Asociación Argentina
de Mecánica Computacional

Mecánica Computacional Vol XXV, pp. 2649-2667
Alberto Cardona, Norberto Nigro, Victorio Sonzogni, Mario Storti. (Eds.)
Santa Fe, Argentina, Noviembre 2006

SPECTRAL-ELEMENT IMPLEMENTATION OF THE KLE METHOD: A (ω, V) FORMULATION OF THE NAVIER–STOKES EQUATIONS

A. D. Otero and F. L. Ponta

Grupo ISEP, Facultad de Ingeniería, UBA, Paseo Colón 850, Buenos Aires C1063ACV, Argentina, email: aotero@fi.uba.ar

Keywords: spectral-element method, vorticity-velocity formulation, Navier-Stokes equations, time-space split algorithm.

Abstract. The kinematic Laplacian equation (KLE) method is a novel procedure belonging to the vorticity-velocity (ω, v) family known as the hybrid formulation of the Navier–Stokes equations. The results of its early classical FEM implementation exhibited satisfactory agreement with experimental measurements. However, thinking on future implementations of the KLE method, it is worth to know its behavior with different space and time discretization techniques. To this end, we started a systematic analysis of the particularities of a high-order implementation of the KLE by spectral-element techniques. Different aspects of the high-order implementation by spectral elements of this novel procedure are discussed in this work. The well-known problem of a semi-infinite region of stationary fluid bounded by an infinite horizontal flat plate impulsively started is used in different ways to conduct comparative evaluation tests. This time-dependant boundary-layer-development problem has an exact analytic solution, allowing to compare the numerical solution against it. Results are analyzed and conclusions presented.

1 INTRODUCTION

Based on what is now known as the hybrid formulation of the Navier–Stokes equations, the vorticity-velocity ($\boldsymbol{\omega}$, $\boldsymbol{v}$) methods evolved as a natural extension of the well-established vorticity-stream function methods which are based on the nonprimitive-variable Navier–Stokes formulation. The ($\boldsymbol{\omega}$, $\boldsymbol{v}$) methods present several advantages compared with the classical formulation on primitive variables (velocity-pressure) or with their vorticity-stream-function cousins (see, Clercx (1997); Ponta (2005); Quartapelle (1993); Speziale (1987), among others). We may single out the elimination of the pressure variable (which simplifies the study of incompressible flows on the inviscid limit and the treatment of boundary conditions at infinity in external flows) Quartapelle (1993), and their intrinsic invariance against acceleration of the frame of reference Speziale (1987). This makes them emerge as a very attractive alternative to the more classical approaches, especially for the solution of flows around bodies in complex rototranslational motion. A comprehensive study of the theoretical basis of the vorticity-velocity formulation in two and three dimensions can be found in chapter 4 of Quartapelle (1993).

Ponta (2005) introduced a novel procedure belonging to the ($\boldsymbol{\omega}$, $\boldsymbol{v}$). This procedure, called the KLE method, is characterized by a complete decoupling of the two variables in a vorticity-in-time/velocity-in-space split approach, thus reducing to three the number of unknowns to solve in the time integration process. This time-space splitting also favors the use of adaptive variable-stepsize/variable-order ODE algorithms, which enhances the efficiency and robustness of the time integration process. The KLE method solves the time evolution of the vorticity as an ordinary differential equation on each node of the spatial discretization. The input for the vorticity transport equation at each time-step is computed by a modified version of the Poisson equation for the velocity, which provides a linear PDE expression in weak form called the *kinematic Laplacian equation* (i.e. , KLE). The input of the KLE is provided by the time integration of the vorticity.

The KLE method is more a mathematical model than a numerical discretization scheme. The first implementation of the KLE method made use of classical low-order finite-element techniques for spatial discretization of the domain. The generality of the KLE method allows further exploration of different techniques for discretization in space and time. A particular point interest is the quality of the approximation of the spatial derivatives and the accuracy of the spatial discretization, specially for nodes that lie at the solid boundary which are the most involved in the critical process commonly known as *vorticity creation*. After good agreement of its first low-order FEM implementation with the experimental data we started a systematic analysis of the particularities of the KLE. In this work we focus on the spatial discretization technique. To this end, we implemented a spectral-element version of the KLE and tested different aspects of the method in order to individualize possible weaknesses and strengths.

2 THE KLE METHOD

In what follows we shall give a brief description of the KLE method, a detailed description of the mathematics and numerical involved in its first low order implementation can be found in Ponta (2005). Starting from the well-known vector identity

$$\nabla^2 \boldsymbol{v} = \boldsymbol{\nabla} \cdot \boldsymbol{\nabla} \boldsymbol{v} = \boldsymbol{\nabla}(\boldsymbol{\nabla} \cdot \boldsymbol{v}) - \boldsymbol{\nabla} \times (\boldsymbol{\nabla} \times \boldsymbol{v}), \tag{1}$$

we found that a variational form of this “Laplacian” expression could be advantageously used as the spatial counterpart of the vorticity transport equation in a new type of vorticity-velocity method.

Let us consider the full three-dimensional incompressible Navier–Stokes equation in vorticity form for a flow domain Ω with solid boundary $\partial\Omega$ and *external* boundary of Ω in the far field, in a moving frame of reference fixed to the solid,

$$\frac{\partial \boldsymbol{\omega}}{\partial t} = -\boldsymbol{v}\cdot\boldsymbol{\nabla}\boldsymbol{\omega} + \nu\nabla^2\boldsymbol{\omega} + \boldsymbol{\omega}\cdot\boldsymbol{\nabla}\boldsymbol{v}. \tag{2}$$

If we have the velocity field $\boldsymbol{v}$ in Ω at a certain instant of time, we can rewrite (2) as

$$\frac{\partial \boldsymbol{\omega}}{\partial t} = -\boldsymbol{v}\cdot\boldsymbol{\nabla}(\boldsymbol{\nabla}\times\boldsymbol{v}) + \nu\nabla^2(\boldsymbol{\nabla}\times\boldsymbol{v}) + (\boldsymbol{\nabla}\times\boldsymbol{v})\cdot\boldsymbol{\nabla}\boldsymbol{v}, \tag{3}$$

and solve for $\boldsymbol{\omega}$ at each point of the discretization of Ω by integration of (3) using an ODE solver.

Now, let us revisit (1) but this time impose a given distribution for the vorticity field and the rate of expansion:

$$\nabla^2\boldsymbol{v} = \boldsymbol{\nabla}\mathcal{D} - \boldsymbol{\nabla}\times\boldsymbol{\omega}, \tag{4}$$

$$\boldsymbol{\nabla}\cdot\boldsymbol{v} = \mathcal{D}, \tag{5}$$

$$\boldsymbol{\nabla}\times\boldsymbol{v} = \boldsymbol{\omega}. \tag{6}$$

Here $\boldsymbol{\omega}$ is the vorticity field in Ω given by (3) and $\mathcal{D}$ is the corresponding rate of expansion (i.e. the divergence field). The KLE is essentially defined as a solution of (4) in its weak form under the simultaneous constraints (5) and (6).

The imposition of the corresponding distributions for both the rate of expansion and the vorticity is needed in order to obtain a unique solution for the complete velocity field from equation (4). The first constraint defines the irrotational-not-solenoidal component of the velocity field, and the latter the solenoidal-not-irrotational component. If those two components are given, the remaining component (which is both solenoidal and irrotational) is uniquely determined for prescribed boundary conditions. A comprehensive treatment of this subject may be found in Batchelor (2000) Sec. 2.4 to 2.7. Usually, in other vorticity-velocity approaches the Poisson equation (4) is solved simultaneously with the vorticity transport equation together with an imposition of the incompressibility condition (i.e. a constant zero rate of expansion). With the KLE, instead, the objective is to uncouple the velocity and vorticity solutions. Hence, the imposition of the vorticity distribution is needed as a second constraint in order to obtain an independent solution of the velocity field.

For incompressible cases, such as discussed here, $\mathcal{D}$ is simply set to zero. For compressible cases, $\mathcal{D}$ can be a general distribution given by a solution analogous to (3) but for the divergence transport equation (i.e. the momentum equation in divergence form) together with a solution of the mass transport equation and adding to (2) and (3) the terms eliminated by the application of the incompressibility condition.

Now, provided that we can find a way of imposing on the velocity field the no-normal-flow condition

$$\boldsymbol{v}\cdot\boldsymbol{n} = 0, \tag{7}$$

and the no-slip condition

$$\boldsymbol{v}\cdot\boldsymbol{\tau} = 0, \tag{8}$$

on the solid boundary $\partial\Omega$ in a way compatible with the vorticity distribution at that time, we obtain a compatible solution for the velocity. Then, from this velocity field we produce the right-hand side of (3) required to advance the time-integration process to the next step. In order to impose the no-normal-flow and no-slip conditions on $\partial\Omega$ together with the correspondingly compatible boundary conditions on the vorticity, we designed an algorithmic sequence based on two consecutive solutions of the KLE: the first under free-slip and the second under no-slip boundary conditions on the solid surface. This algorithmic sequence is repeatedly performed inside the time-iteration process commanded by an adaptive variable-stepsize ODE solver. The solution is checked by the adaptive stepsize control by monitoring of the local truncation error, which proved to be quite stable for this application. During this process we apply the corresponding time-dependent, Dirichlet conditions for the velocity on $\partial\Omega_\infty$, the *external* boundary of Ω in the far field.

It is interesting to note that all the physics of the problem is contained in the time integration of (3) and it is solved as an ODE problem on the vorticity. The rest of the algorithm is concerned with the computation of a discrete spatial solution for the velocity field $\hat{\boldsymbol{V}}$ which is compatible with both: the time-evolved vorticity distribution obtained from (3) and the time-dependent boundary conditions for the velocity. The algorithmic sequence has the advantage of producing a complete decoupling between the time integration of the vorticity transport equation and the space solution of the Poisson equation for the velocity field.

2.1 Vorticity boundary conditions

A common issue to all the methods based on nonprimitive or hybrid variables is the absence of boundary conditions for the vorticity in presence of no-slip boundary conditions for the velocity. In the case of the ($\boldsymbol{\omega}$, $\boldsymbol{\psi}$) formulation it also implies that the Poisson problem for the stream function with both Dirichlet and Neumann conditions is overdetermined. There are several different ways of overcoming this difficulty (see Quartapelle (1993); Anderson (1988); Chorin (1973, 1978); Quartapelle and Valz-Gris (1981); Quartapelle (1981)).

In our approach, the issue of the vorticity boundary conditions on the no-slip surface is dealt with by a sequence of two solutions of the KLE under a different set of velocity boundary conditions. As we saw, inside each time step, we perform two projectional operations of integral character applied on the velocity field that ensure that the vorticity evolves in time in a way compatible with the time-dependent velocity boundary values. This issue is more related with the use of the KLE as the spatial counterpart in a ($\boldsymbol{\omega}$, $\boldsymbol{v}$) scheme than with the KLE in itself as a PDE system.

2.2 Variational formulation for the KLE

The linear spatial solution defined in (4)–(6) (i.e. , the KLE) can be implemented in just one variational formulation. Due to that the variational formulation of the Laplacian operator yields a symmetric and coercive bilinear form we can finding an associated functional. The minimization of that functional leads to an equivalent formulation to that obtained by applying the standard Galerkin method to (4), integrating by parts and considering the boundary conditions. All this also provide this formulation with good stability and convergence properties. The functional is

$$\Phi = \int_\Omega \frac{1}{2} \boldsymbol{\nabla v} : \boldsymbol{\nabla v} \, \mathrm{d}\Omega - \int_\Omega (\boldsymbol{\nabla} \times \boldsymbol{\omega}) \cdot \boldsymbol{v} \, \mathrm{d}\Omega. \tag{9}$$

The next step is the imposition of the constraints (5)–(6). To this end we explored several alternatives and we finally settled on the penalty method for the imposition of the constraints. The penalty method, though less rigorous than other alternatives with regard to the imposition of constraints, appears very well suited to this approach. It provides a solution in one step keeping the positive definiteness of the final bilinear form, it has proven to work properly in this coupled scheme of two simultaneous constraints and shows a wide range of stability for the values of the penalty constant used to impose the constraints without inducing ill-conditioning on the final stiffness matrix.

Thus, we modified the functional (9) by adding the penalty terms related to the constraints (5)–(6),

$$\tilde{\Phi} = \Phi + \int_\Omega \frac{\alpha_\mathcal{D}}{2}(\boldsymbol{\nabla}\cdot\boldsymbol{v})^2 + \frac{\alpha_\omega}{2}(\boldsymbol{\nabla}\times\boldsymbol{v}-\boldsymbol{\omega})\cdot(\boldsymbol{\nabla}\times\boldsymbol{v}-\boldsymbol{\omega})\,\mathrm{d}\Omega, \tag{10}$$

where $\tilde{\Phi}$ is the modified functional and $\alpha_\mathcal{D}$ and α_ω the corresponding penalty constants. We satisfactorily tested values of $\alpha_\mathcal{D}$ from 10^2 to 10^5 (choosing 10^3) and values of α_ω from 10^1 to 10^6 (choosing 10^2).

Invoking the stationarity of $\tilde{\Phi}$ with respect to $\boldsymbol{v}$, and reordering the terms we finally have

$$\begin{aligned}&\int_\Omega \boldsymbol{\nabla v} : \boldsymbol{\nabla\delta v} + \alpha_\mathcal{D}(\boldsymbol{\nabla}\cdot\boldsymbol{v})(\boldsymbol{\nabla}\cdot\boldsymbol{\delta v}) + \alpha_\omega(\boldsymbol{\nabla}\times\boldsymbol{v})\cdot(\boldsymbol{\nabla}\times\boldsymbol{\delta v})\,\mathrm{d}\Omega = \\ &\int_\Omega (\boldsymbol{\nabla}\times\boldsymbol{\omega})\cdot\boldsymbol{\delta v} + \alpha_\omega\boldsymbol{\omega}\cdot(\boldsymbol{\nabla}\times\boldsymbol{\delta v})\,\mathrm{d}\Omega,\end{aligned} \tag{11}$$

which is the expression for the variational formulation corresponding to the system (4)–(6) for the incompressible flow case.

This implementation leads to a global matrix which is independent both of time and of the particular constitutive relation of the continuum media. Then, this matrix can be factorized at the moment of assembling and its triangular factors used as many times as needed so long as we are using the same grid. As we said, this is so even for problems with different constitutive relations because all the physics of the problem is taken into account only in the time-integration process for the vorticity, i.e. the spatial solution is purely *kinematic*. Thus, the spatial solution computed at each time step reduces to a pair of back-substitution processes where we simply change the right-hand side vector of the linear system in order to impose consecutively the boundary conditions (7)–(8). This scheme simplifies the issue of obtaining a vorticity distribution on the no-slip boundaries in a way that satisfies the time-dependant boundary conditions for the velocity. Note that this procedure is not a purely-local manipulation performed on individual points on the boundary, this double solution of the velocity field is calculated over the entire domain involving two projectional operations of nonlocal character. In that sense, our procedure is more related with the abovementioned integral-constraint technique introduced by Quartapelle and Valz-Gris (1981); Quartapelle (1981), than with the purely-local operations of early vorticity-creation approaches.

2.3 Spectral-element implementation of the KLE method

The spectral element method is a particular implementation of the *p*-version of the *hp*-finite element method. It was originally introduced about two decades ago (see Patera (1984); Karniadakis et al. (1985)) in order to avoid the restriction suffered by global spectral methods of being only applicable to simple domains. This multidomain high-order method allows for local refinement, preserving the fast convergence properties of spectral discretizations (Henderson

and Karniadakis (1995)). It may handle complicated geometries while showing an exponential convergence rate which is faster than any algebraic method for smooth solutions. The spectral element method may use any type of Jacobi polynomial to compose its basis functions, the most common options are either Chebyshev or Legendre polynomials. Gauss-Lobatto quadrature points are commonly selected as collocation points leading to orthogonal basis functions which means that mass matrices are diagonal. An advantage of this procedure is that any order polynomial can be generated automatically concurrently with its numerical integration rule. There is also no need to define the basis functions explicitly because we may define implicit relations *a priori* for the inner products of the functions and their derivatives (Giraldo (1998)).

Here, we use typical two-dimensional isoparametric spectral elements where high-order Lagrangian polynomial interpolants are used to approximate the solution variables in each direction. The nodes correspond to the Gauss-Lobatto points, which for high-order elements is far more economical than equispaced nodes (Hourigan et al. (2001)). For a element of order p, $N_{GL} = p + 1$ is the number of nodes in one dimension, so that ${N_{GL}}^2$ is the total number of nodes per element.

The integrals involved are evaluated by Gauss-Legendre-Lobatto (GLL) quadrature. Even though, strictly speaking, GLL is a non-exact integration rule, this approach is particularly economical in computational terms because only a limited number of the element nodes contribute to the equations formed at a certain node. GLL quadrature have been in use for several years showing excellent results (see Thompson et al. (1996, 2001); Hourigan et al. (2001); Sheard et al. (2004), among others). Experiences conducted by Giraldo (1998) indicate that, for polynomial orders $p \geq 4$, results show no differences between non-exact GLL integration and classical exact Gauss-Legendre (GL) integration. We implemented a series of spectral element basis for increasing orders from $p = 2$ to $p = 21$. For $p \geq 4$, our first results showed no differences between non-exact GLL integration and exact GL. Hence, we continued using exact GL integration for $p < 4$ and GLL integration for $p \geq 4$.

Following the standard procedure for finite-element discretization of the velocity field and its gradient we have

$$\boldsymbol{v} = \begin{bmatrix} v_x \\ v_y \end{bmatrix} = \boldsymbol{H} \cdot \hat{\boldsymbol{V}}^{\boldsymbol{e}}, \qquad \boldsymbol{\nabla v} = \begin{bmatrix} \frac{\partial v_x}{\partial x} \\ \frac{\partial v_x}{\partial y} \\ \frac{\partial v_y}{\partial x} \\ \frac{\partial v_y}{\partial y} \end{bmatrix} = \boldsymbol{B} \cdot \hat{\boldsymbol{V}}^{\boldsymbol{e}}, \tag{12}$$

where $\hat{\boldsymbol{V}}^{\boldsymbol{e}}$ is the elemental array of nodal velocity values, $\boldsymbol{H}$ is the interpolation-function array and $\boldsymbol{B}$ the array of interpolation-function derivatives:

$$\hat{\boldsymbol{V}}^{\boldsymbol{e}} = \begin{bmatrix} \hat{v}_x^1 \\ \hat{v}_y^1 \\ \hat{v}_x^2 \\ \vdots \\ \hat{v}_x^{{N_{GL}}^2} \\ \hat{v}_y^{{N_{GL}}^2} \end{bmatrix}, \qquad \boldsymbol{H} = \begin{bmatrix} h^1 & 0 & h^2 & \cdots & h^{{N_{GL}}^2} & 0 \\ 0 & h^1 & 0 & \cdots & 0 & h^{{N_{GL}}^2} \end{bmatrix}, \tag{13}$$

$$\boldsymbol{B} = \begin{bmatrix} \frac{\partial h^1}{\partial x} & 0 & \frac{\partial h^2}{\partial x} & \cdots & \frac{\partial h^{N_{GL}{}^2}}{\partial x} & 0 \\ \frac{\partial h^1}{\partial y} & 0 & \frac{\partial h^2}{\partial y} & \cdots & \frac{\partial h^{N_{GL}{}^2}}{\partial y} & 0 \\ 0 & \frac{\partial h^1}{\partial x} & 0 & \cdots & 0 & \frac{\partial h^{N_{GL}{}^2}}{\partial x} \\ 0 & \frac{\partial h^1}{\partial y} & 0 & \cdots & 0 & \frac{\partial h^{N_{GL}{}^2}}{\partial y} \end{bmatrix}. \tag{14}$$

The partial derivatives of the interpolation functions are given by

$$\begin{bmatrix} \frac{\partial h^k}{\partial x} \\ \frac{\partial h^k}{\partial y} \end{bmatrix} = \boldsymbol{J}^{-1} \cdot \begin{bmatrix} \frac{\partial h^k}{\partial r} \\ \frac{\partial h^k}{\partial s} \end{bmatrix}, \qquad k = 1, \ldots, N_{GL}{}^2, \tag{15}$$

where $\boldsymbol{J}$ is the elemental Jacobian matrix

$$\boldsymbol{J} = \begin{bmatrix} \sum_{k=1}^{N_{GL}{}^2} \frac{\partial h^k}{\partial r} \hat{x}^k & \sum_{k=1}^{N_{GL}{}^2} \frac{\partial h^k}{\partial r} \hat{y}^k \\ \sum_{k=1}^{N_{GL}{}^2} \frac{\partial h^k}{\partial s} \hat{x}^k & \sum_{k=1}^{N_{GL}{}^2} \frac{\partial h^k}{\partial s} \hat{y}^k \end{bmatrix}, \tag{16}$$

and $(\hat{x}^k, \hat{y}^k)$ the geometrical coordinates of the nodes. For the divergence of the velocity field we have

$$\boldsymbol{\nabla \cdot v} = \boldsymbol{m \cdot B \cdot \hat{V}^e}, \qquad \boldsymbol{m} = \begin{bmatrix} 1 & 0 & 0 & 1 \end{bmatrix}, \tag{17}$$

and for the velocity curl the only non-zero component is ω_z, obtained as

$$\boldsymbol{\nabla \times v} = \boldsymbol{r \cdot B \cdot \hat{V}^e}, \qquad \boldsymbol{r} = \begin{bmatrix} 0 & -1 & 1 & 0 \end{bmatrix}. \tag{18}$$

Following a similar procedure for the discretization of the vorticity field and the x and y components of its curl we have

$$\boldsymbol{\omega} = \boldsymbol{H_\omega \cdot \hat{\omega}^e}, \qquad \boldsymbol{\nabla \times \omega} = \begin{bmatrix} \frac{\partial \omega}{\partial y} \\ -\frac{\partial \omega}{\partial x} \end{bmatrix} = \boldsymbol{B_\omega \cdot \hat{\omega}^e}, \tag{19}$$

where $\hat{\boldsymbol{\omega}}^{\boldsymbol{e}}$ is the nine-node elemental array of nodal vorticity values provided by the time-integration process, $\boldsymbol{H_\omega}$ is the vorticity interpolation-function array and $\boldsymbol{B_\omega}$ the array of interpolation-function derivatives for the computation of x and y components of the vorticity curl:

$$\hat{\boldsymbol{\omega}}^{\boldsymbol{e}} = \begin{bmatrix} \hat{\omega}^1 \\ \hat{\omega}^2 \\ \vdots \\ \hat{\omega}^{N_{GL}{}^2} \end{bmatrix}, \qquad \boldsymbol{H_\omega} = \begin{bmatrix} h^1 & h^2 & \cdots & h^{N_{GL}{}^2} \end{bmatrix}, \tag{20}$$

$$\boldsymbol{B_\omega} = \begin{bmatrix} \frac{\partial h^1}{\partial y} & \frac{\partial h^2}{\partial y} & \cdots & \frac{\partial h^{N_{GL}{}^2}}{\partial y} \\ -\frac{\partial h^1}{\partial x} & -\frac{\partial h^2}{\partial x} & \cdots & -\frac{\partial h^{N_{GL}{}^2}}{\partial x} \end{bmatrix}. \tag{21}$$

Now, considering (11) at each elemental subdomain (Ω^e) and substituting the velocity and vorticity fields and their differentiated magnitudes by their discretized counterparts we have

$$\boldsymbol{\delta \hat{V}^e}^T \cdot \underbrace{(\boldsymbol{K_L^e} + \boldsymbol{K_\mathcal{D}^e} + \boldsymbol{K_\omega^e})}_{\boldsymbol{K^e}} \cdot \boldsymbol{\hat{V}^e} = \boldsymbol{\delta \hat{V}^e}^T \cdot \underbrace{(\boldsymbol{R_L^e} + \boldsymbol{R_\omega^e})}_{\boldsymbol{R^e}} \cdot \boldsymbol{\hat{\omega}^e}, \tag{22}$$

where

$$\boldsymbol{K}_{\boldsymbol{L}}^{\boldsymbol{e}} = \int_{\Omega^e} \boldsymbol{B}^T \cdot \boldsymbol{B} \, \mathrm{d}\Omega = \int_{-1}^{1}\int_{-1}^{1} \boldsymbol{B}^T \cdot \boldsymbol{B} \, |\boldsymbol{J}| \, \mathrm{d}r\mathrm{d}s,$$

$$\boldsymbol{K}_{\boldsymbol{\mathcal{D}}}^{\boldsymbol{e}} = \int_{-1}^{1}\int_{-1}^{1} \alpha_{\mathcal{D}} \, \boldsymbol{B}^T \cdot \boldsymbol{m}^T \cdot \boldsymbol{m} \cdot \boldsymbol{B} \, |\boldsymbol{J}| \, \mathrm{d}r\mathrm{d}s,$$

$$\boldsymbol{K}_{\boldsymbol{\omega}}^{\boldsymbol{e}} = \int_{-1}^{1}\int_{-1}^{1} \alpha_{\omega} \, \boldsymbol{B}^T \cdot \boldsymbol{r}^T \cdot \boldsymbol{r} \cdot \boldsymbol{B} \, |\boldsymbol{J}| \, \mathrm{d}r\mathrm{d}s,$$

$$\boldsymbol{R}_{\boldsymbol{L}}^{\boldsymbol{e}} = \int_{-1}^{1}\int_{-1}^{1} \boldsymbol{H}^T \cdot \boldsymbol{B}_{\boldsymbol{\omega}} \, |\boldsymbol{J}| \, \mathrm{d}r\mathrm{d}s,$$

$$\boldsymbol{R}_{\boldsymbol{\omega}}^{\boldsymbol{e}} = \int_{-1}^{1}\int_{-1}^{1} \alpha_{\omega} \, \boldsymbol{B}^T \cdot \boldsymbol{r}^T \cdot \boldsymbol{H}_{\boldsymbol{\omega}} \, |\boldsymbol{J}| \, \mathrm{d}r\mathrm{d}s,$$

and $\boldsymbol{\delta}\hat{\boldsymbol{V}}^{\boldsymbol{e}}$ is the elemental array of nodal values for the arbitrary function $\boldsymbol{\delta v}$.

Assembling the elemental matrices and arrays defined in (22) and taking into account that $\boldsymbol{\delta v}$ is arbitrary and so it is its discretized counterpart $\boldsymbol{\delta}\hat{\boldsymbol{V}}$, we arrive to the global system

$$\boldsymbol{K} \cdot \hat{\boldsymbol{V}} = \boldsymbol{R} \cdot \hat{\boldsymbol{\omega}}. \tag{23}$$

In section 4.1, we test the accuracy of the spectral implementation of the KLE, by performing a *p*-refinement study, comparing against the analytical solution of a canonical boundary-layer problem described in section 3.

Substantial economy is additionally achieved, by the use of the static condensation procedure. Static condensation is particularly attractive for spectral element methods because of the natural division between the equations associated with element-boundary nodes and those associated with element-interior nodes. The condensed system is essentially the Schur complement of the interior-node submatrix in the non-condensed original system (Henderson and Karniadakis (1995)). This technique reduces the size and complexity of the stiffness matrices arising in finite-element and spectral-element methods and improves the condition number of the final condensed system (see Sec. 5.4.2 in Barrett et al. (1994), among others). The Schur complement inherits the symmetric positive definiteness of the original system, which guarantees good convergence properties for iterative solvers. Applying a preconditioned conjugated gradient (PCG) solver to the Schur complement constitutes the basis of the nonoverlapping substructuring approach (Warburton et al. (2000)). Preconditioners may range from simple diagonal scaling or incomplete factorizations to more efficient domain decomposition methods. The latter family includes the overlapping Schwartz methods (see Pavarino and Warburton (2000) and references therein) and the nonoverlapping substructuring approaches based on Schur-complement techniques applied at subdomain level. Due to their independent handling of the subdomains, domain decomposition methods have received a great deal of attention in recent years because they are very attractive for implementation in coarse-grain parallel systems (Barrett et al. (1994)). When implemented in parallel applications, domain-decomposition techniques increase the number of in-core operations but reduce the interprocessor data transfers. The current generation of massively parallel machines is *communications-limited* (Boyd (2000)). The rate of interprocessor transfers is considerably slower than the rate of flops each processor is capable to perform. Thus, the actual limiting factor is interprocessor communication rather than CPU time. In section 4.3, we study the condition number of the KLE stiffness matrix and the number of iterations of a PCG solution, for different types of preconditioning, polynomial order and mesh local refinement.

The above mentioned static condensation of interior nodes is made by following the classical procedure for elemental condensation (see Bathe (1996)), i.e. we rewrite the system $\boldsymbol{K}^{\boldsymbol{e}} \cdot \hat{\boldsymbol{V}}^{\boldsymbol{e}} = \boldsymbol{R}^{\boldsymbol{e}} \cdot \hat{\boldsymbol{\omega}}^{\boldsymbol{e}}$ as

$$\begin{bmatrix} \boldsymbol{K}^{\boldsymbol{e}}{}_{aa} & \boldsymbol{K}^{\boldsymbol{e}}{}_{ab} \\ \boldsymbol{K}^{\boldsymbol{e}}{}_{ba} & \boldsymbol{K}^{\boldsymbol{e}}{}_{bb} \end{bmatrix} \cdot \begin{bmatrix} \hat{\boldsymbol{V}}^{\boldsymbol{e}}{}_{a} \\ \hat{\boldsymbol{V}}^{\boldsymbol{e}}{}_{b} \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}^{\boldsymbol{e}}{}_{a} \\ \boldsymbol{R}^{\boldsymbol{e}}{}_{b} \end{bmatrix} \cdot \hat{\boldsymbol{\omega}}^{\boldsymbol{e}} \tag{24}$$

where subindex a indicates the velocity degrees of freedom associated with the exterior nodes and subindex b the velocity degrees of freedom associated with the interior nodes. From the second row of (24) we have

$$\hat{\boldsymbol{V}}^{\boldsymbol{e}}{}_{b} = \underbrace{(\boldsymbol{K}^{\boldsymbol{e}}{}_{bb})^{-1} \cdot \boldsymbol{R}^{\boldsymbol{e}}{}_{b}}_{\tilde{\boldsymbol{R}}^{\boldsymbol{e}}{}_{b}} \cdot \hat{\boldsymbol{\omega}}^{\boldsymbol{e}} - \underbrace{(\boldsymbol{K}^{\boldsymbol{e}}{}_{bb})^{-1} \cdot \boldsymbol{K}^{\boldsymbol{e}}{}_{ba}}_{\tilde{\boldsymbol{K}}^{\boldsymbol{e}}{}_{ba}} \cdot \hat{\boldsymbol{V}}^{\boldsymbol{e}}{}_{a}, \tag{25}$$

substituting this result into the first row of (24) and reordering terms,

$$\begin{aligned} &\underbrace{\left(\boldsymbol{K}^{\boldsymbol{e}}{}_{aa} - \boldsymbol{K}^{\boldsymbol{e}}{}_{ab} \cdot (\boldsymbol{K}^{\boldsymbol{e}}{}_{bb})^{-1} \cdot \boldsymbol{K}^{\boldsymbol{e}}{}_{ba}\right)}_{\tilde{\boldsymbol{K}}^{\boldsymbol{e}}} \cdot \hat{\boldsymbol{V}}^{\boldsymbol{e}}{}_{a} = \\ &\underbrace{\left(\boldsymbol{R}^{\boldsymbol{e}}{}_{a} - \boldsymbol{K}^{\boldsymbol{e}}{}_{ab} \cdot (\boldsymbol{K}^{\boldsymbol{e}}{}_{bb})^{-1} \cdot \boldsymbol{R}^{\boldsymbol{e}}{}_{b}\right)}_{\tilde{\boldsymbol{R}}^{\boldsymbol{e}}} \cdot \hat{\boldsymbol{\omega}}^{\boldsymbol{e}}, \end{aligned} \tag{26}$$

which defines the new condensed system to solve. Assembling the elemental matrices and arrays defined in (25) and (26) we finally arrive to the global condensed system

$$\tilde{\boldsymbol{K}} \cdot \hat{\boldsymbol{V}}_{a} = \tilde{\boldsymbol{R}} \cdot \hat{\boldsymbol{\omega}}, \tag{27}$$

$$\hat{\boldsymbol{V}}_{b} = \tilde{\boldsymbol{R}}_{b} \cdot \hat{\boldsymbol{\omega}} - \tilde{\boldsymbol{K}}_{ba} \cdot \hat{\boldsymbol{V}}_{a}, \tag{28}$$

which gives the solution for the complete velocity field $\hat{\boldsymbol{V}}$.

Neither $\tilde{\boldsymbol{K}}$, $\tilde{\boldsymbol{R}}$, $\tilde{\boldsymbol{R}}_b$ nor $\tilde{\boldsymbol{K}}_{ba}$ depend on $\hat{\boldsymbol{\omega}}$ nor t, so they can be computed once for a given mesh, stored and used as many times as needed to compute the solution for $\hat{\boldsymbol{V}}$. Matrix $\tilde{\boldsymbol{K}}$ is symmetric and positive definite, so it lends to factorization by Cholesky decomposition and its triangular factor is repeatedly used to solve $\hat{\boldsymbol{V}}_a$ through back-substitution.

Due to their high accuracy, spectral methods are *memory-minimizing* (Boyd (2000)). Even when a relatively-crude accuracy is needed, the high order of spectral approximations makes it possible to attain the modest error required with a considerably lower number of nodes. Hence, even though spectral elements generally require more computations per degree of freedom than low-order approximations, when an in-core solution is needed (either for a sequential procedure or as a subdomain computation within a parallel scheme), the use of spectral elements may be advantageous.

Shock waves in compressible flow and other singularities which induce the so-called Gibbs phenomenon constitute a source of trouble for both the spectral and the spectral-element methods (Pathria and Karniadakis (1994)). Irregularities due to the presence of nonsmooth coefficients, nonsmooth forcings and abrupt changes in boundary shape or boundary conditions, degrade the accuracy of the spectral element method and exponential convergence is lost. This is a consequence of the intrinsic problem of using high-order polynomial interpolations for nonsmooth functions. In section 4.1, we have payed special attention to this subject, analyzing the error induced at the early stages of a boundary-layer development where an impulsive start introduces a singularity at the solid surface.

2.4 Evaluation of the right-hand side of the ODE system

For the two-dimensional implementation of the time-integration procedure, we rewrote the vorticity transport equation (3) in a more convenient way,

$$\frac{\partial \boldsymbol{\omega}}{\partial t} = \boldsymbol{F}\left(\boldsymbol{\omega}, t\right) = \boldsymbol{\nabla} \times \left(\nu\, \boldsymbol{\nabla} \cdot \boldsymbol{\nabla} \boldsymbol{v} - \boldsymbol{v} \cdot \boldsymbol{\nabla} \boldsymbol{v}\right). \tag{29}$$

We evaluated the right-hand side of (29) applying the corresponding differential operators onto the discrete velocity field $\hat{\boldsymbol{V}}$ that was computed by the algorithmic sequence described in section (2).

For the low-order FEM implementation, we applied the normal procedure to calculate derivatives on the nodes of a mesh of isoparametric elements consisting in computing the derivatives at the standard Gaussian points adjacent to each node and interpolate their results using area-weighing interpolation which prove to be very effective. For the spectral-element case, the Gauss-Lobatto points coincide with the nodes. Thus, for the nodes on the inter-element boundaries, we used a simple average of the values from adjacent elements. For the interior nodes, calculation is straightforward.

The contribution of each Gaussian point to its corresponding node only depends on the geometry of the mesh and can be calculated at the moment of assembling. Hence, a set of arrays is assembled simultaneously with the finite-element matrices. Those arrays perform the differential operations on any vector or tensor field, as a dot product with the corresponding discrete solution of that field, implying a derivation/smoothing process. For instance, the discrete form of the curl of the velocity field $\boldsymbol{\nabla} \times \boldsymbol{v}$ is given by the dot product $\hat{\boldsymbol{C}}_{\boldsymbol{url}} \cdot \hat{\boldsymbol{V}}$. Thus, the right-hand side of (29) takes the discrete form

$$\boldsymbol{F}(\hat{\boldsymbol{\omega}}, t) = \hat{\boldsymbol{C}}_{\boldsymbol{url}} \cdot \left(\nu\, \hat{\boldsymbol{D}}_{\boldsymbol{iv}} - \hat{\boldsymbol{V}}_{\boldsymbol{adv}}\right) \cdot \hat{\boldsymbol{G}}_{\boldsymbol{rad}} \cdot \hat{\boldsymbol{V}}, \tag{30}$$

where $\hat{\boldsymbol{C}}_{\boldsymbol{url}}$, $\hat{\boldsymbol{G}}_{\boldsymbol{rad}}$ and $\hat{\boldsymbol{D}}_{\boldsymbol{iv}}$ are respectively the arrays that compute the curl, the gradient and the divergence of the gradient, and $\hat{\boldsymbol{V}}_{\boldsymbol{adv}}$ is simply a reordering of $\hat{\boldsymbol{V}}$ array to perform the product $\boldsymbol{v} \cdot \boldsymbol{\nabla} \boldsymbol{v}$ in the advective term. The accuracy of these operators constitutes a particular point of interest for us and is currently the subject of our research.

Neither $\hat{\boldsymbol{C}}_{\boldsymbol{url}}$, $\hat{\boldsymbol{G}}_{\boldsymbol{rad}}$ nor $\hat{\boldsymbol{D}}_{\boldsymbol{iv}}$ depend on $\hat{\boldsymbol{\omega}}$ nor t, so they can also be computed once for a given mesh, stored and used as many times as needed to provide evaluation of (30) for an advanced package ODE solver. We choose a multivalue variable-order Adams-Bashforth-Moulton predictor-corrector (ABM-PECE) solver with adaptive stepsize control which proved to be quite efficient for this application. We also tried a fifth order adaptive-stepsize Runge-Kutta algorithm with good results. For the first DNS low-Reynolds-number applications of the KLE method, the function prove to be smooth enough for the adaptive ABM-PECE algorithm to work very efficiently, in these smooth cases the predictor-corrector outperforms other alternatives like the Bulirsch-Stoer method (Press et al. (2002)).

3 BENCHMARK: BOUNDARY-LAYER DEVELOPMENT.

To conduct our comparative evaluation tests, we choose the well-known problem of a semi-infinite region of stationary fluid bounded by an infinite horizontal flat plate at $y = 0$, which is suddenly given a velocity U in its own plane and thereafter maintained at that speed. This problem has an exact analytic solution (see Batchelor (2000), Sec.4.3, among others). The velocity field described in a frame of reference fixed to a plate moving in the minus-x direction

is

$$u(y,t) = U \,\mathrm{erf}\left(\frac{y}{\sqrt{4\,\nu\,t}}\right), \tag{31}$$

where erf is the error function and y is the vertical coordinate. Rewriting (31) in terms of the normalized velocity u/U, the normalized vertical coordinate y/Y, and the parameter $\tau = \sqrt{4\,\nu\,t}/Y$, we have

$$\frac{u}{U} = \mathrm{erf}\left(\frac{y/Y}{\tau}\right), \tag{32}$$

where Y is the height of the test mesh.

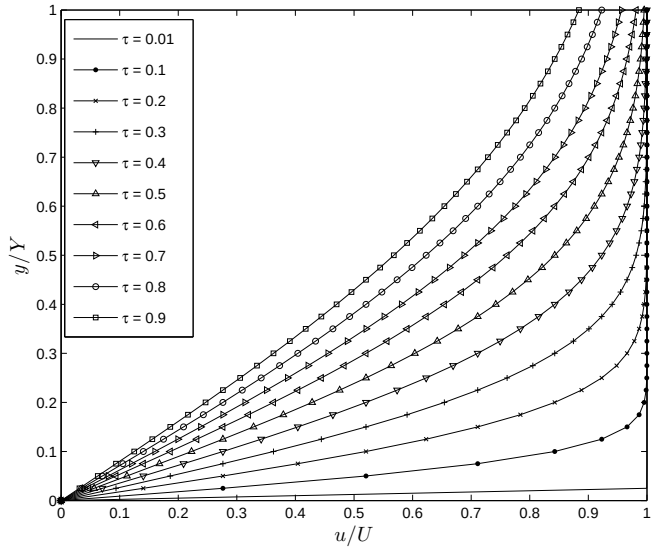

Figure 1: Velocity profile at successive values of the viscous time τ for the exact solution given by (32). For each value of τ the normalized boundary-layer thickness δ is also given.

Figure 1 shows the velocity profile at successive values of the parameter τ for the exact solution given by (32). The normalized vorticity distribution for this incompressible flow is given by the Gaussian function

$$\frac{\omega}{U/Y} = \frac{2}{\tau\,\sqrt{\pi}}\, e^{-\left(\frac{y/Y}{\tau}\right)^2}. \tag{33}$$

For a specified time, the analytic solution for the velocity and vorticity fields are given, respectively, by the Gaussian and the error function of the spatial coordinate. The latter prevents the occurrence of the trivial case in which the analytic solution coincides exactly with any of the polynomial interpolant functions associated to the spectral-element technique.

4 NUMERICAL TESTS

4.1 Accuracy of the spatial solution of the KLE

This series of tests are aimed to exploring the accuracy of the spatial solution provided by the KLE as a PDE system. Expression (4) by itself is no more than the classical Poisson equation for the velocity, which represents a special case of the well known Helmholtz equation that

has been thoroughly studied (and even used as a benchmark) for spectral-element applications. Nevertheless, with two penalty forms applied simultaneously on the variational formulation, a more detailed analysis is required in order to verify if the ability of exponential convergence of the spectral-element discretization is hindered in any way.

To this end, we designed an experiment in which we compared the theoretical velocity distribution given by expression (32) with the numerical solution for the velocity field provided by the KLE when the corresponding vorticity distribution (33) is used as source term. We solved the spatial problem at several stages of the development of the boundary layer computing the infinity norm of the error over the nodes of the mesh. We used a *minimal* regular mesh of two spectral elements in each dimension in order to keep the multidomain nature of the discretization, see figure 5 (left). Figure 2 shows the error curves for a *p*-refinement study for $2 \leq p \leq 20$ and successive values of τ at the early stages of development of the boundary layer, where the effects of the impulsive-start discontinuity are present. The *h*-refinement curves for a Q_2-FEM discretization for $\tau = 0.01$ and $\tau = 0.15$ are included for comparison purposes. In low-order classical finite elements it is usual to plot results against the internodal distance h, but spectral elements has not equidistant-nodes so it is meaningless to speak about an internodal distance. Consequently, we ploted our results against the number of intervals N^* in one dimension, which is one less the number of nodes and the inverse of the mean internodal distance.

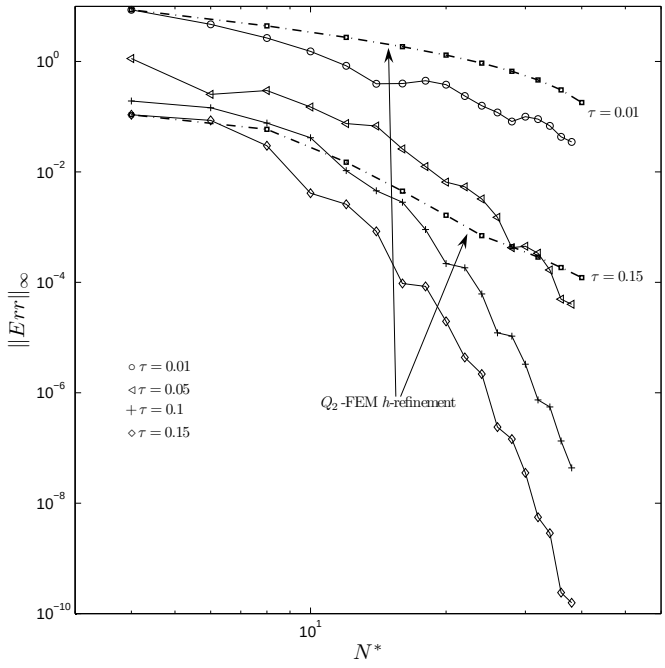

Figure 2: *p*-refinement error curves for $2 \leq p \leq 20$ and successive values of τ at the early stages of development of the boundary layer, where the effects of the impulsive-start discontinuity are present. *h*-refinement curves for a Q_2-FEM discretization for $\tau = 0.01$ and $\tau = 0.15$ are included for comparison purposes.

Here it is worthwhile to note that, when used inside the time-marching process of the vorticity-velocity scheme, the source term for the KLE solution at a given time is provided by a computation made by the ODE integrator from an approximation in weak form of the velocity field at the previous time step. This has the tendency to smooth out the shock introduced at the initial

stage. Hence, forcing the theoretical vorticity distribution given by expression (33) at the initial stages as a source term for the KLE posses a very strict trial on the KLE solution. This sharp forcing is actually more challenging than KLE's normal *operational* requirements as the spatial counterpart in a vorticity-velocity scheme. Taking into account the well-known difficulties of spectral elements when approximating nonsmooth functions, a particular point of interest here is the response of the spectral-element discretization to the initial discontinuity. As it was expected, at the early stage $\tau = 0.01$ the *p*-refinement curve shows a rather modest improvement (of roughly one order of magnitude) with respect to the Q_2-FEM *h*-refinement curve. Nevertheless, this situation quickly changes, and for $\tau = 0.15$ the improvement is already of six orders of magnitude.

Figure 3 shows the error curves for a *p*-refinement study for $2 \leq p \leq 20$ and successive values of τ at several stages of development of the boundary layer. The *h*-refinement curves for a Q_2-FEM discretization for $\tau = 0.2$ and $\tau = 0.9$ are included for comparison purposes. From $\tau = 0.2$ on, all the curves show exponential convergence. As the function gets more smooth with increasing values of $\tau = 0.2$, the rate of convergence increases and the curves become more and more smooth. All the curves progress with a decreasing of the error up to the point where they *hit* the lower limit imposed by the accumulation of roundoff error involved in the matrix-inversion procedure. In this case, we used a Cholesky decomposition followed by backwards substitution, which locates the lower limit in the range of $10^{-14} \approx 10^{-13}$. From that point on, all the curves follow the same path.

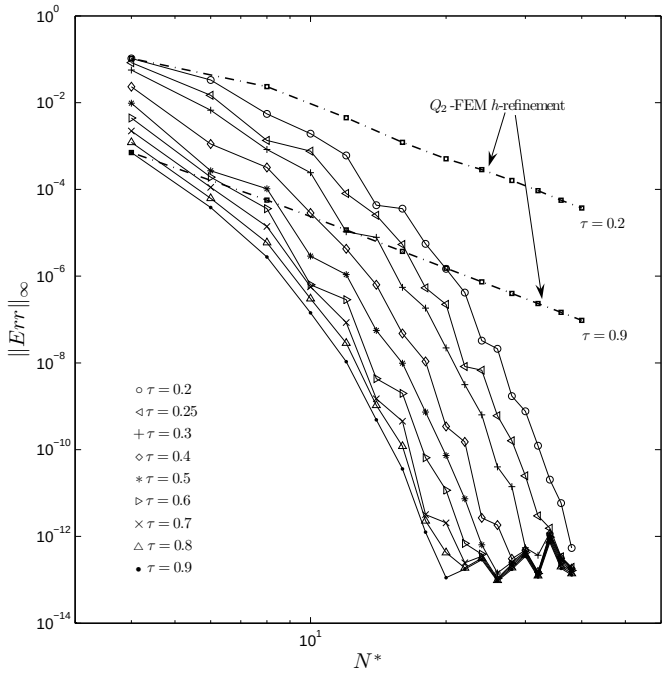

Figure 3: *p*-refinement error curves for $2 \leq p \leq 20$ and successive values of τ at several stages of development of the boundary layer. *h*-refinement curves for a Q_2-FEM discretization for $\tau = 0.2$ and $\tau = 0.9$ are included for comparison purposes.

4.2 Effects of domain truncation

Domain truncation represents the simplest strategy for solving problems on unbounded regions because it requires no modifications of the standard techniques for bounded domains. In particular, if the solution decays exponentially with the geometrical distance as one moves outwards form the origin, then the error in approximating the infinite domain by a finite (but sufficiently large) one will decrease exponentially with the domain size Boyd (2000). The benefits of using what is essentially the same code to solve both bounded and unbounded domains are huge, and domain truncation is actually a widespread adopted strategy. Following the definition given in Boyd (2000), a suitable estimate for the domain-truncation error for the problem presented in section 3 is

$$E_{DT}(Y) \equiv \max_{y \geq Y} \left(|1 - u(y)/U| \right). \tag{34}$$

This "estimate" is based on a definition of the domain-truncation error as merely the maximum value of the exact solution outside the limits of the truncated domain. The error in the approximate solution to a differential equation ($u_N(y)/U$) is another matter, and may be much larger than $E_{DT}(Y)$ (Boyd (2000)). Thus, $E_{DT}(Y)$ should rather be understood as an approximation of the theoretical lower bound for the domain-truncation error. Even though the *Assumption of Equal Errors* asserts that $\max |(u(y) - u_N(y))/U| \sim \mathcal{O}\left(E_{DT}(Y)\right)$ for most real-world problems, one can contrive examples for which this is not true (Boyd (2000)). Although the multidomain nature of the discretization technique employed here is not likely to produce oscillations associated with the Gibbs phenomenon, it is important for us to asses the actual effects of domain truncation on our solution, specially as p increases. We designed a numerical experiment to test the response of the spatial solution of the KLE when non-exact free-stream boundary conditions are imposed on the external pseudo-infinite border of the mesh, in place of the exact boundary conditions used in section 4.1 for both vorticity and velocity. Figure 4 (left) shows comparisons of the error curves for solutions on a 2x2-element mesh using exact and non-exact boundary conditions for $0.1 \leq \tau \leq 0.5$ and $p = 2, 4, 8, 17, 21$. As it was expected, the value of τ at which the effect of domain truncation starts to be felt increases with p as the approximation becomes more and more accurate. Nevertheless, in all cases both curves show no difference whatsoever until they *hit* the curve of $E_{DT}(Y)$ in function of τ. From that point on, the expected influence of the inexactitude of the boundary conditions is evident as all the curves closely follow $E_{DT}(Y)$, which provides the *theoretical* lower bound for the error. Taking into account that the boundary-layer thickness covers 55% of the entire domain for $\tau = 0.3$, 73% for $\tau = 0.4$, and 91% for $\tau = 0.5$, this results show that the KLE solution is quite tolerant to the use of non-exact boundary conditions. For comparison purposes, we conducted an *h*-refinement study on using our Q_2-FEM discretization. The latter results are shown in figure 4 (right).

4.3 Numerical properties of the KLE stiffness matrix

Regarding the size of the problem to be solved, the solution of equation (23) may be carried out using different techniques. For problems which for their size lends to be solved *in-core*, direct methods are advantageous. In case of big distributed problems, where a direct solution is impossible, it is mandatory to turn to iterative solvers. In both cases, the precision of the solution and the number of operations needed may be greatly improved by the use of some preconditioning for the matrix $\boldsymbol{K}$. For direct solvers, one common technique to improve the solution is to use the Schur complement of the matrix obtained by means of static condensation of the interior nodes. For the case of iterative solvers there are many types of preconditioners to use and one must find the better balance between the effort needed to obtain the preconditioning

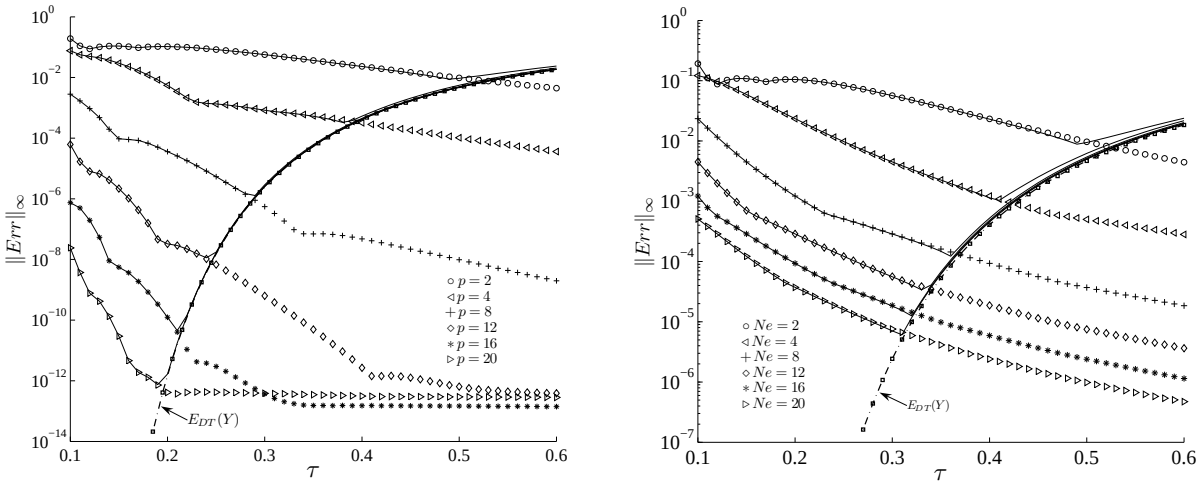

Figure 4: Comparative charts of the error curves for solutions using exact and non-exact boundary conditions for $0.1 \leq \tau \leq 0.6$. Left: for $p = 2, 4, 8, 17, 21$ and right: for an *h*-refinement of a Q_2-FEM discretization.

matrix and the saving in reducing the number of iterations. Here we tried two simple ones: the diagonal and the block diagonal preconditioners. Advanced preconditioning techniques, e.g. Schwartz overlapping methods among others, are a huge field and is our intention to study their properties when applied to the solution of the KLE in the future. At this point, we are interested in establishing the properties of the spectral–element discretization in the conventional schemes.

We use the *Condition Number*

$$\kappa = \lambda_{max}/\lambda_{min} \tag{35}$$

as an index to measure the quality of matrix $\boldsymbol{K}$ after preconditioning, against inversion. In (35), λ_{max} and λ_{min} are the maximun and minimun eigenvalue of $\boldsymbol{K}$ respectively. It is known that the largest eigenvalue of a finite-element matrix is related with the inverse of the small distance between nodes in the mesh. On the other hand, the smallest eigenvalue is related with some characteristic lenght of the mesh. So, using smaller elements in order to improve the solution makes the condition number to worsen. Taking this into account, we used structured non-regular meshes with the element edges located according to

$$\tilde{x} = x^\alpha \qquad \tilde{y} = y^\alpha$$

where x and y are the position of the edges in a mesh with equal square elements (i.e. uniform). Then changing the parameter α we can vary the properties of matrix $\boldsymbol{K}$. Here, we used two different meshes, constructed with $x, y = 0, 1/2, 1$: the first one with $\alpha = 1$ so that it is an uniform mesh of 4 square elements and the second, an irregular mesh with $\alpha = 2$. Both meshes are shown in figure 5.

We compared four different techniques for preconditioning the matrix comparing them with the non preconditioned matrix, so the five cases we tested are identified as

I - None

II - Diagonal preconditioner

III - Block Diagonal preconditioner

IV - Schur complement preconditioner

V - Schur complement + diagonal preconditioner

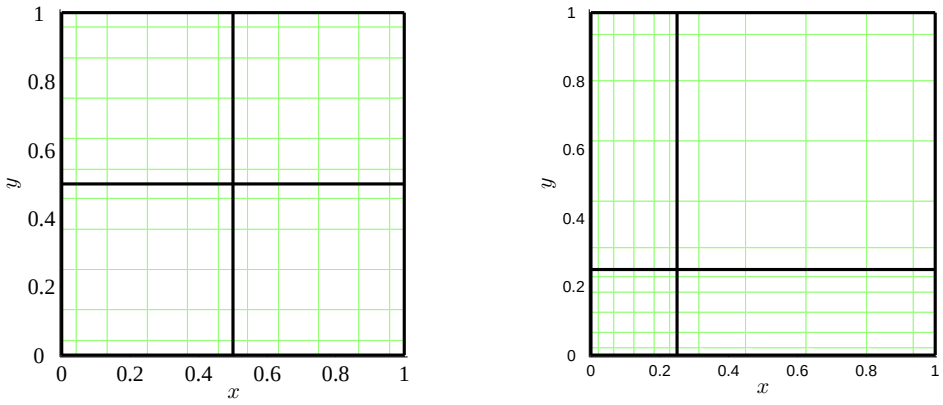

Figure 5: Two meshes of 2x2 elements with $p = 6$. Left $\alpha = 1$ and right $\alpha = 2$ with elements aspect- ratio 3. The thick lines indicate the element edge and the green line intersections show the location of the interior nodes.

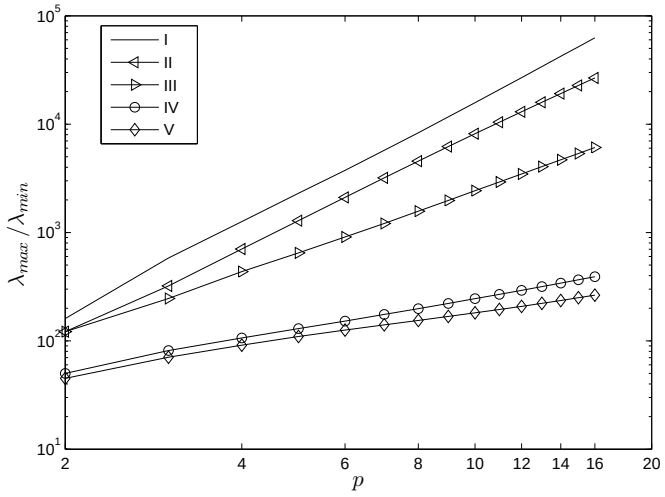

Figure 6: Condition number versus p, for the mesh with $\alpha = 2$.

In figure 6 we show a plot of condition number against p for the different preconditioned matrices in logaritmic scale for a mesh with deformation factor $\alpha = 2$. This implies an aspect-ratio 3 between the the largest and smallest dimension of elements. The scope of each curve in logarithmic scale represents the exponent of the dependance of κ on p. In the case of the matrix without preconditioning (I) the scope is almost 2.9, which may be considered the reference curve for this mesh. With the diagonal and block diagonal preconditioners the scope decreases

to almost 2.6 and 1.9, respectively. In the case of Schur complement preconditioning the scope is very near to 1 meaning a linear dependance of κ on p for this 4 element mesh. It further improves to almost 0.8 combined with a diagonal preconditioning of the condensed matrix.

Thinking on future aplications in distributed memory systems we also tested the behaviour of matrix $\boldsymbol{K}$ in a iterative *pcg* solver counting the number of iterations needed to achieve the convergence. The results are shown in table 1 for the meshes shown in figure 5.

	$\alpha = 1$					$\alpha = 2$				
p	I	II	III	IV	V	I	II	III	IV	V
2	8	8	8	6	6	17	15	15	11	11
4	42	37	29	14	13	96	69	57	27	27
6	99	90	50	22	21	198	149	89	37	32
8	161	148	74	28	25	321	234	125	43	36
10	230	211	95	32	30	462	325	154	47	41
12	311	279	114	35	33	612	411	187	51	45
14	405	346	136	37	36	765	498	214	56	46
16	493	414	155	40	37	931	587	242	60	52

Table 1: Number of iterations of the *pcg* solver.

It can be seen in table 1 that the number of iterations are substantially reduced with the two Schur complement techniques. It is also clear that matrices preconditioned with both techniques are less sensitive to aspect ratio increment. Moreover, with larger p this behaviour is more stressed. So, besides being specially suitable for multidoimain solution of problems, Schur complement techniques have the advantage of reducing the number of iterations needed in the solver with a low dependance on *p*-refinement. Moreover, as the order p increases, the number of condensable nodes into an element increases more than the number of non-condensable ones. Then, the relation of between non-condensed and condensed degrees of freedom diminishes, reducing the relative size of the the system to be solved in equation (23).

5 CONCLUDING REMARKS AND FUTURE PROSPECTS

In this work we presented the spectral element implementation of the KLE and showed the improvement achieved by this mean. Spectral convergence is attained in spite of the presence of the two penalizations, one for the divergence of the velocity and other for its curl.

We analized the solution by means of both direct and indirect methods of the resulting system of equations, showing the advantages of using subdomain techniques as the Schur complement to improve the condition number. These techniques can effectively reduce the communications and the number of *out-of-core* computations in distributed problems as well as within *in-core* solutions.

A particular point to address is the quality of the approximation of the spatial derivatives on the inter-elemental borders and the accuracy of the interpolation, specially for nodes that lie at the boundary. This last issue should be specially taken into account in the choice of space-discretization techniques and that is where our present effort is aimed to. We are conducting experiments to establish the way this spatial discretization modifies the calculation of the derived variables and the consecutive aplication of the $\hat{\boldsymbol{C}}_{\boldsymbol{url}}$, $\hat{\boldsymbol{G}}_{\boldsymbol{rad}}$ and $\hat{\boldsymbol{D}}_{\boldsymbol{iv}}$ arrays to compute the right-hand of equation (30).

The results shown in this work encourage us to implement a spectral-element discretization

on unstructured meshes. This could be done combining the tri–quadrilateral elements used in the former low-order implementation of the KLE with the high-order discretization exposed here.

In view to solve problems in complex geometries in three-dimensional applications which will require a substantial number of nodes (leading to large sparse systems) for the spatial discretization, it will be necessary to turn to a parallel version of the KLE code. The spectral element discretization combined with domain decomposition techniques are specially suited to that end, based in the posibility of condensing more nodes per element as the order p grows.

The generality of the KLE method allows further exploration of different techniques for discretization in space and time, which is the authors' intention. With this work we start dealing with the spatial discretization, we expect to take the ODE integration matter in the near future.

6 ACKNOWLEDGMENTS

We would like to acknowledge the financial support made available by the University of Buenos Aires through grant UBACyT-Pr.2004/07(I-56) and by Fundación Antorchas. A. Otero wants to thank the University of Buenos Aires and its College of Engineering for the support for his PhD studies.

REFERENCES

C. R. Anderson. Observations on vorticity creation boundary conditions. In R. E. Caflisch, editor, *Mathematical Aspects of Vortex Dynamics*, pages 144–159. SIAM, 1988.

R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. Van der Vorst. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods, 2nd Edition*. SIAM, Philadelphia, PA, USA, 1994.

G. K. Batchelor. *An introduction to fluid dynamics*. Cambridge University Press, Cambridge, UK, 2000.

K. J. Bathe. *Finite element procedures*. Prentice Hall, Englewood Cliffs, New Jersey, USA, 1996.

J. P. Boyd. *Chebyshev and Fourier spectral methods*. Dover, Mineola, New York, USA, 2000.

A. J. Chorin. Numerical study of slightly viscous flow. *J. Fluid Mech.*, 57:785–796, 1973.

A. J. Chorin. Vortex sheet approximation of boundary layers. *J. Comput. Phys.*, 27:428–442, 1978.

H. J. H. Clercx. A spectral solver for the Navier–Stokes equations in the velocity-vorticity formulation for flows with two nonperiodic directions. *J. Comput. Phys.*, 137:186–211, 1997.

F. X. Giraldo. The Lagrange–Galerkin spectral element method on unstructured quadrilateral grids. *J. Fluid Mech.*, 147:114–146, 1998.

R. D. Henderson and G. E. Karniadakis. Unstructured spectral element methods for simulation of turbulent flows. *J. Comput. Phys.*, 122:191–217, 1995.

K. Hourigan, M. C. Thompson, and B. T. Tan. Self–sustained oscilations in flows around long blunt plates. *J. Fluids Struct.*, 15:387–398, 2001.

G. E. Karniadakis, E. T. Bullister, and A. T. Patera. A spectral element method for solution of two- and three-dimensional time-dependent incompressible navier-stokes equations. In *Finite Element Methods for Nonlinear Problems*, page 803, New York/Berlin, 1985. Springer-Verlag.

A. T. Patera. A spectral element method for fluid dynamics: laminar flow in a channel expansion. *J. Comput. Phys.*, 54:468–488, 1984.

D. Pathria and G. E. Karniadakis. Spectral element method for elliptic problems in nonsmooth domains. *J. Comput. Phys.*, 122:83–95, 1994.

L. F. Pavarino and T. Warburton. Overlapping Schwraz methods for unstructured spectral elements. *J. Comput. Phys.*, 160:298–317, 2000.

F. L. Ponta. The kinematic Laplacian equation method. *J. Comput. Phys.*, 207:405–426, 2005.

W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical recipies in C, second edition*. Cambridge University Press, Cambridge, UK, 2002.

L. Quartapelle. Vorticity conditioning in the computation of two-dimensional viscous flows. *J. Comput. Phys.*, 40:453–477, 1981.

L. Quartapelle. *Numerical solution of the incompressible Navier-Stokes equations*. Birkäuser, Basel, Switzerland, 1993.

L. Quartapelle and F. Valz-Gris. Projections conditions on the vorticity in viscous incompressible flows. *Int. J. Numer. Meth. Fluids*, 1:129–144, 1981.

G. J. Sheard, M. C. Thompson, and K. Hourigan. From spheres to circular cylinders: non-axisymmetric transitions in the flow past rings. *J. Fluid Mech.*, 506:45–78, 2004.

C. G. Speziale. On the advantages of the velocity-vorticity formulation of the equations of fluid dynamics. *J. Comput. Phys.*, 73:476–480, 1987.

M. C. Thompson, K. Hourigan, and J. Sheridan. Three–dimensional instabilities in the wake of a circular cylinder. *Exp. Therm. and Fluid Science*, 12:190–196, 1996.

M. C. Thompson, T. Leweke, and C. H. K. Williamson. The physical mechanism of transition in bluff body wakes. *J. Fluids Struct.*, 15:607–616, 2001.

T. Warburton, L. F. Pavarino, and J. S. Hesthaven. A pseudo–spectral scheme for the incompresible Navier–Stokes equations using unstructured nodal elements. *J. Comput. Phys.*, 164: 1–21, 2000.