > 1$, v_h^{n+1} and $X_h^{n-l,n+1}$ depend on each other, so a way to proceed avoiding fixed point considerations is to use an explicit procedure to calculate $X_h^{n-l,n+1}$. Furthermore, the numerical solution to ((14)) should be calculated by a method of order equal to or larger than the order of the backward scheme employed to discretize the term $\frac{Dv}{Dt}$. The numerical solution to (14) is frequently calculated by explicit Runge-Kutta schemes of order two or higher, see for instance, [12], [7], [19], [30]. More recently, and based on the good properties of the fixed point implicit multi-step method of order 2 proposed in [29], we have developed and adaptive version of this method that works very well. We first describe a Runge-Kutta scheme of order 2

A Runge-Kutta method of order 2

For $n = 0$, calculate

$$(17a) \quad \begin{cases} K_1 = v_h^0(x), \\ K_2 = v_h^0(x - \Delta t K_1), \\ X_h^{0,1}(x) = x - \Delta t \left(\frac{K_1}{2} + \frac{K_2}{2} \right). \end{cases}$$

For $n = 1, 2, \dots, N-1$, and $l = 0$

$$(17b) \quad \begin{cases} K_1 = \bar{v}_h^{n+1}(x), \\ K_2 = v_h^n(x - \Delta t K_1), \\ X_h^{n,n+1}(x) = x - \Delta t \left(\frac{K_1}{2} + \frac{K_2}{2} \right). \end{cases}$$

When $l = 1$, calculate

$$(17c) \quad \begin{cases} K_1 = v_h^n(X_h^{n,n+1}(x), t_n), \\ K_2 = v_h^{n-1}(X_h^{n,n+1}(x) - \Delta t K_1), \\ X_h^{n-l,n+1}(x) = X_h^{n,n+1}(x) - \Delta t \left(\frac{K_1}{2} + \frac{K_2}{2} \right). \end{cases}$$

Here,

$$\bar{v}_h^{n+1}(\cdot) = 2v_h^n(\cdot) - v_h^{n-1}(\cdot),$$

is a second order approximation to v_h^{n+1} . In this algorithm the crucial steps are the calculations of K_1 and K_2 . Noting that v_h is known only at the mesh points $\{x_i\}$ at time steps t_n, t_{n-1}, \dots, t_0 , then it follows that, in general, $v_h^n(y)$ and $v_h^n(y - \Delta t K_1)$ are unknowns because both y and $y - \Delta t K_1$ do not coincide with mesh points, so they are calculated by finite element interpolation on the elements where the points y and $y - \Delta t K_1$ are located. A search-locate algorithm to identify such elements and simultaneously perform the finite element interpolations is described in [2]. The algorithm also informs if the points lie outside the domain.

Remark 3.1. For all $x \in \Omega$, the departure points $X_h^{n-l, n+1}(x)$ cannot leave the computational domain through the solid boundaries because on such boundaries either $v_h = 0$ or $v_h \cdot \mathbf{n} = 0$, \mathbf{n} being the unit outward normal, so that it can be proved that the trajectories of (14) cannot cross the solid boundaries. However, in many cases, in particular when Δt is not small enough and points y are in elements close to the solid boundaries of Ω , the numerical errors may cause some points $y - \Delta t K_1$ to be outside the computational domain. One way to alleviate this trouble is presented in the following adaptive algorithm.

An adaptive fixed point implicit multi-step method of order 2

We present the algorithm for the calculation of the points $X_h^{n, n+1}(x)$ because, as we have said above, we calculate $X_h^{n-1, n+1}(x)$ in two steps: first, we compute $X_h^{n, n+1}(x)$, and then $X_h^{n-1, n} \circ X_h^{n, n+1}(x)$. For the sake of clarity, we temporally go back to the notation $X_h(x, t_{n+1}; t_n)$ and present the non-adaptive version of the algorithm first. Since the solution of (14) can be expressed by the formula

$$X_h(x, t_{n+1}; t_n) = x - \int_{t_n}^{t_{n+1}} v_h(X_h(x, t_{n+1}; t), t) dt,$$

then, setting

$$\alpha = \int_{t_n}^{t_{n+1}} v_h(X_h(x, t_{n+1}; t), t) dt = x - X_h(x, t_{n+1}; t_n)$$

and evaluating the integral by the mid-point rule, we obtain a formula to approximate α up to order $O(\Delta t^3)$ such as

$$\alpha = \Delta t v_h\left(X(x, t_{n+1}; t_n + \frac{\Delta t}{2}), t_n + \frac{\Delta t}{2}\right).$$

LG methods for Navier-Stokes Equations

Moreover, using the second order approximation

$$X_h(x, t_{n+1}; t_n + \frac{\Delta t}{2}) \simeq \frac{1}{2}(x + X_h(x, t_{n+1}; t_n)) = x - \frac{\alpha}{2},$$

the second order extrapolation

$$v_h(\cdot, t_n + \frac{\Delta t}{2}) = \frac{3v_h(\cdot, t_n)}{2} - \frac{v_h(\cdot, t_{n-1})}{2},$$

and defining the operator $G : \Omega \rightarrow \Omega$ as

$$G(\alpha) = \frac{\Delta t}{2} \left(3v_h(x - \frac{\alpha}{2}, t_n) - v_h(x - \frac{\alpha}{2}, t_{n-1}) \right),$$

we can calculate α as the fixed point of the operator G if

$$\Delta t \max_{(x, t_n) \in B_j \times (0, T)} |\nabla v_h^n(x)| < 2,$$

because under this condition G is a contractive operator. B_j is a neighborhood of the point x such that $X_h^{n, n+1}(x) \in B_j$. A fixed point iterative procedure to calculate α at time t_{n+1} is the following.

Given ϵ , a real number such that $0 < \epsilon \ll 1$, $k_{max} \in \mathbb{N}$ ($k_{max} \geq 1$), v_h^n and v_h^{n-1} :

(1) Set

$$\alpha^{(0)} = \Delta t \left(\frac{3}{2}v_h^n(x) - \frac{1}{2}v_h^{n-1}(x) \right).$$

(2) For $k = 0, 1, \dots, k_{max}$

$$\alpha^{(k+1)} = \Delta t \left(\frac{3}{2}v_h^n(x - \frac{1}{2}\alpha^{(k)}) - \frac{1}{2}v_h^n(x - \frac{1}{2}\alpha^{(k)}) \right).$$

The iterative procedure stops when $k \leq k_{max}$ and

$$\frac{|\alpha^{(k+1)} - \alpha^{(k)}|}{|\alpha^{(k)}|} \leq \epsilon$$

(3) Set

$$X_h^{n, n+1}(x) = x - \alpha^{(k+1)}.$$

If Δt is so large that either the iterative procedure does not converge or the point $X_h^{n, n+1}(x)$ lies outside the domain, then we successively halve

m times Δt until convergence or $X_h^{n,n+1}(x)$ is inside the domain; thus, this yields and **adaptive fixed point iterative procedure**. To describe such a procedure, we consider that $\Delta t \rightarrow \frac{\Delta t}{2^m}$, $m \geq 0$, then $X_h(x, t_{n+1}, t_{n+1} - \frac{\Delta t}{2^m})$ is given by

$$X_h(x, t_{n+1}, t_{n+1} - \frac{\Delta t}{2^m}) = x - \int_{t_{n+1} - \frac{\Delta t}{2^m}}^{t_{n+1}} v_h(X_h(x, t_{n+1}, t), t) dt \equiv x - \alpha.$$

Noting that

$$\begin{aligned} & \int_{t_{n+1} - \frac{\Delta t}{2^m}}^{t_{n+1}} v_h(X_h(x, t_{n+1}, t), t) dt = \\ & = \frac{\Delta t}{2^m} v_h(X_h(x, t_{n+1}, t_{n+1} - \frac{\Delta t}{2^{m+1}}, t_{n+1} - \frac{\Delta t}{2^{m+1}}) + O((\frac{\Delta t}{2^m})^3), \end{aligned}$$

and

$$X_h(x, t_{n+1}, t_{n+1} - \frac{\Delta t}{2^{m+1}}) = \frac{1}{2}(x + X_h(x, t_{n+1}, t_{n+1} - \frac{\Delta t}{2^m})) + O((\frac{\Delta t}{2^m})^2),$$

it follows that

$$\begin{aligned} & X_h(x, t_{n+1}, t_{n+1} - \frac{\Delta t}{2^m}) \\ (18) \quad & = x - \frac{\Delta t}{2^m} v_h(X_h(x, t_{n+1}, t_{n+1} - \frac{\Delta t}{2^{m+1}}, t_{n+1} - \frac{\Delta t}{2^{m+1}}) + O((\frac{\Delta t}{2^m})^3), \end{aligned}$$

and

$$X_h(x, t_{n+1}, t_{n+1} - \frac{\Delta t}{2^{m+1}}) = x - \frac{\alpha}{2} + O((\frac{\Delta t}{2^m})^2).$$

Now, making use of the extrapolation formula

$$v_h^{n+1-2^{-m-1}} = (2 - 2^{-m-1})v_h^n - (1 - 2^{-m-1})v_h^{n-1} + O((\frac{\Delta t}{2^m})^2)$$

we can set

$$\begin{aligned} X_h(x, t_{n+1}, t_{n+1} - \frac{\Delta t}{2^m}) & = x - 2^{-m} \Delta t ((2 - 2^{-m-1}) v_h^n \\ & - (1 - 2^{-m-1}) v_h^{n-1}) (x - \frac{\alpha}{2}) + O((\frac{\Delta t}{2^m})^3) \end{aligned}$$

Thus, neglecting $O((\frac{\Delta t}{2^m})^3)$ terms and defining $G^m : \Omega \rightarrow \Omega$ as

$$(19) \quad G^m(\alpha) = 2^{-m} \Delta t ((2 - 2^{-m-1}) v_h^n - (1 - 2^{-m-1}) v_h^{n-1}) (x - \frac{\alpha}{2}),$$

LG methods for Navier-Stokes Equations

it follows that

$$\alpha = G^m(\alpha).$$

Thus, after calculating α as the fixed point of G^m , we use (18) and obtain $X_h(x, t_{n+1}, t_n)$ by performing the following backward procedure: for $i=m-1, m-2, \dots, 0$

$$\begin{aligned} X_h(x, t_{n+1}, t_{n+1} - \frac{\Delta t}{2^i}) &= x - 2^{-i} \Delta t (2 - 2^{-i-1}) v_h^n(X_h(x, t_{n+1}, t_{n+1} - \frac{\Delta t}{2^{i+1}})) \\ &\quad - 2^{-i} \Delta t (1 - 2^{-i-1}) v_h^{n-1}((X_h(x, t_{n+1}, t_{n+1} - \frac{\Delta t}{2^{i+1}})). \end{aligned}$$

An algorithmic description of the adaptive fixed point iterative procedure is as follows:

$$0 < m \leq M_{max} \in \mathbb{N}$$

Setting $m=1$, then

(1) Calculate

$$\alpha^{(0)} = 2^{-m+1} \Delta t \left(\frac{3}{2} v_h^n(x) - \frac{1}{2} v_h^{n-1}(x) \right).$$

If $x - \frac{1}{2} \alpha^{(0)}$ leaves the computational domain through a solid boundary, set $m = m + 1$ and repeat (1).

(2) For $k = 0, 1, \dots, k_{max}$ calculate

$$\alpha^{(k+1)} = 2^{-m+1} \Delta t v_h(x - \frac{1}{2} \alpha^{(k)}, t_{n+1} - \frac{\Delta t}{2^m}).$$

(a) If $x - \frac{1}{2} \alpha^{(k+1)}$ leaves the computational domain through a solid boundary, set $m = m + 1$ and repeat (1).

(b) If $k = k_{max}$ and

$$\frac{|\alpha^{(k+1)} - \alpha^{(k)}|}{|\alpha^{(k)}|} > \varepsilon,$$

then set $m = m + 1$ and repeat (1).

(c) If

$$\frac{|\alpha^{(k+1)} - \alpha^{(k)}|}{|\alpha^{(k)}|} \leq \varepsilon$$

stop the iterations.

(3) Set

$$X_h(x, t_{n+1}; t_{n+1} - \frac{\Delta t}{2^m}) = x - \alpha^{(k+1)} \left(t_{n+1} - \frac{\Delta t}{2^m} \right).$$

(3) For $i = m - 1, \dots, 0$ calculate

$$\begin{aligned} & X_h \left(x, t_{n+1}; t_{n+1} - \frac{\Delta t}{2^i} \right) \\ &= x - 2^{-i} \Delta t v_h \left(X_h \left(x, t_{n+1}; t_{n+1} - \frac{\Delta t}{2^{i+1}} \right), t_{n+1} - \frac{\Delta t}{2^{i+1}} \right). \end{aligned}$$

Note that when $i = 0$, $X_h^{n,n+1}(x) = X_h(x, t_{n+1}; t_n)$. In the above formula

$$v_h(\cdot, t_{n+1} - \frac{\Delta t}{2^k}) = (2 - 2^{-k})v_h(\cdot, t_n) - (1 - 2^{-k})v_h(\cdot, t_{n-1}), \quad 0 \leq k \leq m.$$

It may happen that despite the adaptive algorithm there are still some points leaving the domain, we artificially move these points to the barycenter of the elements through which they are leaving the domain. It can be shown that the adaptive iterative procedure converges if

$$\Delta t \max_{(x, t_n) \in B_j \times (0, T)} |\nabla v_h^n(x)| < 2^m.$$

Substepping

Another second order scheme, termed substepping, was proposed by Buscaglia and Dari (1992) with the idea of not leaving points outside the domain. Such an algorithm is implemented in two steps as follows:

1) For $t \in [t_{n-l}, t_{n+1})$ set

$$(20) \quad \frac{dX_h(x, t_{n+1}; t)}{dt} = \tilde{v}_h(X_h(x, t_{n+1}; t), t),$$

where $\tilde{v}_h(X_h(x, t_{n+1}; t), t)$ is an approximation to $v_h(X_h(x, t_{n+1}; t))$ by the second order extrapolation formula

$$\tilde{v}_h(\cdot, t) = v_h(\cdot, t_{n-1}) + \frac{t - t_{n-1}}{\Delta t} (v_h(\cdot, t_n) - v_h(\cdot, t_{n-1})).$$

2) Set $NP = 1$ and solve ((20)) by a second order predictor-corrector scheme with step $\Delta t^* = \Delta t / NP$; if $X_h(x, t_{n+1}; t_{n-l})$ is outside Ω , set $NP = NP + 1$ and repeat the procedure until $X_h(x, t_{n+1}; t_{n-l}) \in \Omega$.

4. Error bounds for LG-BDF1 and LG-BDF2 methods

Süli [27] develops a methodology based on mathematical induction on the index n to calculate the error estimates for LG-BDF1 methods, such an approach can be extended to the calculation of the error estimates for LG-BDF2 methods as well, see [5]. Recalling that m and m_1 are the degree of

LG methods for Navier-Stokes Equations

the polynomials of \mathbf{X}_h and M_h respectively, one proves the error estimates by making the following assumptions:

- (A1) $v^0 \in \mathbf{H}^{m+1}(\Omega) \cap \mathbf{V}$,
- (A2) $v \in L^\infty(\mathbf{V} \cap \mathbf{H}^{m+1}(\Omega)) \cap C(\mathbf{C}^{0,1}(\overline{\Omega}))$,
- (A3) $v_t \in L^2(\mathbf{V} \cap \mathbf{H}^{m+1}(\Omega))$, $D_t^2 v$ and $D_t^3 v \in L^2(\mathbf{L}^2(\Omega))$,
- (A4) $p \in L^\infty(H^{m_1+1}(\Omega) \cap L_0^2(\Omega) \cap L^\infty(\Omega))$ and $p_t \in L^2(H^{m_1+1}(\Omega))$;
- (A5) the mesh restriction to get optimal error estimates $\Delta t = o(h^{d/4})$;
- (A6) for $l = 0, 1$ there exists a constant \bar{c} independent of h and Δt , such that

$$\|v^l - v_h^l\|_{\mathbf{L}^2(\Omega)} + h \left[\|v^l - v_h^l\|_{\mathbf{H}^1(\Omega)} + \|p^l - p_h^l\|_{L^2(\Omega)} \right] = \bar{c}(h^{m+1} + l\Delta t^3);$$

(A7) (Induction hypothesis) for all n , such that $0 \leq n < N$, there exist constants $h_s < 1$, and $C > 0$ independent of Δt , h and n such that for $h \in (0, h_s)$

$$\|v - v_h\|_{l^\infty(0, t_n; \mathbf{L}^2(\Omega))} \leq C (h^{m+1} + \Delta t^\omega),$$

$$\|v - v_h\|_{l^\infty(0, t_n; \mathbf{H}^1(D))} \leq C (h^m + \Delta t^\omega),$$

where $\omega = 1$ for LG-BF1 and $\omega = 2$ for LG-BDF2.

Notice that for Taylor-Hood element $\mathbb{P}_2/\mathbb{P}_1$, $m = 2$ and $m_1 = 1$, and for the mini-element, i.e., the element \mathbb{P}_1 -bubble/ \mathbb{P}_1 , $m = 1$ and $m_1 = 1$. A consequence of the induction hypothesis is that, see [27] and [1], there is a constant $h_1 \in (0, h_s)$ independent of Δt and n such that

$$(21) \quad \Delta t |v_h^n|_{\mathbf{W}^{1, \infty}(D)} \leq \varepsilon_d(h) < 1 \quad \forall h \in (0, h_1].$$

Condition (21) allows us to show that the mapping $x \rightarrow X_h^{n-l, n+1}$ is, for all n , a homeomorphism from Ω onto Ω .

With these assumptions we have the following result.

Theorem 4.1. *Assume that the trajectories are calculated by a numerical method of order $r \geq 2$ and (A1)-(A7), ((10a))-((10d)), and $\Delta t = O(h^\sigma)$ as $h \rightarrow 0$, with $\sigma > \frac{d-1}{2}$, hold. Then there are constants C and K of the form $C = \max(\bar{c}, K(\nu^{-1}, \nu, p, T)\exp(\kappa T))$, where the constant κ depends on $\|\nabla v\|_{L^\infty(\mathbf{L}^\infty)}$, such that*

$$(22) \quad \begin{cases} \|v - v_h\|_{l^\infty(0, t_n; \mathbf{L}^2(D))} \leq C (h^{m+1} + \Delta t^\omega), \\ \|v - v_h\|_{l^\infty(0, t_n; \mathbf{H}^1(D))} \leq C (h^m + \Delta t^\omega), \\ \|p - p_h\|_{l^2(0, t_n; L^2(D))} \leq C (h^{m_1+1} + \Delta t^\omega) \end{cases}$$

5. Numerical results

We illustrate the performance of LG-BDF methods running some benchmark tests of three dimensional flows.

5.1. Lid-driven cavity flow

This is a widely used benchmark test to validate numerical schemes. The first accurate solutions for the three dimensional cavity problem were reported by Ku *et al.* (1987) and Tang *et al.* (1995) among others; more recent results employing stabilized finite elements for high Reynolds numbers are presented in Hachen *et al.* (2010). Complex phenomena appear in the cubic cavity and Taylor-Görter like vortices are formed at relatively low Reynolds numbers. We simulate the flow in the cavity, which is plotted in Figure 1, using the time marching LG-BDF2 method with the \mathbb{P}_1 - bubble/ \mathbb{P}_1 element and a time step $\Delta t = 0.05$, until the flow reaches the steady state; we consider that the steady state is reached when

$$(23) \quad \frac{\|v_h^{n+1} - v_h^n\|_{\mathbf{L}^2(\Omega)}}{\Delta t} \leq 10^{-6}, \quad \frac{\|p_h^{n+1} - p_h^n\|_{L^2(\Omega)}}{\Delta t} \leq 10^{-6}.$$

The calculations are carried out with $Re=1000$, at this value of Re steady vortices are expected. The numerical simulations are performed on a non-uniform mesh of 350085 elements and 64396 vertices, with the particularity that the mesh parameter h takes the value $h = 0.01$ near the wall, and then increases as we move towards the center of the cavity. A section of the three-dimensional mesh is displayed in Figure 2.

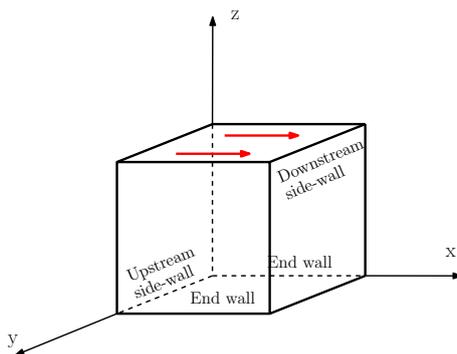


Figure 1. Flow configuration of the 3D cavity problem

To avoid discontinuities, we follow to Hachem *et al.* [16] and define the velocity of the moving boundary by the expression

$$(24) \quad v_1 = (1 - x^{18})^2(1 - y^{18})^2, \quad v_2 = 0, \quad v_3 = 0.$$

LG methods for Navier-Stokes Equations

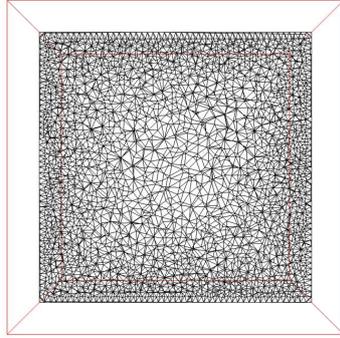


Figure 2. A cross-section of the mesh for the lid-driven cavity at $y = 0.5$

The velocity is assumed to be zero on the rest of the walls. The initial condition is $v = 0$ in the whole domain. The departure points are calculated by the adaptive fixed point implicit multi-step method of order 2. The integrals (15) are approximated with a quadrature rule of order 6 (24 points). The steady state is reached after 1354 time steps. Figure 3 shows the projections of velocity vectors on the planes $x = 0.5$, $y = 0.5$ and $z = 0.5$. In the graphics corresponding to the plane $x = 0.5$, we can see the corner eddies caused by the presence of the walls in the planes $y = 0$ and $y = 1$; moreover, we observe, in the plane $y = 0.5$, the existence of the main circulation cell and the downstream secondary eddy formed in the bottom left corner, whereas we can see the Taylor-Göter-like vortices in the plane $z = 0.5$. These results are in perfect agreement with those reported in Hachem *et al* [16] and Tang *et al* [28].

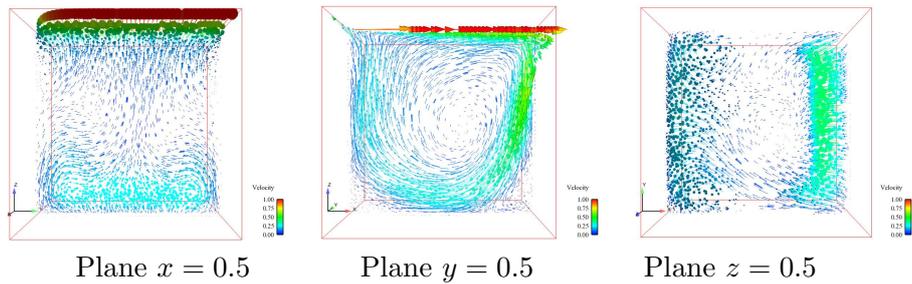


Figure 3. Velocity vectors on the three middle planes

The pressure contours on the three middle planes are displayed in Figure 4, they are similar to those obtained by Tang *et al* [28].

In Figure 5 we represent the profiles of the first component of the velocity at the vertical centerline and the third component of the velocity at

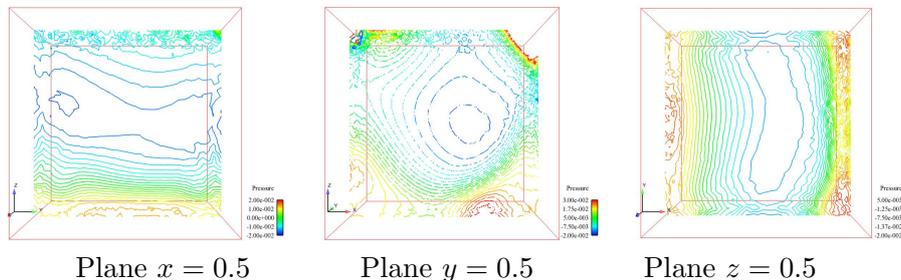


Figure 4. Pressure contours on the three middle planes

the horizontal centerline obtained by different methods. We see that our results compare very well with the results of Tang *et al* [28], Shu *et al* [26] and Hachem *et al* [16], although ours have been calculated on a mesh that is coarser than the meshes used in those papers. Also, to add a piece of further information, we include the profiles obtained with the commercial code FLUENT using a second order finite volume method and the same mesh.

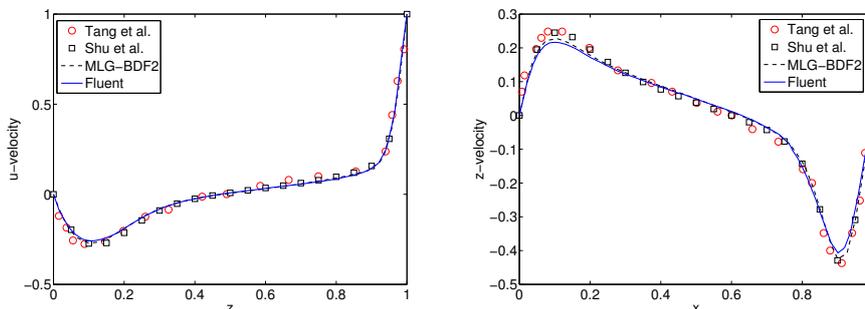


Figure 5. Comparison of velocity profiles v_x and v_z along central axes on plane $y = 0.5$.

5.2. Three dimensional flow around a cylinder

We present the results obtained with the LG-BDF2 method using the $\mathbb{P}_1 - \text{bubble}/\mathbb{P}_1$ element for the 3D stationary incompressible flow around a cylinder, see John [17] for the definition of this example. The geometry of the test is shown in Figure 6. We employ a mesh composed of 476560 elements and 87631 vertices, with the mesh parameter h being equal to 0.005 near the cylinder. A section of the mesh is plotted in Figure 7.

The boundary conditions are zero velocity on the solid boundaries, the inflow velocity is given by

$$(25) \quad v_{in} = (7.2x_1x_3(H - x_2)(H - x_3)/H^4, 0, 0),$$

LG methods for Navier-Stokes Equations

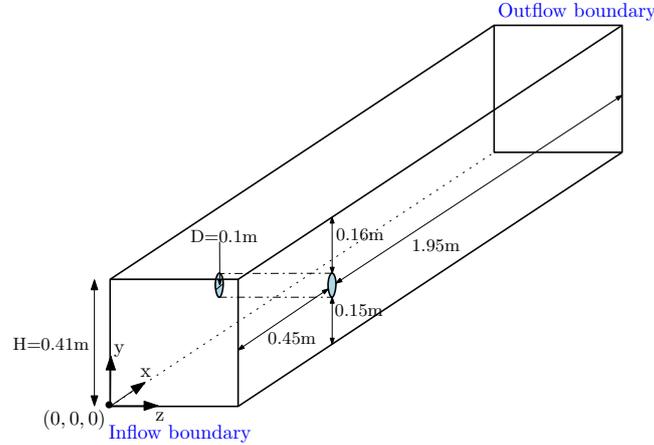


Figure 6. Computational domain

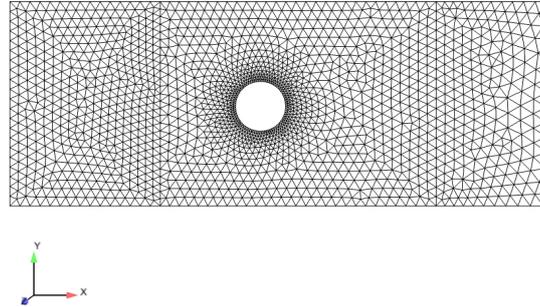


Figure 7. Mesh around the cylinder

where $H = 0.41$ m is the height of the channel, and the conventional do-nothing condition ($-\nu \frac{\partial v}{\partial \mathbf{n}} + p \mathbf{n} = 0$) on the outflow boundary. The kinematic viscosity of the fluid is $\nu = 10^{-3}$ m²/s, and $\text{Re}(= \frac{v_{in} D}{\nu})$ is 20.

The quantities chosen to represent the numerical solution are the pressure difference Δp between the points $(0.45, 0.2, 0.205)$ and $(0.55, 0.2, 0.205)$, and the drag (c_d) and lift (c_l) coefficients. These coefficients are given by the formulas:

$$c_d = \frac{2}{\rho v_0^2 D H} F_d = \frac{500}{0.41} F_d, \quad c_l = \frac{2}{\rho v_0^2 D H} F_l = \frac{500}{0.41} F_l,$$

where F_d and F_l denote the drag and lift forces respectively, which are

expressed as

$$(26) \quad F_d = \int_C \left(\nu \frac{\partial(v \cdot \tau_1)}{\partial \mathbf{n}} n_y - p n_x \right), \quad F_l = - \int_C \left(\nu \frac{\partial(v \cdot \tau_1)}{\partial \mathbf{n}} n_x + p n_y \right).$$

Here, C denotes the lateral surface of the cylinder, $\mathbf{n} = (n_x, n_y, 0)^T$ is the inward pointing unit normal vector with respect to Ω , and $\tau_1 = (n_y, -n_x, 0)^T$ and $\tau_2 = (0, 0, 1)^T$ are the tangent vectors. We compute the integrals ((26)) following the method presented in [17], which evaluates the surface integrals as volume integrals over the whole domain. This way of evaluating the integrals is easy to implement and gives more accurate results than the direct numerical calculation of the surface integrals. The results of Table 1 have been calculated with a time step $\Delta t = 0.005$, employing the adaptive fixed point implicit multi-step algorithm for the calculation of the departure points. We consider that the steady state is reached when conditions ((23)) are fulfilled. The Min and Max values in Table 1 define the interval of values for c_l and c_d according to [25], however, the most accurate values for c_l , c_d and Δp are $c_d=6.1853$, $c_l=0.0094$, and $\Delta p=0.1713$, and they are calculated in [8] with a DWR adaptive finite element method using $\mathbb{Q}_2/\mathbb{Q}_2$ elements and local projection stabilization . We also show in this table the results obtained (with the same mesh as the one used in the LG method) by the commercial software FLUENT using a second order finite volume scheme.

Coefficients	Min.	Max.	LG	Fluent
c_d	6.05	6.25	6.2172	6.1956
c_l	0.008	0.01	0.0096	0.0104
Δp	0.165	0.175	0.1753	0.1637

Finally, we represent in Figure 8 the streamlines on section $y = 0.2$; it is worth noting that a vortex is generated behind the cylinder, revealing thus the three-dimensionality of this flow.

All these calculations were carried out on a node with an Intel Xeon processor X5570.

6. Projection/LG methods

As we mention in the introductory section, LG methods yield every time step a linear Stokes problem, in the previous numerical tests we solve such Stokes problems by a direct method; however, Achdou and Guermond (2000) and Guermond and Mineev (2003) apply LG methods in combination with fractional steps schemes to decouple velocity and pressure in a

LG methods for Navier-Stokes Equations

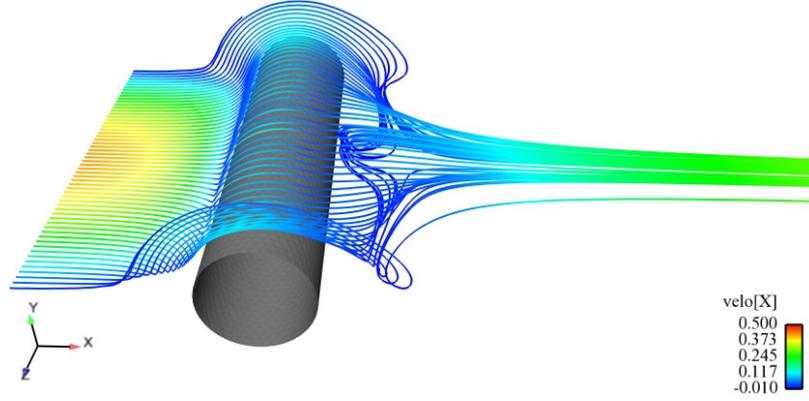


Figure 8. Stream lines at $y = 0.2$ colored by v_x

way that in the first step a velocity is calculated by solving a viscous equation satisfying the boundary conditions, then the pressure is obtained by solving a Poisson equation with homogeneous Neumann boundary conditions if the boundary conditions for the velocity are of Dirichlet type on Γ (see [14] for the case in which the velocity is also subject to open boundary conditions), and finally the divergence free velocity is calculated by a projection of the viscous velocity onto a divergence free subspace. Specifically, the projection/LG method of Guermond, Achdou and Minev to calculate a numerical solution to ((1))-((3)) is the following.

Given $v_h^0 \in \mathbf{X}_h$, for $n = 0, 1 \dots N - 1$ find $(\bar{v}_h^{n+1}, v_h^{n+1}, p_h^{n+1})$ such that

$$\begin{aligned} (\bar{v}_h^{n+1}, u_h) + \Delta t \nu (\nabla \bar{v}_h^{n+1}, \nabla u_h) &= \left(v_h^n \circ X_h^{n,n+1}, u_h \right) + \Delta t (p_h^n, \operatorname{div} u_h) \\ &\quad + \Delta t (f^{n+1}, u_h), \quad \forall u_h \in \mathbf{X}_h, \end{aligned}$$

$$(\nabla (p_h^{n+1} - p_h^n), \nabla q_h) = \frac{-1}{\Delta t} (\operatorname{div} \bar{v}_h^{n+1}, q_h), \quad \forall q_h \in M_h,$$

$$v_h^{n+1} = \bar{v}_h^{n+1} - \Delta t \nabla (p_h^{n+1} - p_h^n).$$

Notice that $\operatorname{div} v_h^{n+1} = 0$, and the L^2 projection of this velocity onto \mathbf{X}_h is the velocity employed to calculate $X_h^{n,n+1}$. Guermond and Minev proved the following result

Theorem 6.1. *Under the hypotheses of Theorem 1 there are constants, c_e , c_s , independent of Δt and h , and h_s such that for $h \in (0, h_s]$ and $\Delta t =$*

$c_s h^{d/3}$

$$(27) \quad \begin{aligned} \|v - v_h\|_{l^\infty(0,t_n;\mathbf{L}^2(D))} + \|v - \bar{v}_h\|_{l^\infty(0,t_n;\mathbf{L}^2(D))} &\leq c_e(h^{m+1} + \Delta t), \\ \|v - v_h\|_{l^2(0,t_n;\mathbf{H}^1(D))} &\leq c_c(h^m + \Delta t). \end{aligned}$$

Recently, Galán del Sastre and Bermejo (2011) have implemented the LG-BDF2 method as the rotational incremental velocity-correction scheme, see [15], combined with hp finite elements that use modal basis functions. The formulation of this implementation is as follows.

Let v_h^0 given, choose v_h^1 to be a good approximation to v^1 , then for $n \geq 1$ calculate $(v_h^{n+1}, p_h^{n+1}) \in \mathbf{X}_h \times M_h$ through the following steps:

- (1) Set $v_{he}^{n+1} = 2v_h^n - v_h^{n-1}$, then
- (2) calculate p_h^{n+1} by solving the equation

$$\begin{aligned} (\nabla p_h^{n+1}, \nabla q_h) &= -\frac{1}{2\Delta t} \left(3v_{he}^{n+1} - 4v_h^n \circ X_h^{n,n+1} + v_h^{n-1} \circ X_h^{n-1,n+1} \right) \\ &\quad -\nu (\nabla \times \nabla \times v_{he}^{n+1}, \nabla q_h) + (f^{n+1}, \nabla q_h) \quad \forall q_h \in M_h \end{aligned}$$

- (3) calculate v_h^{n+1} by solving the system

$$\begin{aligned} \frac{3}{2} (v_h^{n+1}, u_h) + \nu \Delta t (\nabla v_h^{n+1}, \nabla u_h) &= \left(2v_h^n \circ X_h^{n,n+1} - \frac{1}{2}v_h^{n-1} \circ X_h^{n-1,n+1} \right) \\ &\quad + \Delta t (-\nabla p_h^{n+1} + f^{n+1}, u_h) \quad \forall u_h \in \mathbf{X}_h. \end{aligned}$$

In this paper, the authors also study, via numerical examples, the efficiency and accuracy of LG methods versus SL methods when both methods are combined with quadrilateral spectral/hp elements. The conclusions with respect to these issues are: (1) for the same degree p of the polynomials defining the finite element spaces, both LG and SL methods have the same order of asymptotic convergence, however, LG methods show to be more accurate than SL methods for any p ; (2) concerning the CPU time, LG methods are more efficient than SL methods when p is large, say, $p \geq 4$; this conclusion may depend on the problem.

A different, although equivalent, form of implementing the velocity-correction -projection BDF2 scheme is given in Xiu and Karniadakis (2001) where they also use spectral/hp elements, but the material derivative is calculated by interpolation (semi-Lagrangian method) instead of Galerkin

LG methods for Navier-Stokes Equations

projection as in LG methods. This implementation is as follows:

Let v_h^0 given, choose v_h^1 to be a good approximation to v^1 , then for $n \geq 1$ calculate $(v_h^{n+1}, p_h^{n+1}) \in \mathbf{X}_h \times M_h$ through the following steps:

(1) Set $\hat{v}_h = 2\bar{v}_h^n - \frac{1}{2}\bar{v}_h^{n-1}$, where \bar{v}_h^n and \bar{v}_h^{n-1} denote the values of v_h^n and v_h^{n-1} at the feet $X_h^{n,n+1}(x_i)$ and $X_h^{n-1,n+1}(x_i)$ respectively, x_i being mesh points; these values are calculated by polynomial interpolation instead of Galerkin projection.

(2) Calculate p_h^{n+1} by solving the equation

$$(\nabla p_h^{n+1}, \nabla q_h) = \frac{-1}{\Delta t} (\operatorname{div} \hat{v}_h, q_h) + \left(\frac{\partial p_h^{n+1}}{\partial \mathbf{n}}, q_h \right), \quad \forall q_h \in M_h.$$

(3) Set

$$\hat{\hat{v}}_h = \hat{v}_h - \Delta t p_h^{n+1}$$

and calculate v_h^{n+1} by solving the system

$$(3) \quad \frac{3}{2} (v_h^{n+1}, u_h) + \Delta t \nu (\nabla v_h^{n+1}, \nabla u_h) = \left(\hat{\hat{v}}_h, u_h \right) + \Delta t (f^{n+1}, u_h), \quad \forall u_h \in \mathbf{X}_h,$$

where

$$\frac{\partial p_h^{n+1}}{\partial \mathbf{n}} = -\nu \mathbf{n} \cdot (\hat{\hat{v}}_h + \nabla \times (\nabla \times v_h^{n+1})).$$

7. LG methods for very high Reynolds numbers

In general, the LG methods presented so far have serious difficulties in dealing with flows at high Reynolds numbers, unless the mesh is very fine and the quadrature rules to evaluate the integrals (15) are very accurate. As in Eulerian methods, one technique to make LG methods suitable for such Reynolds numbers is to add stabilizing terms to the LG formulation of the Navier-Stokes equations. Bermejo and Saavedra (2015) have introduced a local projection stabilized formulation of LG methods that stabilizes the LG formulation symmetrically, maintaining the Stokes problem structure of the conventional LG formulation; moreover, the local projection stabilization approach can be identified with a variational multi-scale method. This new stabilized LG method is valid for Taylor-Hood finite element spaces (\mathbf{X}_h, M_h) so that $m_1 = m - 1$, with $m \geq 2$ in two-dimensional problems and $m \geq 3$ in three-dimensional problems, and the generalized min-element

with $m \geq 1$ in two- or three-dimensional problems. A good reference for the local projection stabilization technique is the textbook [24].

The formulation of the stabilized LG-BDF2 method of [3] reads as follows:

for $j = 1, 2, \dots, N - 1$, find $(v_h^{j+1}, p_h^{j+1}) \in \mathbf{X}_h \times M_h$ such that for any $v_h \in \mathbf{X}_h$ and $q_h \in M_h$ they are solution of the discrete problem

$$\begin{cases} \frac{1}{2\Delta t} \left(3v_h^{j+1} - 4v_h^j \circ X_h^{j,j+1} + v_h^{j-1} \circ X_h^{j-1,j+1}, u_h \right) \\ + \nu \left(\nabla v_h^{j+1}, \nabla u_h \right) - (p_h^{j+1}, \operatorname{div} u_h) + S_h(v_h^{j+1}, u_h) = (f^{j+1}, u_h), \\ \left(\operatorname{div} v_h^{j+1}, q_h \right) = 0, \end{cases}$$

here, $S_h(v_h, v_h)$ is the stabilization term given by the expression

$$S_h(v_h, u_h) = \sum_{K \in \Omega_h} \tau_K (\kappa_h^1 \nabla v_h, \kappa_h^1 \nabla u_h)_K + \mu_K (\operatorname{div} v_h, \operatorname{div} u_h)_K,$$

where K denotes a generic element of the mesh, and τ_K and μ_K are coefficients that depend on the mesh size and their optimal values are determined by the error analysis; specifically, in the examples we show below, $\tau_K = c_1 h_K^2$, $\mu_K = c_2$, c_1 and c_2 are constants. Moreover, $\kappa_h^1 = \mathbf{id} - \pi_h^1$ is the so called fluctuation operator, with $\mathbf{id} : \mathbf{L}^2(\Omega) \rightarrow \mathbf{L}^2(\Omega)$ being the identity operator and $\pi_h^1 : \mathbf{L}^2(\Omega) \rightarrow \mathbf{G}_h^1$ being an orthogonal projector; the finite dimensional space \mathbf{G}_h^1 is defined as

$$\mathbf{G}_h^1 := \{u_h \in \mathbf{L}^2(\Omega) : u_h|_K \in \mathbf{P}_0(K)\},$$

where $\mathbf{P}_0(K)$ denotes the space of polynomials of degree zero defined in K . In [3] it is proved the following result.

Theorem 7.1. *Under the hypotheses of Theorem 1 and with $\tau_K = O(h_K^{2(m-1)})$ there are constants C_1 , which is independent of h and Δt , but depending on $\|\nabla v\|_{L^\infty(L^\infty(\Omega))}$, and C_2 independent of h and Δt , such that*

$$\| \|v^N - v_h^N\| \|_\nu \leq C_1(h^m + \Delta t^2),$$

and

$$\|p - p_h\|_{l^2(L^2(\Omega))} \leq C_2(\|v - v_h\|_{l^\infty(\mathbf{L}^2(\Omega))} + h^m + \Delta t^2),$$

where $\| \cdot \|_\nu$ is a mesh dependent norm given by the expression

$$\| \|u^N\| \|_\nu = \left(\|u^N\|_{\mathbf{L}^2(\Omega)} + \nu \Delta t \sum_{j=0}^N \|\nabla u^j\|_{\mathbf{L}^2(\Omega)} + \Delta t \sum_{j=0}^N S_h(u^j, u^j) \right)^{1/2}.$$

LG methods for Navier-Stokes Equations

7.1. Numerical test with stabilized LG methods

To illustrate the performance of the stabilized LG methods, we show some simulations of the flow past the NACA0012 airfoil at zero angle of attack for $Re = 10^5$ using the stabilized LG-BDF1 method with $\mathbb{P}_2/\mathbb{P}_1$ elements and $\Delta t = 10^{-3}$. The domain $\Omega := (-5, 10) \times (-5, 5)$ and the NACA0012 profile is defined in $0 \leq x \leq 1$. The triangular mesh has 71785 elements of different size with 182893 velocity nodes and 37036 pressure nodes. A detail of the mesh is displayed in Figure 9.

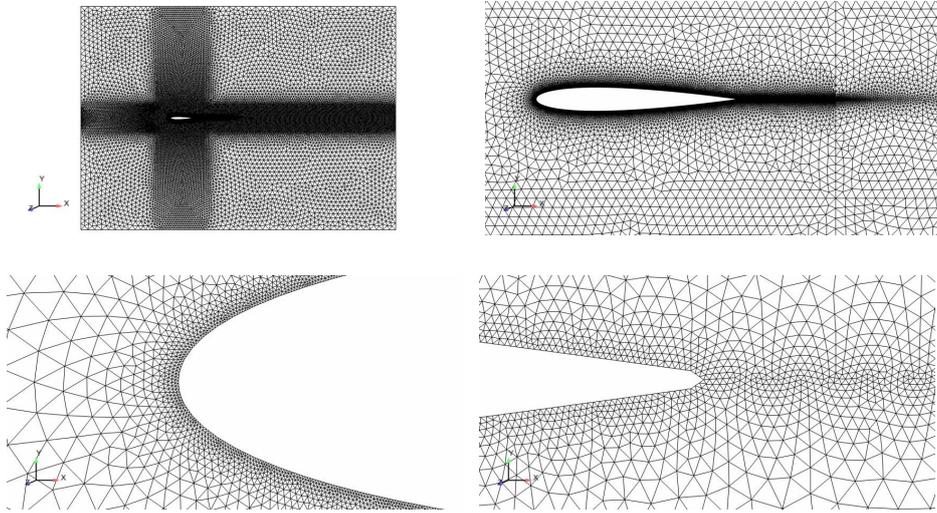


Figure 9. Mesh of the whole domain and around the airfoil (top), and near the leading and trailing edges (bottom)

Since in two dimensional flows the width of the boundary layer developed around the foil is $Re^{-1/2}$, then we choose a mesh size $h = 10^{-3}$ in a region around the foil to properly resolve the boundary layer. The boundary conditions are the following: (1) no-slip on the foil, (2) a potential velocity $\mathbf{U}_\infty = (1, 0)$ on the boundary $\{x = -5, -5 \leq y \leq 5\} \cup \{-5 \leq x \leq 10, y = \pm 5\}$, whereas a do-nothing boundary condition is enforced on the outflow boundary. The initial condition is zero in $\Omega \cup \{x = 5, -5 < y < 5\}$, and \mathbf{U}_∞ on $\{x = 5, -5 \leq y \leq 5\} \cup \{-5 \leq x \leq 5, y = \pm 5\}$. We show in Figure 10 velocity contours of the solutions obtained with both the conventional LG method and the stabilized LG method, the latter with $\tau_K = 0.1h_K^2$ and $\tau_K = h_K^2$, at $t = 5s$. A simple inspection of the graphics reveals that the solution of the conventional LG method is unstable, in contrast with the solutions of the local projection stabilized LG method which are stable; as expected, the solution with $\tau_K = h_K^2$ is smoother than the one with $\tau_K = 0.1h_K^2$. Further results with a Reynolds number as high as $Re = 3 \times 10^6$ are reported in [3].

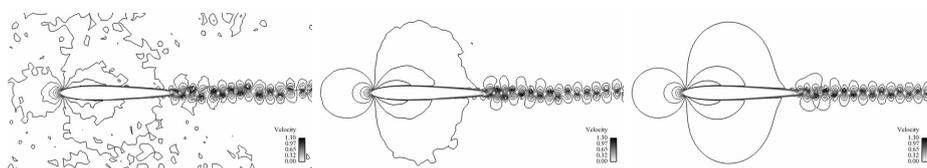


Figure 10. Velocity contours at $t = 5s$ obtained with conventional LG method (left) and with stabilized LG method: $\tau_K = 0.1h_K^2$ (middle) and $\tau_K = h_K^2$ (right)

REFERENCES

1. Y. Achdou, J.-L. Guermond, Convergence analysis of a finite element projection/Lagrange-Galerkin method for the incompressible Navier-Stokes equations, *SIAM Journal on Numerical Analysis*, vol. 37, pp. 799-826, 2000.
2. A. Allievi, R. Bermejo, A generalized particle search-locate algorithm for arbitrary grids, *Journal of Computational Physics*, vol. 132, pp. 157-166, 1997.
3. R. Bermejo, L. Saavedra, A second order in time local projection stabilized Lagrange-Galerkin method for Navier-Stokes equations at high Reynolds numbers, *Computers and Mathematics with Applications*, to appear, 2015.
4. R. Bermejo, L. Saavedra, Modified Lagrange-Galerkin methods to integrate time dependent incompressible Navier-Stokes equations, *SIAM Journal on Scientific Computing*, to appear, 2015.
5. R. Bermejo, P. Galán del Sastre and L. Saavedra, A second order in time modified Lagrange-Galerkin finite element method for the incompressible Navier-Stokes equations, *SIAM Journal on Numerical Analysis*, vol. 50, pp. 3084-3109, 2012.
6. R. Bermejo and L. Saavedra, Modified Lagrange-Galerkin methods of first and second order in time for convection-diffusion problems, *Numerische Mathematik*, vol. 120, pp. 601-638, 2012.
7. K. Boukir, Y. Maday, B. Métivet and E. Razanfindrakoto, A high-order characteristics/finite element method for the incompressible Navier-Stokes equations, *International Journal on Numerical Methods in Fluids*, vol. 25, pp. 1421-1454, 1997.
8. M. Braack and T. Richter, Solutions of 3D Navier-Stokes benchmark problems with adaptive finite elements, *Computers and Fluids*, vol. 35, pp. 372-392, 2006.
9. G.C. Buscaglia, A. Dari, Implementation of the Lagrange-Galerkin method for the incompressible Navier-Stokes equations, *International Journal of Numerical Methods in Fluids*, vol. 15, pp. 23-36, 1992
10. J. Douglas and T.F. Russell, Numerical methods for convection-

LG methods for Navier-Stokes Equations

- dominated diffusion problems based on combining the method of characteristics with finite element or finite difference procedures, *SIAM Journal on Numerical Analysis*, vol. 19, pp. 871-885, 1982.
11. M. El-Amrani, M. Seaid, An L^2 -projection for the Galerkin-Characteristic solution for incompressible flows, *SIAM Journal on Scientific Computing*, vol. 33, pp. 3110-3131, 2011.
 12. P. Galán del Sastre, R. Bermejo, A comparison of semi-Lagrangian and Lagrange-Galerkin hp-FEM methods in convection-diffusion problems, *Communications in Computational Physics*, vol. 9, pp. 1020-1039, 2011.
 13. J.-L. Guermond, P. Minev, Analysis of a projection/characteristic scheme for incompressible flow, *Communications in Numerical Methods in Engineering*, vol. 19, pp. 535-550, 2003.
 14. J.-L. Guermond, P. Minev, J. Shen, Error analysis of pressure-correction schemes for the Navier-Stokes equations with open boundary conditions, *SIAM Journal on Numerical Analysis*, vol. 43, pp. 239-258, 2005.
 15. J.-L. Guermond, J. Shen, On the error estimates for the rotational pressure-correction projection methods, *Mathematics of Computation*, vol. 73 (248), pp. 1719-1737, 2004.
 16. E. Hachem, B. Rivaux, T. Kloczko, H. Digonnet and T. Coupez. Stabilized finite element method for incompressible flows with high Reynolds number, *Journal of Computational Physics*, vol. 229, pp. 8643-8665, 2010.
 17. V. John. Higher order finite element methods and multigrid solvers in a benchmark problem for the 3D Navier-Stokes equations, *International Journal of Numerical Methods in Fluids*, vol. 40, pp. 775-798, 2002.
 18. H. C. Ku, R. S. Hirsh and T. D. Taylor. A pseudospectral method for solution of the three-dimensional incompressible Navier-Stokes equations, *Journal of Computational Physics*, vol. 70, pp. 439-462, 1987.
 19. P.D. Minev, C.R. Ethier, A characteristic/finite element algorithm for the 3D Navier-Stokes equations using unstructured grids, *Computer Methods in Applied Mechanics and Engineering*, vol. 178, pp. 39-50, 1999.
 20. K. W. Morton, A. Priestley and E. Süli, Stability of the Lagrange-Galerkin method with non-exact integration, *M2AN Mathematical Modelling and Numerical Analysis*, vol. 22, pp. 625-653, 1988.
 21. H. Notsu and M. Tabata, A single-step characteristic-curve finite element scheme of second order in time for the incompressible Navier-Stokes equations, *Journal of Scientific Computing*, vol. 38, pp. 1-14, 2009.
 22. A. Priestley, Exact projections and the Lagrange-Galerkin method: a realistic alternative to quadrature, *Journal of Computational Physics*, vol. 112, pp. 316-333, 1994.

23. O. Pironneau, On the transport-diffusion algorithm and its applications to the Navier-Stokes equations, *Numerische Mathematik*, vol. 38, pp. 309-332, 1982.
24. H.-G. Roos, M. Stynes, *Robust Numerical Methods for Singularly Perturbed Differential Equations*, Springer Series in Computational Mathematics, Springer-Verlag, Berlin, 2008.
25. M. Schäfer and S. Turek, *Benchmark computations of laminar flow around a cylinder (With support by Durst F, Krause E, Rannacher R)*. In: Schäfer E, editor. Flow simulation with high-performance computers II. DFG priority research program results 1993-1995. Notes numer. fluid mech, no. 52. Wiesbaden Vieweg; 547-566, 1996.
26. C. Shu, L. Wang and Y. T. Chew Numerical computation of three-dimensional incompressible Navier-Stokes equations in primitive variable form by DQ method, *International Journal for Numerical Methods in Fluids*, vol.43, pp. 345-368, 2003.
27. E. Süli, Convergence and nonlinear stability of the Lagrange-Galerkin method for the Navier-Stokes equations, *Numerische Mathematik*, vol.54, pp. 459-483, 1988.
28. L.Q. Tang, T. Cheng and T.T.H. Tsang Transient solutions for three-dimensional lid-driven cavity flows by a least-squares finite element method, *International Journal for Numerical Methods in Fluids*, vol. 21, pp. 413-432, 1995.
29. C. Temperton, A. Staniforth, An efficient two-time-level semi-Lagrangian semi-implicit integration scheme, *Quarterly Journal of the Royal Meteorological Society*, vol. 113, pp. 1025-1039, 1987.
30. D.Xiu and G.E.Karniadakis, A semi-Lagrangian high order method for Navier-Stokes equations, *Journal of Computational Physics*, vol.172, pp. 658-684, 2001.