Journal of Computational Physics **149,** 32–58 (1999)
Article ID jcph.1998.6135, available online at http://www.idealibrary.com on IDEAL®

A Penalty Method for the Vorticity–Velocity Formulation

James Trujillo and George Em Karniadakis[1]

Center for Fluid Mechanics, Division of Applied Mathematics, Brown University, Providence, Rhode Island 02912
E-mail: gk@cfm.brown.edu

Received March 17, 1998; revised July 20, 1998

We present a new vorticity–velocity formulation and implementation for the unsteady three-dimensional Navier–Stokes equations, based on a penalty method. It relies on an equivalence theorem that employs exact boundary conditions and the vorticity definition on the domain boundary. This approach is particularly attractive for high-order methods for which the often-used influence matrix method fails to converge for $\Delta t \to 0$. The accuracy and the robustness of the new method is demonstrated in the context of several spectral element simulations of unsteady two- and three-dimensional internal and external flows. In particular, the flow past a finite span cylinder attached to end-plates is studied in some detail in order to evaluate the effects of the aspect ratio on the formation length.

1. INTRODUCTION

The vorticity–velocity formulation of the Navier–Stokes equations has emerged as an attractive alternative to the velocity–pressure formulation in simulating incompressible flows [8, 16, 32, 40]. Several general advantages of this formulation are often cited in the literature: (1) it deals with the physically relevant variables of vortex dominated flows; (2) it works in both two- and three-dimensions; (3) it eliminates the pressure term, which leads to a simple diffusion operator rather than the Stokes operator; (4) boundary conditions can be easier to implement in external flows where the vorticity at infinity is easier to set than the pressure boundary condition; and (5) no additional computational work is required to evaluate noninertial terms since all noninertial effects arising from rotation and translation of the reference frame enter into solution through the initial and boundary conditions [45]. Specifically, in the finite element context, the vorticity–velocity formulation produces a *vorticity field* that is $C^0(\Omega)$ continuous across elemental interfaces. This is unlike

[1] Corresponding author.

0021-9991/99 $30.00

the velocity–pressure formulation, where continuity of vorticity across elemental interfaces is achieved only upon convergence.

In this paper, a new vorticity–velocity numerical formulation, based on a penalty method, is presented. It is implemented in the context of semi-implicit temporal discretization and spectral element spatial discretization. The penalty method is the key to imposing robust and stable high-order accurate vorticity boundary conditions. Several numerical tests will demonstrate that high-order accuracy is achieved with the penalty method. The exponential convergence, the minimization of the dispersion and dissipation errors, and the geometric flexibility of the spectral element method make it particularly well suited for simulating turbulent flows. However, this method (and, in general, any high-order method) is more sensitive to boundary condition implementation. In particular, the influence matrix method [52] described in Section 3 has been used to impose the vorticity boundary conditions, but it was discovered that it has fundamental numerical limitations for high-order discretizations, as it does not converge as $\Delta t \to 0$.

The formulation presented here expands both the vorticity and velocity fields in the same discrete space (polynomial order, $\mathbf{P_N} - \mathbf{P_N}$ formulation). However, a fundamental theoretical question of which discrete space should be used to expand the vorticity and velocity terms remains. Low-order finite element vorticity–velocity formulations expand the vorticity in a subspace of lower order than the velocity [16, 15, 19, 39, 40] and finite difference vorticity–velocity formulations use a staggered grid [8, 9, 29, 30, 17, 46, 18, 32, 56, 11, 13]. Numerical evidence from our work suggests that, for the proposed splitting formulation, there is no obvious incompatibility in the spaces of velocity and vorticity. This, in turn, allows for an easy implementation of the proposed formulation.

This paper is organized as follows: In Section 2, we present the equivalence theorem for the vorticity–velocity formulation. In Section 3 we analyze the influence matrix method. In Section 4, the accuracy of the penalty method is demonstrated with several analytical problems, and the scaling of the error with the penalty term is determined by numerical experiments. In Section 5, a discussion of treatment of corner singularities and the effect of rounding the corners are presented. In Section 6, the simulation of flow past a two-dimensional cylinder at Reynolds number $\mathrm{Re} = 1000$ is compared with a velocity–pressure flow solver, and a three-dimensional simulation of a cylinder with end-plates shows the effect of side boundaries on the formation length. We conclude in Section 7 with a brief summary of the results. Finally, in Section 8 we include some details of the implementation, as well as representative parallel timings.

2. EQUIVALENCE THEOREM

We first state the canonical velocity–pressure form of the unsteady incompressible three-dimensional Navier–Stokes equations and the two proposed forms of the vorticity–velocity systems—rotational and Laplacian. The canonical velocity–pressure form of the incompressible Navier–Stokes equations is

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\nabla p + \frac{1}{\mathrm{Re}}\nabla^2 \mathbf{u} \quad \text{in } \Omega \tag{1a}$$

$$\nabla \cdot \mathbf{u} = 0 \quad \text{in } \Omega \tag{1b}$$

$$\mathbf{u} = \mathbf{u}_\Gamma \quad \text{on } \Gamma_D; \quad \frac{\partial \mathbf{u}}{\partial n} = 0 \quad \text{on } \Gamma_N, \tag{1c}$$

where the velocity boundary conditions must satisfy the constraint $\int_\Gamma \mathbf{u}\cdot\mathbf{n}\,d\Gamma = 0$, and the initial conditions for the velocity must be supplied, $\mathbf{u}(\mathbf{x}, t = 0)$. Here Γ_D is the Dirichlet boundary and Γ_N is the Neumann boundary.

The proposed **rotational form** of the vorticity–velocity formulation of the incompressible Navier–Stokes equations is

$$\frac{\partial \omega}{\partial t} + \nabla \times (\omega \times \mathbf{u}) = -\frac{1}{\mathrm{Re}}\nabla \times \nabla \times \omega \quad \text{in } \Omega \tag{2a}$$

$$\nabla^2 \mathbf{u} = -\nabla \times \omega \quad \text{in } \Omega \tag{2b}$$

with boundary constraints and initial conditions

$$\omega = \nabla \times \mathbf{u} \quad \text{on } \Gamma \tag{2c}$$

$$\oint_{c_k} \left(\frac{\partial \mathbf{u}}{\partial t} + \omega \times \mathbf{u} + \frac{1}{\mathrm{Re}}\nabla \times \omega \right) \cdot ds = -\oint_{c_k} d\left(p + \frac{1}{2}\mathbf{u}\cdot\mathbf{u} \right), \quad k = 1, \ldots, p, \tag{2d}$$

$$\mathbf{u} = \mathbf{u}_\Gamma \quad \text{on } \Gamma_D; \quad \frac{\partial \mathbf{u}}{\partial n} = 0 \quad \text{on } \Gamma_N \tag{2e}$$

$$\int_\Gamma \mathbf{u}\cdot\mathbf{n}\,d\Gamma = 0 \quad \text{or} \quad \nabla\cdot\mathbf{u} = 0 \quad \text{at one point on } \Gamma \tag{2f}$$

$$\omega = \nabla \times \mathbf{u} \quad \text{at } t = 0 \quad \text{in } \Omega, \tag{2g}$$

where the domain is p-multiply connected and c_k's are the p independent contours and the initial conditions for the velocity must be supplied, $\mathbf{u}(\mathbf{x}, t = 0)$. The equivalence between the rotational form of the vorticity–velocity equations, Eqs. (2), and the canonical velocity–pressure form, Eqs. (1), will be demonstrated by Theorems Ia, II, III, and IV.

The proposed **Laplacian form** of the vorticity–velocity formulation of the incompressible Navier–Stokes equations is

$$\frac{\partial \omega}{\partial t} + \nabla \times (\omega \times \mathbf{u}) = \frac{1}{\mathrm{Re}}\nabla^2 \omega \quad \text{in } \Omega \tag{3a}$$

$$\nabla^2 \mathbf{u} = -\nabla \times \omega \quad \text{in } \Omega \tag{3b}$$

with boundary constraints and initial conditions

$$\omega = \nabla \times \mathbf{u} \quad \text{on } \Gamma \tag{3c}$$

$$\nabla \cdot \omega = 0 \quad \text{on } \Gamma \tag{3d}$$

$$\oint_{c_k} \left(\frac{\partial \mathbf{u}}{\partial t} + \omega \times \mathbf{u} + \frac{1}{\mathrm{Re}}\nabla \times \omega \right) \cdot ds = -\oint_{c_k} d\left(p + \frac{1}{2}\mathbf{u}\cdot\mathbf{u} \right), \quad k = 1, \ldots, p, \tag{3e}$$

$$\mathbf{u} = \mathbf{u}_\Gamma \quad \text{on } \Gamma_D; \quad \frac{\partial \mathbf{u}}{\partial n} = 0 \quad \text{on } \Gamma_N \tag{3f}$$

$$\int_\Gamma \mathbf{u}\cdot\mathbf{u}\,d\Gamma = 0 \quad \text{on } \nabla\cdot\mathbf{u} = 0 \quad \text{at one point on } \Gamma \tag{3g}$$

$$\omega = \nabla \times \mathbf{u} \quad \text{at } t = 0 \quad \text{in } \Omega, \tag{3h}$$

where the domain is p-multiply connected and c_k's are p independent contours and the initial conditions for the velocity must be supplied, $\mathbf{u}(\mathbf{x}, t = 0)$. The equivalence between

the Laplacian form of the vorticity–velocity equations, Eqs. (3) and the canonical velocity–pressure form, Eqs. (1), will be demonstrated by Theorems Ib, II, III, and IV.

The equivalence theorem states that the vorticity and the velocity obtained from systems (2) and (3) are identical to the vorticity and velocity from system (1). The only difference in the proof of equivalence between the rotational and Laplacian form of the vorticity–velocity equations is in the way the divergence-free vorticity condition is enforced. In Theorem Ia, the rotational form of the equations, Eqs. (2), implicitly sets the derivatives of the divergence to zero by using the *curl* form of the diffusion term. In Theorem Ib, the Laplacian form of the equations, Eqs. (3), explicitly sets the divergence of vorticity to zero on the boundary.

THEOREM Ia. *A necessary and sufficient condition for the vorticity–velocity equations* (2) *to satisfy the condition that the vorticity is divergence-free at all times in the domain* Ω *is that the definition of vorticity must be satisfied in the domain initially* (*Shen and Loc* [41]).

Remark on Theorem Ia. The rotational form of the vorticity transport equation, Eq. (2a), is a convenient form to prove that vorticity is divergence-free. However, the rotational form couples all three components of the vorticity. This means that all three components of the vorticity must be computed simultaneously.

THEOREM Ib. *A necessary and sufficient condition for the vorticity–velocity equations* (3) *to satisfy the condition that the vorticity is divergence-free in the domain is that the definition of vorticity must be satisfied in the domain initially and on the boundary at all times* (*Quartapelle* [37]).

Remark on Theorem Ib. The equation governing the evolution of the divergence of vorticity is parabolic with homogeneous Dirichlet boundary conditions. The homogeneous Dirichlet boundary conditions can be replaced by homogeneous Neumann boundary conditions, which are just $\nabla(\nabla \cdot \omega) = 0$ on the boundary Γ. This term, $\nabla(\nabla \cdot \omega)$, is implicitly set to zero in Eq. (2a) of the rotational form in the entire domain Ω. Setting either the Dirichlet or Neumann divergence-free vorticity boundary conditions explicitly on the boundary is perhaps preferred over solving the rotational form because later the penalty method will be shown to be well suited to imposing complicated constraints on the boundary. Also, by using the Laplacian form of the vorticity–velocity equations, one is able to use standard solvers and avoid the nonsymmetric coupled solvers needed for the vorticity in the rotational form.

It will be proven here that the definition of vorticity is governed by a Laplace equation with the definition of vorticity enforced on the boundary Γ in the proposed vorticity–velocity formulations, implying the definition of vorticity is satisfied in the domain Ω. The proof is based on a vector identity and on Eq. (2b) or (3b).

THEOREM II. *Necessary and sufficient conditions for the definition of the vorticity to be satisfied everywhere in the domain are that the definition of vorticity is satisfied on the boundary and the vorticity is divergence-free* (*Theorem Ia or Theorem Ib*).

Proof. Consider the vector identity in terms of the velocity,

$$\nabla^2 \mathbf{u} = \nabla(\nabla \cdot \mathbf{u}) - \nabla \times (\nabla \times \mathbf{u}) \tag{4}$$

and Eq. (2b) or (3b)

$$\nabla^2 \mathbf{u} = -\nabla \times \omega. \tag{5}$$

Equating the right sides of these two equations and manipulating gives

$$\nabla \times (\omega - \nabla \times \mathbf{u}) = -\nabla(\nabla \cdot \mathbf{u}). \tag{6}$$

Taking the *curl* of this equation gives

$$\nabla \times \nabla \times (\omega - \nabla \times \mathbf{u}) = -\nabla \times \nabla(\nabla \cdot \mathbf{u}), \tag{7}$$

where the right-hand side is zero. Applying the vector identity to the left-hand side gives

$$\nabla[\nabla \cdot (\omega - \nabla \times \mathbf{u})] - \nabla^2(\omega - \nabla \times \mathbf{u}) = 0. \tag{8}$$

The first term drops out because the vorticity is divergence-free in the domain from Theorem Ia or Ib, and the second term is zero because derivatives are interchangeable. Thus,

$$\nabla^2(\omega - \nabla \times \mathbf{u}) = 0. \tag{9}$$

Hence, by the minmax principle for the Laplace equation and the fact that the definition of vorticity is satisfied on the boundary, the definition of vorticity is satisfied everywhere in the domain. ■

THEOREM III. *A necessary and sufficient condition for the velocity to be divergence-free in the domain is for the definition of vorticity to be satisfied in the domain and for either global mass balance to be satisfied or the velocity to be divergence-free at one point on the boundary* (*Daube* [8]).

We also require that the total pressure (i.e., static plus dynamic pressure) is single-valued. Stella and Guj [46] derived a constraint from a pressure single-valuedness argument in multiply-connected domains and showed, by example, that the constraint is necessary by considering the Taylor–Couette problem. A more general constraint for multiply-connected domains was derived by Daube [9].

THEOREM IV. *A necessary and sufficient condition for the total pressure,* $p + \frac{1}{2}\mathbf{u} \cdot \mathbf{u}$, *to be single-valued if the domain is p-multiply connected is for* $\oint_{c_k} (\partial \mathbf{u}/\partial t + \omega \times \mathbf{u} + (1/\mathrm{Re})\nabla \times \omega) \cdot ds = 0$ *for* $k = 1, \ldots, p$, *on p independent contours.*

The constraint can be reduced to

$$\oint_{c_k} (\nabla \times \omega) \cdot ds = 0, \tag{10}$$

if the contour is on a solid surface where the *no-slip* condition is required. For a 2D flow past a cylinder, the constraint applied to the cylinder surface with the no-slip condition further reduces to

$$\oint_c \frac{\partial \omega}{\partial n}\, ds = 0, \tag{11}$$

where n is the normal to the surface of the cylinder and c is the contour around the surface of the cylinder. For a 3D flow past a cylinder the side boundary conditions need to be taken into account. In practice, it is convenient to impose the constraint in Eq. (10) on a no-slip surface

instead of the more general form of the constraint in Theorem IV, because the expression takes on a simpler form that just involves the vorticity.

One of the key steps in proving equivalence for both the rotational and Laplacian forms is enforcing the definition of vorticity on the boundary. This important result from the equivalence theorem provides a linear coupling between the vorticity and velocity on the boundary and is necessary to guarantee that Eqs. (2) and (3) give the correct vorticity and velocity fields.

3. UNRESOLVED ISSUES IN THE INFLUENCE MATRIX METHOD

The influence matrix technique has often been used to impose linear constraints on the boundary implicitly. Kleiser and Schuman [27, 28] were among the first to use the influence matrix technique to impose pressure boundary conditions in their channel simulation for the velocity–pressure equations. Vanel *et al*. [54] used an influence matrix technique to solve the Navier–Stokes equations based on a vorticity-streamfunction spectral method formulation. Daube simulated axi-symmetric flow [8] in a cylindrical tank by solving the vorticity–velocity equations using an influence matrix technique identical to the one presented here.

The influence matrix method relies on the *linearity* of the semi-discrete equations, with the nonlinear terms treated explicitly. We can separate the 2D vorticity–velocity equations into a time-dependent and time-independent problem,

$$\omega(\mathbf{x}, t) = \tilde{\omega}(\mathbf{x}, t) + \sum_{k=1}^{N_\Gamma} \lambda_k \hat{\omega}_k(\mathbf{x}) \tag{12a}$$

$$\mathbf{u}(\mathbf{x}, t) = \tilde{\mathbf{u}}(\mathbf{x}, t) + \sum_{k=1}^{N_\Gamma} \lambda_k \hat{\mathbf{u}}_k(\mathbf{x}), \tag{12b}$$

where λ_k's will be determined by enforcing the definition of vorticity on the boundary and N_Γ is the number of nodes on the boundary. The time-independent equations need only be solved once and are

$$\left(\frac{\mathrm{Re}}{\Delta t} - \nabla^2\right)\hat{\omega}_k = 0; \quad \hat{\omega}_k(\gamma_j) = \delta_{kj} \quad \forall \gamma_j \in \Gamma \tag{13a}$$

$$\nabla^2 \hat{\mathbf{u}}_k = -\nabla \times \hat{\omega}_k, \quad \hat{\mathbf{u}} = 0. \tag{13b}$$

The time-dependent problem is

$$\left(\frac{\mathrm{Re}}{\Delta t} - \nabla^2\right)\tilde{\omega}^{n+1} = \frac{\mathrm{Re}}{\Delta t}\omega^n - \mathrm{Re}(\mathbf{u}^n \cdot \nabla)\omega^n, \quad \tilde{\omega}_\Gamma^{n+1} = 0 \tag{14a}$$

$$\nabla^2 \tilde{\mathbf{u}}^{n+1} = -\nabla \times \tilde{\omega}^{n+1}, \quad \tilde{\mathbf{u}}^{n+1} = \mathbf{u}_\Gamma. \tag{14b}$$

The influence of the boundary vorticity on the interior flow is represented by the influence matrix. The influence matrix equations are constructed by substituting the time dependent and independent solutions into the definition of vorticity on the boundary and then solving for λ_k's. More details about this method can be found in [51].

3.1. *Numerical Boundary Layers*

The cost to store the time-independent solution for the influence matrix technique is typically very large. This cost can be offset by increasing the computational cost of the flow solver, but the influence matrix will still have to be stored [50]. Furthermore, the rank of the influence matrix (i.e. degrees-of-freedom on the physical boundary) can be rather large and the matrix is not symmetric. It is important to note that the influence matrix technique is limited to solving problems with fixed grids and time steps, which excludes the class of problems with moving boundaries or adaptive remeshing and time stepping. However, all these difficulties are not nearly as discouraging as the intrinsic *stiff* numerical boundary layers in the formulation that become more severe as Δt goes to zero.

The numerical stiffness caused by the influence matrix technique has previously gone unnoticed. It will be shown that this numerical stiffness increases as the Reynolds number is raised and as the numerical time step decreases. A possible explanation of why the numerical stiffness in the influence matrix technique has been overlooked is that the low-order schemes previously used to solve the vorticity–velocity equations have a tendency to smooth sharp boundary layers artificially. On the other hand, the high-order method presented here must resolve the steep boundary layers because high-order methods have minimal artificial dissipation. Researchers using the technique have limited their studies to low Reynolds numbers [6], where the numerical stiffness is not as severe. Another possible reason for the omission is the influence matrix technique has been successfully used to impose pressure boundary conditions in the velocity–pressure formulation of the Navier–Stokes equations. In the velocity–pressure formulation, numerical stiffness does not occur unless a Fourier expansion is used in at least one direction, and even then, only if the wave number is large does the problem manifest itself.

To demonstrate the numerical boundary layers in the influence matrix technique, we will consider a one-dimensional model problem. The spatial discretization does not affect the size of the numerical boundary layers but, as mentioned earlier, it affects the way in which poorly resolved numerical boundary layers are handled. For example, high-order schemes will allow Gibbs oscillations, while low-order methods will exhibit a local smeared shock-type profile. This is the reason why high-order schemes are more susceptible to instability than low-order schemes that seem to be more robust.

Consider the linear one-dimensional advection–diffusion equation on a semi-infinite domain,

$$\begin{aligned} &\frac{\partial \omega}{\partial t} = U_c \frac{\partial \omega}{\partial x} + \nu \frac{\partial^2 \omega}{\partial x^2}, \quad 0 < x < \infty \\ &\omega(0,t) = 1, \quad \lim_{x\to\infty} \omega(x,t) = 0. \end{aligned} \tag{15}$$

A simple Euler-forward approximation on the advection term, and an Euler-backward approximation on the diffusion term give

$$\left(\frac{1}{\nu\Delta t} - \frac{\partial^2}{\partial x^2}\right)\omega^{n+1} = \frac{1}{\nu\Delta t}\omega^n + \frac{U_c}{\nu}\frac{\partial \omega^n}{\partial x}. \tag{16}$$

Note that, in practice, Euler-forward approximation would not be used for the advection term because of stability problems, but for this model problem it will suffice. The time-independent influence matrix problem is constructed by solving the equation with a pulse

on the boundary but no forcing function. So, the time-independent Helmholtz equation is

$$\left(\frac{1}{\nu\Delta t} - \frac{\partial^2}{\partial x^2}\right)\hat{\omega} = 0, \quad \hat{\omega}(0) = 1. \tag{17}$$

The exact solution to this problem is $\hat{\omega}(x) = e^{-(x/\sqrt{\nu\Delta t})}$. We are interested in small values of the viscosity, ν, because we are interested in high Reynolds number flows. The time step will be small also because we want to accurately simulate unsteady flows and satisfy the CFL stability limit. Therefore, the thickness of the numerical boundary layer, $\mathcal{O}(\sqrt{\nu\Delta t})$, will be small. The boundary layer based on Eq. (15) is $\mathcal{O}(\nu/U_c)$, and typical physical laminar boundary layers scale as $\mathcal{O}(\sqrt{\nu x/U_c})$. So, the numerical boundary layers can always be made smaller than the physical boundary layers by decreasing the time step Δt. Usually, it is desirable for the numerical boundary layer to be small. In fact, many times, the smaller the numerical boundary layer the more accurate the solution. For instance, the pressure boundary conditions in the splitting scheme for the velocity–pressure form of the Navier–Stokes equations are chosen to minimize the numerical boundary layer around the walls [25]. Another example, where one wants to minimize the numerical normal boundary layers is at outflow boundaries, where an artificial boundary condition has been applied. However, in our case the numerical boundary layer *must be resolved* in order to obtain accurate vorticity boundary conditions. If the numerical boundary layer is smaller than the physical boundary layer, then the spatial resolution needed is determined by the influence matrix technique, instead of the physical problem. The situation where the numerical technique increases the stiffness of the problem is highly undesirable.

Now let us use the Kovasznay flow test case, a 2D steady Navier–Stokes flow solution with a perturbation in the initial conditions, to demonstrate that the influence matrix technique is *not convergent in time*. The Kovasznay flow with $\mathcal{O}(\epsilon)$ perturbation is

$$u(x, y) = 1 - e^{\beta x}\cos 2\pi y + \epsilon \sin^2\left(\frac{\pi}{2}x\right)\sin^2\left(\frac{\pi}{2}\left(y + \frac{1}{2}\right)\right) \tag{18a}$$

$$v(x, y) = \frac{\beta}{2\pi}e^{\beta x}\sin 2\pi y + \epsilon \sin^2\left(\frac{\pi}{2}x\right)\sin^2\left(\frac{\pi}{2}\left(y + \frac{1}{2}\right)\right), \tag{18b}$$

where $\beta = \mathrm{Re}/2 - \sqrt{\mathrm{Re}^2/4 + 4\pi^2}$. We are going to examine the error as a function of time of the perturbed Kovasznay flow solution. It will be shown that as the time step gets smaller, then the solution will blow up, due to underresolution of the numerical boundary layers. In an attempt to simplify the problem as much as possible, we will use one element to approximate the domain of $x \in (0, 2)$ and $y \in (-0.5, 1.5)$ with fixed spatial resolution. Details of the time-splitting scheme and of implementation are given in the Appendix.

We start with a set of parameters that lead to convergence when using the perturbed Kovasznay initial conditions, Eq. (18). For the following tests, the Reynolds number is fixed at 40 and the amplitude of perturbation, ϵ, is fixed at 10^{-2}. The only parameter varied is the time step, Δt. The left plot in Fig. 1 shows the $\mathcal{L}_2$ error in the velocity converges, using a 40th-order polynomial and a time step of 10^{-3}. Next, in the right plot in Fig. 1 the time step is reduced to 10^{-4} with all other parameters fixed. In this case, the $\mathcal{L}_2$ error of the velocity diverges in time. The reason is the length of the numerical boundary layers arising when computing the vorticity boundary conditions using the influence matrix technique have

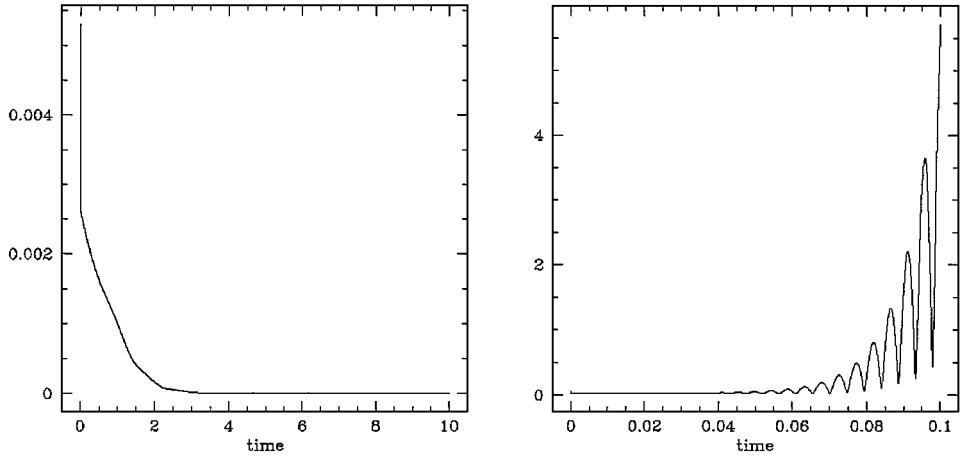

FIG. 1. Time history of the $\mathcal{L}_2$ error in the velocity of the Kovasznay flow, 2D steady Navier–Stokes flow, at Reynolds number 40 for one spectral element from $x \in (0, 2)$ and $y \in (-0.5, 1.5)$ with a 40th-order polynomial and a time step of $\Delta t = 10^{-3}$ for the left plot and $\Delta t = 10^{-4}$ for the right plot.

decreased with the time step. As a result, the spatial resolution of 40th-order polynomial is not sufficient to resolve the numerical boundary layers.

Another way to verify that the source of the problem is the construction of the vorticity boundary conditions using the influence matrix technique is by enforcing the correct vorticity at the boundary for the case that diverges. If the influence matrix technique is the source of the problem, then the solution will converge because the solution containing the numerical boundary layers is circumvented. In general, this test is not possible because the vorticity on the boundary is not known, but for the Kovasznay test case we know the steady solution. The test reveals that the $\mathcal{L}_2$ error of the velocity converges when the vorticity on the boundary is fixed, so our suspicions about the influence matrix technique are confirmed. A more systematic study of this instability phenomenon for high-order discretizations is presented in [51], where other such examples are also included.

A solution to this problem is to map out the steep boundary layers in the time-independent problem. Mappings of this kind have been constructed for high-order methods [5]. This is only economical if the numerical boundary layers are not present in the time-dependent equations, so the cost of the mapping can be limited to the preprocessing stage. However, we will not attempt to correct the numerical boundary layer problem in the influence matrix technique. Instead, we will consider an alternative technique to impose the vorticity boundary conditions—the penalty method.

4. PENALTY METHOD

Recently, penalty methods have been used to successfully implement boundary conditions in high-order discretizations. For example, multidimensional asymptotically stable finite difference schemes on complex geometries have been developed by Abarbanel and Ditkoedki [2, 1], using a penalty method to impose Dirichlet boundary conditions. Also, Hesthaven and Gottlieb [22, 20, 21] developed a penalty method to enforce boundary conditions for shock-free compressible Navier–Stokes simulations. This penalty method enforces the boundary conditions, as well as accounting for the governing equation at the boundary. In the classical version, the equation is penalized everywhere in the domain. Such a *global*

penalty method developed by Temam [48] and successfully used by Hughes [23] imposes a penalty term in the *entire* domain, which leads to excessive stiffness and poor efficiency. The computational advantage of imposing the penalty term on the *boundary* highlights the importance of the equivalence theorem.

4.1. *Multidimensional Formulation*

The vorticity transport equation with the penalty boundary conditions is

$$\frac{\partial \omega}{\partial t} + \nabla \times (\omega \times \mathbf{u}) = \frac{1}{\mathrm{Re}} \nabla^2 \omega - \tau\, Q(\mathbf{x}) D_\omega - \tau_c\, Q_c(\mathbf{x}_c) F_\omega, \tag{19}$$

where $D_\omega = \omega - \nabla \times \mathbf{u}$, $F_\omega = \int_c (\nabla \times \omega) \cdot d\mathbf{s}$, τ is the penalty term that imposes the definition of vorticity, and τ_c is the penalty term that imposes the multiply-connected constraint on the body. The function

$$Q(\mathbf{x}) = \begin{cases} 1, & \text{if } \mathbf{x} \text{ is on } \Gamma, \\ 0, & \text{otherwise,} \end{cases} \tag{20}$$

ensures that the definition of vorticity is imposed only on the boundary, and the function

$$Q_c(\mathbf{x}_c) = \begin{cases} 1, & \text{if } \mathbf{x} \text{ is on } \Gamma_c, \\ 0, & \text{otherwise,} \end{cases} \tag{21}$$

ensures that the multiply-connected condition is imposed only on the body.[1]

The weak form of the vorticity transport equation using the penalty method to impose the vorticity boundary conditions is

$$\begin{aligned} &\left(\psi, \frac{\partial \omega}{\partial t}\right) + (\psi, \nabla \times (\omega \times \mathbf{u})) \\ &\quad = -\frac{1}{\mathrm{Re}} (\nabla \psi, \nabla \omega) - (\psi, \tau\, Q(\mathbf{x}) D_\omega) - (\psi, \tau_c\, Q_c(\mathbf{x}_c) F_\omega) + \int_\Gamma \psi \frac{\partial \omega}{\partial n}\, d\Gamma, \end{aligned} \tag{22}$$

where $(\cdot, \cdot) = \int_\Omega d\Omega$ and $\psi \in H^1(\Omega)$. The last term in the weak form of the vorticity equation arises from the integration by parts of the diffusion term. The dominant boundary terms in (22) are the penalty terms. The boundary term arising from the integration by parts is subdominant. Leaving out the integration by parts boundary term can be justified as the penalty terms, τ and τ_c, become larger and dominate the boundary. Later it will be shown that spectral convergence can be achieved using the penalty method when the subdominant boundary term is neglected. However, neglecting this term can explain why such a large penalty parameter is needed. Some test cases demonstrating this will be shown later in this section.

Now the vorticity and velocity are coupled on the boundary by the vorticity boundary conditions and the multiply-connected constraint. This strong coupling is undesirable with the current numerical formulation because it requires both the vorticity and velocity to be solved simultaneously. In this case, fast linear solvers, based on the semi-discrete equations, cannot be used. Hence, a time extrapolation of the penalty terms on the boundary of the form

$$D_\omega^{n+1} = \omega^{n+1} - \nabla \times \mathbf{u}^n \tag{23a}$$

$$F_\omega^{n+1} = \int_c (\nabla \times \omega^n) \cdot d\mathbf{s} \tag{23b}$$

[1] We consider the case where the boundary of the domain is stationary.

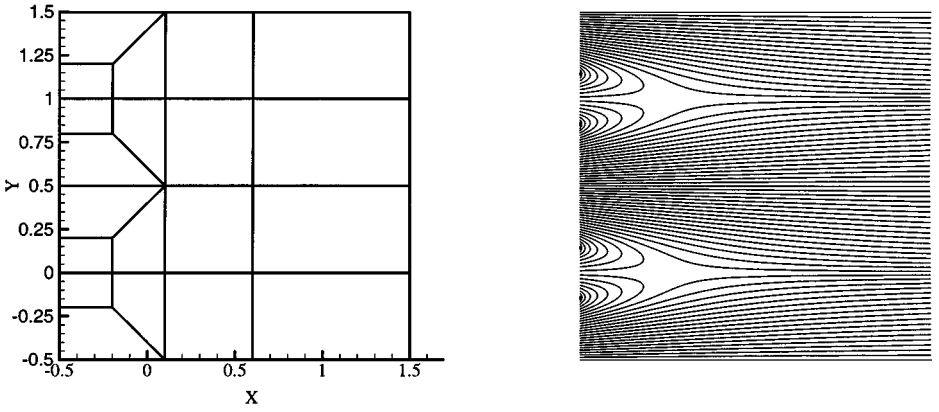

FIG. 2. Mesh and streamlines for 2D Kovasznay flow test case at Reynolds number 40. Note the mesh cluster elements in the wake.

is used, so the fast linear solvers can be applied. We accept this time error caused by lagging the vorticity boundary conditions with the understanding that the numerical method already requires a small time step due to the CFL limit caused by treating the nonlinear terms explicitly. Note that the penalty boundary terms, Eq. (23), are imposed using a first-order extrapolation in time. High-order extrapolation schemes in time could be used, but first-order has given satisfactory results.

We will test the penalty method on a steady 2D Navier–Stokes solution. We will address two questions when solving the test case. First, can spectral accuracy be achieved with the penalty method? Second, how does the penalty parameter scale with the mesh parameters, such as polynomial order?

We consider again the Kovasznay flow test case given in Eq. (18) with the perturbation ϵ set to zero for the purpose of demonstrating that high-order accuracy can be achieved using the penalty method. Figure 2 shows a plot of the mesh and the streamlines at Reynolds number 40. Figure 3 shows that the error in the vorticity and velocity scales inversely with

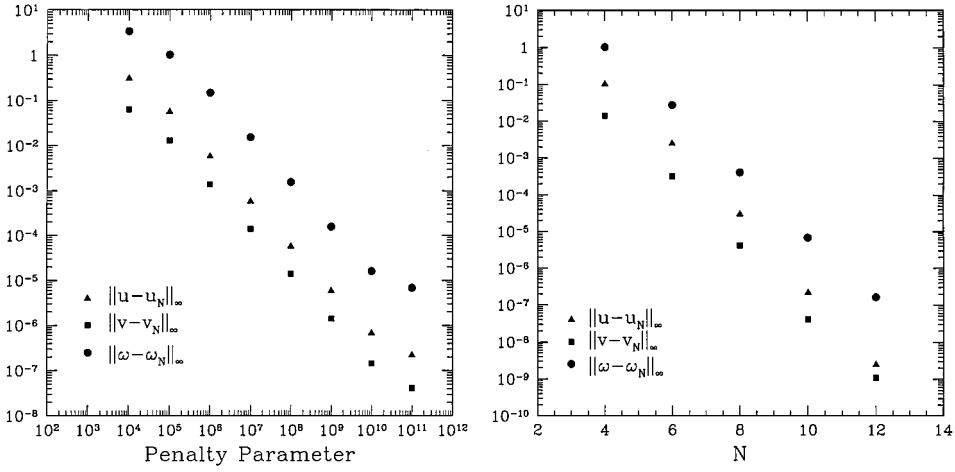

FIG. 3. Plot (left) showing the error as a function of penalty parameter with a 10th order polynomial and a plot (right) demonstrating the exponential convergence of the 2D Kovasznay test case at Reynolds number 40.

the penalty parameter, τ. Exponential convergence is achieved in the vorticity and velocity using the maximum norm.

5. CORNERS, DISCONTINUITIES, AND VORTICITY

Nonsmooth computational domains can give rise to singular solutions, which are especially problematic in the vorticity–velocity formulation. The solution of an elliptic problem in a domain with a corner of angle ($\alpha\pi$) solved in local polar coordinates has the form

$$u(r,\theta) \propto r^{\beta}\zeta(\theta)\chi(r,\theta), \tag{24}$$

where $\zeta(\theta)$ is a smooth function and $\chi(r,\theta)$ is a smooth cutoff function. For these types of problems, the convergence rate estimates for the spectral element solution is

$$\|u - u_N\| \leq CN^{-2\beta-\epsilon}, \tag{25}$$

where u_N is the numerical solution and ϵ is a small positive constant. For most problems, $\beta = \pi/\alpha$, so the convergence rate lies between $\mathcal{O}(N^{-1})$ and $\mathcal{O}(N^{-2})$. These results can be applied to the velocity Poisson equations, Eq. (2b), or Eq. (3b), and the regularity of the vorticity can be estimated from the regularity of the first derivative of the velocity. Several techniques have been used to overcome this difficulty such as auxiliary mapping and supplementary basis functions [36]. One supplementary basis function formulation used to recover high-order convergence which is particularly attractive is an eigenpair representation called the Steklov formulation [57, 47]. Here, we want to study the effect of sharp corners on the vorticity by rounding the corner and vary its curvature systematically. In this case, the main question is how large must the curvature be for the rounded corner to accurately approximate the sharp corner geometry. We recognize that the answer to this question is problem-dependent. Therefore, we will limit our corner study to the 2D backward-facing step and the 3D conduit expansion problem, which are representative of the geometries that are of immediate interest in our work. Furthermore, it is noted that the singularity issue is exaggerated in the vorticity–velocity equations, compared with the velocity–pressure equations, because the vorticity is less regular. In velocity–pressure formulations it is the pressure which is infinite at corners with angles greater than π and thus, in principle, the same sort of difficulty should be encountered with this formulation. However, the elliptic equation for the pressure is supplemented with Neumann boundary conditions, and by defining a *normal* vector at a corner, one effectively rounds the corner and, thus, a finite value of the pressure at the corner results.

5.1. *Flow over Expansions*

First, the nature of the corner problem will be explored by considering the 2D backward-facing step as a test case [24]. The inlet channel height is $h = 1.06$ and the outlet channel height is $H = 2.0$ to match the expansion ratio in the experiments by Armaly *et al.* [3]. The step corner will be rounded and the limit of large curvature will be taken to demonstrate that the features of the solution asymptote to the sharp corner solution. Figure 4 shows the vorticity contours for the backward facing step with an increasingly sharper corner. The vorticity contours look very similar, especially plots (2) and (3) in Fig. 4, with the sharper corners. The meshes used to obtain the results are also shown in Fig. 4. The step problem with a sharp corner is solved using $\boldsymbol{\mathcal{N}\varepsilon\kappa\mathcal{T}\alpha r}$ [55]—an unsteady incompressible

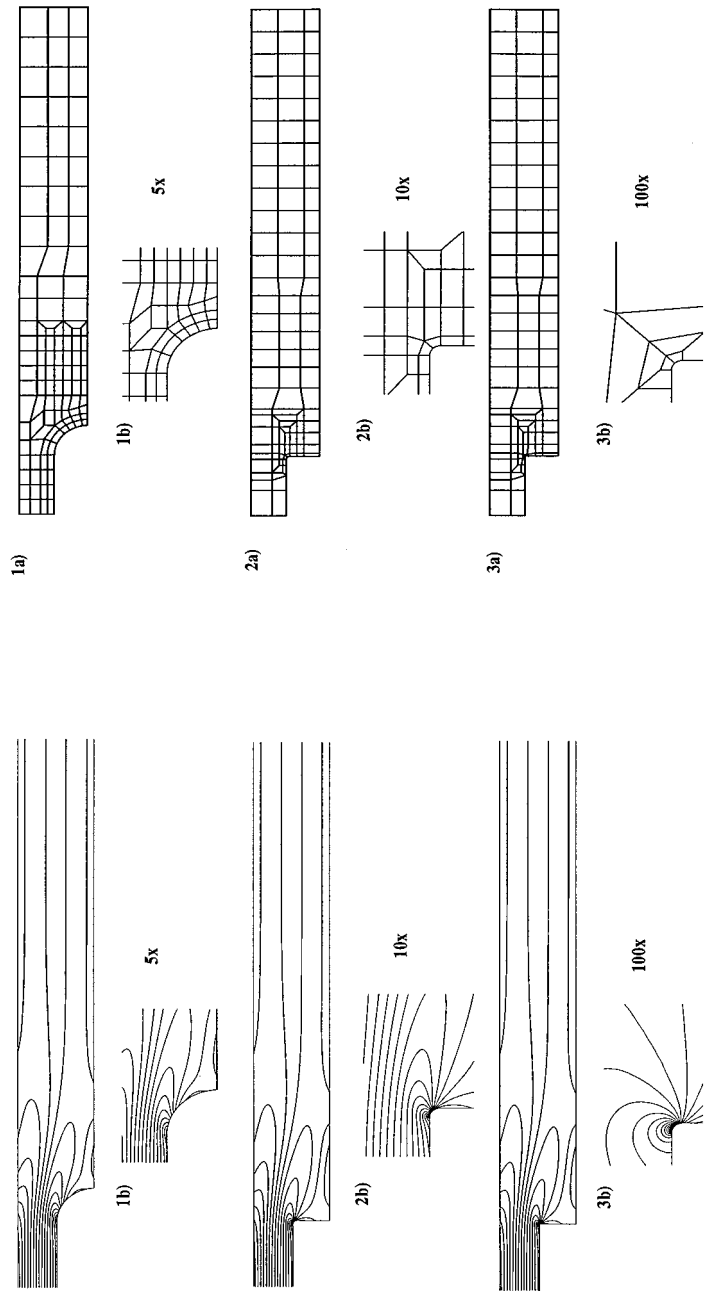

FIG. 4. Vorticity contours and computational mesh of the 2D backward facing step with a rounded corner curvature of (1) 1, (2) 10, (3) 100 normalized with inlet channel height, h. The Reynolds number is 100 based on $\frac{2}{3}(U_{\max}(2h)/\nu)$.

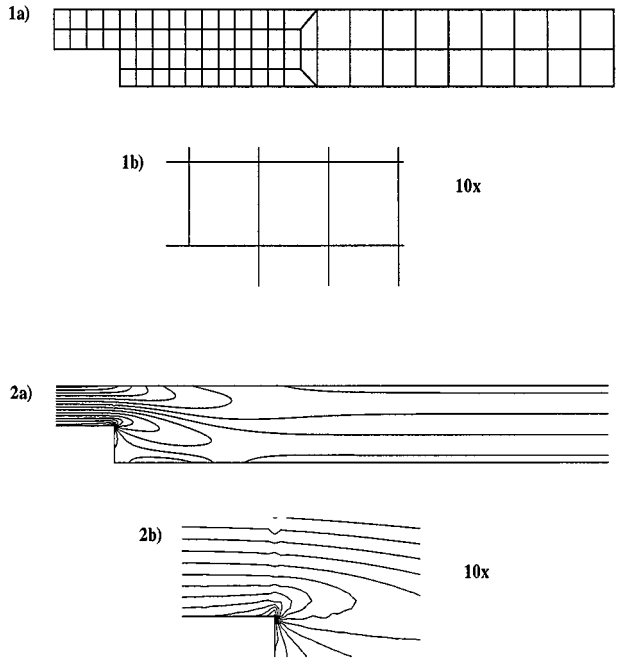

FIG. 5. Vorticity contours generated using $\mathcal{NEKTAR}$—velocity-pressure flow solver for the sharp corner 2D backward facing step problem. In part (2a) of the figure the vorticity contours are shown and in part (2b) a 10-fold magnification of the vorticity contours around the corner is shown with the mesh shown in part (1). The Reynolds number is 100, based on $\frac{2}{3}(U_{\max}(2h)/\nu)$, and an 8th-order polynomial is used within each element.

velocity–pressure Navier–Stokes solver. The vorticity contours and corresponding mesh from the velocity–pressure solver are plotted in Fig. 5. Notice the similarity between the rounded corner result from $\mathcal{IVVA}$—the vorticity–velocity solver, and the sharp corner result from the velocity–pressure solver. Figure 6 shows the separation length as a function of corner curvature predicted by the vorticity–velocity solver and is plotted along with the experimental value and predicted value from the velocity–pressure solver. Both the experiment and the velocity–pressure solver predict a nondimensional separation length of approximately 2.7. Note that the separation length is defined as the distance between the separation point and the reattachment point. For a sharp corner, the separation point is fixed at the corner. However, the location of the separation point for the rounded corner is not fixed.

Figure 6 shows two important results: First, both the experiment by Armaly *et al.* [3] and the numerical simulation using the velocity–pressure solver agree. Second, the separation length predicted by the vorticity–velocity flow solver approaches the above-mentioned experimental and numerical sharp corner result as the corner curvature is increased. It is interesting to see how the corner curvature affects the vorticity along the wall. A comparison of the vorticity along the lower wall of the backward-facing step with the different curvatures is shown in Fig. 7. Notice that the vorticity around the corners with curvature 10 and 100 are almost identical, except at the point at the corner where the vorticity approaches a large negative value. It can be seen in Fig. 7 that the maximum in the vorticity on the wall increases with curvature.

The next corner test case is internal flow in a 3D conduit expansion. The smaller pipe has a diameter of one and the larger pipe has a diameter of two. The domain is divided into 1028 hexahedra elements. Figure 8 shows the computational domain and

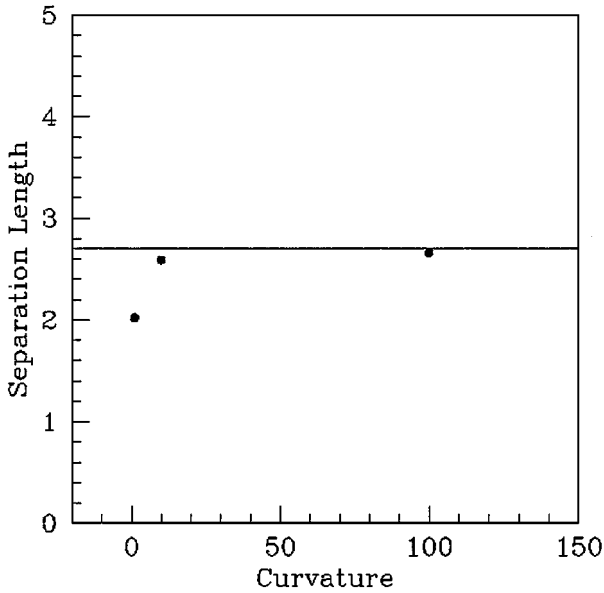

FIG. 6. The solid dots in the figure are the separation length (normalized with the step height) as a function of the corner curvature calculated using the vorticity–velocity flow solver for Reynolds number 100, based on $\frac{2}{3}(U_{\max}(2h)/\nu)$. The separation length is defined as the distance between the reattachment point and the separation point. The straight line corresponds to the separation length found experimentally by Armaly *et al*. [3] and the numerical simulation from the velocity–pressure solver for the sharp corner case.

Fig. 9 shows both the nondimensional reattachment length (also called here the bubble length, L/D_o), and distance of the center of the eddy (or the bubble center), L_2/D_o, as a function of the Reynolds number, based on the small diameter and mean inlet velocity, $\mathrm{Re} = W_o D_o/\nu$. Figure 9 shows the experimental results by Macagno and Hung [31], 3D

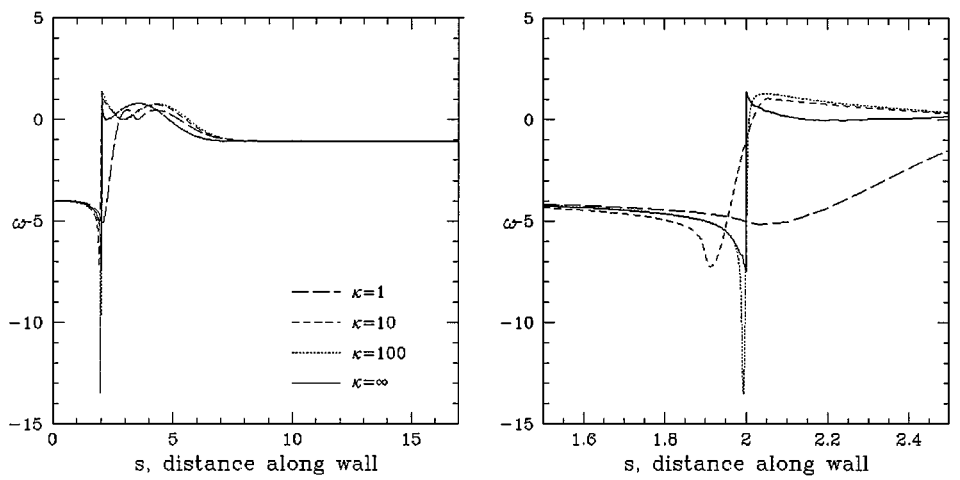

FIG. 7. A plot and zoom (right) of the vorticity along the lower wall as a function of the distance along the wall for Reynolds number 100. The three cases with finite curvature of 1, 10, and 100 are from the vorticity–velocity flow solver. The step flow case with an infinitely sharp corner is from the velocity–pressure solver. The rounded corner of the 2D backward facing step has a steep but continuous vorticity distribution. The minimum in vorticity on the lower wall corresponds to the start of the corner and the maximum corresponds to the end of the corner. Hence, in the sharp corner case, run using the velocity–pressure solver, there is a discontinuity in the vorticity at the corner.

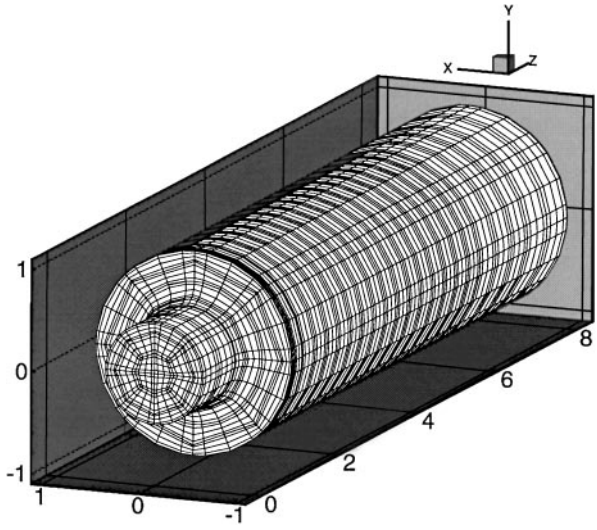

FIG. 8. A plot of the computational domain of the three-dimensional conduit with rounded corners. There are 1028 elements at the fourth-order polynomial in the mesh. Notice the curvature at the rounded corner is torus in nature, where there are two radii of curvature defining the geometry—pipe and turning curvature with the turning curvature five.

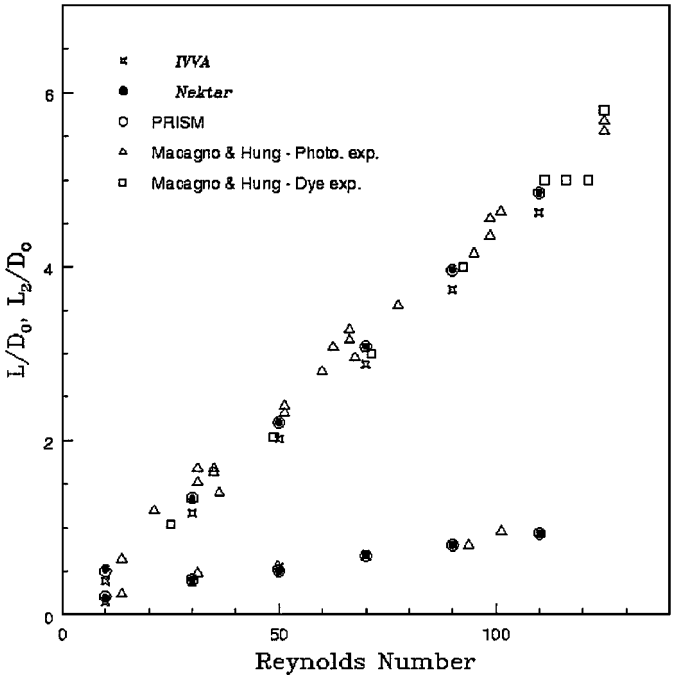

FIG. 9. A plot of the bubble length and bubble center for the three-dimensional conduit as a function of the Reynolds number is shown. The plot compares the experimental results by Macagno and Hung, 3D numerical simulation using the vorticity–velocity solver $\mathcal{IVVA}$, 3D numerical simulation using the tetrahedral velocity–pressure solver $\mathcal{N}\varepsilon\kappa\mathcal{T}\alpha\boldsymbol{r}$ [43], and a 2D axi-symmetric numerical simulation **PRISM** [33].

numerical simulation using the vorticity–velocity algorithm ($\mathcal{IVVA}$), 3D numerical simulation by Sherwin and Karniadakis [42] using tetrahedral velocity–pressure solver ($\mathcal{N}\varepsilon\kappa\mathcal{T}\alpha r$), and 2D axi-symmetric numerical simulation by Newman [33] (**PRISM**).

The curvature at the junction of the conduit expansion is torus, where the turning curvature is five. It is the length and center of the bubble around the torus-shaped corner, calculated using the vorticity–velocity solver that is going to be compared with other experiments and computations in Fig. 9. The experiment and the velocity–pressure computations are going to be for the sharp corner conduit expansion. For the finite curvature geometry, shown in Fig. 8, we expect that the bubble length calculated using the vorticity–velocity solver will underpredict the experimentally measured length, based on the experience from the 2D backward-facing step results shown previously. The amount of underprediction will directly depend on the turning curvature at the junction. For the 3D conduit problem, we are not going to increase the turning curvature. For this test case, the curvature is fixed and the inlet Reynolds number is varied to test the Reynolds number dependence of the recirculation bubble length and center. For the sharp corner case, the bubble length and distance of the center are a linear function of the Reynolds number for the range considered. So, we expect that the bubble length and center will be a linear function of Reynolds number for the rounded corner expanding conduit. Figure 9 confirms the linear dependence of the bubble length and center as a function of the Reynolds number, calculated using the vorticity–velocity solver. It is interesting to note that the bubble length seems to be more sensitive to the rounded corner than the bubble center. The difference between the prediction of the bubble center from the rounded corner and the sharp corner is indistinguishable on the graph, while the rounded corner expanding conduit underpredicts the bubble length determined by the sharp corner case. Again, this underprediction is expected, based on the results from the 2D backward–facing step tests done previously.

6. EXTERNAL FLOWS

6.1. *Two-Dimensional Flow Past a Cylinder*

An interesting point about boundary conditions for cylinder flows or for general external flows, is that the velocity is known on the cylinder, while the vorticity is not. In contrast, one can argue that the vorticity is known on the far field boundary condition, while the velocity is not. So, the vorticity on the cylinder and outflow is calculated using the definition of vorticity. The far field regions are assumed to be irrotational, so the vorticity is set to zero. On the cylinder, the velocity is set to zero to impose the no-slip condition. The flux of the velocity is set to zero at the outflow, and the velocity is set to the free stream condition at the far field boundary. The outer region, where the flow is irrotational, is relatively coarse. Notice that the "far" field boundaries are relatively close to the cylinder at 5 diameters, and the outflow is 15 diameters from the cylinder.

We have performed a detailed comparison of the instantaneous and average vorticity and velocity from the incompressible Navier–Stokes flow codes $\mathcal{IVVA}$ and $\mathcal{N}\varepsilon\kappa\mathcal{T}\alpha r$ for cylinder flow at Reynolds number 1000. ($\mathcal{IVVA}$ is the spectral element vorticity–velocity developed in this work. $\mathcal{N}\varepsilon\kappa\mathcal{T}\alpha r$ is a hybrid spectral/h-p element velocity–pressure solver that uses both triangles and quadrilaterals [43, 55]). For the comparison to follow, the mesh, number of elements, and polynomial order is the same for both solvers. This detailed two-dimensional comparison has the dual purpose of benchmarking the vorticity–velocity

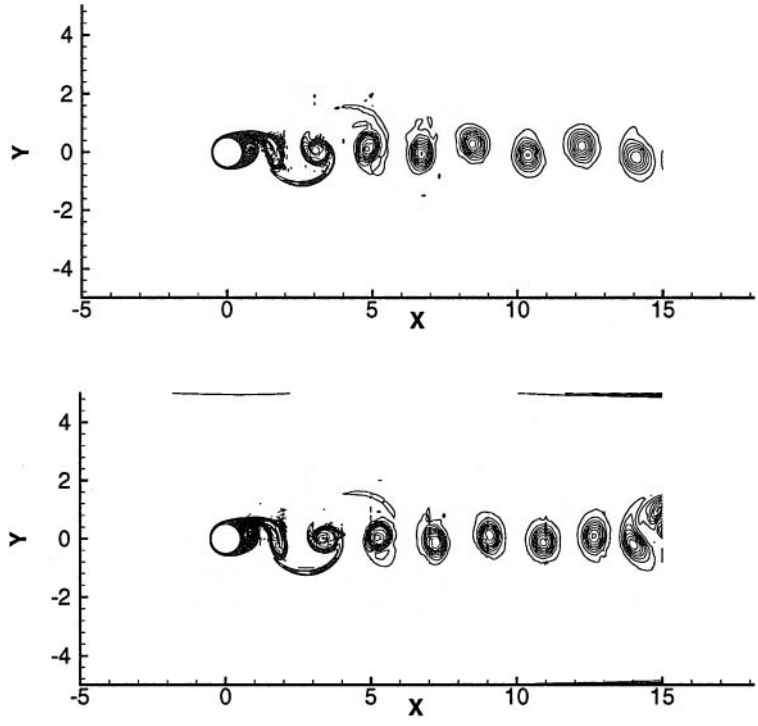

FIG. 10. A comparison of instantaneous vorticity contours for two-dimensional cylinder flow at Reynolds number 1000. The contour plot on the top is from the vorticity–velocity solver $\mathcal{IVVA}$, while the contour plot on the bottom is from the velocity–pressure solver $\mathcal{NEKTAR}$. Both results were obtained using 94 elements with 14th-order polynomial. Identical contours levels are used for both plots. Also, the simulations were started from the same initial conditions and integrated to the same time.

code for an unsteady flow and highlighting the strengths of the method. The intentionally selected small external flow domain favors the vorticity–velocity code because of the method's ability to impose irrotational boundary conditions and robustly handle outflow boundary conditions. The outflow will be a problem for $\mathcal{NEKTAR}$ and will have to be treated with a viscous sponge outflow boundary condition that acts to dampen the waves created by the outflow boundary condition.

Figure 10 compares the instantaneous vorticity calculated by the two codes. The vorticity is similar near the cylinder. However, the differences at the side walls are due to the blocking effects of the velocity–pressure boundary conditions used in the velocity–pressure code. Also, the viscous outflow sponge used to stabilize the velocity–pressure code is affecting the vorticity at the outflow. A comparison of the average streamwise velocity profile at $x/D = 1$ of the average field in Fig. 11 shows the *blockage effect* from the boundary conditions in the velocity–pressure code. Notice that the blockage effect from the boundary conditions is not limited to the exterior flow, but effects the wake profile dramatically. Hence, the difference between the two simulations is significant in the wake. A comparison of the time trace and power spectrum from the shear layer on the cylinder and the near wake show very good agreement between the simulations. The dominant nondimensional frequency (Strouhal number) is 0.242 for the vorticity–velocity code and 0.25 for the velocity–pressure code. The slightly higher frequency of 0.25 predicted by the velocity–pressure code is because the blockage effects due to the close external boundaries is more pronounced.

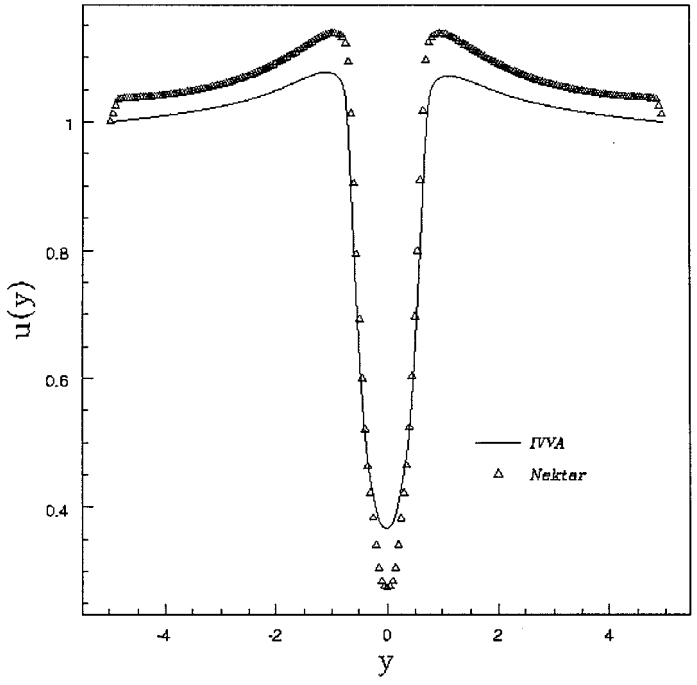

FIG. 11. A comparison of the u-profile at $x/D = 1$ of the average field from $\mathcal{IVVA}$ and $\mathcal{N}\varepsilon\kappa\mathcal{T}\alpha\mathbf{r}$. The effects of blockage for the velocity–pressure formulation are obvious.

6.2. *Three-Dimensional Flow Past a Cylinder with End-Plates*

In this simulation we consider flow past a finite length cylinder mounted on end-plates. This configuration is used in experimental arrangements in order to minimize oblique shedding [10]. However, experiments with relatively small aspect ratio (AR) [34], i.e. cylinder length over diameter, have shown that the formation length is a strong function of this aspect ratio. The formation length here is defined as the maximum length of the recirculating zone in the near-wake.

Here we consider such a model problem with the end-plates located in a region from $-10 < x < 4.5$ and $-10 < y < 10$ at both $z = 0$ and 10 while the exit of the domain goes to $x = 25$. All lengths are normalized by the diameter and the origin is located at the center of the cylinder. In Fig. 12 a sketch of the computational model problem is shown. Note

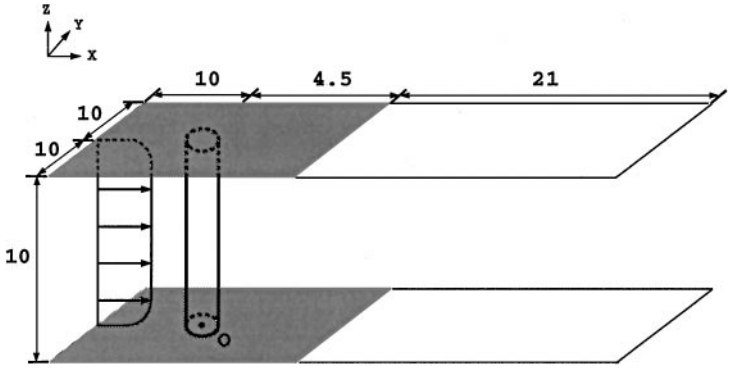

FIG. 12. Sketch of the 3D cylinder flow domain bounded by end-plates (shaded).

the origin of the coordinate system is denoted with O. The Reynolds number based on the cylinder diameter and maximum inlet velocity is 500. At this Reynolds number we expect the cylinder wake to be turbulent for large aspect ratios. Here the ratio of distance between the walls to cylinder diameter is 10, which is on the lower side of what is typically used in an experiment. The Y direction is periodic throughout the entire domain. Downstream past the endplates, the Z direction is also periodic. The usual outflow boundary conditions shown to work in 2D cylinder flow are applied here.

The inlet velocity is

$$u(z) = 1 - e^{-2z} - e^{2(z-10)}, \quad 0 < z < 10, \tag{26a}$$

$$u(0) = u(10) = 0. \tag{26b}$$

We specify such an inlet velocity in order to avoid the singularity in vorticity that occurs in the boundary layers at the leading edge of the walls. The exponential form of the boundary layers is supposed to mimic the shape of a boundary layer. The boundary layer has 10 diameters to "adjust" before it encounters the cylinder. The boundary layers are viewed as a perturbation to the cylinder wake. A horseshoe-type vortex is expected at the junction where the cylinder and walls meet. In addition, a shear layer is expected at the trailing edge of the end-plates. These features break the symmetries seen in the infinite cylinder flow case.

The experience gained with 2D cylinder flow guides us in constructing the mesh in the plane perpendicular to the cylinder. In this cross section, there are 350 elements where most of the elements are concentrated around the body and in the wake. Elements are heavily concentrated in the boundary layers on the surface of the cylinder. There is at least one element in the boundary layer. The mesh is structured in the Z direction. There is a clustering of elements near the end-plates in order to resolve the boundary layers shown in the exploded view in Fig. 13.

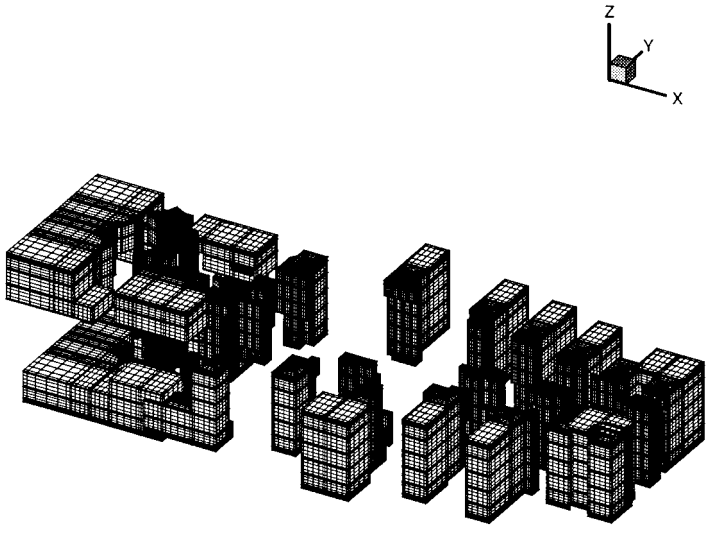

FIG. 13. Domain divided into 32 subdomains using the METIS package (see Appendix). The domain decomposition package tries to minimize the number of cuts while maintaining the same number of elements in each subdomain. By minimizing the number of cuts, the communication time is minimized. By maintaining the same number of elements in each subdomain, load balance across the processors is maintained.

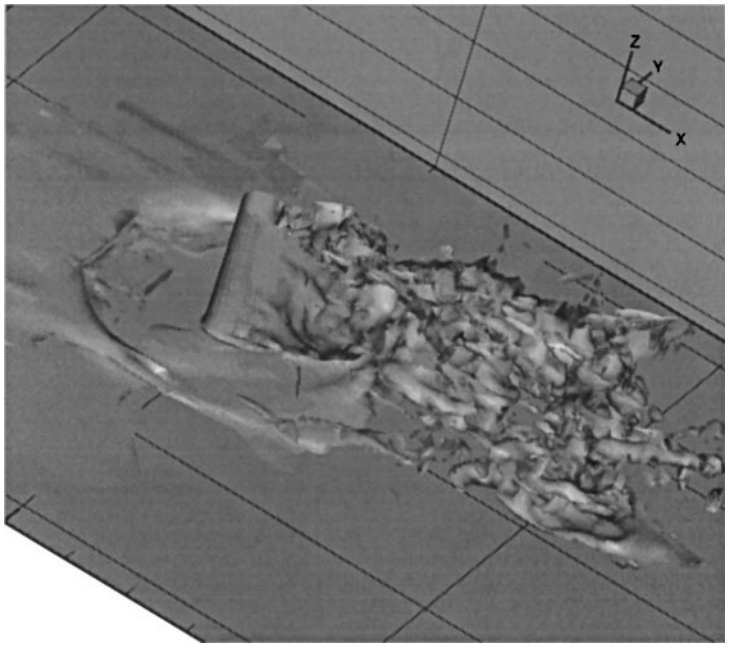

FIG. 14. Plot of the vorticity contours of half the cylinder domain at Reynolds number 500. The incoming boundary layer and developing wake are visible. Also, a horseshoe vortex structure is present at the junction between the cylinder and endplate.

A perspective view of the instantaneous vorticity field from the simulation is shown in Fig. 14. The incoming boundary layer and developing wake is shown. A horseshoe vortex at the junction between the cylinder and end-plates is also visible. However, the flow is still developing. The initial model problem consisted of a cylinder bounded by walls. That model was integrated for 20 convective units before switching to the current model, where the cylinder is bounded by end-plates. The end-plates simulation has been integrated for five convective units. A two-dimensional corss section of the boundary layer shown in Fig. 15 at the planes $y = 0$. The chaotic wake and the large recirculation pattern formed in front of the cylinder can be seen on this plane.

The contours of zero streamwise (instantaneous) velocity are shown in Fig. 16. The contour behind the cylinder is an estimate of the steady formation length. The formation length peaks at the middle of the cylinder. The formation length is decreasing as the wall is approached. The zero streamwise contrours in the front of the cylinder identify the recirculation zone as occurs at the junction between the cylinder and the end-plates. The zero in the streamwise velocity after the cylinder and near the periodic sides is a consequence of the end-plates. From Fig. 16, the maximum nondimensional formation length behind the cylinder can be estimated as approximately 4.3. This is significantly larger than the lengths predicted by experimental and computation results for large to infinity aspect ratio cylinders shown in Table I. The formation length measured in the experiments of Park and Gharib [35] shows the trend of increasing formation length with decreasing aspect ratio. The experiment by Gerrard [14] shows some large variation but it does not state the aspect ratio of the experimental setup. The simulation using $\mathcal{IVA}$ predicts a large instantaneous formation length at the midspan for an aspect ratio of 10, while the length decreases by 25% by the $\frac{1}{4}$-span. The simulation using a Fourier version (in the span) of $\mathcal{NEKTAR}$ has no influence from end-plates and predicts the smallest formation length

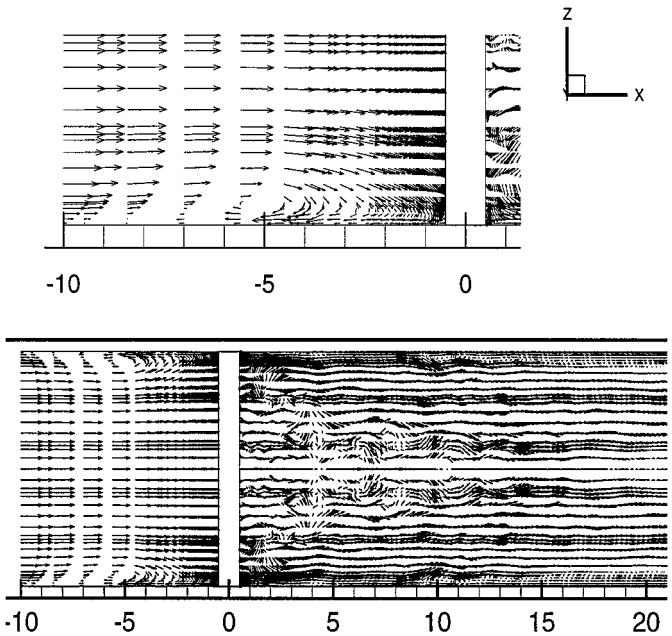

FIG. 15. Two-dimensional slice of the velocity vectors at the centerline, $y = 0$, for Reynolds number 500.

where the zero of the streamwise velocity is the indicator. One expects overprediction of the formation length compared to the infinite cylinder case when the aspect ratio between the walls is only 10. The conclusion from the simulation by $\mathcal{IVVA}$ is that the end-plates tend to increase significantly the formation length behind the cylinder, in agreement with experimental evidence. Note that while the time acuracy in $\mathcal{IVVA}$ is only first-order due to the penalty term, the above results are independent of time discretization errors as verified by performing several simulations corresponding to very small time steps.

7. SUMMARY

A vorticity–velocity algorithm has been presented to solve the incompressible Navier–Stokes equations for complex three-dimensional geometries. The equivalence between the rotational and Laplacian forms of the vorticity–velocity equations and the canonical

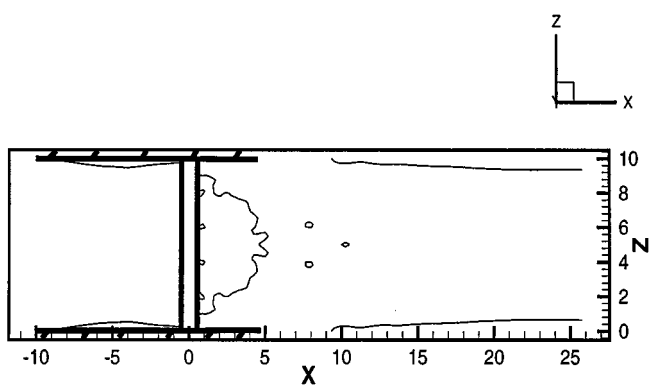

FIG. 16. Instantaneous contours of zero streamwise velocity, $u = 0$, for Reynolds number 500.

TABLE I
Formation Length for Flow Past a Cylinder at Reynolds Number 500

Cases	Velocity $u = 0$		Peak of $\overline{u'u'}$	
	Experiment	Computation	Experiment	Computation
$\mathcal{IVVA}$ midspan AR = 10		4.3		
$\mathcal{IVVA}$ $\frac{1}{4}$-span AR = 10		3.3		
Park & Gharib AR = 15 (1996)	2.5			
Park & Gharib AR = 46 (1997)	1.59		1.57	
Park & Gharib AR = 144 (1997)	1.59		1.45	
Gerrard (1978)	1.70/2.8		1.70	
$\mathcal{N}\varepsilon\kappa\mathcal{T}\alpha\boldsymbol{r}$ AR = ∞		1.52		1.61

Note. The formation length measured in experiments by: (1) Park and Gharib with aspect ratios (AR) of 15, 46, and 144, and (2) Gerrard with unknown aspect ratio reported several different lengths; and in simulations by: (1) $\mathcal{IVVA}$ with aspect ratio of 10, and (2) the Fourier version of $\mathcal{N}\varepsilon\kappa\mathcal{T}\alpha\boldsymbol{r}$ based on the velocity–pressure formulation with infinite aspect ratio.

velocity–pressure equations is stated with emphasis on the boundary constraints and initial conditions. A penalty method used to impose the vorticity boundary conditions is developed and validated on an unsteady 2D flow past a cylinder and a steady 3D flow in a pipe expansion. An analysis of the often-used influence matrix techinque shows that the method does not converge as Δt goes to zero for high-order spatial discretizations. The lack of convergence of the influence matrix technique as the time step decreases is demonstrated by perturbing an exact 2D Navier–Stokes solution. Unsteady flow past a 3D cylinder with end-plates at Reynolds number 500 is simulated for first time. The effect of the end-plates on the formation length is compared with experiments. Parallel timings of the 3D cylinder with end-plates on a Silicon Graphics Origin2000 parallel system shows fairly good parallel scaling.

The numerical formulation of the vorticity–velocity equations can be solved by expanding the vorticity and velocity on nonstaggered grids or on staggered grids. Researchers who have suggested that the numerical formulation should reflect the fact that the vorticity is equal to the derivatives of the velocity have constructed their finite difference [8, 9, 29, 30, 17, 46, 18, 32, 56, 11, 13] and finite element schemes [16, 15, 19, 39, 40] on staggered grids. However, the *necessity* of expanding vorticity/velocity on different grids still remains an unresolved theoretical question. In the splitting formulation presented in this paper, both the vorticity and velocity are expanded and solved on the same grid. No instabilities of any sort were observed even for very long time integration, and several simulations included in [51] confirm this result. Implementations at different interpolation orders did not affect stability, only accuracy. Solving for the vorticity and velocity on nonstaggered grids has several distinct advantages over staggered grids, including ease of implementation, ease of extension to high-order, and ease of extension to unstructured grids.

APPENDIX: IMPLEMENTATION

The vorticity–velocity equations are discretized in time using a stiffly stable time integration scheme developed by Karniadakis *et al*. [25, 49]. The spatial discritization is performed using the spectral element formulation [38]. The fast iterative/direct Schur complement method used here was developed independently by Sherwin and Karniadakis [42]

and Couzy and Deville [7], where the Schur complement is solved iteratively and the interior matrices are solved directly. There are several reasons to work with the Schur complement, also known as the nonoverlapping subdomain method of the spectral element equations: (1) the Schur complement inherits the symmetric positive definiteness of the original system—it guarantees good convergence properties for the iterative solver, (2) denser matrix—vectorization accelerates matrix–vector products, (3) rank is the number of points on the boundary of subdomains—cost per iteration is cheaper, (4) the condition number of the Schur complement matrix is bounded by the original matrix [44], the finite element predicts $\mathcal{O}(h^{-1})$, where the original system is $\mathcal{O}(h^{-2})$ [4]—translates into fewer iterations to solve systems and (5) the interior solves are decoupled and can be solved independently—a benefit on parallel architectures because this reduces communication costs. A projection technique [12] called the successive right-hand side (RHS) accelerator developed by Fischer accelerates the iterative Schur complement solver.

All partitions are generated by a mesh partition package called METIS[2] [26], which uses a multilevel graph partition algorithm that coarsens and then projects backward toward the original finer graph. The communication interface for the "direct stiffness summation" needed in the spectral element formulation is performed by a "Gather-Scatter" library developed by Tufo and Fischer [53] based on the *message passing interface* (MPI). The output from the parallel algorithm is performed so that each processor writes its own self-contained data set. The data can be concatenated, together with a "cat" system call and viewed as a whole, or a subgroup of partitions can be viewed for the economy of postprocessing.

The parallel performance of the unsteady vorticity–velocity Navier–Stokes algorithm is validated by timing a production case problem described in Section 6.2, the 3D cylinder with end-plates at Reynolds number 500. The timings are performed on the Silicon Graphics Origin2000 parallel system at the National Center for Supercomputing Applications. The following results represent a typical timing performance without and with the successive RHS acceleration technique. The saving in wall clock time can be as dramatic as 50% when using the acceleration technique for unsteady flows. The success of the acceleration technique can be attributed to the fact that the dynamics from the previous time steps can be a fairly good approximation to the flow at the current time step. The successive RHS technique has a speed versus memory trade-off. The memory requirements increase as more right-hand sides are stored. Eventually, as more RHS are included, there are diminishing returns in the speedup. The optimal number of RHS is dependent on the unsteady history of the problem and therefore is problem dependent.

The average time/time step is computed by using the last 20 steps of a 23-step run and computing the mean using $\bar{x} = (1/n)\sum_{i=1}^{n} x_i$. The error bars on the average time/time step are computed using the standard deviation, $\sigma = \sqrt{(1/(n-1))\sum_{i=1}^{n}(x_i - \bar{x})^2}$. The nearest neighbor communication required when performing "direct stiffness assembly" is achieved by the "gs" package involving both pairwise and tree communication. The cutoff for pairwise communication is set to 5, so any nodes shared by five or more processors send/receive data using a tree algorithm; otherwise a pairwise communication is used to send/receive data.

The average time/time step scales roughly as $1/P$ shown in Fig. 17. The successive RHS technique consistently gives a 50% reduction in computational time. The problem

[2] METIS is copyrighted by the regents of the University of Minnesota.

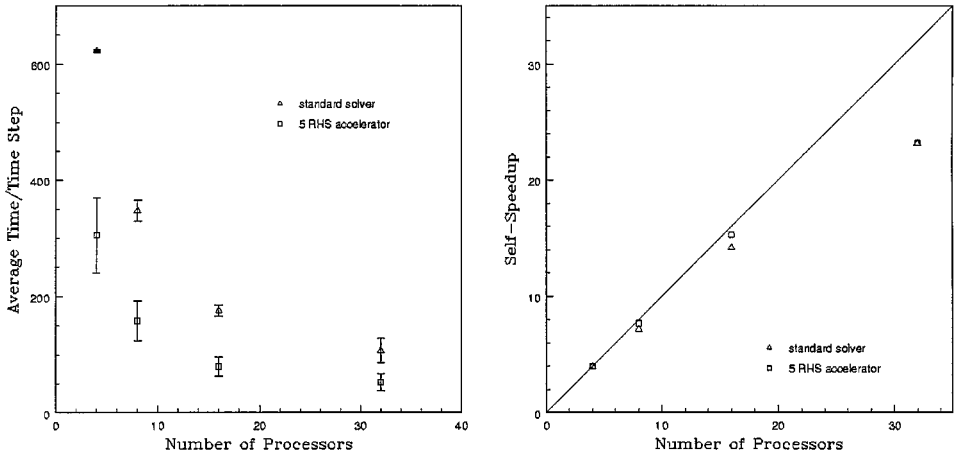

FIG. 17. Average time/time step in seconds, including the standard deviation for a 3D cylinder with end-plates Navier–Stokes calculation with 2366 elements at sixth-order polynomial running on Silicon Graphics Origin2000. The standard solver is the iterative/direct Schur complement solver, while the 5 RHS accelerator is the iterative/direct Schur complement solver with five successive right sides used to accelerate the calculation.

scales nearly linear with processors for small number of processors; see Fig. 17. Note the self-speedup uses the four-processor run as the reference point, because the test case would not fit on one processor due to memory constraints.

ACKNOWLEDGMENTS

We thank Professor David Gottlieb for suggesting the penalty method and Professors Sausors Abarbanel and Paul Fischer for many useful discussions. We also thank the students in our group Tim Warburton, Ma Xia, and Mike Kirby for helping with some of the simulations. James Trujillo received funding from the NASA Graduate Student Fellowship NGT 70289. Partial support was also provided by ONR Grant N00014-90-1315, NSF Contract ECS-90-23362, and AFOSR AASERT Grant F49620-96-1-0267. Computations were performed at the National Center for Supercomputing Applications, University of Illinois at Urbana-Champaign.

REFERENCES

1. S. Abarbanel and A. Ditkowski, *Multi-dimensional Asymptotically Stable 4th-Order Accurate Schemes for the Diffusion Equation*, ICASE Report No. 96-8, February 1996, p. 37.
2. S. Abarbanel and A. Ditkowski, *Multi-dimensional Asymptotically Stable Finite Difference Schemes for the Advection-Diffusion Equation*, ICASE Report No. 96-47, July 1996, p. 36.
3. B. F. Armaly, F. Durst, J. C. F. Pereira, and B. Schonung, Experimental and theoretical investigation of backward-facing step flow, *J. Fluid Mech.* **127**, 473 (1983).
4. 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 ed.* (SIAM, Philadelphia, 1994).
5. A. Bayliss and E. Turkel, Mappings and accuracy for Chebyshev pseudo-spectral approximations, *J. Comput. Phys.* **101**(2), 349 (1992).
6. E. J. Chang and M. R. Maxey, Unsteady flow about a sphere at low to moderate Reynolds number. Part 1. Oscillatory motion, *J. Fluid Mech.* **277**, 347 (1994).
7. W. Couzy and M. O. Deville, A fast Schur complement method for the spectral element discretization of the incompressible Navier–Stokes equations. *J. Comput. Phys.* **116**, 135 (1995).
8. O. Daube, Resolution of the 2D Navier–Stokes equations in velocity–vorticity form by means of an influence matrix technique, *J. Comput. Phys.* **103**, 402 (1992).

9. O. Daube, J. L. Guermond, and A. Sellier, On the velocity-vorticity formulation of Navier-Stokes equations in incompressible flows, *C.R. Acad. Sci. Paris Ser. II* **313**, 377 (1991).
10. H. Eisenlohr and H. Eckelmann, Vortex splitting and its consequences in the vortex street wake of cylinders at low Reynolds number, *Phys. Fluids A* **1**, 189 (1989).
11. A. Ern and M. D. Smooke, Vorticity-velocity formulation for three-dimensional steady compressible flows, *J. Comput. Phys.* **105**, 58 (1993).
12. P. Fischer, Projection techniques for iterative solution of $Ax = b$ with successive right-hand sides, Tech. Rep., Center for Fluid Mechanics, Brown University, 1993.
13. J. Fontaine and L. Ta Phuoc, Wakes past a finite spanwise plate, *C.R. Acad. Sci. Paris Ser. IIb* **320**, 581 (1995).
14. J. H. Gerrard, The wakes of cylindrical bluff bodies at low Reynolds number, *Philos. Trans. R. Soc. London Ser. A* **288**, 351 (1978).
15. G. Guevremont, *Finite Element Vorticity-Based Methods for the Solution of the Incompressible and Compressible Navier–Stokes equations*, Ph.D. thesis, Concordia University, 1993.
16. G. Guevremont, W. G. Habashi, and M. M. Hafez, Finite element solution of the Navier–Stokes equations by a velocity–vorticity method, *Int. J. Numer. Methods Fluids* **10**, 461 (1990).
17. G. Guj and F. Stella, Numerical solutions of high-Re recirculating flows in vorticity–velocity form, *Int. J. Numer. Methods Fluids* **8**, 405 (1988).
18. G. Guj and F. Stella, A vorticity–velocity method for the numerical solution of 3D incompressible flows, *J. Comput. Phys.* **106**, 286 (1993).
19. M. D. Gunzburger and J. S. Peterson, On finite element approximations of the streamfunction-vorticity and velocity-vorticity equations, *Int. J. Numer. Methods Fluids* **8**, 1229 (1988).
20. J. S. Hesthaven, A stable penalty method for the compressible Navier–Stokes equations. II. Open dimensional decomposition schemes, *SIAM J. Sci. Comput.* **18**(3), 658 (1997).
21. J. S. Hesthaven, A stable penalty method for the compressible Navier–Stokes equations. III. Multi dimensional domain decomposition schemes, *SIAM J. Sci. Comput.* **20**(1), 62 (1998).
22. J. S. Hesthaven and D. Gottlieb, A stable penalty method for the compressible Navier–Stokes equations. I. Open boundary conditions, *SIAM J. Sci. Comput.* **17**(3), 579 (1996).
23. T. J. R. Hughes, W. T. Liu, and A. Brooks, Finite element analysis of incompressible viscous flows by the penalty function formulation, *J. Comput. Phys.* **30**, 1 (1979).
24. L. Kaiktsis, G. E. Karniadakis, and S. A. Orszag, Onset of three-dimensionality, equilibria, and early transition in flow over a backward-facing step, *J. Fluid Mech.* **231**, 501 (1991).
25. G. E. Karniadakis, M. Israeli, and S. A. Orszag, High-order splitting methods for the incompressible Navier–Stokes equations, *J. Comput. Phys.* **97**, 414 (1991).
26. G. Karypis and V. Kumar, *Multilevel k-Way Partitioning Scheme for Irregular Graphs*, Tech. Technical Report TR 95-035, Department of Computer Science, University of Minnesota, 1995. [Also Available on WWW at URL http://www.cs.umn.edu/karypis/]
27. L. Kleiser and U. Schumann, Treatment of incompressible and boundary conditions in 3-d numerical spectral simulations of plane channel flows, in *Proc. 3rd GAMM Conf. Numerical Methods in Fluid Mechanics*, edited by E. H. Hirschel (Vieweg, Wiesbaden, 1980), p. 165.
28. L. Kleiser and U. Schumann, Spectral simulation of the laminar-turbulent transition process in plane poiseuille flow, in *Spectral Methods for Partial Differential Equations*, edited by R. G. Voigt, D. Gottlieb, and M. Y. Hussaini (SIAM-CBMS, Philadelphia, 1984), p. 141.
29. W. Labidi and L. Ta Phuoc, Numerical resolution of Navier–Stokes equations in velocity–vorticity formulation: application to the circular cylinder, in *Proceedings of Seventh GAMM Conference on Numerical Methods in Fluid Mechanics, 1988*, p. 159.
30. W. Labidi and L. Ta Phuoc, Numerical resolution of the three-dimensional Navier–Stokes equations in velocity–vorticity formulation, in *Eleventh International Conference on Numerical Methods in Fluid Dynamics, 1989*, p. 354.
31. E. O. Macagno and T. Hung, Computational and experimental study of a captive annular eddy, *J. Fluid Mech.* **28**, 43 (1967).

32. M. Napolitano and L. A. Catalano, A multigrid solver for the vorticity–velocity Navier–Stokes equations, *Int. J. Numer. Methods in Fluids* **13**, 49 (1991).
33. D. Newman, *A Computational Study of Fluid/Structure Interactions: Flow-Induced Vibrations of a Flexible Cable*, Ph.D. thesis, Princeton University, 1996.
34. C. Norberg, An experimental investigation of the flow around a circular cylinder: Influence of aspect ratio, *J. Fluid Mech.* **258**, 287 (1994).
35. H. Park and M. Gharib, personal communication, December 1997.
36. D. Pathria and G. E. Karniadakis, Spectral element methods for elliptic problems in non-smooth domains, *J. Comput. Phys.* **122**, 83 (1995).
37. L. Quartapelle, *Numerical Solution of the Incompressible Navier–Stokes Equations* (Springer-Verlag, Boston, 1993).
38. E. M. Rønquist, *Optimal Spectral Element Methods for the Unsteady Three-Dimensional Incompressible Navier–Stokes Equations*, Ph.D. thesis, Massachusetts Institute of Technology, 1988.
39. V. Ruas, Variational approaches to the two-dimensional Stokes system in terms of the vorticity, *Mech. Res. Commun.* **18**(6), 359 (1991).
40. V. Ruas, A velocity–vorticity formulation of the three-dimensional incompressible Navier–Stokes equations, *Math. Problems Mech. Ser. I*, 639 (1991).
41. W. Z. Shen and T. P. Loc, Numerical method for unsteady 3D Navier–Stokes equations in velocity-vorticity form, *Comput. & Fluids* **26**(2), 193 (1997).
42. S. J. Sherwin, *Triangular and Tetrahedral Spectral/h-p Finite Element Methods for Fluid Dynamics*, Ph.D. thesis, Princeton University, 1995.
43. S. J. Sherwin and G. E. Karniadakis, Tetrahedral *h-p* finite elements: Algorithms and flow simulations, *J. Comput. Phys.* **124**, 14 (1996).
44. B. Smith, P. Bjørstad, and W. Gropp, *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations* (Cambridge Univ. Press, Cambridge, 1996).
45. C. G. Speziale, On the advantages of the vorticity–velocity formulation of the equations of fluid dynamics, *J. Comput. Phys.* **73**, 476 (1987).
46. F. Stella and G. Guj, Vorticity–velocity formulation in the computation of flows in multiconnected domains, *Int. J. Numer. Methods Fluids* **9**, 1285 (1989).
47. B. Szabo and Z. Yosibash, Superconvergent extraction of flux intensity factors and first derivatives from finite element solutions, *Comput. Methods Appl. Mech. Engrg.* **129**, 349 (1996).
48. R. Temam, *Navier–Stokes Equations* (North-Holland, Amsterdam, 1979).
49. A. G. Tomboulides, M. Israeli, and G. E. Karniadakis, Efficient removal of boundary-divergence errors in time-splitting methods, *J. Sci. Comput.* **4**(3), 291 (1989).
50. J. R. Trujillo, A spectral element vorticity-velocity algorithm for the incompressible Navier–Stokes equations, Master's thesis, Princeton University, 1994.
51. J. R. Trujillo, *Effective High-Order Vorticity-Velocity Formulation*, Ph.D. thesis, Princeton University, 1998.
52. J. R. Trujillo and G. E. Karniadakis, A spectral element vorticity-velocity algorithm for the incompressible Navier–Stokes equations, in *Proceedings 5th International Symposium on CFD, Sendai, Japan, 1993*.
53. H. M. Tufo, Generalized communication algorithms for parallel FEM codes, Tech. Rep., Center for Fluid Mechanics, Brown University, 1997.
54. J. M. Vanel, R. Peyret, and P. Bontoux, A pseudo-spectral solution of vorticity-stream function equations using the influence matrix technique, in *Numerical Methods for Fluid Dynamics II*, edited by K. W. Morton (Oxford Univ. Press, Oxford, 1986), p. 463.
55. T. Warburton, *Spectral/hp Methods For Polymorphic Elements*, Ph.D. thesis, Brown University, 1999.
56. X. H. Wu, J. Z. Wu, and J. M. Wu, Effective vorticity–velocity formulations for three-dimensional incompressible viscous flows, *J. Comput. Phys.* **122**, 68 (1995).
57. Z. Yosibash and B. Szabo, Numerical analysis of singularities in two-dimensions, Part 1. Computation of Eigenpairs, *Int. J. Numer. Methods Eng.* **38**, 2055 (1995).