

Fekete-Gauss Spectral Elements for Incompressible Navier-Stokes Flows: The Two-Dimensional Case

Laura Lazar¹, Richard Pasquetti^{1,*} and Francesca Rapetti¹

¹ Lab. J.A. Dieudonné, UMR 7351 CNRS UNS, Université de Nice - Sophia Antipolis, 06108 Nice Cedex 02, France.

Received 18 January 2012; Accepted (in revised version) 11 June 2012

Available online 8 October 2012

Abstract. Spectral element methods on simplicial meshes, say TSEM, show both the advantages of spectral and finite element methods, *i.e.*, spectral accuracy and geometrical flexibility. We present a TSEM solver of the two-dimensional (2D) incompressible Navier-Stokes equations, with possible extension to the 3D case. It uses a projection method in time and piecewise polynomial basis functions of arbitrary degree in space. The so-called Fekete-Gauss TSEM is employed, *i.e.*, Fekete (resp. Gauss) points of the triangle are used as interpolation (resp. quadrature) points. For the sake of consistency, isoparametric elements are used to approximate curved geometries. The resolution algorithm is based on an efficient Schur complement method, so that one only solves for the element boundary nodes. Moreover, the algebraic system is never assembled, therefore the number of degrees of freedom is not limiting. An accuracy study is carried out and results are provided for classical benchmarks: the driven cavity flow, the flow between eccentric cylinders and the flow past a cylinder.

AMS subject classifications: 65M60, 65M70, 76D05

Key words: Spectral elements, simplicial meshes, Fekete-Gauss approximation, Navier-Stokes equations, projection methods, domain decomposition.

1 Introduction

Using high order/spectral/spectral like methods may be of interest for many physical problems, *e.g.*, wave propagation over long distances or hydrodynamic instabilities, for which standard first/second order approximations may completely fail to capture the correct dynamics. As well known, spectral methods are however usually restricted to

*Corresponding author. *Email addresses:* laura.lazar@unice.fr (L. Lazar), richard.pasquetti@unice.fr (R. Pasquetti), francesca.rapetti@unice.fr (F. Rapetti)

simple geometries, *i.e.*, Cartesian, cylindrical, spherical, \dots . Progresses in this field essentially rely on using embedding methods but at the price of a loss of regularity of the solution and so, at least formally, of the so-called spectral accuracy. Spectral element methods (SEMs) are much better adapted to more involved geometries, see *e.g.* [8, 19]. However, since based on using quadrangular elements (2D case), they may also be not adapted to really complex ones for which simplicial meshes are required. This is why going to high order finite element methods (FEMs) or equivalently to SEMs on simplicial meshes is of increasing interest nowadays.

Many works have been carried out in this field during the last two decades, especially on the *hp*-FEM, see *e.g.* [19, 31] and references herein. Here we rather follow approaches proposed in the late 90's and in the 2000's on the basis of "true SEMs" for simplicial meshes, see *e.g.* [12, 15, 16, 28, 35, 40]. Such approaches are of nodal rather than modal type, *i.e.*, the basis functions are the Lagrange polynomials based on a set of carefully selected interpolation points. The choice of the best set of points, based on minimizing the corresponding Lebesgue constant for the reference triangle/tetrahedron, remains an open problem, which is however now more of academical interest. Various sets of interpolation points have indeed been proposed, at least in 2D, all of them showing satisfactory properties as soon as the polynomial interpolation degree on the spectral element remains reasonable (say $N \leq 12$) [26]. Among them we adopt the so-called Fekete points of the triangle, because of some nice properties, such as the Lagrange polynomials based on the Fekete points are maximum at these points, *i.e.*, the Lagrange polynomial φ_i based on the Fekete point F_i is such that $\max \varphi_i(\mathbf{x}) = \varphi_i(F_i) = 1$. Moreover, the Fekete points of the cube coincide with the Gauss-Lobatto-Legendre (GLL) points [2] involved in the standard SEM. This allows the efficient interfacing of triangles and quadrilaterals together in the same mesh, making *e.g.* possible the use of thin quadrilaterals to capture short length scales in boundary layers. Note however that, to our knowledge, Fekete points are only known for the triangle and remain to be determined for the tetrahedron.

As a new contribution to works that we have carried out recently on the so-called Fekete-Gauss *TSEM* for elliptic partial differential equations (PDE), see *e.g.* [24], we focus here on problems governed by the unsteady incompressible Navier-Stokes equations. The Fekete-Gauss *TSEM* (*T* for triangle/tetrahedron) makes use of two sets of points, (i) the Fekete points for the interpolations in *T* and the (ii) Gauss points of *T* for the quadratures. Such sets of points depend of course on the polynomial approximation degree. Adopting two sets of points allows to by-pass the *a priori* not solvable problem of finding in a non-tensorial domain a single set of points with both nice quadrature and interpolation properties, see [37] and references herein. In other words the Fekete points of *T* are not Gauss points, contrary to the GLL points for the cube. Moreover, as detailed in [24], the use of two sets of points provides a larger flexibility and may be handled efficiently.

The paper is organized as follows: Section 2 describes the time scheme, based of an implicit (resp. explicit) treatment of the diffusion (resp. advection) term and an up to date projection method. Section 3 provides details on the *TSEM* approximation. Section

4 presents the Schur complement method that is implemented and the way we use to solve the resulting algebraic systems. We also present a bending procedure to handle curved elements. Section 5 firstly provides some convergence results before addressing some well known benchmarks: Driven cavity flow, flow between two eccentric cylinders and wake of a cylinder, for which we introduce an original outflow boundary treatment. We conclude in Section 6.

2 The time scheme

In dimensionless form the incompressible Navier-Stokes (NS) equations may be written as follows:

$$\partial_t \mathbf{u} + \mathcal{N}(\mathbf{u}) = -\nabla p + \nu \Delta \mathbf{u} + \mathbf{f} \quad \text{in } \Omega \times (0, t_F), \quad (2.1a)$$

$$\nabla \cdot \mathbf{u} = 0, \quad (2.1b)$$

where ν is the inverse of the Reynolds number, \mathbf{f} a given forcing term and where \mathbf{u} and p are the velocity and pressure fields, respectively. Here Ω denotes the computational domain and t_F the final time value. In Eq. (2.1) $\mathcal{N}(\mathbf{u})$ is the non-linear advection term, *e.g.* in convective form $\mathcal{N}(\mathbf{u}) = (\mathbf{u} \cdot \nabla) \mathbf{u}$. To solve the evolution problem governed by the NS equations, we need a divergence free initial velocity field, \mathbf{u}_0 , defined on Ω , and boundary conditions, *e.g.*, for simplicity homogeneous Dirichlet boundary conditions $\mathbf{u}|_{\Gamma} = \mathbf{0}$, with Γ for the boundary of Ω .

For the numerical approximation of the NS system we plan to use a projection method and a second order accurate implicit (resp. explicit) treatment of the linear (resp. non-linear) term. Three steps are involved: (i) Computation of a provisional velocity, (ii) projection to obtain a divergence free field and (iii) update of the pressure. Such an approach is rather classical (projection methods were introduced in the late 60's [5, 38]). However we use some more recent ingredients in order to enhance the accuracy of the splitting [14, 39]. In details:

Step 1: Let $\mathbf{u}^n(\mathbf{x})$ be the numerical approximation of the exact solution $\mathbf{u}(\mathbf{x}, t_n)$, where $t_n = n\tau$ (n is the time advancing index and τ is the time-step). First we compute the provisional velocity $\mathbf{u}^*(\mathbf{x})$ using a second order backward finite difference (BDF2) approximation of the time derivative, a second order extrapolation (EX2) scheme for the convection term and an EX1 one for the pressure. Then, \mathbf{u}^* solves:

$$\frac{1}{\tau} \left(\alpha_0 \mathbf{u}^* + \sum_{j=1}^2 \alpha_j \mathbf{u}^{n+1-j} \right) + \sum_{j=1}^2 \beta_j \mathcal{N}(\mathbf{u}^{n+1-j}) = -\nabla p^n + \nu \Delta \mathbf{u}^* + \mathbf{f}^{n+1}, \quad (2.2a)$$

$$\mathbf{u}^*|_{\Gamma} = \mathbf{0}, \quad (2.2b)$$

where α_j, β_j are the coefficients of the BDF2 and EX2 schemes, respectively (here $\alpha_0 = 3/2, \alpha_1 = -2, \alpha_2 = 1/2$ and $\beta_1 = 2, \beta_2 = -1$).

Step 2: $\mathbf{u}^{n+1}(\mathbf{x})$ is computed from \mathbf{u}^* by a L^2 projection onto a space of divergence free vectors. On the basis of the Helmholtz decomposition theorem, one introduces a scalar potential φ and states that:

$$\mathbf{u}^* = \nabla\varphi + \mathbf{u}^{n+1}, \quad \nabla \cdot \mathbf{u}^{n+1} = 0. \quad (2.3)$$

Taking the divergence of \mathbf{u}^* and its dot product on Γ with the unit outwards orthogonal vector \mathbf{n} , if we enforce the impermeability condition $\mathbf{u}^{n+1} \cdot \mathbf{n}|_{\Gamma} = \mathbf{u}^* \cdot \mathbf{n}|_{\Gamma} = 0$, it turns out that φ solves the Poisson problem:

$$\Delta\varphi = \nabla \cdot \mathbf{u}^*, \quad (2.4a)$$

$$\partial_n \varphi|_{\Gamma} = 0. \quad (2.4b)$$

One easily checks that the divergence free vector field \mathbf{u}^{n+1} is indeed the L^2 projection of \mathbf{u}^* :

$$\begin{aligned} \int_{\Omega} (\mathbf{u}^* - \mathbf{u}^{n+1}) \cdot \mathbf{u}^{n+1} d\Omega &= \int_{\Omega} \nabla\varphi \cdot \mathbf{u}^{n+1} d\Omega = \int_{\Omega} \nabla \cdot (\varphi \mathbf{u}^{n+1}) d\Omega \\ &= \int_{\Gamma} \varphi \mathbf{u}^{n+1} \cdot \mathbf{n} d\Gamma = 0. \end{aligned} \quad (2.5)$$

Note that in case of non-homogeneous Dirichlet conditions, if $\mathbf{u} \cdot \mathbf{n}|_{\Gamma} \neq 0$ the impermeability condition may be recovered by using an irrotational divergence free lifting of \mathbf{u} and then a modified forcing term \mathbf{f} .

Step 3: For the update of the pressure, let us report the expression (2.3) of \mathbf{u}^* in the time-discrete momentum equation (2.2), as well as its Laplacian:

$$\Delta \mathbf{u}^* = \Delta \mathbf{u}^{n+1} + \Delta(\nabla\varphi) = \Delta \mathbf{u}^{n+1} + \nabla(\Delta\varphi) = \Delta \mathbf{u}^{n+1} + \nabla(\nabla \cdot \mathbf{u}^*), \quad (2.6)$$

then we get:

$$\frac{1}{\tau} \left(\sum_{j=0}^2 \alpha_j \mathbf{u}^{n+1-j} \right) + \sum_{j=1}^2 \beta_j \mathcal{N}(\mathbf{u}^{n+1-j}) = -\nabla p^{n+1} + \nu \Delta \mathbf{u}^{n+1} + \mathbf{f}^{n+1}, \quad (2.7a)$$

$$\mathbf{u}^{n+1}|_{\Gamma} = -\nabla\varphi|_{\Gamma}, \quad (2.7b)$$

where

$$p^{n+1} = p^n + \frac{\alpha_0}{\tau} \varphi - \nu \nabla \cdot \mathbf{u}^*. \quad (2.8)$$

Note however that for the present derivation of the pressure update [39], in (2.6) we have used the commutativity property of the gradient and Laplacian operators, which may not hold for the discrete operators. Thanks to the so-called rotational form of the viscous term this may be avoided: Since at the continuous level any gradient is in the kernel of the rotational operator we state that in (2.2)

$$\Delta \mathbf{u}^* \equiv \nabla \times (\nabla \times \mathbf{u}^*) - \nabla(\nabla \cdot \mathbf{u}^*) = \Delta \mathbf{u}^{n+1} - \nabla(\nabla \cdot \mathbf{u}^*). \quad (2.9)$$

The present approach is thus generally called projection method in rotational form of the NS equations [14].

Let us conclude this Section with the following remarks:

- Higher order approximations in time are easily obtained by using higher order finite difference discretizations. Thus, a third order approximation may be obtained by using BDF3 (rather than BDF2) for the time derivative, EX3 (rather than EX2) for the advection term and EX2 (rather than EX1) for the pressure. Such approximation orders are however formal.
- The present projection method is actually $\mathcal{O}(\tau^2)$ for the velocity components and $\mathcal{O}(\tau^{3/2})$ for the pressure [14], which up to our knowledge is the best accuracy result obtained with a projection method. Convergence results are provided in Section 5.
- From the gradient of the pressure update (2.8), one can derive that on the boundary Γ :

$$\partial_n p^{n+1} = \partial_n p^n - \nu \partial_n (\nabla \cdot \mathbf{u}^*), \quad (2.10)$$

so that the additional correction term avoids using for all time-steps the same Neumann condition value (generally zero) for the pressure. As discussed later in Section 3, in the developed TSEM implementation taking into account such an additional term requires however to solve a mass matrix problem.

- The explicit treatment of the convective term may differ from a simple extrapolation. For example, semi-Lagrangian approaches, like the operator integration factor (OIF) [22], method that are a priori better adapted to high Reynolds number flows, could be used here instead of the EX2 scheme.

Thanks to using a projection method, the NS problem splits into three uncoupled elliptic scalar PDEs in 2D (four PDEs in 3D), for the components of \mathbf{u}^* and for the potential φ . Hereafter we shortly describe how we derive the corresponding algebraic systems on the basis of the Fekete-Gauss TSEM approximation.

3 TSEM approximation

As described in the previous Section, by adopting a projection method we must essentially solve elliptic problems. This is why we consider now the 2D model problem:

$$-\nabla \cdot (\nu \nabla u) + \sigma u = f \quad \text{in } \Omega, \quad (3.1a)$$

$$u|_{\Gamma} = 0, \quad (3.1b)$$

where $\nu > 0$, $\sigma \geq 0$ are bounded constants and f belongs to $L^2(\Omega)$, the space of square integrable measurable functions in Ω . For simplicity, homogeneous Dirichlet conditions have been assumed.

The weak formulation of problem (3.1) reads: Given $f \in L^2(\Omega)$, find $u \in E = H_0^1(\Omega)$ (standard notations are used for these spaces, see e.g. [29]) such that

$$a(u, v) := \int_{\Omega} (v \nabla u \cdot \nabla v + \sigma uv) d\Omega = \int_{\Omega} f v d\Omega, \quad \forall v \in E. \quad (3.2)$$

The variational formulation (3.2) is discretized by a conforming spectral element method based on triangles. This is a Galerkin method which employs a discrete space consisting of continuous piecewise polynomials of total degree N .

3.1 Fekete interpolation points

Let $T = \{(r, s) : -1 < r, s \leq +1, r + s < 0\}$ be the reference triangle and $\mathbb{P}_N(T)$ the set of polynomials on T of total degree $\leq N$. We assume that Ω is decomposed into K non overlapping triangular finite elements Ω_k , $\bar{\Omega} = \bigcup_{k=1}^K \bar{\Omega}_k$, each of which is the image of T by means of a suitable mapping, i.e., $\Omega_k = \mathbf{g}_k(T)$. The mesh is conforming, so that the intersection between two distinct $\bar{\Omega}_k$ is either the empty set or a common vertex or a common side. For the approximation space, say $E_{K,N}$, we use continuous, piecewise polynomials of total degree $\leq N$,

$$E_{K,N} = \{v \in C^0(\bar{\Omega}) : v|_{\Omega_k} \circ \mathbf{g}_k \in \mathbb{P}_N(T), 1 \leq k \leq K\} \subset E. \quad (3.3)$$

Let $\{\psi_j\}_{j=1}^n$, with $n = (N+1)(N+2)/2$, be an orthonormal basis of $\mathbb{P}_N(T)$ for the usual $L^2(T)$ inner product (for example, the Koornwinder-Dubiner polynomials may be used to constitute such a basis [9]). Fekete points on T are defined as those points $\{\hat{\mathbf{x}}_i\}_{i=1}^n$ that maximize the determinant of the Vandermonde matrix V with entries $V_{ij} = \psi_j(\hat{\mathbf{x}}_i)$, $1 \leq i, j \leq n$. Among the main properties of Fekete points proved in [1, 2, 36], one has that on the sides of the triangle the Fekete and GLL points coincide and that Fekete points are GLL points for the cube, thus providing a strong link with the usual SEM.

Unlike GLL points, a quadrature formula based on Fekete points is only exact for integrands in $\mathbb{P}_N(T)$. This remark suggests to separate the sets of approximation and quadrature points, using the Fekete points $\{\hat{\mathbf{x}}_i\}_{i=1}^n$ for the first set and other points $\{\hat{\mathbf{y}}_i\}_{i=1}^m$ for the second set, defined by imposing an exact integration of polynomials in $\mathbb{P}_{2N}(T)$ for example [24]. Given the values at the approximation points of a polynomial $u_N \in \mathbb{P}_N(T)$, one can set up interpolation and differentiation matrices to compute the values of u_N and its derivatives, respectively, at the quadrature points. The interpolation matrix is simply $V'V^{-1}$, where V' is a matrix of dimension (m, n) such that $V'_{ij} = \psi_j(\hat{\mathbf{y}}_i)$. For a clever choice of the interpolation points [26] the matrix V has a condition number that slowly increases with N , therefore the computation of its inverse V^{-1} does not involve difficulties. To compute derivatives, e.g., with respect to r , at the quadrature points we use again the Koornwinder-Dubiner polynomials to obtain $D^r = V'^r V^{-1}$, with $(V'^r)_{ij} = \partial_r \psi_j(\hat{\mathbf{y}}_i)$. Once the differentiation matrices D^r and D^s are known it is an easy task to compute derivatives at the quadrature points starting from the values at the approximation points by applying

the chain rule. This yields for the gradient, in the generic element Ω_k , $\nabla_{xy} = J_k^{-t} \nabla_{rs}$, where J_k is the Jacobian matrix of the mapping \mathbf{g}_k (the superscript $-t$ stands for transposition times inversion).

3.2 Quadrature rule

The presented TSEM approach makes use of highly accurate integration rules based on Gauss points [6, 7, 34]. If such integration rules are unknown, *e.g.* for large values of N , at a higher computational cost it is possible to use integration rules based on Gauss points for the quadrangle and then map them to T (*e.g.*, [19, 33]). On a generic triangle $\Omega_k = \mathbf{g}_k(T)$, one may approximate the $L^2(\Omega_k)$ product of functions u and v by

$$(u, v)_{k,N} = \sum_{j=1}^m u(\mathbf{g}_k(\hat{\mathbf{y}}_j)) v(\mathbf{g}_k(\hat{\mathbf{y}}_j)) |J_k(\hat{\mathbf{y}}_j)| \omega_j \approx (u, v)_{\Omega_k}, \quad (3.4)$$

where $\omega_j > 0$, $1 \leq j \leq m$, are the quadrature weights and $|J_k|$ the Jacobian of the mapping \mathbf{g}_k between T and Ω_k . Knowing how to compute derivatives and integrals, we can use the usual FEM methodology to set up the discrete problem

$$\sum_{k=1}^K a_{k,N}(u, v) = \sum_{k=1}^K (f, v)_{k,N}, \quad \forall v \in E_{K,N}, \quad (3.5)$$

where $a_{k,N}(\cdot, \cdot)$ is obtained from $a(\cdot, \cdot)$ by replacing each integral with the quadrature rule (3.4). By using all the involved basis functions, Eq. (3.5) can be written in matrix form as a linear system $A\mathbf{u} = \mathbf{b}$. Note that the TSEM matrix A is less sparse than the standard SEM matrix and more ill-conditioned. As for some related choices of the basis functions, its condition number is expected to grow as $\mathcal{O}(N^4 h^{-2})$, where h is a maximal diameter of the triangular spectral elements (*e.g.*, [17, 23, 27]).

3.3 TSEM discrete system

With $\sigma = \alpha_0/\tau$, the approach presented here directly applies to the equations governing the components of the provisional velocity \mathbf{u}^* . If the boundary conditions are of Dirichlet type but not homogeneous, the procedure is the same as for the standard FEM (or SEM). It consists of using a lifting, which is simple to implement with Lagrange polynomials based on the boundary nodes, to recover the homogeneous Dirichlet condition. If a Neumann condition is considered the space $E_{K,N}$ must be enlarged with the Lagrange polynomials based on the corresponding boundary nodes. Moreover if this Neumann condition is non-homogeneous, then the source term should be completed with the contribution of the boundary integral which results from the integration by part.

For the potential φ we recover the Poisson equation with $\sigma = 0$ and $\nu = 1$. Since Neumann conditions are concerned, the space of basis functions $E_{K,N}$ additionally includes the Lagrange polynomials based on the boundary nodes. Here a difficulty arises as the

solution can only be computed up to an additive constant. In practice we simply set to 0 the value of φ at one of the boundary element nodes.

As it stands, the pressure update Eq. (2.8) has been shown to induce instabilities, because the $\nabla \cdot \mathbf{u}^*$ term is not C^0 continuous and so not regular enough. Such numerical instabilities are overcome if the equation is firstly projected onto the TSEM basis functions, which means that the pressure must solve:

$$\int_{\Omega} p^{n+1} v d\Omega = \int_{\Omega} \left(p^n + \frac{\alpha_0}{\tau} \varphi - \nu \nabla \cdot \mathbf{u}^* \right) v d\Omega, \quad \forall v \in E_{K,N}, \quad (3.6)$$

where again $E_{K,N}$ is enlarged with the Lagrange polynomials based on the boundary nodes, since no boundary conditions are here required. We then obtain a mass matrix problem, *i.e.* that requires the inversion of the mass matrix, of the form (3.2) with $\sigma = 1$ and $\nu = 0$, and the Lagrange basis functions for u and v .

3.4 Isoparametric elements

If the computational domain Ω is not polygonal, at the boundary Γ using isoparametric elements is necessary to preserve the spectral accuracy of the approximation. Isoparametric elements are easy to implement as soon as the images of the interpolation points by the mapping \mathbf{g}_k are known: The components of \mathbf{g}_k being themselves approximated by polynomials of total degree N , one can indeed use the differentiation matrices D^r and D^s to compute the elements of the Jacobian matrix J_k . Then, the problem reduces to define the images of the Fekete points in the generic element Ω_k . Some procedures may be found in the literature to set up a transformation from a reference element to a generic element [13]. Here we use an in-house bending procedure defined as follows: First, we use a standard mesh generator to discretize Ω by triangles; This allows to define linear mappings and for each triangle a set of interpolation points. Then, considering successively each triangle, say ABC , if one edge, say BC , should approximate a curved boundary, we apply to each interpolation point, say F_i , $1 \leq i \leq n$, the linear transformation such that the image of the intersection point of AF_i with BC is on the curved edge. Note that if F_i belongs to AB or AC , then it is not moved (the B and C vertices are on Γ) and of course that F_i may belong to BC . In this bending procedure it is assumed that each deformed triangle has at most one edge on the boundary Γ and that the opposite vertex is inside Ω . Its main advantage is that the $\mathbf{g}_k : T \rightarrow \Omega_k$ isoparametric mapping is not explicitly formulated.

3.5 Advection term

To conclude this Section let us focus on the advection term. In convective form such a term writes $\mathcal{N}(\mathbf{u}) = (\mathbf{u} \cdot \nabla) \mathbf{u}$ whereas in conservative form we have $\mathcal{N}(\mathbf{u}) = \nabla \cdot (\mathbf{u} \otimes \mathbf{u})$. In both cases, it is required to differentiate quantities which are not continuous at the edges of the elements. Indeed, in Eq. (2.3) the TSEM approximations of \mathbf{u}^* and φ are only C^0 continuous, so that $\nabla \varphi$ and consequently \mathbf{u}^{n+1} jump at the edges of the elements. Despite

that, because a weak formulation is involved good results may be obtained. However, we prefer to use the rotational form of the convective term, so that the momentum equation writes:

$$\partial_t \mathbf{u} + \boldsymbol{\omega} \times \mathbf{u} = -\nabla P + \nu \Delta \mathbf{u}, \quad (3.7a)$$

$$P = p + \frac{\mathbf{u}^2}{2}, \quad (3.7b)$$

where P is the total pressure and $\boldsymbol{\omega}$ the vorticity. Then, since $\boldsymbol{\omega}^{n+1} = \nabla \times \mathbf{u}^{n+1} = \nabla \times \mathbf{u}^*$, we have only to differentiate continuous quantities. Note that the pressure correction φ still appears as the Lagrange multiplier that allows the velocity field to be divergence free, so that nothing has to be changed in the projection algorithm previously described.

4 The solution algorithm

The main drawback of the usual FEM approach is that the matrix A may be very large and ill conditioned. Following works that we have previously carried out [25,27], we are going to use a Schur complement method in the particular case where each mesh element is a subdomain. The proposed approach presents the following advantages:

- The condition number of the matrix A is $\mathcal{O}(N^4 h^{-2})$ whereas the condition number of its Schur complement S is $\mathcal{O}(N h^{-2})$ [23,25].
- The dimension of the Schur complement matrix S is $\mathcal{O}(N)$ whereas the dimension of matrix A is $\mathcal{O}(N^2)$.

Thus, using the Schur complement method the algebraic system we have to solve is both better conditioned and smaller, since we only compute the unknowns associated to nodes that are located on the boundaries of the spectral elements. Such an approach is sometimes called static condensation [30]. The unknowns at the inner nodes are then computed a posteriori and the resolution matrix is never assembled, as explained in what follows.

Let us consider the matrix form of Eq. (3.5) and restrict it to each element Ω_k . Using as test functions $v \circ \mathbf{g}_k$ the Lagrange polynomials based on the Fekete points, the elemental matrix system reads

$$A_k \mathbf{u}_k = \mathbf{b}_k + \mathbf{r}_k, \quad (4.1)$$

where \mathbf{u}_k is the vector of the unknowns at the interpolation nodes in Ω_k whereas \mathbf{r}_k stands for the contribution of an (unknown) Neumann condition at the edges shared by two elements. Note however that such terms compensate when assembled, *i.e.*

$$\sum'_k \mathbf{r}_k = 0, \quad (4.2)$$

where \sum' is used to denote the assembling procedure.

By reordering (if necessary) first the boundary nodes and then the interior ones (this is not required in our implementation), and since \mathbf{r}_k has no contribution to inner nodes, the matrix system $A_k \mathbf{u}_k = \mathbf{b}_k + \mathbf{r}_k$ can be rewritten as

$$\begin{pmatrix} A_{k,\gamma\gamma} & A_{k,\gamma I} \\ A_{k,I\gamma} & A_{k,II} \end{pmatrix} \begin{pmatrix} \mathbf{u}_{k,\gamma} \\ \mathbf{u}_{k,I} \end{pmatrix} = \begin{pmatrix} \mathbf{b}_{k,\gamma} + \mathbf{r}_{k,\gamma} \\ \mathbf{b}_{k,I} \end{pmatrix}, \quad (4.3)$$

where the subscript (k,γ) (resp. (k,I)) refers to the boundary (resp. inner) nodes of the element Ω_k . Here, we have used the fact that $\mathbf{r}_{k,I} = 0$. Assuming now that $A_{k,II}$ is not singular, we can eliminate the variables $\mathbf{u}_{k,I}$ and set up the following equation for $\mathbf{u}_{k,\gamma}$:

$$S_k \mathbf{u}_{k,\gamma} = \mathbf{g}_k, \quad (4.4a)$$

with

$$S_k = (A_{k,\gamma\gamma} - A_{k,\gamma I} A_{k,II}^{-1} A_{k,I\gamma}), \quad (4.4b)$$

$$\mathbf{g}_k = \mathbf{b}_{k,\gamma} + \mathbf{r}_{k,\gamma} - A_{k,\gamma I} A_{k,II}^{-1} \mathbf{b}_{k,I}. \quad (4.4c)$$

The elementary Schur complement matrix S_k is of smaller dimension than matrix A_k , *i.e.*, $3N$ rather than $(N+1)(N+2)/2$. Moreover, S_k is also symmetric as A_k .

By the assembling procedure and taking into account the compensatory equation (4.2) one obtains:

$$S \mathbf{u}_\gamma = \mathbf{g}_\gamma, \quad (4.5a)$$

with

$$S = \sum_k' S_k, \quad (4.5b)$$

$$\mathbf{g}_\gamma = \sum_k' (\mathbf{b}_{k,\gamma} - A_{k,\gamma I} A_{k,II}^{-1} \mathbf{b}_{k,I}), \quad (4.5c)$$

where γ refers to the union of all element boundaries, so that in the present formulation of the Schur complement method, $\Gamma \subset \gamma$.

In practice we assemble the source term \mathbf{g}_γ but do not assemble the Schur complement matrix S , for memory space reasons. Because the Schur complement system is solved by using a PCG method, we indeed only need to realize matrix vector product, which is easy from:

$$S \mathbf{u}_\gamma = \sum_k' S_k \mathbf{u}_{k,\gamma}. \quad (4.6)$$

This is an alternative to the more usual approach based on the use of low storage algorithms for sparse matrices. For the preconditioner, we simply use the diagonal term of matrix S , which is also assembled to this end. Following our previous works [25, 27], interesting improvements could be achieved with the implementation of more sophisticated preconditioners.

Boundary conditions must however be enforced and this may appear problematic if matrix S is not assembled. Let us go into the details by first considering Neumann, then homogeneous Dirichlet and finally non-homogeneous Dirichlet conditions.

- Neumann conditions: Since they are taken into account in the source terms \mathbf{b}_k and since they only infer on $\mathbf{b}_{k,\gamma}$ (just like the \mathbf{r}_k), they are in fact correctly taken into account in the assembled term \mathbf{g}_γ .
- Homogeneous Dirichlet conditions: If the matrix $S = \sum_k' S_k$ was assembled, then each i_γ -line and each i_γ -column of S associated to a Dirichlet boundary point should be set to 0 and the (i_γ, i_γ) diagonal element should be set to 1, so that if the i_γ component of \mathbf{g}_γ is set to 0, then the boundary condition is correctly prescribed. One may notice that this strategy preserves the dimension of the algebraic system (4.5), is easy to implement and also preserves the symmetry of matrix S , which is important in view of using a CG method. The algorithm that we use exactly realizes that without modifying the elemental matrices: Each time a matrix vector product of the form $S\mathbf{u}_\gamma = \sum_k' S_k \mathbf{u}_{k,\gamma}$ is made, we first set to zero the i_γ -component of \mathbf{u}_γ , which is equivalent to cancel the i_γ -column of S ; Then, once the product is done, we set to zero the i_γ -component of $S\mathbf{u}_\gamma$, which is equivalent to cancel the i_γ -line of S except the diagonal term.
- Non-homogeneous Dirichlet conditions: We start by the splitting of \mathbf{u}_γ such as, $\mathbf{u}_\gamma = \mathbf{u}_\gamma^h + \mathbf{u}_\gamma^d$, where \mathbf{u}_γ^h solves an homogeneous Dirichlet problem and \mathbf{u}_γ^d gathers the Dirichlet data. This is in fact the usual FEM implementation of the solution lifting. First, one subtracts to the source term the sum $\sum_k' S_k \mathbf{u}_{k,\gamma}^d$ (obvious notations are used). Then the procedure used for the homogeneous Dirichlet case is applied to compute $S\mathbf{u}_\gamma^h = \sum_k' S_k \mathbf{u}_{k,\gamma}^h$. Finally it remains to set the Dirichlet data to get $S\mathbf{u}_\gamma$.

In our implementation all operators specific to each element, *i.e.* $A_{k,II}^{-1}$, $A_{k,I\gamma}$, $A_{k,\gamma I} A_{k,II}^{-1}$ and S_k , are computed and stored in a preliminary calculation. The storage requirement is then $\mathcal{O}(KN^2)$. Such storage capacity remains reasonable and guarantees of an efficient resolution during the time loop. It must however be done for each (σ, ν) pair, *i.e.*, for the \mathbf{u}^* -components, for the potential φ and also for the pressure update.

5 Applications

On the basis of the algorithms presented in the previous sections, a Fortran code has been developed. In this Section we present some examples of applications to well known benchmark problems: (i) the driven cavity flow, (ii) the flow between two eccentric cylinders and finally (iii) the flow past a cylinder. Computations have been done on a sequential Dell computer with processor Intel Xeon 5570 at frequency 2.93GHz. Before going to these examples, it is however of interest to check the accuracy of the projection method described in Section 2.

5.1 Accuracy results

As, *e.g.*, outlined in [14], the formal accuracy of projection methods generally fails. The approach described in Section 2 should at least avoid the formation of a numerical boundary layer if the computational domain is smooth, *i.e.* without corners. Another point generally emphasized is that in order to satisfy the inf-sup condition, and so to obtain a well posed discrete problem, the polynomial approximation degree for the pressure must be lower than for the velocity components. As a preliminary step, in the present study we use same polynomial degrees for all variables. It is then interesting to check the accuracy of the method and provide some convergence rates with respect to the time-step τ and to the polynomial approximation degree N .

To this end we simply consider the unsteady Stokes problem, *i.e.*, the non-linear convective term is dropped. The goal is to recover the exact solution:

$$\mathbf{u}_{ex} = (\sin(cx)\sin(cy+t), \cos(cx)\cos(cy+t)), \quad (5.1)$$

$$p_{ex} = \cos(cx)\sin(cy+t), \quad (5.2)$$

where the parameter c controls the spatial frequency (with $c = 1$ this solution is the one used in [14]). The Stokes equations are completed by a compatible forcing term. The computational domain Ω is the polygon of 18 equal sides contained in a circle of radius 1. Eq. (5.1) restricted to the boundary provides the Dirichlet condition. Computations are performed over the time interval $[0, t_F]$, with $t_F = 1$, on a mesh composed of $K = 48$ triangles.

First, in order to check if the spectral convergence is obtained, we have set the time-step $\tau = 10^{-3}$ and $c = 5$, so that the spatial error dominates the temporal one. For the polynomial degree we have either $N = 3$ (number of degrees of freedom per scalar field, $dof = 244$), or $N = 6$ ($dof = 919$) or $N = 9$ ($dof = 2026$). Results obtained at $t = t_F$ are provided in Fig. 1 (left). In the semi-log setting, the linear decrease of the “max norm” (l^∞ norm at the interpolation points) of the error clearly points out the so-called spectral accuracy for both the velocity components and the pressure. Note that to eliminate the pressure arbitrary additional constant, the pressure error is defined as $(\max(p - p_{ex}) - \min(p - p_{ex}))/2$.

Second, we have studied the convergence with respect to the time-step. For $N = 3$, $N = 6$ and $N = 9$, results for the pressure and the x -component of the velocity are displayed in Fig. 1 (right). The exact solution is (5.1)-(5.2) with $c = 1$, so that the temporal error is now dominant. Despite that, one observes that for $N = 3$ the error quickly saturates, meaning that the spatial error dominates for $\tau < 10^{-2}$. For $N = 6$ and $N = 9$, the error curves of the velocity coincide, which means that the time error is indeed dominant. These error curves clearly show the second order convergence rate that results from the BDF2-EX2 approximation. In agreement with the remarks of Section 2, things are more tricky for the pressure. For $N = 6$ and $N = 9$ minima of the errors are obtained for $\tau \approx 410^{-2}$ and $\tau \approx 10^{-3}$, respectively, whereas for $N = 3$, such a minimum does not lie in the time-step range. Beyond this N -dependent critical value of the time-step, the errors increase. The

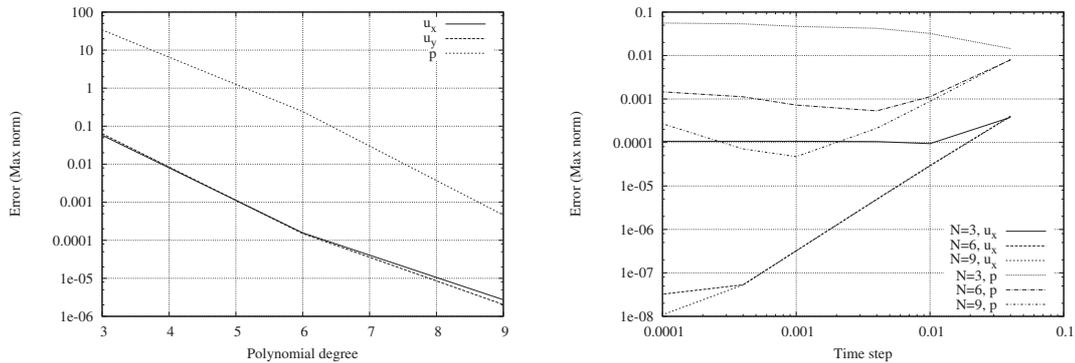


Figure 1: Quasi-circular domain. Left panel: Error curves for the velocity components and for the pressure vs the polynomial degree. The error on p dominates and the curves for u_x and u_y nearly coincide. Right panel: Error curves for u_x and p vs the time-step τ for $N = \{3, 6, 9\}$. For $N=3$, when decreasing the time-step the error on u_x quickly saturates and the error on p increases. For $N=6$ and $N=9$, the error curves for p show a minimum which shifts to the smaller time-steps when increasing N .

curves obtained for $N=3$ and $N=6$ seem however to show that the pressure field is not “blowing up”. This is typical of the spurious modes of pressure that affect the standard $\mathbb{P}_N - \mathbb{P}_N$ spectral approximations. As previously discussed, in the limit $N = \infty$ it seems that one recovers the exact pressure, so that one should rather invoke pseudo-spurious pressure modes. The present study is not contradictory with [14], where the authors guess a pressure blow up in grid size limit. Thus, differently to the h -refinement one may conjecture that the N -refinement allows to recover the exact pressure.

Tests have also been carried out in the square domain $\Omega = (-1, 1)^2$, using again the analytical solution (5.1)-(5.2) with $c = 5$. The mesh is composed of $K = 42$ elements, with polynomial approximation degree $N = 3$ ($dof = 220$), $N = 6$ ($dof = 817$) or $N = 9$ ($dof = 1792$). From the results of [14], one expects to observe accuracy failures in the corners despite the fact the solution is smooth. For the three different time-steps $\tau = 10^{-3}$, $\tau = 4 \cdot 10^{-4}$ and $\tau = 10^{-4}$, Fig. 2 (left) shows the decay of the max norm of the error for the x -component of the velocity and for the pressure. For the smallest value of the time step the results are not polluted by the time error, and one clearly discerns an exponential rate of convergence. For $N = 3$ and $N = 6$, just like for the quasi-circular domain one observes that the error is increasing when decreasing the time-step, so that one may again think that pseudo-spurious modes of pressure are present with the considered mesh. It may be observed that the error essentially results from one of the four corners of the computational domain. This is pointed out in Fig. 2 (right) which shows the field $p - p_{ex}$, obtained at $t = t_F$ for $N = 3$ and $\tau = 10^{-4}$, together with the mesh and the “micromesh” associated to each element. It turns out that the error is governed by the only corner that is not shared by two spectral elements. The spectral accuracy is however preserved.

We conclude this Section by providing an insight on the computational efficiency of our Fortran TSEM code. The computation time is of course essentially associated to the iterative solutions of the algebraic systems obtained for the velocity components, the pres-

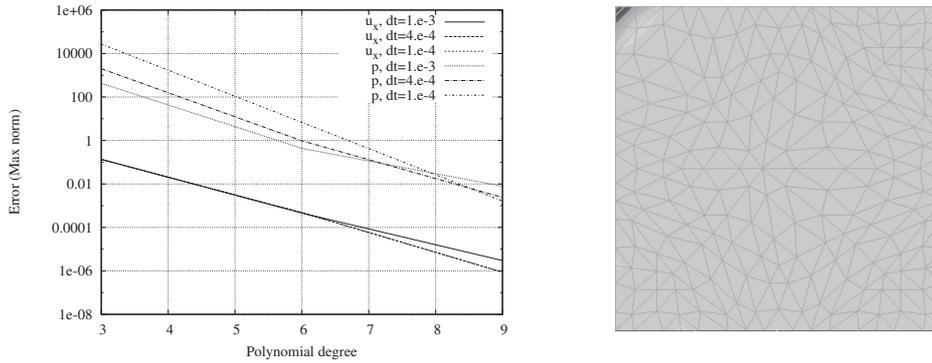


Figure 2: Square domain. Left panel: Error curves for u_x and p vs the polynomial degree N for the three different time-steps $\tau = \{10^{-3}, 4 \cdot 10^{-4}, 10^{-4}\}$. For p (the three upper curves), if $N \leq 6$ the smaller is the time-step the greater is the error. Right panel: Pressure error visualization for $N=3$ and $\tau=10^{-4}$; A peak occurs in the upper-left corner.

sure correction and the pressure update. The iterative process stops when the relative error on the residual equals 10^{-9} . Keeping fixed the mesh and the number of time-steps (10000, with $\tau = 10^{-4}$ and $t_F = 1$), Table 1 provides the computational time associated to the time advancing part of the unsteady Stokes problem, for different values of the polynomial approximation degree N . The results have been obtained for the quasi-circular domain. It is interesting to observe that the variations of the computational time per time step and degree of freedom (*dof*) are weak and also that a minimum is obtained for $N=6$.

Table 1: Computational time (Time, s) and computational time per time-step and degree of freedom (Time₁, μ s) for the unsteady Stokes problem.

| N | <i>dof</i> | Time (s) | Time ₁ (μ s) |
|-----|------------|----------|------------------------------|
| 3 | 244 | 12.66 | 5.19 |
| 6 | 919 | 45.87 | 4.99 |
| 9 | 2026 | 114.4 | 5.65 |

5.2 Lid-driven cavity flow

The flow within the square cavity $\Omega = (0,1)^2$ is driven from above by a lid moving with a given velocity. In order to provide accuracy comparisons with reference results, we first consider the regularized driven cavity problem, see *e.g.* [10], with boundary conditions $\mathbf{u} = (-16x^2(1-x)^2, 0)$, at $y = 1$, and no-slip condition ($\mathbf{u} = \mathbf{0}$) elsewhere. For the initial condition, the fluid is assumed to be at rest ($\mathbf{u}_0 = \mathbf{0}$).

Computations have been carried out for two values of the Reynolds number, $Re = 100$ and $Re = 400$, till obtaining a steady flow. Fig. 3 (top) shows the vorticity field obtained with $N=9$ for $Re=400$ at the final time of the computation together with the mesh which is composed of $K = 159$ elements.

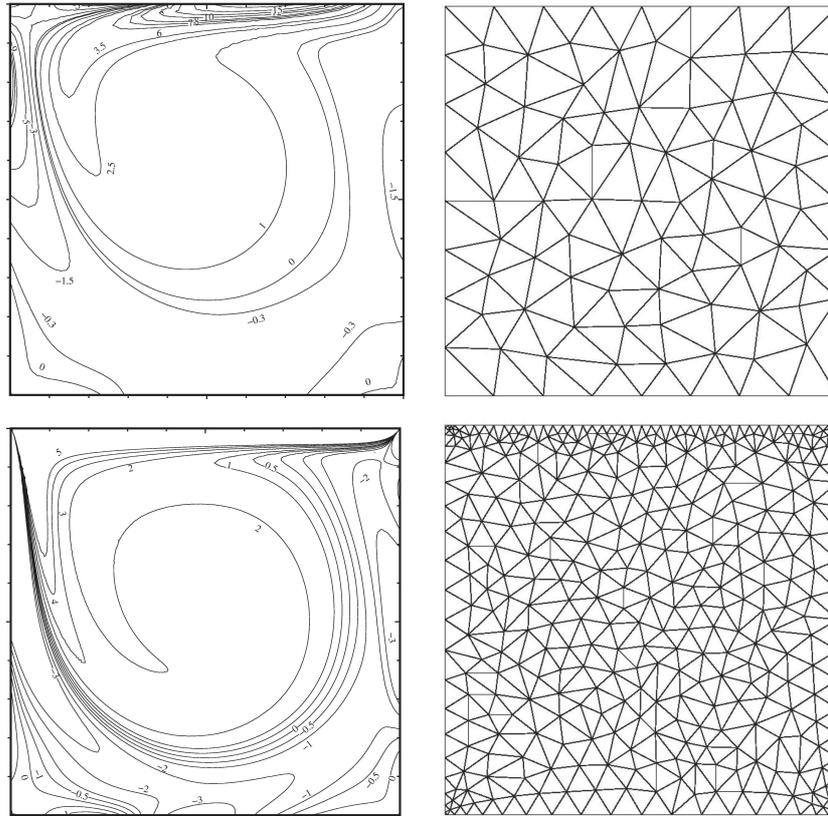


Figure 3: Vorticity (at left) with the mesh (at right) for the regularized driven cavity flow at $Re=400$ (top) and in the non-regularized case for $Re=1000$ (bottom); Computations done with $(K=159, N=9)$ and $(K=682, N=9)$, respectively.

Table 2 gives some significant results, *i.e.* extrema of the vorticity, obtained with different values of the polynomial approximation degree N . Comparisons are also provided with the spectral Chebyshev calculations [3, 10, 18]. A good convergence to the spectral results may be observed, especially knowing that in our case the extrema are those

Table 2: Lid-driven cavity flow: Extrema of the vorticity at $Re=100$ and $Re=400$.

| Ref. | N | dof | $Re = 100$ | | $Re = 400$ | |
|------|-----|-------|----------------|------------|----------------|------------|
| | | | $\max \omega $ | x_{\max} | $\max \omega $ | x_{\max} |
| | 3 | 763 | 13.25673 | 0.6250000 | 23.77490 | 0.6250000 |
| | 6 | 2956 | 13.44357 | 0.6143893 | 25.06365 | 0.6356107 |
| | 9 | 6580 | 13.44452 | 0.6199709 | 24.91301 | 0.6300291 |
| [3] | 32 | 1089 | 13.4448 | 0.620 | 24.9111 | 0.630 |
| [18] | 32 | 1089 | 13.4447 | 0.620 | 24.9107 | 0.630 |
| [10] | 32 | 1089 | 13.4447 | 0.620 | 24.9110 | 0.630 |

obtained at the Fekete points, *i.e.*, no polynomial interpolation is used to better localize these extrema, contrarily to what is done for the Chebyshev results. In Table 1 we have also mentioned the *dof* number per unknown variable. As expected the Chebyshev method appears here more efficient, because of the natural accumulation of the Chebyshev-Gauss-Lobatto grid points at the boundaries. Moreover, for the TSEM computation a quasi-uniform mesh is used, see Fig. 1 (top).

We now consider the non-regularized problem, *i.e.* with boundary condition $\mathbf{u} = (-1, 0)$ at $y=1$. Such a driven cavity flow is much more challenging to be captured with a high-order method, since involving singularities in the upper corners of Ω , where u_x is not continuous. This means that the vorticity, $\omega = \partial_x u_y - \partial_y u_x$, blows up at that points.

The calculation has been done for $Re = 1000$ and with $N = 9$. The mesh makes use of $K = 682$ elements, so that $dof = 28090$, and is adapted to the singularities, *i.e.*, smaller elements are used at the upper corners. At these points we simply enforce the no-slip condition, *i.e.* no sophisticated singularity treatment, like the one proposed in [4], is implemented. The vorticity field together with the mesh is shown in Fig. 3 (bottom). As it may be observed, despite the singular behavior of the solution the result is quite satisfactory, see *e.g.* [4] where the same isolines are shown. In this calculation, the value of the vorticity at the upper left corner is $\omega \approx 1000$.

5.3 Flow between eccentric cylinders

The computational domain Ω is now defined by two eccentric cylinders of radius R_1 and R_2 ($R_1 < R_2$), with eccentricity e defined as the distance between the two cylinders' centers. The case of a rotating inner cylinder is considered. As usual for this kind of flow, the clearance $c = R_2 - R_1$ is chosen as the characteristic length. For the sake of comparison, the geometry and flow parameters are identical to those given in [11, 32]: $R_1 = 1$, $R_2 = 2$, $e = 0.5$ and $Re = 37.2$, the Reynolds number being based on the inner cylinder tangential velocity. The main features of this flow are known to be controlled by the Taylor number defined as $T_n = Re\sqrt{c/R_1}$. When the inner cylinder is the one rotating, the flow remains two dimensional and laminar up to $T_n = 41.6$. Note that for the case considered $c = R_1$ and consequently $T_n = Re$. Essentially we plan here to demonstrate the necessity of using isoparametric elements.

Computations have been carried out, starting from the fluid at rest and with Dirichlet boundary conditions, with $N=9$ and $K=222$ standard triangles or isoparametric elements ($dof = 9315$). In the former case the cylinders are approximated by polygons whereas, in the latter case, polynomial approximations of degree $N=9$ are used to better approximate the geometry. Fig. 4 (top left) and Fig. 4 (top right) show the vorticity, as computed with triangles and with isoparametric elements, respectively. It is remarkable that if triangular elements are used, then maxima of vorticity appear at the vertices of the triangle. This is of course a numerical effect that can be avoided by using isoparametric elements. Fig. 4 (bottom left) shows the pressure and velocity field. The u_y velocity component is shown in Fig. 4 (bottom right) to point out the expected recirculation zone.

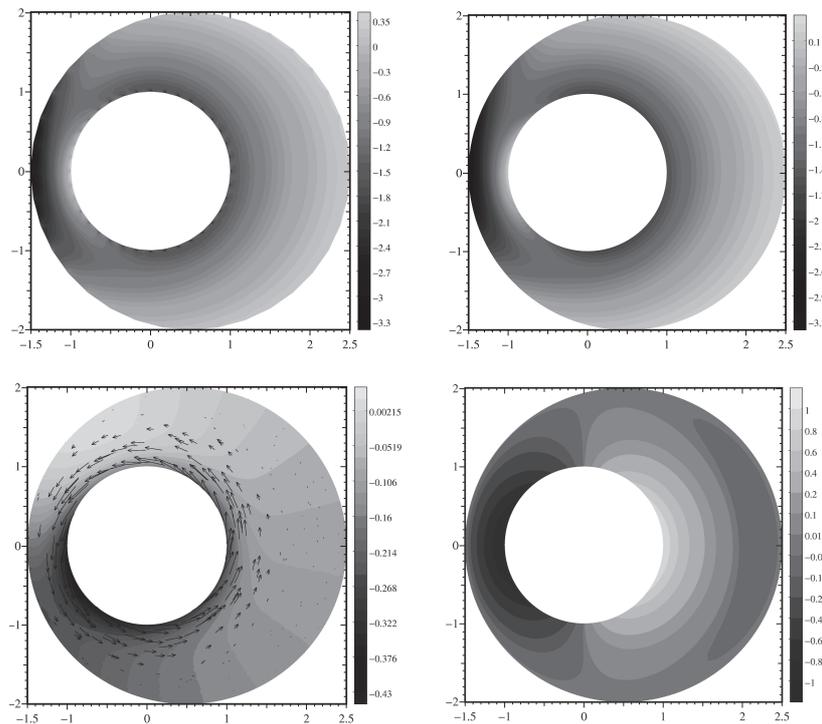


Figure 4: Eccentric cylinders: Top: Vorticity with straight triangles (left) and vorticity with isoparametric elements (right). Bottom: Pressure and velocity field (left) and y -component of velocity (right). Computations done with $(N=9, K=222)$.

5.4 Flow past a cylinder

Flow past a cylinder represents a good test to verify the capabilities of an unsteady NS code. For $Re \geq 48$, vortex shedding occurs at the cylinder and a von Karman street of vortices appears in the wake of the cylinder. For Reynolds numbers up to approximately 190, the flow is two-dimensional. Above this threshold, three-dimensional instabilities occur. Here we consider the 2D wake of a cylinder at Reynolds number ranging from $Re = 100$ to $Re = 200$ (the upstream flow velocity and the cylinder diameter are used as reference velocity and length, respectively). The computational domain Ω is the box $(-5, 23) \times (-6, 6)$, from which a circle of unit diameter and centered at the origin is extracted. Free slip boundary conditions are used at the upper and lower parts of the boundary, *i.e.* $\partial_y u_x = 0$ and $u_y = 0$ at $y = \pm 6$, and no-slip boundary conditions are used at the cylinder. At the inlet, say at $x = x_{in}$, the velocity is constant and equal to $\mathbf{u}_{in} = (1, 0)$. Such a flow is known to be laminar but unsteady, with a vortex shedding phenomenon characterized by the Strouhal number, St , which is the dimensionless shedding frequency (the ratio of the reference length and inflow velocity is used as reference time).

Here the main problem is that we have an open flow, so that soft outflow boundary

conditions (OBC) should be implemented at the outlet, say at $x = x_{out}$. It has turned out that computations done with natural boundary conditions for the velocity components, *i.e.* $\partial_x \mathbf{u} = 0$, yield unstable calculations. This may result from the fact that enforcing $\partial_x u_x = 0$ at $x = x_{out}$ implies $u_y = 0$ through the continuity equation, $\partial_x u_x + \partial_y u_y = 0$, and the boundary conditions at $y = \pm 6$, $u_y = 0$. Moreover, using a stress free boundary condition is not trivial in the frame of projection methods, because involving a coupling between the pressure and the velocity components that precisely we want to avoid.

To overcome this difficulty we prefer using an “outlet zone” within which the flow is softly enforced to become parallel and constant, just as it is at the inlet. Then one can use at $x = x_{out}$ the same Dirichlet condition used at $x = x_{in}$, with full consistency with the continuity equation. This is done by introducing in the Navier-Stokes equations an appropriate forcing term. Here, we have used:

$$\partial_t \mathbf{u} + \boldsymbol{\omega} \times \mathbf{u} = -\nabla P + \nu \Delta \mathbf{u} + \mathbf{f}, \quad (5.3a)$$

$$\mathbf{f} = C \sin\left(\frac{x-x_0}{x_{out}-x_0} \frac{\pi}{2}\right) (\mathbf{u}_{in} - \mathbf{u}), \quad \text{if } x > x_0, \quad (5.3b)$$

$$\mathbf{f} = \mathbf{0}, \quad \text{if } x \leq x_0, \quad (5.3c)$$

where (x_0, x_{out}) defines the outlet zone and with C a control parameter.

Computations have been done with polynomial degree $N = 6$ and $K = 2358$ elements. We have then $dof = 42882$. The CFL number, based on a minimum value of the grid-size, is set to 0.5, so that the time-step $\tau \approx 2.610^{-3}$. The computational time per time-step and degree of freedom is about $36.5 \mu s$. For the outflow treatment we have taken $x_0 = 18$ and $C = 2$. Fig. 5 shows the vorticity field obtained for $Re = 200$, together with the mesh, once

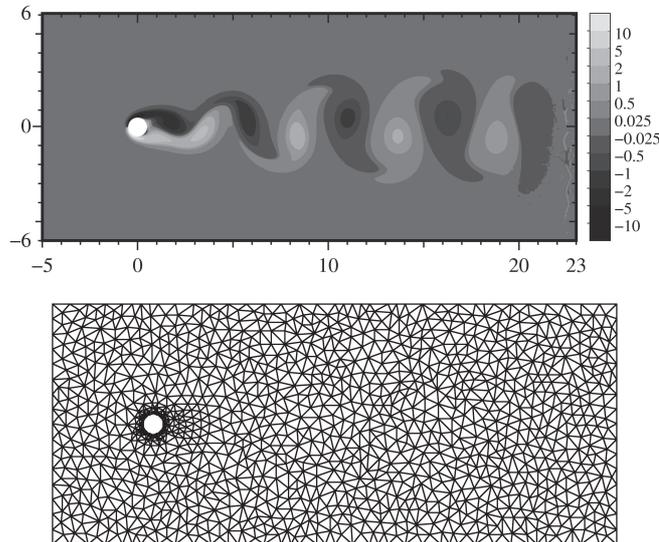


Figure 5: Vorticity (top) and mesh (bottom); Computation done with $(N = 6, K = 2358)$.

the unsteady flow is well established. The Strouhal number for $Re = 100$ and $Re = 200$ is equal to $St = 0.1775$ and $St = 0.2075$, respectively. Such values being a little higher than expected for the wake of a cylinder in an open domain, we have also enforced in the outlet zone a velocity profile showing a velocity deficit equal to 0.1 at $y = 0$ and whose integral preserves the flow rate. But for $Re = 100$ and $N = 6$ the Strouhal number is $St = 0.1771$, so that one may think that the outflow treatment has a negligible influence on the dimensionless frequency. One may rather suspect a confinement effect, the blocking factor being greater than 8 %. Thus, for $Re = 100$ and $N = 6$, computations done with a blocking factor 5 % has yielded $St = 0.1762$. Such a Strouhal number remains a little higher than expected, with respect to the experimental result and the numerical ones compiled in [21]. Note however that the SEM computation [20,21] has even yielded slightly larger values than ours, whereas less sophisticated methods, *e.g.*, based on penalization techniques to model the cylinder, seem presently to yield results in better agreement with the experiment [42].

6 Conclusion

Algorithms developed to set up a high order FEM for the incompressible Navier-Stokes equations have been carefully described. With respect to the standard \mathbb{P}_1 -FEM, the TSEM offers the possibility of much higher accuracy for the same number of degree of freedom whereas with respect to the SEM, it is much easier to handle complex geometries. The main goal being efficiency, one uses an implicit/explicit treatment of the diffusion/advection terms and a projection method, in order to avoid the pressure-velocity coupling. Then, the NS system splits into scalar PDE which are solved on the basis of the Fekete-Gauss TSEM approximation. Beyond the fact that simplicial meshes are supported, one important advantage of the TSEM approximation is its great flexibility, since the polynomial degrees for the interpolation and the quadrature are not linked as it is the case with the SEM. The TSEM differentiation matrices are however of larger size, but this is not really a drawback for reasonable polynomial approximation degrees. Isoparametric elements are employed to preserve the spectral accuracy when curved boundaries are involved. A Schur complement method is used to solve the resulting algebraic systems, which may be very large since the associated matrices are never assembled. Examples of applications have been presented. All elements of the proposed methodology were described for 2D problems but extend naturally to 3D ones. Nowadays one should however use interpolation points different from the Fekete ones, *e.g.* the warp and blend points [41], since up to our knowledge Fekete points remain to be determined for the tetrahedron. Progresses are also expected for efficient quadrature rules in the tetrahedron.

References

- [1] L. Bos, On certain configurations of points in R^n which are uniresolvent for polynomial interpolation, J. Approx. Theory, 64 (1991), 271-280.

- [2] L. Bos, M.A. Taylor, B.A. Wingate, Tensor product Gauss-Lobatto points are Fekete points for the cube, *Math. Comp.*, 70 (2001), 1543-1547.
- [3] O. Botella, On the solution of the Navier-Stokes equations using Chebyshev projection schemes with third order accuracy in time, *Comput. Fluids*, 116 (1997), 107-116.
- [4] O. Botella, R. Peyret, Benchmark spectral results on the lid driven cavity flow, *Comput. Fluids*, 27(4) (1998), 421-433.
- [5] A.J. Chorin, Numerical simulation of the Navier-Stokes equations, *Math. Comp.*, 22 (1968), 745-762.
- [6] R. Cools, Advances in multidimensional integration, *J. of Comput. and Appl. Math.*, 149 (2002), 1-12.
- [7] L. Demkowicz, T. Walsh, K. Gerdes, A. Bajer A., 2D hp-adaptative finite element package Fortran 90 implementation (2Dhp90), TICAM Report, 98-14 (1998).
- [8] M.O. Deville, P.F. Fischer, E.H. Mund, High-order methods for incompressible flows, Cambridge University Press, 2002.
- [9] M. Dubiner, Spectral methods on triangles and other domains, *J. Sci. Comput.*, 6 (1991), 345-390.
- [10] U. Ehrenstein, R. Peyret, A Chebyshev collocation method for the Navier-Stokes equations with application to double diffusive convection, *Int. J. Numer. Methods Fluids*, 9 (1989), 427-452.
- [11] M. Elghaoui, R. Pasquetti, Mixed spectral - boundary element embedding algorithms for the Navier-Stokes equations in the vorticity stream function formulation, *J. Comput. Phys.*, 153 (1999), 82-100.
- [12] F.X. Giraldo, T. Warburton A nodal triangle-based spectral element method for the shallow water equations on the sphere, *J. of Comput. Phys.*, 207 (1) (2005), 129-150.
- [13] W.J. Gordon, C. Hall, Transfinite element methods: Blending function interpolation over arbitrary curved element domains, *Numer. Math.*, 21 (1973), 109-129.
- [14] J.L. Guermond, P. Mineev, J. Shen, An overview of projection methods for incompressible flows, *Comput. Methods Appl. Mech. Engrg.*, 165 (2006), 6011-6045.
- [15] J.S. Hesthaven, From electrostatic to almost optimal nodal sets for polynomial interpolation in a simplex, *SIAM J. Numer. Anal.*, 35 (1998), 655-676.
- [16] J.S. Hesthaven, C.H. Teng, Stable spectral methods on tetrahedral elements, *SIAM J. Sci. Comput.*, 21 (2000), 2352-2380.
- [17] N.H. Hu, X.Z. Guo, I.N. Katz, Bounds for eigenvalues and condition numbers in the p -version of the finite element method. *Mathematics of Computation*, 67 (1998), 1423-1450.
- [18] S. Hugues, A. Randriamampianina, An improved projection scheme applied to pseudospectral methods for the incompressible Navier-Stokes equations, *Int. J. Numer. Meth. Fluids*, 28 (1998), 501-521.
- [19] G.E. Karniadakis, S.J. Sherwin, Spectral hp element methods for CFD, Oxford Univ. Press, London (1999).
- [20] G.E. Karniadakis, Numerical simulation of forced convection heat transfer from a cylinder in crossflow, *Int. J. Heat & Mass Transfer*, 31 (1988), 107-118.
- [21] K. Khadra, P. Angot, S. Parneix, J.P. Caltagirone, Fictitious domain approach for numerical modelling of the Navier-Stokes equations, *Int. J. Numer. Meth. Fluids*, 34 (2000), 651-684.
- [22] Y. Maday, A.T. Patera, E.M. Ronquist, An operator-integration-factor splitting method for time-dependent problems: application to incompressible fluid flow, *J. of Sci. Comput.*, 5 (4) (1990), 263-292.
- [23] J.M. Melenk, On condition numbers in hp-FEM with Gauss-Lobatto based shape functions,

- J. of Comp. and Appl. Math. 139 (2002) 21-48.
- [24] R. Pasquetti, F. Rapetti, Spectral element methods on unstructured meshes: comparisons and recent advances, *J. Sci. Comp.*, 27 (1-3) (2006), 377-387.
 - [25] R. Pasquetti, F. Rapetti, L. Pavarino, E. Zampieri, Neumann-Neumann Schur complement methods for Fekete spectral elements, *J. of Eng. Math.*, 56(3) (2006), 323-335.
 - [26] R. Pasquetti, F. Rapetti, Spectral element methods on unstructured meshes: which interpolation points ?, *Numerical Algorithms*, 55(2) (2010), 349-366.
 - [27] L.F. Pavarino, E. Zampieri, R. Pasquetti, F. Rapetti, Overlapping Schwarz methods for Fekete and Gauss-Lobatto spectral elements, *SIAM J. Sci. Comput.*, 29 (3) (2007), 1073-1092.
 - [28] J.P. Pontaza, A spectral element least-squares formulation for incompressible Navier-Stokes flows using triangular nodal elements, *J. Comput. Phys.*, 221 (2007) 649-665.
 - [29] V. Girault, P.A. Raviart, Finite element methods for Navier-Stokes equations. Theory and algorithms, Springer Verlag, Berlin (1986).
 - [30] B. F. Smith, P. E. Bjørstad, W. D. Gropp – Domain Decomposition. Parallel multilevel methods for elliptic partial differential equations, Cambridge University Press (1996).
 - [31] P. Solin, K. Segeth, I Dolezel, Higher-Order Finite Element methods Chapman & Hall/CRC Press (2003).
 - [32] D. Sood, H. Elrod, Numerical solution of the incompressible Navier-Stokes equations in doubly connected regions, *AIAA Journal*, 12(5) (1974), 636-641.
 - [33] A.H. Stroud, D. Secrest, Gaussian quadrature formulas, Prentice Hall (1966).
 - [34] A.H. Stroud, Approximate calculations of multiple integrals, Prentice Hall (1971).
 - [35] M.A. Taylor, B.A. Wingate, A generalized diagonal mass matrix spectral element method for non-quadrilateral elements, *Appl. Num. Math.*, 33 (2000), 259-265.
 - [36] M.A. Taylor, B.A. Wingate, R.E. Vincent, An algorithm for computing Fekete points in the triangle, *SIAM J. Numer. Anal.*, 38 (2000), 1707-1720.
 - [37] M.A. Taylor, B.A. Wingate, L.P. Bos A cardinal function algorithm for computing multivariate quadrature points, *SIAM J. Numer. Anal.* 45(1) (2007), 193205.
 - [38] R. Temam, Sur l'approximation de la solution des équations de Navier-Stokes par la méthode des pas fractionnaires II, *Archiv. rat. Mech. Anal.*, 33 (1969), 377-385.
 - [39] L.J. Timmermans, P.D. Mineev, F.N. Van De Vosse, An approximate projection scheme for incompressible flow using spectral elements, *Int. J. Numer. Math.*, 22 (1996), 673-688.
 - [40] T. Warburton, L. Pavarino, J.S. Hesthaven, A pseudo-spectral scheme for the incompressible Navier-Stokes equations using unstructured nodal elements, *J. Comput. Phys.*, 164 (2000), 1-21.
 - [41] T. Warburton, An explicit construction of interpolation nodes on the simplex, *J. Engrg. Math.* 56(3) (2006), 247-262.
 - [42] C.H.K. Williamson, Oblique and parallel modes of vortex shedding in the wake of a circular cylinder at low Reynolds numbers, *J. of Fluid Mechanics*, 206 (1989), 579-627.