Computers & Fluids 76 (2013) 11–22

Contents lists available at SciVerse ScienceDirect

Computers & Fluids

journal homepage: www.elsevier.com/locate/compfluid

High-order implementation of the *kinematic Laplacian equation* method by spectral elements

F.L. Ponta [a,*], A.D. Otero [b,a]

[a] Department of Mechanical Engineering – Engineering Mechanics, Michigan Technological University, Houghton, MI 49931, USA
[b] CONICET & College of Engineering, University of Buenos Aires, Argentina

ARTICLE INFO

Article history:
Received 12 September 2012
Received in revised form 16 January 2013
Accepted 29 January 2013
Available online 16 February 2013

Keywords:
Navier–Stokes equations
Vorticity–velocity formulation
Spectral-element method

ABSTRACT

A novel high-order implementation for the Navier–Stokes equations in the vorticity–velocity formulation is presented. It is based on the kinematic Laplacian equation (KLE) method introduced in a previous work as a low-order finite-element approach. Different aspects of the high-order implementation by spectral elements of this novel procedure are discussed. 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 dependent boundary-layer-development problem has an exact analytic solution, and may be regarded as a canonical problem for the subject of generation of vorticity boundary conditions in vorticity–velocity approaches. Results are analyzed and conclusions presented.

1. Introduction

The emergence of vorticity–velocity methods might be considered one of the most recent innovations in the computational solution of time-dependent viscous flows. Even though the appearance of what could be regarded as the first $(\boldsymbol{\omega},\boldsymbol{v})$ approach may be traced as early as 1976 [1], it is rather during the last decade or so that a systematic research effort was applied to the development of this family of methods (see [2,3] for a complete list of references). Based on what is now known as the hybrid formulation of the Navier–Stokes equations, they 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 [2–5], 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) [4], and their intrinsic invariance against acceleration of the frame of Ref. [5]. 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 [4]. This reference also includes a series of theorems proving the equivalence between the $(\boldsymbol{\omega},\boldsymbol{v})$ formulation of the incompressible Navier–Stokes equations and their classical formulation in primitive variables (i.e. the velocity–pressure formulation). The equivalence theorem may also be found at [6].

In a previous paper, a novel procedure belonging to the $(\boldsymbol{\omega},\boldsymbol{v})$ family was introduced [3]. 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, especially for nodes that lie at the solid boundary which are the most involved in the critical

* Corresponding author. Tel.: +1 906 487 3563; fax: +1 906 487 2822.
E-mail address: flponta@mtu.edu (F.L. Ponta).

0045-7930/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.compfluid.2013.01.028

process commonly known as *vorticity creation*. Notwithstanding the results of its early low-order FEM implementation exhibit a satisfactory agreement with the experimental measurements, the abovementioned issues must be carefully taken into account in the choice of space-discretization techniques for future implementations of the KLE method. Following this line of reasoning, the purpose of this paper is to systematically analyze the particularities of a high-order implementation of the KLE by spectral-element techniques, focusing on each different aspect of this novel method to individualize possible weaknesses and strengths.

1.1. 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. Some earlier approaches like the *boundary vorticity formula* or the *vorticity creation* methods use different techniques to define the boundary values of vorticity in terms of the stream function (or the velocity) by means of some approximate formula applied locally at the no-slip boundary. They are roughly equivalent, however their implementation may differ remarkably depending on the type of discretization used (see [4,7–9]). An alternative viewpoint have been introduced by Quartapelle and Valz-Gris [10,11]. They showed that in order to satisfy the no-slip boundary conditions for the velocity, the vorticity should be subject to an integral constraint. This condition is a direct consequence of the boundary conditions on the velocity, and ensures satisfaction of essential conservation laws for the vorticity. An important aspect of the integral vorticity conditions is their nonlocal character: the vorticity distribution in the interior of the domain and on its boundary is affected at each time by the instantaneous values of the tangential and normal components of the velocity along the entire boundary. A detailed description of the mathematical basis and the different numerical implementations of the orthogonal-projection operation of the vorticity field for the $(\boldsymbol{\omega},\boldsymbol{v})$ formulation can be found in [4].

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 shall see, 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, which 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, will be discussed in following sections.

1.2. Outline of the KLE method

A detailed description of the mathematics and numerical implementation of the KLE method can be found in [3], including validation tests performed against analytical solutions and experimental measurements. Here, we shall give a brief outline of our method in order to make this article self contained. 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 Eq. (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 [12] Sections 2.4–2.7. 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. To clarify this point, let us consider the orthogonal decomposition of the velocity field in its irrotational not-solenoidal component $\boldsymbol{v}_{\mathcal{D}}$, its solenoidal not-irrotational component $\boldsymbol{v}_{\omega}$ and its irrotational and solenoidal (i.e., harmonic) component $\boldsymbol{v}_h$. Under prescribed boundary conditions for the normal component of the velocity and given distributions for the vorticity $\boldsymbol{\omega}$ and the rate of expansion $\mathcal{D}$, this decomposition $\boldsymbol{v} = \boldsymbol{v}_{\mathcal{D}} + \boldsymbol{v}_{\omega} + \boldsymbol{v}_h$ is uniquely determined [12]. Constraints (5) and (6) ensure that $\boldsymbol{v}_{\mathcal{D}}$ and $\boldsymbol{v}_{\omega}$ are properly solved:

$$\boldsymbol{\nabla}\cdot\boldsymbol{v} = \boldsymbol{\nabla}\cdot\boldsymbol{v}_{\mathcal{D}} = \mathcal{D}, \tag{7}$$

$$\boldsymbol{\nabla}\times\boldsymbol{v} = \boldsymbol{\nabla}\times\boldsymbol{v}_{\omega} = \boldsymbol{\omega}. \tag{8}$$

Now, applying the orthogonal decomposition to the total velocity field $\boldsymbol{v}$ in (4) we have,

$$\begin{aligned}\nabla^2(\boldsymbol{v}_h + \boldsymbol{v}_{\mathcal{D}} + \boldsymbol{v}_{\omega}) &= \nabla^2\boldsymbol{v}_h + \boldsymbol{\nabla}(\boldsymbol{\nabla}\cdot\boldsymbol{v}_{\mathcal{D}}) - \boldsymbol{\nabla}\times(\boldsymbol{\nabla}\times\boldsymbol{v}_{\omega})\\ &= \boldsymbol{\nabla}\mathcal{D} - \boldsymbol{\nabla}\times\boldsymbol{\omega}.\end{aligned} \tag{9}$$

Substituting (7) and (8) in (9) yields,

$$\nabla^2\boldsymbol{v}_h = 0, \tag{10}$$

which provides the solution of the harmonic component $\boldsymbol{v}_h$. Thus, the KLE construction ensures that all three components of the velocity field are properly solved.

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,

$$\boldsymbol{v}\cdot\boldsymbol{n}=0, \tag{11}$$

and the no-slip condition,

$$\boldsymbol{v}\cdot\boldsymbol{\tau}=0, \tag{12}$$

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.

Our algorithmic sequence is the analog of the *vorticity creation* process typically found in the early hybrid and nonprimitive methods mentioned above, where vorticity is *created* in response to the induced slip [7–9]. But, in our case, we produce our compatible vorticity boundary conditions on the solid surface by sequence of two solutions of the KLE under a different set of velocity boundary conditions. These two projectional operations of integral character applied on the velocity field (and performed inside each time step) ensure that the vorticity evolves in time in a way compatible with the time-dependent velocity boundary values.

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 $\widehat{\boldsymbol{V}}$ which is compatible with both: the time-evolved vorticity distribution obtained from (3) and the time-dependent boundary conditions for the velocity. This 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. A detailed scheme of the complete algorithmic sequence may be found in [3].

The linear spatial solution defined in (4)–(6) (i.e., the KLE) can be implemented in just one variational formulation. We start by applying the standard Galerkin method to (4). Then, integrating by parts, applying the Gauss theorem, and taking into account the boundary conditions, we arrive to a basic variational formulation associated to (4). One important property of the Laplacian operator is that its variational formulation yields a symmetric and coercive bilinear form with good stability and convergence properties. It also has an equivalent minimization formulation, which gives, for the variational form of (4), the following associated functional:

$$\Phi=\int_\Omega\frac{1}{2}\nabla\boldsymbol{v}:\nabla\boldsymbol{v}\,\mathrm{d}\Omega-\int_\Omega(\nabla\times\boldsymbol{\omega})\cdot\boldsymbol{v}\,\mathrm{d}\Omega. \tag{13}$$

The next step is the imposition of the constraints (5) and (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 (13) by adding the penalty terms related to the constraints (5) and (6),

$$\widetilde{\Phi}=\Phi+\int_\Omega\frac{\alpha_{\mathcal{D}}}{2}(\nabla\cdot\boldsymbol{v})^2+\frac{\alpha_\omega}{2}(\nabla\times\boldsymbol{v}-\boldsymbol{\omega})\cdot(\nabla\times\boldsymbol{v}-\boldsymbol{\omega})\mathrm{d}\Omega, \tag{14}$$

where $\widetilde{\Phi}$ 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).

Before applying the KLE scheme to the solution of flows around complex geometries, we conducted a series of experiments using the so-called *Patch-Test* technique to verify the accuracy of the imposition of constrains (5) and (6), and the absence of spurious modes when the mesh is distorted. Originally proposed by Irons and Razzaque [13] in the context of structural mechanics, the idea of the Patch-Test involves solving a structural problem subjected to boundary conditions that lead to a constant-stress state, with the minimum set of restrictions in order to avoid rigid translations. This problem is solved on a simple domain or *patch* discretized by an arbitrarily-distorted mesh of a small number of elements. If for any arbitrary patch of elements given the stresses and displacements are correctly predicted, the element is said to have passed the test. The idea originated from the observation that some incompatible elements (i.e. elements which present inter-elemental discontinuities) typically used in the analysis of plates and shells, still worked perfectly fine despite those incompatibilities, and the patch-test offered a way of determining which elements satisfy that condition (for a detailed description of the Patch-Test technique see [14], among others).

In the case of displacement-based elements correctly assembled in a compatible mesh, the satisfaction of the test is automatic. This is what happens in our case, even though our interpolated variable is the velocity, the analogy is a direct one. Nevertheless, conducting a patch-test in the case of a new type of element is a healthy practice to ensure that no spurious modes are going to appear when the mesh is distorted and, particularly for the KLE, to verify the correct satisfaction of constrains (5) and (6). To this end, we conducted an extensive series of tests over 1000 different patches consistent of meshes which were arbitrarily-distorted by adding randomly-generated displacements to the original locations of the nodes, subjected to boundary conditions of uniform stream. Fig. 1 shows two examples of 9-node isoparametric-element meshes used. The mean of the error in the divergence of the flow, i.e. the satisfaction of constrain (5), was 5.5536×10^{-15} with a maximum of 1.6135×10^{-14}. For the curl of the flow, i.e. the satisfaction of constrain (6), the results were 2.6207×10^{-14} for the mean and 1.3805×10^{-13} for the maximum. These results show that both conditions were satisfied almost up to the limit of precision of the floating-point representation, and no spurious modes were induced.

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

$$\int_\Omega\nabla\boldsymbol{v}:\nabla\boldsymbol{\delta v}+\alpha_{\mathcal{D}}(\nabla\cdot\boldsymbol{v})(\nabla\cdot\boldsymbol{\delta v})+\alpha_\omega(\nabla\times\boldsymbol{v})\cdot(\nabla\times\boldsymbol{\delta v})\mathrm{d}\Omega$$
$$=\int_\Omega(\nabla\times\boldsymbol{\omega})\cdot\boldsymbol{\delta v}+\alpha_\omega\boldsymbol{\omega}\cdot(\nabla\times\boldsymbol{\delta v})\mathrm{d}\Omega, \tag{15}$$

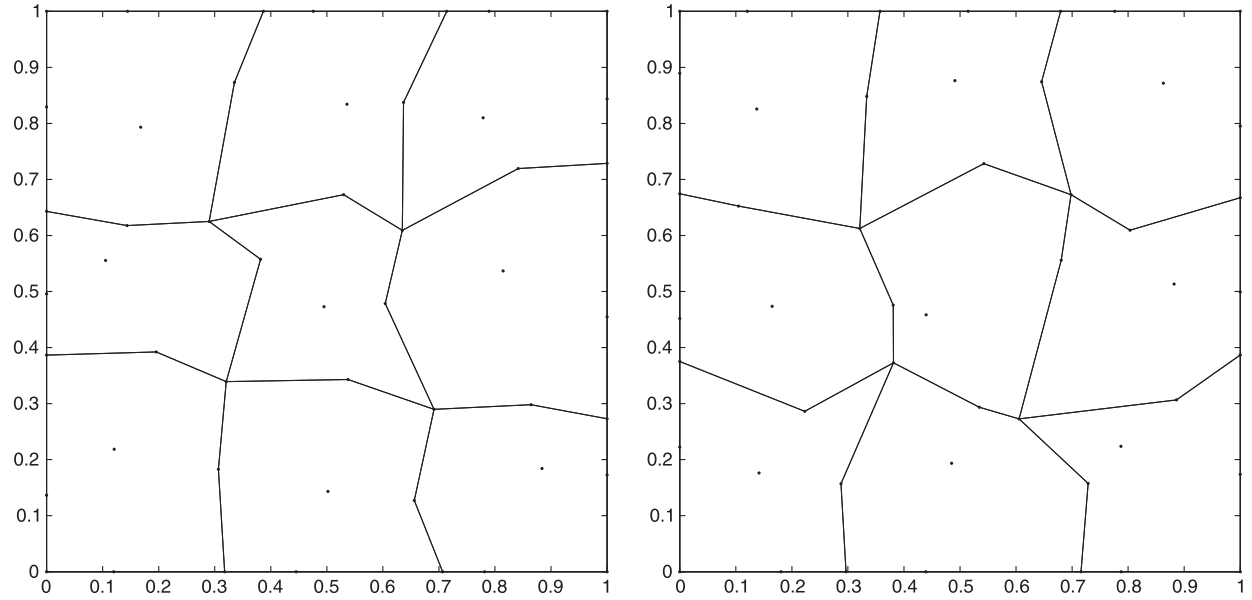

Fig. 1. Two examples of arbitrarily-distorted meshes used in the Patch-Test analysis obtained by adding randomly-generated displacements to the original locations of the nodes.

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 (11) and (12). This scheme simplifies the issue of obtaining a vorticity distribution on the no-slip boundaries in a way that satisfies the time-dependent 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 [10,11], than with the purely-local operations of early vorticity-creation approaches.

The basic formulation of the PDE system (4)–(6), i.e. the KLE, is three-dimensional. Then, the extension of the spatial solution provided by the KLE to three-dimensional problems is straightforward (the theoretical basis of the $(\boldsymbol{\omega}, \boldsymbol{v})$ formulation in three dimensions and its equivalence with the classical velocity–pressure formulation may be found in Quartapelle [4], chapter 4). The fact that in $(\boldsymbol{\omega}, \boldsymbol{v})$ approaches the velocity is supplemented by unique boundary conditions, substantially simplifies the question of computing a velocity field compatible with the vorticity boundary conditions. This is not the case with the vorticity–stream function methods, where the variety of boundary conditions that may be chosen for the velocity potentials due to the nonuniqueness of the velocity representation is much more complicated in three-dimensions than in two [4]. The KLE method has no special requirements on the rate-of-expansion distribution imposed. It implies that the method can be extended to the analysis of compressible flows, provided that we implemented an algorithmic sequence to compute compatible boundary conditions for the rate of expansion in an analog way as we do with the vorticity.

Even though in previous paragraphs we referred to the KLE method as a "vorticity-in-time/velocity-in-space *split* approach", this is more a general description of its time–space/vorticity–velocity uncoupled nature than a strict definition of its algorithmic structure. Strictly speaking, time-marching *splitting* or *fractional-step* methods replace simultaneous processes by sequential steps as a means to increase efficiency [15]. Split may be by dimensions (e.g. a three-dimensional process split into three one-dimensional substeps), or by physics (e.g. advection on one fractional step, pressure adjustment on another, and diffusion on a third). For the hydrodynamic equations, the advantage of splitting-by-process is that the nonlinear advection process can be treated by a different algorithm than pressure adjustment, which in turn can be different from diffusion, the latter two involve a linear solution each. The advective step is usually advanced explicitly and the adjustment of fields, is integrated implicitly. A typical example of this technique is the very successful AB3CN (third-order Adams–Bashforth/Crank–Nicholson) three-step scheme (see [16,17], among others). The choice of appropriate boundary conditions is important in minimizing the splitting error [18].

On the other hand, there is no splitting in the KLE method. All terms in the physical problem are solved simultaneously during time integration of the vorticity field, and all the spatial components of the velocity are solved together by the KLE. The fact that the linear spatial solution provided by the KLE is purely kinematic, with all the nonlinearities and the material constitutive properties remitted to the high-order adaptive time integration, favors the solution of problems with more complex nonlinear constitutive relations like non-Newtonian, plastic or viscoplastic flows. And the same argument may be applied to the adoption of turbulence models for a future LES implementation of the method. The KLE is based on a universal vectorial relation, so it can be used to solve any vector field provided that we may solve a transport equation for its divergence and curl. This, together with the fact that time

is the only iteration variable present, makes it possible to extend its application to other physical problems like, for instance, electromagnetic fields. It is also possible to couple the fluid analysis with other physical processes (e.g., heat transfer or chemical reaction) by adding more equations to the ODE system. In that sense, the flexibility of the KLE to deal with different problems, using what is in fact the same spatial formulation, substantially simplifies the analysis of its numerical performance: What we are doing is essentially solving the same PDE with different source terms. Hence, several conclusions obtained here regarding the numerical behavior of the KLE as a PDE system acquire a more general character which goes beyond the Navier–Stokes problem.

KLE also shows a substantial tolerance to the use of unstructured meshes which allows a more suitable meshing of complex geometries than structured-mesh approaches would permit. The KLE method was applied successfully to the study of the Strouhal–Reynolds number relationship for vortex streets [19], to the analysis of vortex structures in the wake of oscillating cylinders [20], and as a basis of a vortex-identification technique [21].

2. 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 [22,23]) 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 [24]. It may handle complicated geometries while showing an exponential convergence rate which is faster than any algebraic method for smooth solutions. Due to their high accuracy, spectral methods are *memory-minimizing* [15]. 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.

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 [25].

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 Galerkin method is applied to form the KLE equations for the solution of the velocity field at the nodal points (see the implementation of the KLE variational formulation included in [3]). The nodes correspond to the Gauss–Lobatto points, which for high-order elements is far more economical than equispaced nodes [26]. 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 [17,26–28], among others). Experiences conducted by Giraldo [25] indicate that, for polynomial orders $p \geqslant 4$, results show no differences between non-exact GLL integration and classical exact Gauss–Legendre integration. We implemented a series of spectral element basis for increasing orders from $p = 2$ to $p = 21$. For $p \geqslant 4$, 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 $p \geqslant 4$.

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

$$\boldsymbol{v} = \begin{bmatrix} v_x \\ v_y \end{bmatrix} = \boldsymbol{H} \cdot \widehat{\boldsymbol{V}}^e, \quad \nabla \boldsymbol{v} = \begin{bmatrix} \frac{\partial v_x}{\partial x} \\ \frac{\partial v_x}{\partial y} \\ \frac{\partial v_y}{\partial x} \\ \frac{\partial v_y}{\partial y} \end{bmatrix} = \boldsymbol{B} \cdot \widehat{\boldsymbol{V}}^e, \tag{16}$$

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

$$\widehat{\boldsymbol{V}}^e = \begin{bmatrix} \hat{v}_x^1 \\ \hat{v}_y^1 \\ \hat{v}_x^2 \\ \vdots \\ \hat{v}_x^{N_{GL}^2} \\ \hat{v}_y^{N_{GL}^2} \end{bmatrix}, \quad \boldsymbol{H} = \begin{bmatrix} h^1 & 0 & h^2 & \cdots & h^{N_{GL}^2} & 0 \\ 0 & h^1 & 0 & \cdots & 0 & h^{N_{GL}^2} \end{bmatrix}, \tag{17}$$

$$\boldsymbol{B} = \begin{bmatrix} \frac{\partial h^1}{\partial x} & 0 & \frac{\partial h^2}{\partial x} & \cdots & \frac{\partial h^{N_{GL}^2}}{\partial x} & 0 \\ \frac{\partial h^1}{\partial y} & 0 & \frac{\partial h^2}{\partial y} & \cdots & \frac{\partial h^{N_{GL}^2}}{\partial y} & 0 \\ 0 & \frac{\partial h^1}{\partial x} & 0 & \cdots & 0 & \frac{\partial h^{N_{GL}^2}}{\partial x} \\ 0 & \frac{\partial h^1}{\partial y} & 0 & \cdots & 0 & \frac{\partial h^{N_{GL}^2}}{\partial y} \end{bmatrix}, \tag{18}$$

where $N_{GL} = p + 1$ is the number of nodes of the Gauss–Lobatto interpolation.

The partial derivatives of the interpolation functions are given by

$$\begin{bmatrix} \frac{\partial h^k}{\partial x} \\ \frac{\partial h^k}{\partial y} \end{bmatrix} = \boldsymbol{J}^{-1} \cdot \begin{bmatrix} \frac{\partial h^k}{\partial r} \\ \frac{\partial h^k}{\partial s} \end{bmatrix}, \quad k = 1, \ldots, N_{GL}^2, \tag{19}$$

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

$$\boldsymbol{J} = \begin{bmatrix} \sum_{k=1}^{N_{GL}^2} \frac{\partial h^k}{\partial r} \hat{x}^k & \sum_{k=1}^{N_{GL}^2} \frac{\partial h^k}{\partial r} \hat{y}^k \\ \sum_{k=1}^{N_{GL}^2} \frac{\partial h^k}{\partial s} \hat{x}^k & \sum_{k=1}^{N_{GL}^2} \frac{\partial h^k}{\partial s} \hat{y}^k \end{bmatrix}, \tag{20}$$

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

$$\nabla \cdot \boldsymbol{v} = \boldsymbol{m} \cdot \boldsymbol{B} \cdot \widehat{\boldsymbol{V}}^e, \quad \boldsymbol{m} = [1 \quad 0 \quad 0 \quad 1], \tag{21}$$

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

$$\nabla \times \boldsymbol{v} = \boldsymbol{r} \cdot \boldsymbol{B} \cdot \widehat{\boldsymbol{V}}^e, \quad \boldsymbol{r} = [0 \quad -1 \quad 1 \quad 0]. \tag{22}$$

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

$$\boldsymbol{\omega} = \boldsymbol{H}_{\omega} \cdot \hat{\boldsymbol{\omega}}^{\boldsymbol{e}}, \quad \nabla \times \boldsymbol{\omega} = \begin{bmatrix} \frac{\partial \omega}{\partial y} \\ -\frac{\partial \omega}{\partial x} \end{bmatrix} = \boldsymbol{B}_{\omega} \cdot \hat{\boldsymbol{\omega}}^{\boldsymbol{e}}, \tag{23}$$

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

$$\hat{\boldsymbol{\omega}}^{\boldsymbol{e}} = \begin{bmatrix} \hat{\omega}^1 \\ \hat{\omega}^2 \\ \vdots \\ \hat{\omega}^{N_{GL}^2} \end{bmatrix}, \quad \boldsymbol{H}_{\boldsymbol{\omega}} = \begin{bmatrix} h^1 & h^2 & \cdots & h^{N_{GL}^2} \end{bmatrix}, \tag{24}$$

$$\boldsymbol{B}_{\boldsymbol{\omega}} = \begin{bmatrix} \frac{\partial h^1}{\partial y} & \frac{\partial h^2}{\partial y} & \cdots & \frac{\partial h^{N_{GL}^2}}{\partial y} \\ -\frac{\partial h^1}{\partial x} & -\frac{\partial h^2}{\partial x} & \cdots & -\frac{\partial h^{N_{GL}^2}}{\partial x} \end{bmatrix}. \tag{25}$$

Now, considering (15) at each elemental subdomain (Ω^e) and substituting the velocity and vorticity fields and their differentiated magnitudes by their discretized counterparts we have

$$\delta\hat{\boldsymbol{V}}^{\boldsymbol{e}^T} \cdot \underbrace{\left(\boldsymbol{K}_{\boldsymbol{L}}^{\boldsymbol{e}} + \boldsymbol{K}_{\mathscr{D}}^{\boldsymbol{e}} + \boldsymbol{K}_{\boldsymbol{\omega}}^{\boldsymbol{e}}\right)}_{\boldsymbol{K}^{\boldsymbol{e}}} \cdot \hat{\boldsymbol{V}}^{\boldsymbol{e}} = \delta\hat{\boldsymbol{V}}^{\boldsymbol{e}^T} \cdot \underbrace{\left(\boldsymbol{R}_{\boldsymbol{L}}^{\boldsymbol{e}} + \boldsymbol{R}_{\boldsymbol{\omega}}^{\boldsymbol{e}}\right)}_{\boldsymbol{R}^{\boldsymbol{e}}} \cdot \hat{\boldsymbol{\omega}}^{\boldsymbol{e}}, \tag{26}$$

where

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

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

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

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

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

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

Assembling the elemental matrices and arrays defined in (26) and taking into account that $\delta\boldsymbol{v}$ is arbitrary and so is its discretized counterpart $\delta\hat{\boldsymbol{V}}$, we arrive to the global system

$$\boldsymbol{K} \cdot \hat{\boldsymbol{V}} = \boldsymbol{R} \cdot \hat{\boldsymbol{\omega}}. \tag{27}$$

Neither $\boldsymbol{K}$ nor $\boldsymbol{R}$ depend on $\hat{\omega}$ 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 $\boldsymbol{K}$ is symmetric and positive definite, so it lends to factorization by Cholesky decomposition and its triangular factor is repeatedly used to solve $\hat{\boldsymbol{V}}$ through back-substitution.

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 [24]. 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 Section 5.4.2 in [29], among others). The Schur complement inherits the symmetric positive definiteness of the original system. Those are desirable properties for application of both iterative and direct solvers. We use static condensation at elemental level during the final assembly of the global system.

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.

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 [30]. 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.

2.1. 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 Eq. (3) in a more convenient way,

$$\frac{\partial \boldsymbol{\omega}}{\partial t} = \boldsymbol{F}(\boldsymbol{\omega}, t) = \boldsymbol{\nabla} \times (\nu \boldsymbol{\nabla} \cdot \boldsymbol{\nabla}\boldsymbol{v} - \boldsymbol{v} \cdot \boldsymbol{\nabla}\boldsymbol{v}). \tag{28}$$

We evaluated the right-hand side of (28) applying the corresponding differential operators onto the discrete velocity field $\hat{\boldsymbol{V}}$ that was computed by the algorithmic sequence described in Section 1.2. 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. A detailed description of the this procedure can be found in [31]. In our case we used 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. 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 (28) takes the discrete form,

$$\boldsymbol{F}(\hat{\boldsymbol{\omega}}, t) = \hat{\boldsymbol{C}}_{\boldsymbol{url}} \cdot \left(\nu \hat{\boldsymbol{D}}_{\boldsymbol{iv}} - \hat{\boldsymbol{V}}_{\boldsymbol{adv}}\right) \cdot \hat{\boldsymbol{G}}_{\boldsymbol{rad}} \cdot \hat{\boldsymbol{V}}, \tag{29}$$

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. In Section 4.3, we show accuracy tests for a p- and h-refinement study.

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 (29) 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 outper-

forms other alternatives [32]. In Section 4.4, we analyze how the use of the spectral-element discretization affects the accuracy of the time-dependent solution by the adaptive-stepsize ABM-PECE solver. We shall also discuss some future options for the time-integration process further on.

3. Boundary-layer evolution as a canonical problem for testing (ω,v) approaches

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 [12], Section 4.3, among others). The velocity field described in a frame of reference fixed to a plate moving in the minus-x direction is

$$u(y,t) = U\mathrm{erf}\left(\frac{y}{\sqrt{4\nu t}}\right), \tag{30}$$

where erf is the error function and y is the vertical coordinate. Rewriting (30) 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

$$\frac{u}{U} = \mathrm{erf}\left(\frac{y/Y}{\tau}\right), \tag{31}$$

where Y is the height of the test mesh. Fig. 2 shows the velocity profile at successive values of the parameter τ for the exact solution given by (31). For comparison purposes, the normalized boundary-layer thickness δ is given for each value of τ. Here we follow the classical definition of δ as the height at which $u/U = 0.99$ (i.e. the height at which the velocity deficit is 1%). The normalized vorticity distribution for this incompressible flow is given by the Gaussian function

$$\frac{\omega}{U/Y} = \frac{2}{\tau\sqrt{\pi}} e^{-\left(\frac{y/Y}{\tau}\right)^2}. \tag{32}$$

This problem is closely related with the key process of ($\boldsymbol{\omega}$,$\boldsymbol{v}$) methods, i.e. the vorticity generation at a solid surface due to the induced slip and its further propagation to the body of the fluid. And in that sense, may be regarded as a canonical problem on the subject. 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. Thus, besides analyzing the time-dependent problem, we used the spatial analytic solution at specific times to test several aspects on the spatial discretization of the KLE.

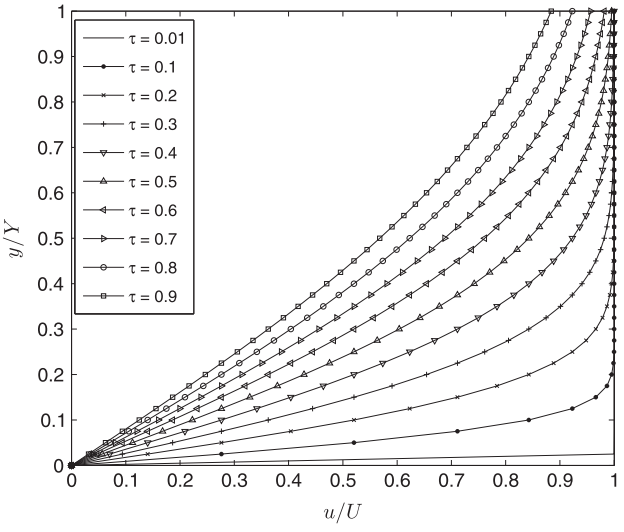

Fig. 2. Velocity profile at successive values of the viscous time τ for the exact solution given by (31). For each value of τ the normalized boundary-layer thickness δ is also given.

4. Numerical tests

In this section we shall describe several tests we performed to analyze the particularities of the high-order implementation of the KLE method by spectral-element techniques, focusing on different aspects of the method.

4.1. Accuracy of the spatial solution of the KLE

The first series of tests are aimed to exploring the accuracy of the spatial solution provided by the KLE as a PDE system. As it was described in Section 1.2, the KLE consist in a reformulation of (4) in its weak form subject to the simultaneous constraints (5) and (6). In its actual implementation, the variational formulation of the KLE makes use of the penalty method to impose both constraints (5) and (6). The penalty method, though less rigorous than other alternatives regarding the imposition of constraints, is very well suited to this coupled scheme of two simultaneous constraints of the KLE approach, producing results that are in good agreement with experiments (a detailed description of these issues may be found in [3]). 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 in hindered in any way. To this end, we designed an experiment in which we compared the theoretical velocity distribution given by expression (31) with the numerical solution for the velocity field provided by the KLE when the corresponding vorticity distribution (32) 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. Fig. 3 shows the error curves for a p-refinement study for $2 \leqslant p \leqslant 20$ 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. As in spectral elements the nodes are not equidistant, here we plotted the error against the number of intervals in each dimension N^*, which is equal to the number of nodes minus one and represents the inverse of the average internodal distance.

Previous experiences with Q_2-FEM approximations on impulsively-started cylinders [3,19–21] showed that, after a small overshot at the very early stages immediately following the initial shock, the algorithm quickly recovers and the solution continues quite satisfactorily up to the end for long simulations involving, for instance, the development of 70-diameter-long vortex-street wakes. 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

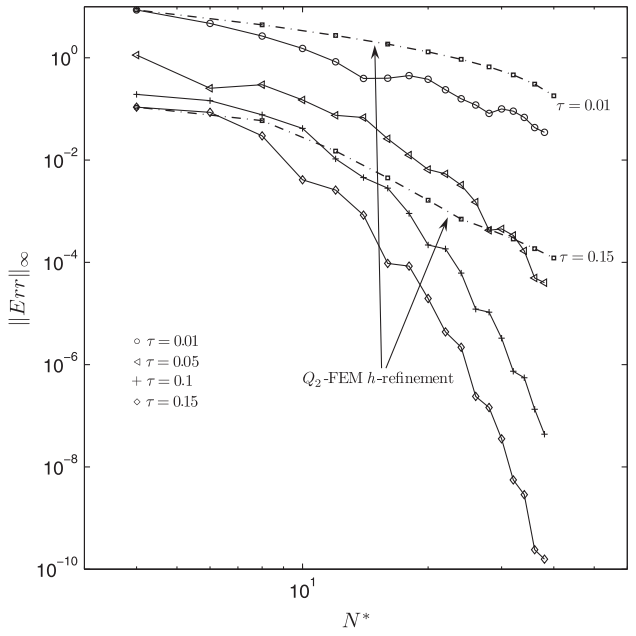

Fig. 3. Chart of the p-refinement error curves for $2 \leqslant p \leqslant 20$ and successive values of τ at the early stages of development of the boundary layer, where the effects of the impulsive-start discontinuity are present. The h-refinement curves for a Q_2-FEM discretization for $\tau = 0.01$ and $\tau = 0.15$ are included for comparison purposes.

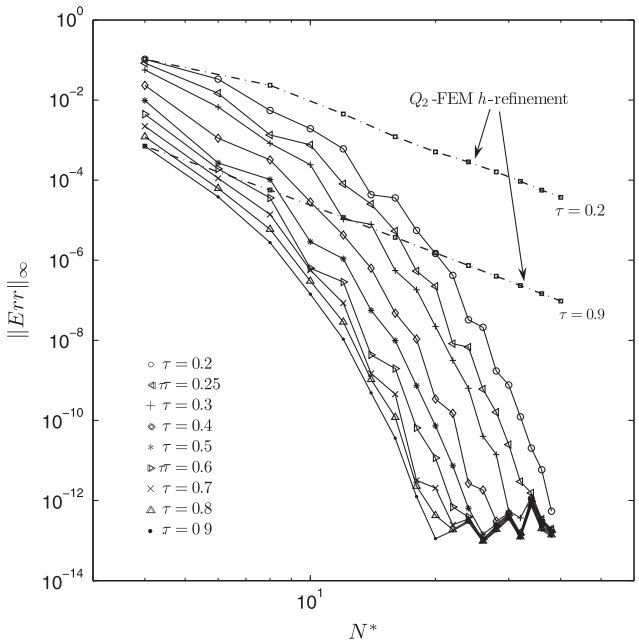

Fig. 4. Chart of the p-refinement error curves for $2 \leqslant p \leqslant 20$ and successive values of τ at several stages of development of the boundary layer. The h-refinement curves for a Q_2-FEM discretization for $\tau = 0.2$ and $\tau = 0.9$ are included for comparison purposes.

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 (32) 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.

Fig. 4 shows the error curves for a p-refinement study for $2 \leqslant p \leqslant 20$ 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.

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 [15]. 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 [15], a suitable estimate for the domain-truncation error for the problem presented in Section 3 is

$$E_{DT}(Y) \equiv \max_{y \geqslant Y}(|1 - u(y)/U|). \tag{33}$$

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 $E_{DT}(Y)$ [15]. 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 $\max|(u(y) - u_N(y))/U| \sim \mathcal{O}(E_{DT}(Y))$ for most real-world problems, one can contrive examples for which this is not true [15]. For instance, in some applications of global-spectral methods (see [33], among others), boundary-condition discontinuities, even those fairly small in magnitude compared to the speed of the unperturbed free stream, are enough to generate oscillations associated with the Gibbs phenomenon. Due to the global nature of the computations, the effect of those oscillations is felt throughout the computational domain, and viscous-sponge-like "filter functions" are needed in order to damp the oscillations. Although the multidomain nature of the discretization technique employed here is not likely to produce such a behavior, it is important for us to assess the actual effects of domain truncation on our solution, specially as p increases. Hence, 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

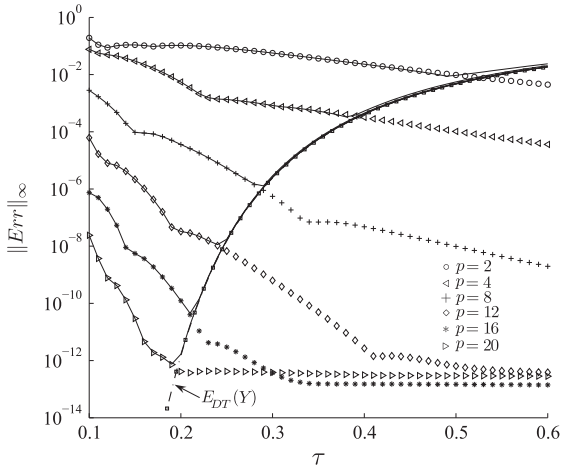

Fig. 5. Comparative charts of the error curves for solutions using exact and non-exact boundary conditions for $0.1 \leqslant \tau \leqslant 0.5$ and $p = 2,4,8,17,21$.

the mesh, in place of the exact boundary conditions used in Section 4.1 for both vorticity and velocity. Fig. 5 shows comparisons of the error curves for solutions on a 2-element mesh using exact and non-exact boundary conditions for $0.1 \leqslant \tau \leqslant 0.5$ 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 noticed decreases 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 using our Q_2-FEM discretization. The latter results are shown in Fig. 6. In Sections 4.1 and 4.2 we used the infinity norm in order to be able to compare with the *Assumption of Equal Errors* mentioned above, in the following sections we revert to the classical 2-norm for the evaluation of the error.

4.3. Numerical properties of the differential operators

In this section we shall present results from the experimentation carried on in order to assess the accuracy of the differential

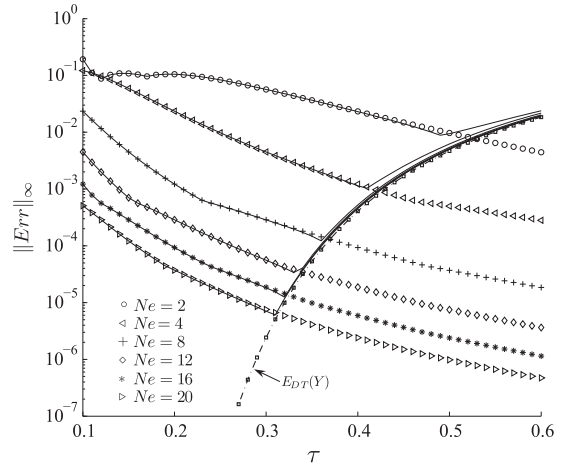

Fig. 6. Comparative charts of the error curves for solutions using exact and non-exact boundary conditions for an h-refinement of a Q_2-FEM discretization.

operators $\widehat{\boldsymbol{C}}_{\boldsymbol{url}}$, $\widehat{\boldsymbol{G}}_{\boldsymbol{rad}}$ and $\widehat{\boldsymbol{D}}_{\boldsymbol{iv}}$ used to compute the right-hand side of (28) as explained in Section 2.1. We systematically applied the operators computed for different meshes over known velocity fields and compared the results against the exact derived fields, i.e. the curl of the velocity, convective term and diffusive term of the Navier–Stokes equations. In order to avoid the trivial case in which the derived fields belong to the functional space generated by the discretization, we used test velocity fields based on the error function, which is equivalent to the solution of the flat plate problem described in Section 3, and on sinusoidal functions.

We assembled the differential operator matrices $\widehat{\boldsymbol{C}}_{\boldsymbol{url}}$, $\widehat{\boldsymbol{G}}_{\boldsymbol{rad}}$ and $\widehat{\boldsymbol{D}}_{\boldsymbol{iv}}$ for uniform meshes of square elements. p-refinement studies were carried on meshes of 2×2 and 4×4 elements of order $2 \leqslant p \leqslant 21$ which were compared with h-refinement for classical Q_2-FEM discretization. Periodic boundary conditions were applied between lateral edges of the mesh in order to simulate an infinite domain.

For the studied meshes we assembled the differential operator matrices $\widehat{\boldsymbol{C}}_{\boldsymbol{url}}$, $\widehat{\boldsymbol{G}}_{\boldsymbol{rad}}$ and $\widehat{\boldsymbol{D}}_{\boldsymbol{iv}}$ and starting from a known velocity field, with nodal values arranged in the array $\widehat{\mathbf{V}}$, we compute the discrete versions of

- the curl of the velocity as $\widehat{\boldsymbol{C}}_{\boldsymbol{url}}\widehat{\mathbf{V}}$,
- the curl of the convective term as $\widehat{\boldsymbol{C}}_{\boldsymbol{url}}\ \widehat{\boldsymbol{V}}_{\boldsymbol{adv}}\ \widehat{\boldsymbol{G}}_{\boldsymbol{rad}}\ \widehat{\mathbf{V}}$, and
- the curl of the diffusive term as $\widehat{\boldsymbol{C}}_{\boldsymbol{url}}\ \widehat{\boldsymbol{D}}_{\boldsymbol{iv}}\ \widehat{\boldsymbol{G}}_{\boldsymbol{rad}}\ \widehat{\mathbf{V}}$.

This way we were able to study the convergence properties of our derivation technique in variables involving 1st, 2nd and 3rd order derivatives, using the spectral element discretization.

Velocity field based on the error function:
In the first test we evaluated our derivation technique using a velocity field similar to that obtained as a solution of the flat plate boundary layer problem given by Eq. (31) and $\nu = 0$. This velocity field depends on the parameter τ and for smaller values it becomes less smooth as seen in Fig. 2. The z component of its curl, which is the only nonzero, is given by expression (32). In the right-hand side of Eq. (28), the curl of the convective term of this given velocity field is

$$[\nabla \times (\boldsymbol{v} \cdot \nabla \boldsymbol{v})]_z = 0, \tag{34}$$

and the curl of the diffusive term results

$$[\nabla \times (\nabla \cdot \nabla \boldsymbol{v})]_z = \frac{4U}{\sqrt{\pi}\tau^3 Y^3}\left(1 - \frac{2y^2}{\tau^2 Y^2}\right) e^{-\left(\frac{y/Y}{\tau}\right)^2}. \tag{35}$$

In this case, the exact curl of the convective term is identically zero. Thus, the error measured gives an idea of how our derivation technique, when applied together with the spectral element, gives rise to spurious noise which comes from the round-off error of the operations involved.

Velocity field based on sinusoidal functions:
The second test was aimed to test the behavior of the derivation technique in the case of fields with variation in both directions (x,y). to this end we choose a velocity field based on sinusoidal functions given by

$$\begin{aligned} u &= \sin(2\omega_y \pi y), \\ v &= \sin(2\omega_x \pi x). \end{aligned} \tag{36}$$

This field depends on the two parameters ω_x and ω_y that give the frequency of the oscillation in each direction. The z component of its curl is given by

$$(\nabla \times \boldsymbol{v})_z = 2\omega_x \pi \cos(2\omega_x \pi x) - 2\omega_y \pi \cos(2\omega_y \pi y). \tag{37}$$

In the right-hand side of Eq. (28), the curl of the convective term of the velocity field of expression (36) is

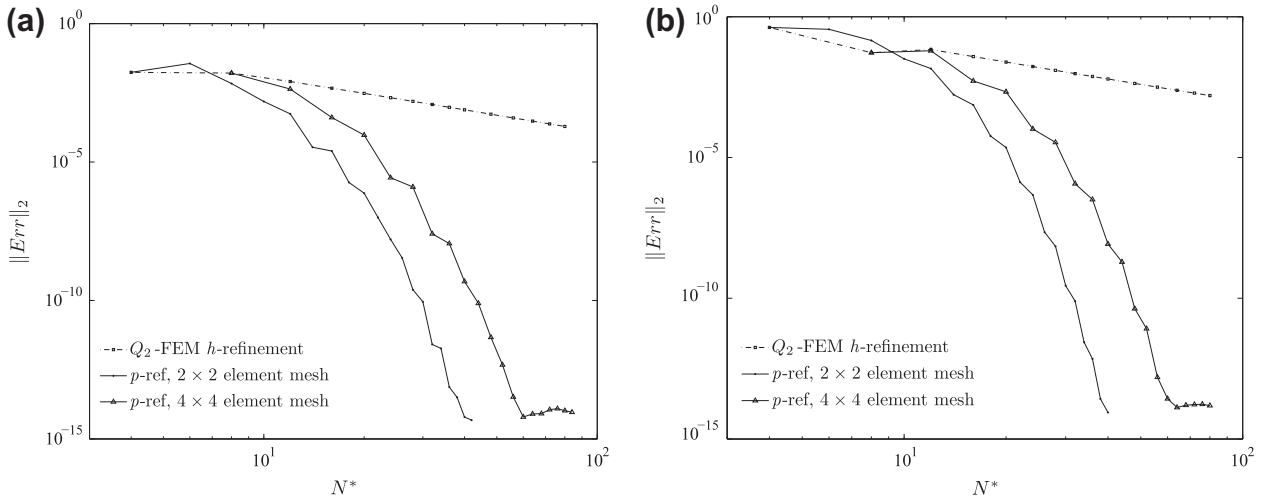

Fig. 7. Comparative charts of the error curves for the curl of the velocity field for (a) the error function based field from (31) and (b) the sinusoidal function from (36). p-refinement curves for meshes of 2×2 and 4×4 elements are compared against h-refinement curve for classical Q_2-FEM.

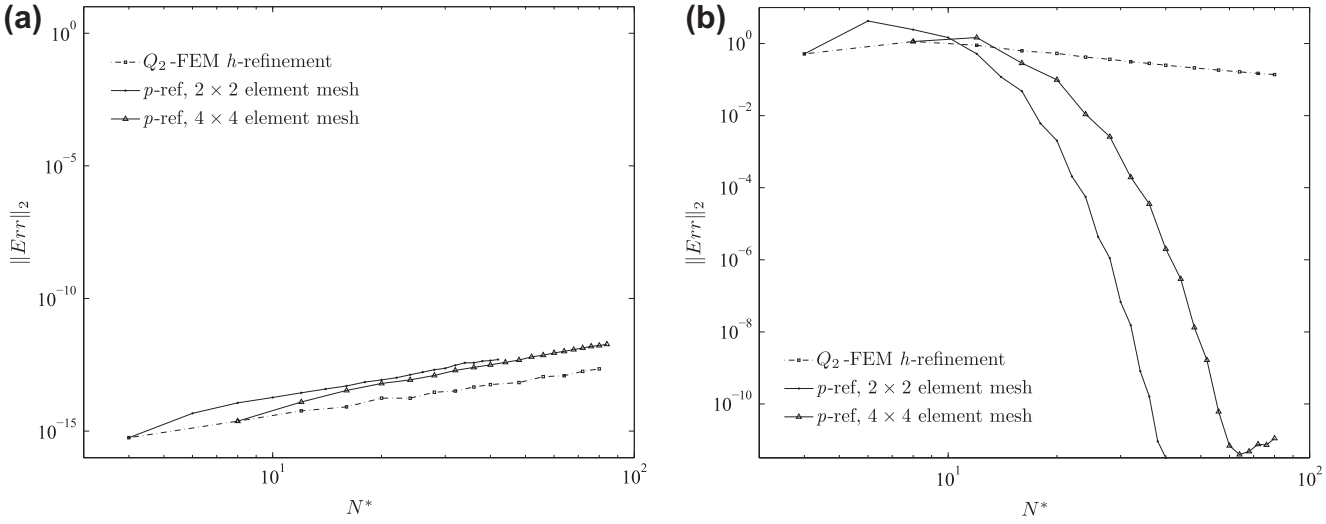

Fig. 8. Comparative charts of the error curves for the curl of the convective term of Navier–Stokes equations corresponding to velocity fields for (a) the error function based field from (31) and (b) the sinusoidal function from (36). p-refinement curves for meshes of 2×2 and 4×4 elements are compared against h-refinement curve for classical Q_2-FEM.

$$[\nabla \times (\boldsymbol{v} \cdot \nabla \boldsymbol{v})]_z = [(2\omega_y \pi)^2 - (2\omega_x \pi)^2] \sin(2\omega_x \pi x) \times \sin(2\omega_y \pi y), \tag{38}$$

and the curl of its diffusive term

$$[\nabla \times (\nabla \cdot \nabla \boldsymbol{v})]_z = -(2\omega_x \pi)^3 \cos(2\omega_x \pi x) + (2\omega_y \pi)^3 \times \cos(2\omega_y \pi y). \tag{39}$$

Fig. 7 shows charts comparing error curves for the curl of the two test fields and different meshes: the error function based field with $\tau = 0.3$ and the sinusoidal field with $\omega_x = 2y\omega_y = \frac{2\pi}{3}$. The values of the parameters in the sinusoidal field were chosen to avoid the trivial case of null curl of the convective term when $\omega_x = \omega_y$. Figs. 8 and 9 show charts comparing error curves for the curl of the convective and diffusive terms for the two test fields and different meshes.

The error curves for h-refinement are straight line with slope decreasing with the derivation order. On the other hand, curves for p-refinement show the typical spectral convergence though as the derivation order increases it can be notice a slightly different behavior between even and odd order elements. This results in irregular decreasing curves with increasing slopes when moving to higher order elements. In these cases as shown in Section 4.1 there is a lower limit were the curves "hit". This limit changes for the different derivation order being located around 10^{-14} for the curl of the velocity, 10^{-12} for the curl of the convective term and 10^{-11} for the curl of the diffusive term. In the particular case shown in Fig. 8a where the exact solution is zero, the error grows for increasing order but remains under the limit observed in the case.

4.4. Accuracy of the time-dependent solution

The complete KLE method for the solution of the time-dependent problem combines the successive application of the KLE and the linear operators on each time step under in the integration process controlled by the adaptive ODE algorithm. To test the complete algorithmic sequence of the method as described in Section 1 we solved the problem of the infinite flat plate suddenly started from rest as mentioned in Section 3 for different meshes obtained by both h- and p-refinement. In order to let the boundary layer

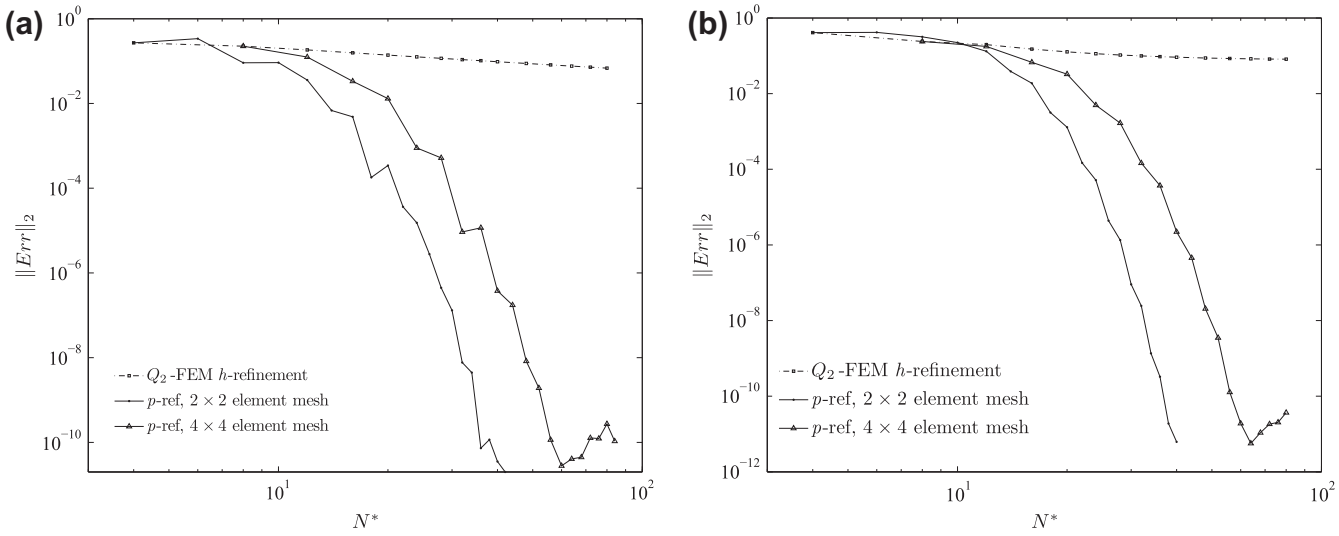

Fig. 9. Comparative charts of the error curves for the curl of the diffusive term of Navier–Stokes equations corresponding to velocity fields for (a) the error function based field from (31) and (b) the sinusoidal function from (36). p-refinement curves for meshes of 2×2 and 4×4 elements are compared against h-refinement curve for classical Q_2-FEM.

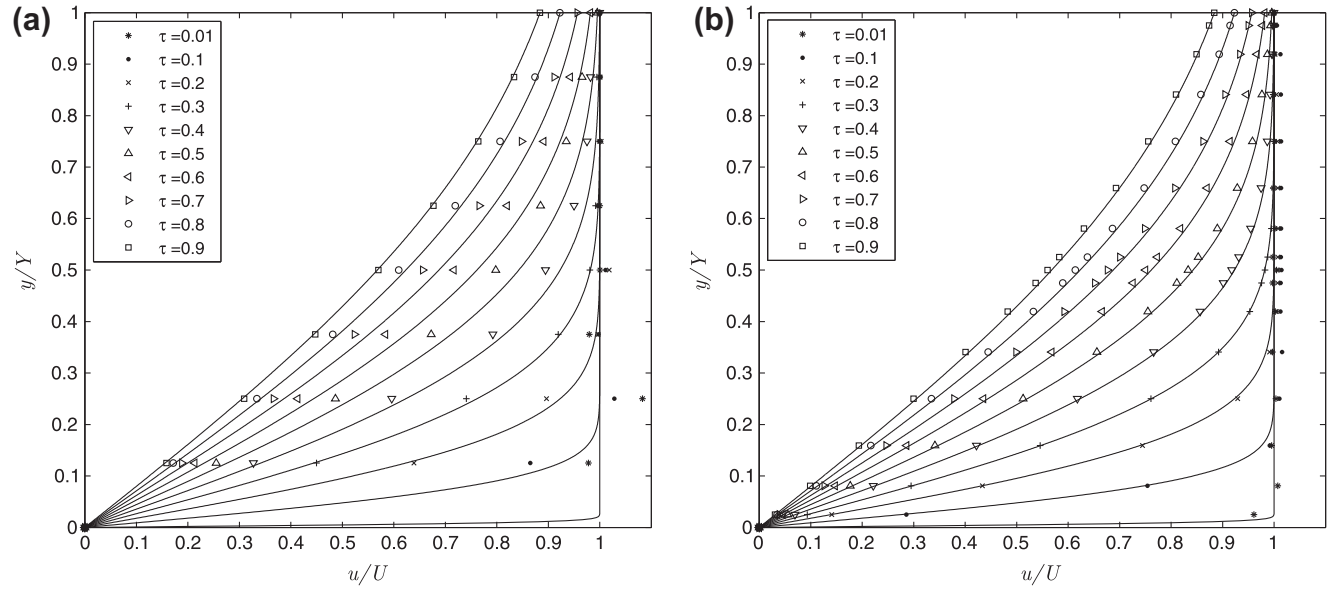

Fig. 10. Results of the time-dependent solution obtained for meshes composed by low and high order elements in: (a) a mesh with 4×4 elements of order $p = 2$ and (b) a mesh with 2×2 elements of order $p = 8$.

fully develop without the effect of domain truncation we impose exact velocity and vorticity boundary conditions on the top face of the domain and we run our method from $\tau = 0$ to $\tau = 1$.

In Fig. 10 the solution at different values of the time parameter τ ranging from 0.01, where the exact solution is almost a step, to 0.9, where the solution is sensible more smooth, is presented. The solution shown in Fig. 10a corresponds to a mesh of 16 (4×4) elements of order $p = 2$, i.e. the classical Q_2-FEM whereas in Fig. 10b the solution shown correspond to a mesh of 4 (2×2) elements of order $p = 9$.

In order to analyze the spatial convergence of the complete method when refining the meshes we show in Fig. 11 the error curves for h- and p-refinement at a particular intermediate value of the time parameter, $\tau = 0.38$. The behavior is clearly different in each curve: the h-refinement curve presents fixed slope corresponding to the second order elements; on the other hand, the p-refinement curve appears to resemble the spectral convergence up to order 7 while for higher orders it seems to converge with fixed slope with higher absolute value than that of the Q_2-FEM h-refinement

5. Concluding remarks and outlook for further work

We have analyzed systematically the particularities of a high-order implementation of the KLE by spectral-element techniques. First, we have focused our study on the accuracy of each of the two main sub-algorithms that compose this novel method, namely, the spatial solution of the KLE, and the linear operators used to evaluate the right-hand side of the ODE integration. Using the solution for the boundary-layer evolution on an infinite horizontal flat plate impulsively started as a canonical problem for testing, we found that the spatial solution of the KLE exhibits spectral convergence up to the lower limit of accuracy given by the round-off er-

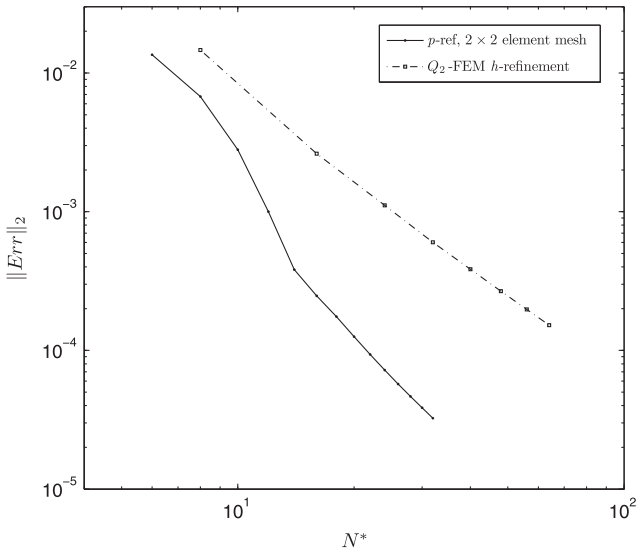

Fig. 11. Chart of the error curves for p-refinement and h-refinement obtained in the time-dependent solution process at a specific value of the time parameter $\tau = 0.38$. p-Refinement curves were obtained for a 2×2 mesh with elements of order $3 \leqslant p \leqslant 16$ and h-refinement, for a Q_2-FEM discretization with several meshes from 4×4 to 32×32 elements.

ror. We also tested the effects of domain truncation verifying that the maximum error induced on the whole domain is bounded by the minimum error given by the theoretical limit. We tested the accuracy of the second sub-algorithm (i.e. the linear operators). Comparing with two different families of test functions for the velocity distribution, we verified that the successive application of the linear operators does not ruin spectral convergence. We also verified that the error accumulation of successive application of the linear operators during the chain of derivations required for each term of the ODE's right-hand side is kept to a minimum. Especially, for the case of the convective term in the error function, which should be theoretically zero, the error induced by the applications of the linear operators is kept at the level of the round-off error (see Fig. 8a).

In Section 4.4 we show the results for the accuracy test for the complete solution of the time-dependent problem, which combines the successive application of the KLE and the linear operators on each time step under in the integration process controlled by the adaptive ODE algorithm. We could see that the numerical solution for the velocity profile is in good agreement with the theoretical counterpart, closely following it during the evolutive process of the boundary layer. Fig. 11 shows that, even after the successive application of the main sub-algorithms and the computations of the ODE algorithm itself, spectral convergence is maintained up to an error of 4×10^{-4}. Afterwards, the error curve seams to switch to an algebraic-convergence behavior. Taking into account the results of the previous sections on the individual behavior of each of the main sub-algorithms, this indicates that future efforts to further improve the accuracy of the whole method should be centered in the interaction between the main sub-algorithms into the ODE process. Namely, we are going to focus on the accuracy of the vorticity-creation process associated with the algorithmic sequence of two KLE solutions performed on each time-step (see Section 1.2; for a detailed description see [3]), and on the accuracy aspects of the ODE integration algorithm itself. Another point that would be interesting to analyze in future stages of the development of our method will be the usage of zero-stress open conditions on the outflow boundary.

Acknowledgments

The authors are very grateful for the financial support made available by the National Science Foundation through Grants CEBET-0933058 and CEBET-0952218, and University of Buenos Aires through Grant 20020100100536 UBACyT 2011/14.

References

[1] Fasel H. Investigation of the stability of boundary layers by a finite difference model of the Navier–Stokes equations. J Fluid Mech 1976;78:355–83.
[2] Clercx HJH. A spectral solver for the Navier–Stokes equations in the velocity–vorticity formulation for flows with two nonperiodic directions. J Comput Phys 1997;137:186–211.
[3] Ponta FL. The kinematic Laplacian equation method. J Comput Phys 2005;207:405–26.
[4] Quartapelle L. Numerical solution of the incompressible Navier–Stokes equations. Basel (Switzerland): Birkäuser; 1993.
[5] Speziale CG. On the advantages of the velocity–vorticity formulation of the equations of fluid dynamics. J Comput Phys 1987;73:476–80.
[6] Trujillo J, Em Karniadakis G. A penalty method for the vorticity–velocity formulation. J Comput Phys 1999;149(1):32–58.
[7] Anderson CR. Observations on vorticity creation boundary conditions. In: Caflisch RE, editor. Mathematical aspects of vortex dynamics. SIAM; 1988. p. 144–59.
[8] Chorin AJ. Numerical study of slightly viscous flow. J Fluid Mech 1973;57:785–96.
[9] Chorin AJ. Vortex sheet approximation of boundary layers. J Comput Phys 1978;27:428–42.
[10] Quartapelle L, Valz-Gris F. Projections conditions on the vorticity in viscous incompressible flows. Int J Numer Meth Fluids 1981;1:129–44.
[11] Quartapelle L. Vorticity conditioning in the computation of two-dimensional viscous flows. J Comput Phys 1981;40:453–77.
[12] Batchelor GK. An introduction to fluid dynamics. Cambridge (UK): Cambridge University Press; 2000.
[13] Irons B, Razzaque A. Experience with the patch test for convergence of finite elements. In: The mathematical foundations of the finite element method with applications to partial differential equations (Proc. Sympos., Univ. Maryland). New York: Academic Press; 1972. p. 557–87.
[14] Taylor RL, Simo JC, Zienkiewicz OC, Chan ACH. The patch test – a condition for assessing FEM convergence. Int J Numer Methods Eng 1986;22(1):39–62.
[15] Boyd JP. Chebyshev and Fourier spectral methods. Mineola (NY, USA): Dover; 2000.
[16] Sherwin SJ, Karniadakis GE. Tetrahedral hp finite elements: algorithms and flow simulations. J Comput Phys 1996;124:14–45.
[17] Thompson MC, Hourigan K, Sheridan J. Three-dimensional instabilities in the wake of a circular cylinder. Exp Therm Fluid Sci 1996;12:190–6.
[18] Karniadakis GE, Israeli M, Orszag SA. High-order splitting methods for the incompressible Navier–Stokes equations. J Comput Phys 1991;97:414–43.
[19] Ponta FL, Aref H. Strouhal–Reynolds number relationship for vortex streets. Phys Rev Lett 2004;93:084501.
[20] Ponta FL, Aref H. Numerical experiments on vortex shedding from an oscillating cylinder. J Fluids Struct 2006;22:327–44.
[21] Ponta FL. Analyzing the vortex dynamics in bluff-body wakes by Helmholtz decomposition of the velocity field. Fluid Dynam Res 2006;38:431–51.
[22] Patera AT. A spectral element method for fluid dynamics: laminar flow in a channel expansion. J Comput Phys 1984;54:468–88.
[23] Karniadakis GE, Bullister ET, Patera AT. A spectral element method for solution of two- and three-dimensional time-dependent incompressible Navier–Stokes equations. In: Finite element methods for nonlinear problems. New York/Berlin: Springer-Verlag; 1985. p. 803.
[24] Henderson RD, Karniadakis GE. Unstructured spectral element methods for simulation of turbulent flows. J Comput Phys 1995;122:191–217.
[25] Giraldo FX. The Lagrange–Galerkin spectral element method on unstructured quadrilateral grids. J Fluid Mech 1998;147:114–46.
[26] Hourigan K, Thompson MC, Tan BT. Self-sustained oscillations in flows around long blunt plates. J Fluids Struct 2001;15:387–98.
[27] Thompson MC, Leweke T, Williamson CHK. The physical mechanism of transition in bluff body wakes. J Fluids Struct 2001;15:607–16.
[28] Sheard GJ, Thompson MC, Hourigan K. From spheres to circular cylinders: non-axisymmetric transitions in the flow past rings. J Fluid Mech 2004;506:45–78.
[29] Barrett R, Berry M, Chan TF, Demmel J, Donato J, Dongarra J, et al. Templates for the solution of linear systems: building blocks for iterative methods. 2nd ed. Philadelphia (PA, USA): SIAM; 1994.
[30] Pathria D, Karniadakis GE. Spectral element method for elliptic problems in nonsmooth domains. J Comput Phys 1994;122:83–95.
[31] Bathe KJ. Finite element procedures. Englewood Cliffs (NJ, USA): Prentice Hall; 1996.
[32] Press WH, Teukolsky SA, Vetterling WT, Flannery BP. Numerical recipies in C. 2nd ed. Cambridge (UK): Cambridge University Press; 2002.
[33] Mittal R, Balachandar S. Direct numerical simulation of flow past elliptic cylinders. J Comput Phys 1996;124(2):351–67.