JOURNAL OF COMPUTATIONAL PHYSICS 105, 354–366 (1993)

Modeling a No-Slip Flow Boundary with an External Force Field

D. GOLDSTEIN

Center for Fluid Mechanics, Brown University, Providence, Rhode Island 02912

R. HANDLER

Naval Research Laboratory, Center for Advanced Space Sensing, Washington, DC 20375

AND

L. SIROVICH

Center for Fluid Mechanics, Brown University, Providence, Rhode Island 02912

Received November 20, 1991; revised June 15, 1992

A novel technique related to Peskin's immersed boundary approach is used to introduce solid surfaces into a simulated flow field. The Navier–Stokes equations permit the presence of an externally imposed body force that may vary in space and time. Forces are chosen to lie along a desired surface and to have a magnitude and direction opposing the local flow such that the flow is brought to rest on an element of the surface. For unsteady viscous flow the direct calculation of the needed force is facilitated by a feedback scheme in which the velocity is used to iteratively determine the desired value. In particular, we determine the surface body force from the relation $\mathbf{f}(\mathbf{x}_s, t) = \alpha \int_0^t \mathbf{U}(\mathbf{x}_s, t')\, dt' + \beta \mathbf{U}(\mathbf{x}_s, t)$ for surface points $\mathbf{x}_s$, velocity $\mathbf{U}$, time t, and negative constants α and β. Examples are presented which include 2D flow around cylinders, 3D turbulent channel flow where one boundary is simulated with a force field, and turbulent channel flow over a riblet-covered surface. While the new method may be applied to complex geometries on a non-Cartesian mesh, we have chosen to use a simple Cartesian grid. All simulations are done with a spectral code in a single computational domain without any mapping of the mesh.

1. INTRODUCTION

Computational techniques for simulating fluid flow model solid surfaces in a variety of ways. In a finite difference or spectral method a velocity or vorticity field is specified or calculated explicitly on a set of boundary points (Roache [27]). For the vortex blob approach, image vortices enforce a no-through-flow condition while vorticity embedded in surface panels can satisfy the no-slip condition (Leonard [19], Pepin [21]). Generally in these approaches no-slip or no-through-flow can be exactly enforced at selected grid points at every time step. In a molecular dynamics scheme the molecular structure of the surface can be represented directly; approaching molecules feel the surface through intermolecular potential force fields (Koplik [18]). The no-slip condition occurs as a direct consequence of the surface structure; surface penetration and adhesion may occur. The direct simulation Monte Carlo model (Bird [3]) simulates molecules statistically and these scatter off surfaces either diffusely or specularly, depending on whether no-slip or no-through-flow is the desired boundary condition. Velocity slip occurs for both molecular methods in a manner consistent with experimental observations. However, of these techniques only in the molecular dynamics approach does the surface have a force that reaches out into the flow to make its presence felt.

When fluid flows over a body it exerts a normal (pressure) force on the surface and, if the surface is no-slip, the fluid also exerts a shear force. Conversely, the surface exerts a force of opposite sign on the fluid; in the no-slip case, this localized force is what brings the fluid to rest on the body. That is, the fluid simply *sees* a body through the forces of pressure and shear that exist along the body surface. (In non-isothermal and non-equilibrium flows other surface effects occur as well.) In an unsteady flow, the surface forces vary in time. We can imagine that, in a simulation, if one were to apply the *correct* set of forces to the model fluid, the fluid would flow as though it were passing over a solid object. That is, the effect of certain boundary conditions can be modeled with an *external* force field rather than with a specification of boundary parameter values. Hence, a flow about a body can be computed in a simple domain on a regular grid in the course of the solution of the overall fluid equations with the body's effect imposed by a force field. It is this point of view that we now pursue.

0021-9991/93 $5.00

The concept of modeling a boundary immersed in the fluid with a force field has been used before although most, but not all, of the works involved two-dimensional flows and few if any of the flows were continuously turbulent. Also, none of the authors appear to have used a pseudo-spectral approach to compute the flow. The immersed boundary ideas were applied extensively to model moving boundaries of: heart valves (Peskin [22], Peskin and McQueen [25]) a beating heart (Peskin [23], Peskin and McQueen [24], McQueen and Peskin [20]) a swimming fish-like structure (Fauci and Peskin [10]) and flexible particles in suspension (Fogelson and Peskin [12]). The basis of these models is that the solid being modeled is defined by a set of connected boundary points which move (Lagrangian) through a fixed (Eulerian) mesh. The boundary points are connected by elastic links which create internal forces (to the body) which are transmitted to the surrounding fluid since the boundary points and the links are massless. The interior solid forces, calculated by an approximately implicit scheme or an implicit scheme (Tu and Peskin [32]; Peskin [26]) provide part of the force density in the Stokes or Navier–Stokes equations. The forces and velocities are interpolated between the boundary points and the fixed mesh. Sulsky and Brackbill [31] modeled elastic particles in a suspension, where the force density in the particles is related to the displacement field computed from the stress–strain constitutive equations for an elastic solid. The flow is computed with a finite volume technique; interior as well as surface points represent the solid body. Unverdi and Tryggvason [33] modeled droplets using a localized body force to simulate the effects of surface tension. They used front-tracking particles to maintain the separation between two fluids and calculated the whole flow with a finite difference approach.

The works cited above were primarily concerned with boundaries moving through a fixed mesh and the flow field was computed with a finite difference/element approach. The boundaries move with the local flow velocity and the force density is determined from internode forces within the body. The forces between boundary points rather than the relative (or absolute) location of the boundary points are specified. The solid body was modeled as a set of massless nodes interconnected by springs; the fluid mediates the internode forces. In these approaches the force computation is fairly complex as it must model the interior stresses and strains of the solid. If the locations of the boundary points (rather than the forces between them) are specified, the problem is simpler. The force produced by each point may then be computed independently (Peskin [22], Fauci [11]). In [11] this approach has been combined with a vortex blob technique and has been shown to provide qualitatively accurate solutions. An earlier related approach is that of Viecelli [34–35] in which a shear-free surface is modeled with an adaptive pressure field in a marker and cell technique. The pressure along some desired boundary is used to enforce the no-through-flow condition; if fluid flows through the boundary the pressure is increased on the boundary until the through-flow ceases.

2. PRESENT APPROACH

We write the incompressible Navier–Stokes equations with an external force field as

$$\rho \frac{\partial \mathbf{U}}{\partial t} = \rho \mathbf{U} \times \boldsymbol{\Omega} - \nabla \left(p + \rho \frac{\mathbf{U} \cdot \mathbf{U}}{2} \right) + \mu \nabla^2 \mathbf{U} + \mathbf{T}_{ij} n_j \delta(S) \tag{1a}$$

and

$$\nabla \cdot \mathbf{U} = 0. \tag{1b}$$

Here, t is time, $\mathbf{U} = (u, v, w)$ is the velocity, $\boldsymbol{\Omega}$ is the vorticity, p is the pressure, ρ is the density, and μ is the viscosity.

The *force* term on the right-hand side of (1a) requires some explanation. $S = 0$ denotes the surfaces which delimit the *body* and $\delta(S)$ is the three-dimensional delta function having the property

$$\int_{\text{all x}} \phi \delta(S)\, d\mathbf{x}^3 = \int_S \phi\, dS. \tag{2}$$

In this context $\mathbf{T}_{ij} n_j$ is the force exerted by the body on the fluid, where $\mathbf{n}$ is the normal pointing out of the fluid domain. This formalism, which was derived in Refs. [28–30], justifies and demonstrates several aspects of the formalism now being developed. First, for a given flow with no-slip boundary conditions, say, there exists a force system on the surface which if applied at the boundary will bring the flow to rest at each point of the bounding surface. In the customary context of fluid mechanics this force is calculated after the flow has been determined. As will be seen, however, we shall obtain the force on each element of the surface during the course of calculation which will enforce the no-slip boundary condition. A second feature of this formalism is that (1) is derived for the fluid occupying all space. Thus, if $S = 0$ separates an *inside* from an *outside*, the force field at the surface sets up a fluid flow in a domain which is not of immediate interest. This feature will emerge in our calculations. Finally, we observe that the formulation given in (1) also demonstrates that each element of the surface may be treated as a separate entity, and the surface is a linear superposition of the correct forces appropriately located.

We will denote by $\mathbf{f}(\mathbf{x}_s, t)$, where $\mathbf{x}_s$ is a boundary point, the force on the element of surface. In general, in an

unsteady viscous flow the method of choosing $\mathbf{f}(\mathbf{x}_s, t)$ such that $\mathbf{U}(\mathbf{x}_s, t) = 0$ is not self-evident. We will permit the force to adapt itself to the local flow field. In particular, consider the force governed by the following feedback loop:

$$\mathbf{f}(\mathbf{x}_s, t) = \alpha \int_0^t \mathbf{U}(\mathbf{x}_s, t')\, dt' + \beta \mathbf{U}(\mathbf{x}_s, t). \tag{3}$$

The quantities α and β are negative constants having dimensions of $M/(L^3T^2)$ and $M/(L^3T)$, respectively. This force field represents a straightforward feedback of the velocity field information and in the realm of control systems theory it is called a two mode controller (Dorf [9]). Here the error being fed back is the velocity integral and the velocity itself. One might at first think that using concepts from linear control theory in simulations of turbulent and highly non-linear flows would be difficult, especially because the control is digital rather than analog (Dorf [9]). Yet, as will be seen, we only need to control small regions of the flow where near-linear responses might be expected. The first term with integral feedback is alone sufficient to create a force field that will bring the flow to rest on the surface points. If $\mathbf{U}$ is, say, positive along one of the Cartesian axes, the force will increase with time to oppose the flow until the flow comes to rest. The second term can be thought of as representing a force created by the Stokes drag of an obstacle located at $\mathbf{x}_s$ but which is too small to be resolved, i.e., it might represent the drag on a fine hair located at $\mathbf{x}_s$. Other terms may be contemplated as well. For example, a term involving derivative feedback, i.e., $\gamma(d\mathbf{U}(\mathbf{x}_s, t)/dt)$, might be used to more closely maintain $\mathbf{U}(\mathbf{x}_s, t) = 0$: if $\mathbf{U}$ is slightly perturbed from zero, $\mathbf{f}$ will change abruptly to force the velocity back to zero. A force proportional to $\mathbf{U}|\mathbf{U}|$ might represent the turbulent drag of an unresolved obstacle. A term related to a spatial derivative or integral of $\mathbf{U}$ might also be considered. As will be shown below, however, simply using the time integral and direct velocity feedback is adequate for solving the unsteady incompressible Navier–Stokes equations for flows around a variety of bodies.

Using simple feedback on boundary points may be thought of as creating a force field that *learns* to simulate the boundary condition desired. Hence, other boundary conditions can be simulated as well. For example, if $(\mathbf{U} - \mathbf{v})$ replaces $\mathbf{U}$ in Eq. (3) the surface boundary points will end up having velocity $\mathbf{v}$. If $\mathbf{v}$ is not a tangential velocity, the locations of $\mathbf{x}_s$ will have to change with time.

We can show in a heuristic way that such a feedback scheme may work. Suppose we temporarily consider the momentum equation with forcing but without the first three terms on the right-hand side:

$$\rho \frac{d\mathbf{U}(\mathbf{x}_s, t)}{dt} \approx \mathbf{f}(\mathbf{x}_s, t) = \alpha \int_0^t \mathbf{U}(\mathbf{x}_s, t')\, dt' + \beta \mathbf{U}(\mathbf{x}_s, t). \tag{4}$$

This is the equation for the velocity of a damped simple harmonic oscillator of a mass $\approx \rho\, d\mathbf{x}^3$ with a spring constant α. The oscillation (natural) frequency is $(1/2\pi)\sqrt{|\alpha|/\rho}$ and the damping ratio is $-\beta/(2\sqrt{|\alpha|\rho})$. The integral feedback acts like a spring force on the small volume of fluid near $\mathbf{x}_s$ which, in a sense, tethers with an elastic string that volume of fluid to $\mathbf{x}_s$ and $\beta\mathbf{U}$ provides damping. A larger damping ratio implies greater dissipation of flow perturbations near the surface. An important assumption throughout is that the force acts where the velocity is measured. It is not clear what may result if the two locations are substantially separate. The value of $\mathbf{U}(\mathbf{x}_s, t)$ is most conveniently evaluated if $\mathbf{x}_s$ coincides with a grid point. In practical calculations the computational cell volume $\Delta\mathbf{x}^3$ will be small and the temporarily neglected terms of Eq. (1a) will provide additional damping for these oscillations. There will, however, be a somewhat ill-defined apparent mass term since the volume of fluid feeling $\mathbf{f}$ will impart motion to the surrounding flow. This additional apparent mass will cause a decrease in the oscillation frequency. In an unsteady flow, α must be large enough so that the resulting natural frequency is higher than the most energetic flow frequencies in order that the force field can track the changing flow. The method by which the time integral is evaluated will determine the maximum magnitude of α.

In the simulations discussed below the time integral is approximated simply as a Riemann sum

$$\int_0^t \mathbf{U}(\mathbf{x}_s, t')\, dt' \approx \sum_{j=1}^{N} \mathbf{U}(\mathbf{x}_s, j)\, \Delta t, \tag{5}$$

where N is the number of steps and Δt is the size of the time step. This integration scheme, for large gain, is clearly unstable for the simple harmonic oscillator problem. In practice, we find that the solution to Eqs. (1a), (1b), and (3) is still unstable for large enough α or β but is stable for moderate values and is insensitive to their exact values. In particular, in the codes discussed below, the time marching of the forcing term is done with a second-order accurate Adams–Bashforth scheme. We find that the stability limit for the time step is approximately given by

$$\Delta t < \frac{-\beta - \sqrt{(\beta^2 - 2\alpha k)}}{\alpha}, \tag{6}$$

where k is a problem dependent constant of order one.

The forcing scheme is introduced into a spectral method solver of the incompressible Navier–Stokes equations and the following discussion will be specific to that approach. We note, however, that the scheme can be used in other methods that permit the introduction of local forces, even if the method solves more complex equations such as the Boltzmann equation or higher order equations describing liquid molecular dynamics.

3. SPECTRAL METHOD

We use a pseudo-spectral method in two or three spatial directions to solve the incompressible Navier–Stokes equations and use body forces to simulate solid obstacles in the flow. The spectral method code, developed by Handler *et al.* [15] is based on that of Kim *et al.* [17] in which the Navier–Stokes equations are rewritten for channel flow as a fourth-order equation for the wall-normal velocity and a second-order equation for the wall-normal vorticity. In this method the incompressibility condition is satisfied exactly at each time step. We first will consider a rectangular three-dimensional channel, periodic in the streamwise direction (x) and the spanwise direction (z) and bounded by impermeable surfaces in the vertical direction (y). We can impose no-slip or shear-free boundary conditions on either wall. The flow quantities are represented with a Fourier expansion in the periodic direction(s) and a Chebyshev expansion in the wall-normal direction. The code was originally created to study turbulent channel flow.

It is convenient, because of the numerical method used, to add the force $\mathbf{f}$ to the non-linear term $\mathbf{U}\times\boldsymbol{\Omega}$ since this term is most efficiently evaluated in physical space. De-aliasing in the x and z directions occurs during the plane by plane evaluation of $\mathbf{U}\times\boldsymbol{\Omega}$ by interpolating $\mathbf{U}$ and $\boldsymbol{\Omega}$ onto a grid having $\frac{3}{2}$ as many colocation points in each direction. The force field required is then also calculated on this expanded grid. Because these colocation points may not correspond to the regular grid sites, the surfaces created with the force field are properly interpreted only on the expanded grid.

In order to generate a *smooth* surface rather than a step-like surface which would result from applying the force only to the grid sites of the rectangular grid, we smooth the force field in space. In the two periodic directions the force at each surface point $\mathbf{x}_s$ is defined by a narrow Gaussian distribution, $\mathbf{f}(\mathbf{x},t)=\mathbf{f}(\mathbf{x}_s,t)e^{-[(i-i_s)^2+(j-j_s)^2+(k-k_s)^2]}$, where the surface point $\mathbf{x}_s$ is located at grid site (i_s,j_s,k_s). The immediately adjacent points receive about 37% of the central force while points two grid sites further away receive essentially none. The effect of this local smoothing is to blur the location of the surface and extend its reach slightly. Hence, the exact location of the surface may be uncertain to within one grid site. If sufficient spatial resolution has been used we find that the exact smoothing function is not crucial provided it has similarly compact support. In the simulations of three-dimensional flow discussed below, such spatial smoothing is not done in the Chebyshev direction. In the papers based on Peskin's approach discussed earlier a similar approach is used to distribute the forces of the boundary nodes to the nearby grid sites.

Introducing point forces into a spectral representation of the flow posed some unique problems because singularities tend to produce significant global spatial oscillations even in a temporally steady flow. These oscillations are distinct from the temporal oscillations discussed above (see Eq. (4)). Those oscillations come from the actual solution to Eqs. (1) and (3) while the presence of singularities causes unrealistic oscillations due to the global nature of the expansion functions used in the spectral method. Even though the force field is a body force applied throughout the cell volume and not a point force, the effective force has a sharp peak. We find, however, that the spatial oscillations did not grow in an unstable manner in time (see also Gottlieb *et al.* [13]) and, if left alone, they simply remained at about constant amplitude during the calculation. As might be expected, the oscillations as reflected in the energy spectra of the velocity field were by far the greatest at the highest wavenumbers and were insignificant at somewhat lower wavenumbers. The unaltered spectral representation is simply unsuited to handle sharp gradients of, say, velocity across a single computational cell. As greater spatial resolution is used, the energy in the highest modes declines and the amplitude of the oscillations declines as well. Yet, at the highest practical resolution, the oscillations were still present.

It was found that two partial remedies could effectively remove these oscillations without either remedy itself being too harsh. In the Chebyshev direction as well as in the Fourier directions, mild spectral smoothing is used. The spectral coefficients of the quantity $\mathbf{U}\times\boldsymbol{\Omega}+\mathbf{f}$ at every time step are multiplied by

$$e^{-(n_x/N_x)^{20}}e^{-(n_y/N_y)^{20}}e^{-(n_z/N_z)^{20}},$$

where (n_x, n_y, n_z) are grid point indices in the (x, y, z) directions and (N_x, N_y, N_z) are the number of modes in those directions. This is a low pass filter. The decay constant 20 is chosen so that there will be a sharp cutoff of the highest modes and so that the highest wavenumber will be reduced by $1/e$. This filtering leaves the first 86% of the modes of $\mathbf{U}\times\boldsymbol{\Omega}+\mathbf{f}$ with over 95% of their original energy at each step. The exact values of the constants should not be important as long as the natural energy cascade has generated little energy at the highest wavenumbers (i.e., all scales are resolved). Similar spectral smoothing has been employed by others as well (Gottlieb *et al.* [14]). Of course, with greater resolution this smoothing will have less influence on the flow because there will naturally be less energy in the highest resolvable modes. It does not appear necessary to remove the highest modes more completely. At very early times when the fluid first begins to flow around the body, the smoothing is most necessary because the boundary layers near any solid surface are thin. As the simulation continues and boundary layers thicken, the gradients across a cell diminish, and the relative amount of energy in the highest modes decreases. (Another possibility, not pursued by us, would be to spectrally smooth the force field itself.)

A second, and perhaps unusual smoothing method, is to create a flow *inside* the solid bodies to reduce kinks that

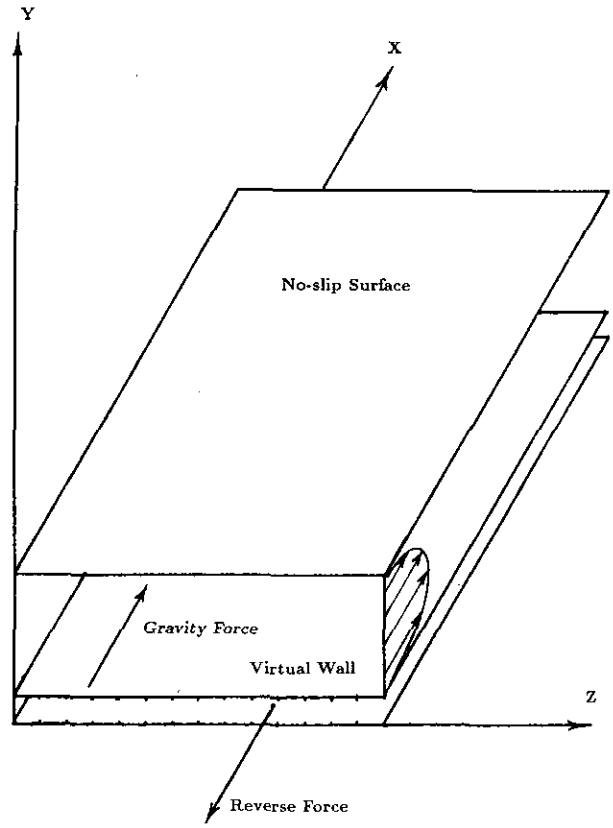

FIG. 1. Geometry for flow simulation of channel bounded by two impermeable boundaries and containing a virtual wall. Uniform gravity is applied above the wall and a reverse force below.

otherwise develop in the mean velocity profile. The idea is analogous to potential flows past bodies for which an internal flow, usually not of interest, is created in the course of solving the problem. In the present instance, the force field maintains the no-slip impermeable conditions on surfaces and the unwanted internal flows are created as part of obtaining a globally smooth velocity field. Slight unforced internal flows may also appear in the natural development of the integration.

As an illustration consider laminar flow between two flat plates (Fig. 1). We suppose that the top plate is a no-slip boundary and the bottom is a shear free boundary, $(d\mathbf{U}/dy) = \mathbf{U} \cdot \mathbf{n} = 0$. Next, a virtual flat plate is inserted near the bottom of the channel with time integral feedback forcing on all grid points at index $n_y = 2$. In the bulk of the channel between the no-slip wall and the modeled wall a uniform pressure gradient or gravitational force G is applied. If no force is applied below the modeled plate the velocity $\mathbf{U}$ and velocity derivative $d\mathbf{U}/dy$ will both remain approximately zero there. Just above this plate $d\mathbf{U}/dy$ will be large and there will thus be a substantial kink in the velocity profile that will cause large oscillations. Figure 2a shows these oscillations and the development with time of the flow toward the theoretical parabolic velocity profile. If, however, in the small space between the virtual plate and the lower wall we apply a large force in the direction opposite to G, a small back flow will develop and the kink is substantially reduced. An appropriate choice of back force below the plate can effectively eliminate the

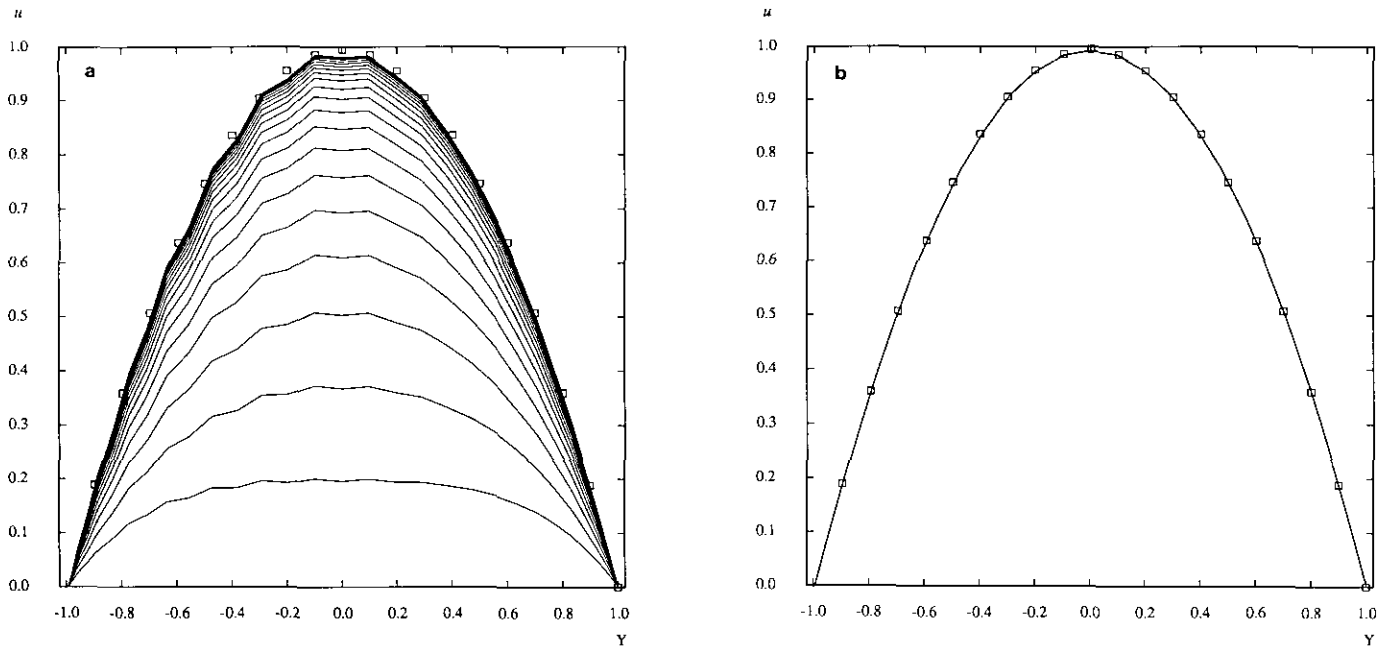

FIG. 2. (a) Time development of u velocity profile in the channel. Note the spatial oscillations. Back force $= 0$. Symbols represent the equilibrium theoretical profile. (b) Equilibrium velocity profile with non-zero backforce. Note the absence of spatial oscillations. The symbols represent the equilibrium theoretical profile.

undesirable oscillations as seen in Fig. 2b. A parabolic flow is obtained (in the sense that the RMS difference between a parabola and the solution integrated over the height of the channel goes nearly to zero) when the back force is such that the energy in the highest Chebyshev u mode is minimized. Minimizing the energy in the highest modes of a time-averaged flow can be used in the turbulent cases as well. This simulation is discussed further in Section 4.

4. RESULTS

We now present solutions to a variety of problems that demonstrate by example the practicality and flexibility of the technique. In the absence of rigorous proofs, these examples support the validity of the methodology presented here.

4.1. *Laminar Flow*

In the above simulation of laminar channel flow we overlay the theoretical parabolic velocity profile and find that it is the same as the simulated result (Fig. 2b). The theoretical profile is: $u = y(h - y)\, G/(2\mu)$, where h is the channel height, $G = 0.002$, and $\mu = 0.001$. The presence of the spatial oscillations did not affect the approach to the solution nor did they provoke a growing instability. In this case the Reynolds number based on channel half width and the centerline velocity is 1000. Even though this is a one-dimensional problem, we chose to compute it on a reduced three-dimensional mesh; $4 \times 32 \times 6$ in the x, y, and z directions, respectively. If we consider a lower Reynolds number or use higher resolution in y, the size of the oscillations would be smaller. The gains in Eq. (3) were taken as $\alpha = -20$ and $\beta = -0.5$; and $\Delta t = 0.1$. After the flow had reached equilibrium, a backforce $(-160G)$ below the virtual plane was turned on. A large reverse force is required because the virtual plane is so close to the shear-free boundary. The spatial oscillations then rapidly disappear as the velocity below this plane reaches -0.00311, a value too small to be seen on Fig. 2b. No spectral or $x - z$ spatial smoothing is required.

Consider next the response of uniform flow at a single grid site suddenly being subjected to the no-slip condition. A two-dimensional flow has velocity $(u, v) = (1, 0)$ at $t = 0$. We use a square spatial mesh 32 grid sites on a side, Fourier in x, and Chebyshev in y. At $t = 0^+$ we turn on the forcing at one site in the center of the flow field and observe the evolution of the u velocity at that site. The computational domain is also 32×32, so a cell has dimensions $\Delta x = \Delta y = 1$. If we choose $\mu/\rho = 1/100$, the Reynolds number of a single cell, $\mathrm{Re}_{\mathrm{cell}}$, based on the initial velocity and Δx, is 100 and viscosity damps out the velocity oscillations slowly (in roughly 80 time steps, Fig. 3). Here $\alpha = -100$, $\beta = 0$, and $\Delta t = 0.1$. If α is varied the frequency response follows well the simple harmonic oscillator prediction of a $\sqrt{\alpha}$ dependence. The differences are likely due to the neglected fluid effects, particularly the added mass, from the first three terms on the right-hand side of Eq. (1a). If we choose $\mathrm{Re}_{\mathrm{cell}} = 2$ so that the viscosity strongly damps the oscillations, the oscillations disappear more quickly but still not quickly enough (about 50 time steps) with $\beta = 0$. If, however, we also use proportional feedback, $\beta = -10$, it is possible to nearly eliminate the oscillations; in Fig. 3 we see that this case is highly damped. This suggests that with a judicious choice of gains and time step it should be possible to simulate unsteady flow around a solid object.

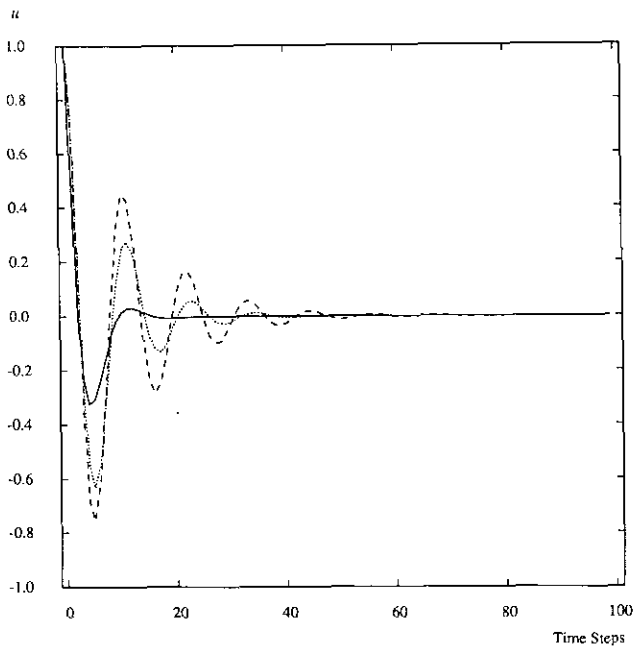

FIG. 3. Time response of velocity u for a single cell. Dashes $-\alpha = -100$, $\beta = 0$, $\mathrm{Re}_{\mathrm{cell}} = 100$; dots $-\alpha = -100$, $\beta = 0$, $\mathrm{Re}_{\mathrm{cell}} = 2$; solid $-\alpha = -100$, $\beta = -10$, $\mathrm{Re}_{\mathrm{cell}} = 2$.

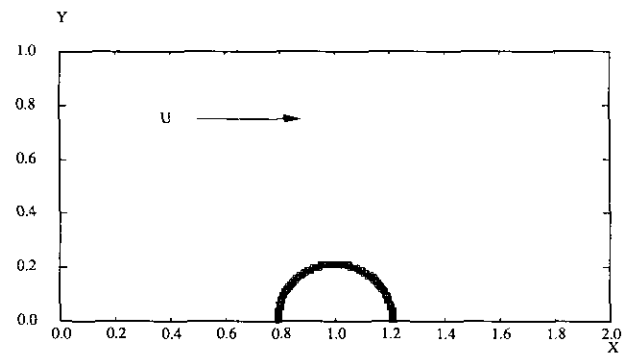

FIG. 4. Geometry for flow simulation of two-dimensional startup flow around a cylinder.

4.2. *Startup Flow around a cylinder*

The startup flow past an impulsively accelerated cylinder has been well documented experimentally (Bouard and Coutanceau [4]) and computationally (Pepin [21]). In the experimental setup a solid cylinder is dragged through a large towing tank in which the tank walls are far enough away that they do not influence the flow. Photographs of flow tracers show the wake structure and various separation events that occur at different Reynolds numbers. In Pepin's computational technique the cylinder surface consists of 576 straight panels and the flow is modeled with nearly 50,000 interacting and overlapping vortex blobs by the end of his simulation. The external flow boundaries are at infinity and the blobs exchange circulation to model viscous diffusion. The surface is modeled with image vortices inside the cylinder and vortex panels on its surface. Pepin found good agreement with the experiments at Reynolds numbers based on cylinder diameter and the velocity at infinity of 550, 3000, and 9500. At Reynolds numbers 3000 and 9500 greater spatial resolution was used.

In our computations using the spectral code we have a more limited computational domain. Since we are at present only interested in the startup case in which the cylinder wake remains symmetric about the cylinder centerline, we only simulate half the domain and place a half cylinder against a shear free boundary at $y=0$. The top ($y=1$) boundary is also shear free. A sketch of the domain is given in Fig. 4. We use a square domain with sides of length two in which a cylinder of diameter 0.44 is centered. The grid resolution is 192 by 129 in the Fourier (x) and Chebyshev (y) directions, respectively. Of course, our grid remains rectangular and $\mathbf{x}_s$ lie on grid points so the cylinder cross section is slightly rough; it approximates a circle with a diameter of 41 grid spacings in x and 80 in y. That represents 62 grid spacings in x on the $\frac{3}{2}$'s expanded grid and the same number (80) in y, since there is no de-aliasing done in the Chebyshev direction. The 306 small squares are drawn in Fig. 4 indicating the locations of $\mathbf{x}_s$.

A simple method for starting the flow is to create an initially uniform velocity in the x direction and at $t=0$ turn on the force field along the boundary points. With an appropriate choice of gains ($\alpha=-4\times10^6$, $\beta=-6\times10^3$) the cylinder gradually appears in the center of the channel as the force field adjusts to the oncoming flow. We use a very small initial time step ($\Delta t=5\times10^{-5}$) corresponding to the stability limits implied by α and β and find that the cylinder is fully formed within $O(10^2)$ time steps and before the flow has progressed more than one-tenth of a diameter downstream. In referring to the flow development we use

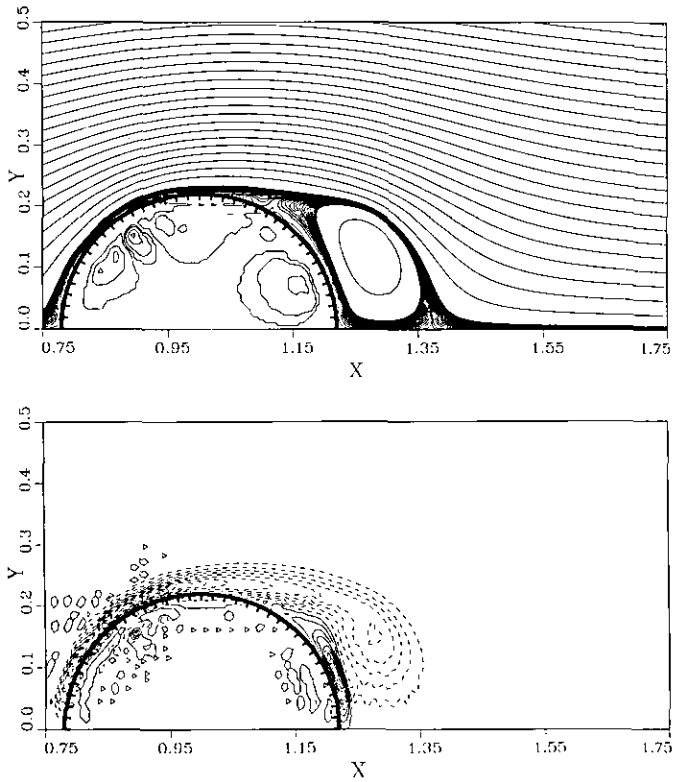

FIG. 5. Closeup view of flow around a cylinder, $\tau=2.12$. Cylinder is indicated by heavy line: (a) Select streamlines; (b) Contours of constant vorticity; solid—positive; dashes—negative.

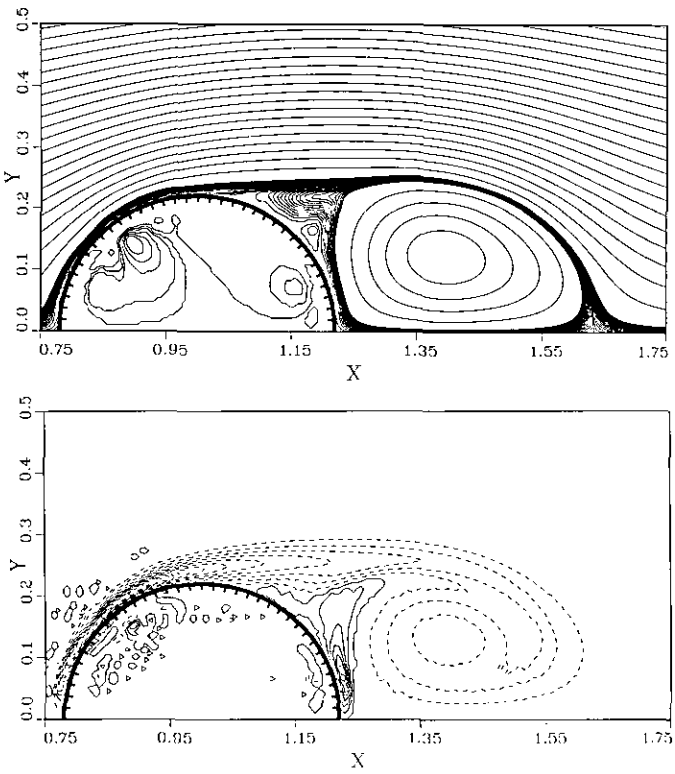

FIG. 6. Closeup view of flow around a cylinder, $\tau=5.12$. Cylinder is indicated by heavy line: (a) Select streamlines; (b) Contours of constant vorticity; Solid—positive; dashes—negative.

the dimensionless time $\tau = tU/R$, where U is the upstream velocity, t is the computational time, and R is the cylinder radius. For comparison, Bouard and Coutanceau estimate that the startup time in their experiment is $\tau < 0.08$. This relaxed state can then be used as the initial flow field and the computation may be continued with a more reasonable time step which more closely approaches the Courant limit ($\Delta t = 5 \times 10^{-4}$ and $\alpha = -4 \times 10^5$, $\beta = -6 \times 10^2$). Alternatively, we can specify a potential flow about a cylinder as the initial state but the force field required to maintain it would be unknown a priori. In addition, we wish to create a general computational technique and the potential flow field around other shapes is not always readily available. There is little or no penalty in using the simple relaxation approach employed here.

The gains α and β are chosen based on the stability limit (6) and the oscillation response time. This, in combination with the normal Courant number restriction, gives us an estimate of the time step that should be used. We choose gain values with a safe margin of error since we do not know for certain the value of k in (6). If the time step is then increased, the gains should be decreased appropriately. We note that for most of the cylinder flow simulations we also use feedback, in much the same manner as for the surface points, to adjust the gravitational force G in such a way so as to maintain a constant mass flux.

We will consider a flow with a Reynolds number of 550. At a time $\tau = 2.12$ the streamlines and contours of constant vorticity are shown in Figs. 5a and b for a portion of the flow field. The cylinder is indicated with a thick line and tick marks spaced by 5° increments. Inside the cylinder a slight flow develops due to the spectral and spatial smoothing of the force field but the magnitude of the flow velocities is small. At this time a recirculation bubble has formed with a primary separation angle of about 60° measured from the three o'clock position. It is clear that the negative vorticity of the primary vortex draws positive vorticity off the cylinder trailing edge. At $\tau = 5.12$ (Fig. 6) the primary vortex has grown to almost the same size as the cylinder itself and has spawned a secondary vortex near the 45° location. The primary separation point has moved forward to about 75°. These flow patterns, including the details of the secondary separation, correspond remarkably well with the calculations of Pepin and the photographs of Bouard and Coutanceau.

In Fig. 7 the plot of the aft stagnation point location versus time shows that our results are somewhat higher than experiment, while Pepin's are slightly lower. The geometrical parameters are defined in the figure. In Fig. 8 it is seen that our streamwise location of the vortex core (defined as a local extremum of the stream function) again agrees well with the other results. At early times the uncertainty in the location of our cylinder surface and the close proximity of the core to that surface is the cause of discrepancy. Similar uncertainty applies to the vertical location of the core (Fig. 9). Our location of the separation angle generally agrees with Pepin's results although we predict a slightly earlier initial separation (Fig. 10).

Figure 11 shows a comparison of drag coefficient ($C_D = D/(\rho R U^2)$, where D is the drag) results. To find D we let $G = 0$ and compute the loss of fluid momentum in the channel with time. Here we do not maintain a fixed upstream mean velocity and we compute τ and C_D based on

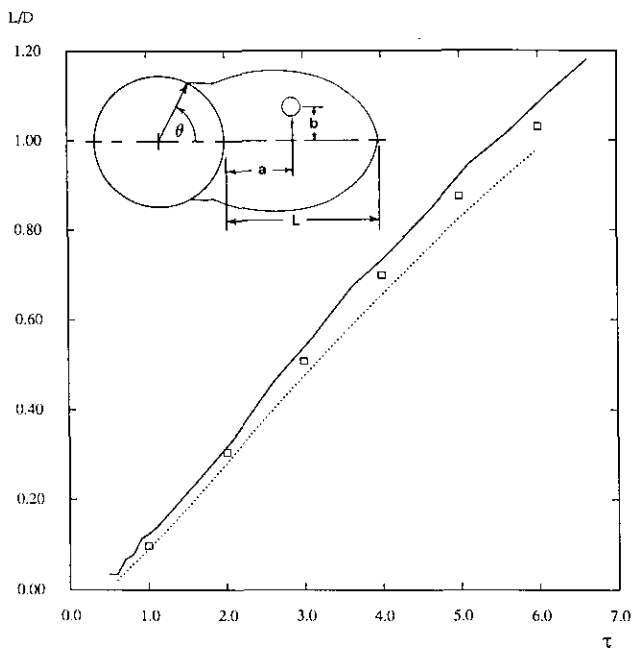

FIG. 7. Location of aft stagnation point vs. time: Solid—present calculations; dots—Pepin; symbols—Bouard and Coutanceau.

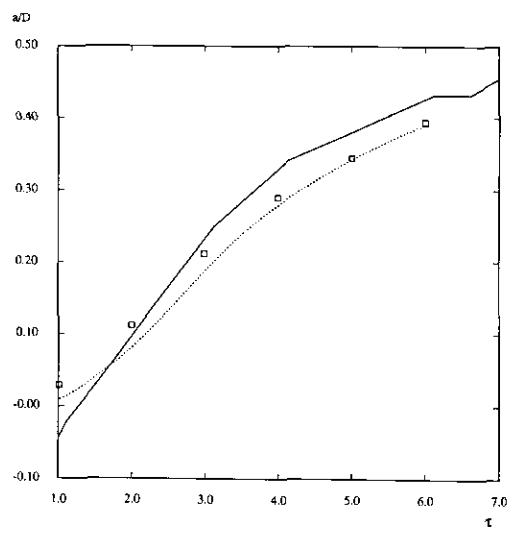

FIG. 8. Horizontal displacement of vortex core vs. time: Solid—present calculations; dots—Pepin; symbols—Bouard and Coutanceau.

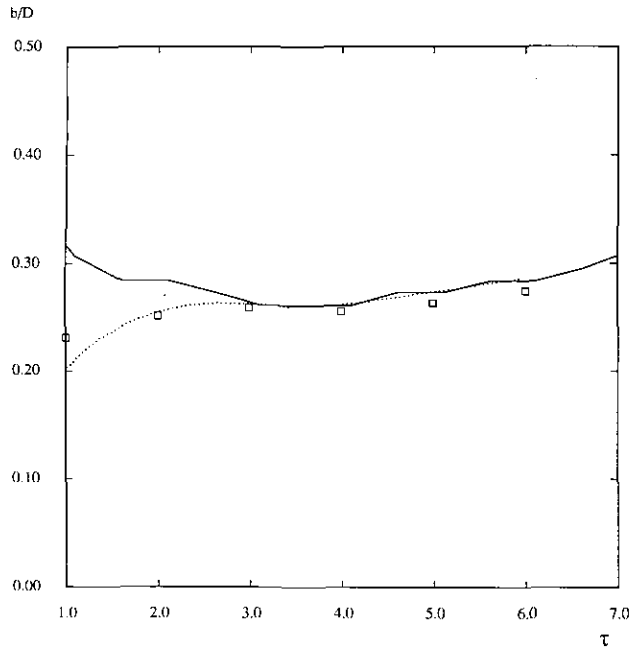

FIG. 9. Vertical displacement of vortex core vs. time: Solid—present calculations; dots—Pepin; symbols—Bouard and Coutanceau.

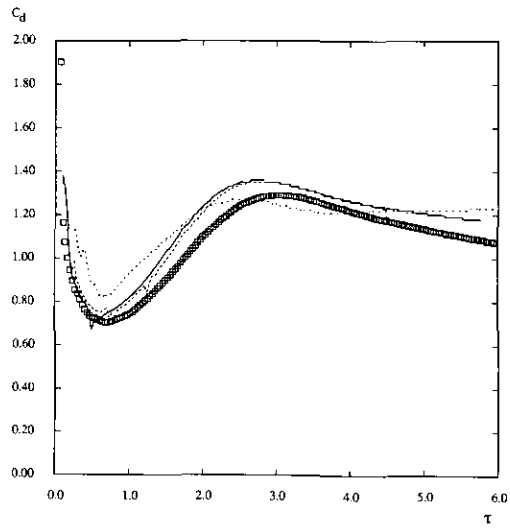

FIG. 11. Coefficient of drag vs. time: Dots—present calculations on 96×64 mesh; solid—present calculations on 192×128 mesh; dashes—present calculations on 300×256 mesh; symbols—Pepin.

the actual mean velocity at $x = 0$, not simply the initial velocity. We use the drag as a measure of the convergence of the simulation. With a spatial resolution of 96×64 grid sites in x and y (only 69 points on the half circle) we find that the drag erratically follows the correct trends (see Fig. 11) and that greater resolution of 192×129 is needed to obtain a converged solution. The 192×129 calculation appears to be close to convergence since there is little difference between it and the higher resolution solution using 300×256 grid points (956 points on the half circle), particularly at later times when the boundary layers are thicker. The spectral solution appears to adequately track the vortex solution at early times even considering that the boundary layers are thin and that there is a $1/\sqrt{\tau}$ drag singularity near $\tau = 0$ (Bar-Lev and Yang [1]). The discrepancy in the results at later times between our method and that of Pepin, though not large, is possibly due to the non-constancy of our U or that Pepin had an infinite domain while ours is finite.

It appears that the proximity of the periodic and shear free boundaries has otherwise little influence at these early times. Also, the effect of the slightly *rough* cylinder surface due to the rectangular grid is of little significance except in

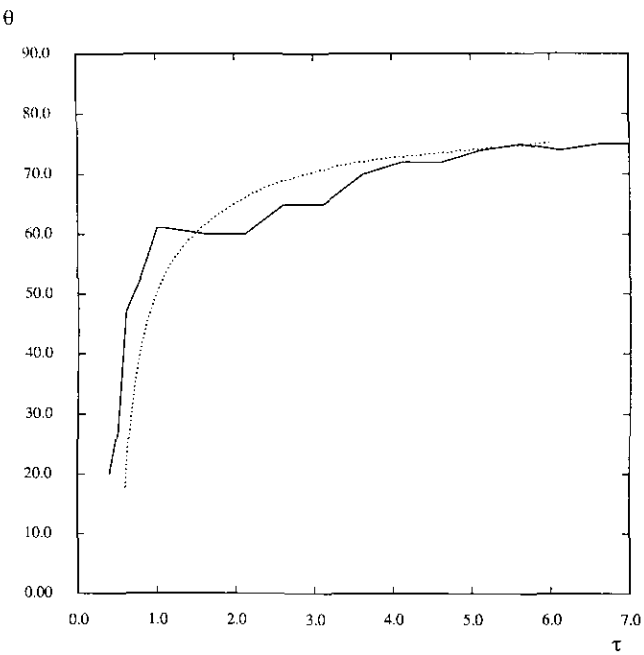

FIG. 10. Separation angle, measured from the 3 o'clock location vs. time: Solid—present calculations; dots—Pepin.

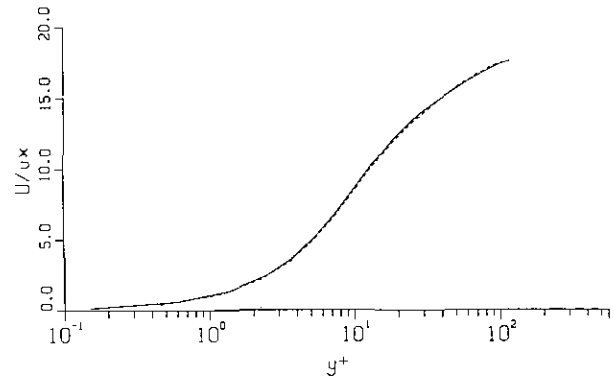

FIG. 12. Mean velocity profile in turbulent channel flow: Solid—Handler *et al.*; dashes—present results.

the lowest resolution simulation. This is probably because the scale of the roughness in the most highly resolved calculations was less than the boundary layer thickness near the separation region. In these calculations no internal flow was needed to smooth the solution although several flows resembling internal potential flow solutions were tried. The minor oscillations that are seen in the vorticity contours do not appear to affect the overall development of the flow. As mentioned above, slight internal flows (velocities $<\frac{1}{10}U$) appear naturally owing to the smoothing of the force which introduces perturbation forces inside the cylinder. The maximum velocities on the virtual cylinder itself, however, remain three to four orders of magnitude less than U.

4.3. *Turbulent Flow*

In a larger calculation we use a fully three-dimensional geometry to simulate turbulent channel flow. The resolution of the calculation is now $48 \times 65 \times 64$ grid points in x, y, and z, respectively, and the geometry is that of Fig. 1. The channel dimensions are 1250:250:625 viscous units (l^*) in x, y, and z, respectively, the same as those used by Handler *et al.* [15]. Here the wall shear velocity, u^*, is given by

$$u^* = \sqrt{\nu(d\bar{u}/dy)|_{u=\mp h}} \tag{7}$$

and the viscous scale length by

$$l^* = \nu/u^*. \tag{8}$$

Here, an overbar indicates a time average. The Reynolds number based on the wall shear velocity and the channel half height ($\mathrm{Re} = u^*h/\nu$) is 125, where $\nu = \mu/\rho$. During the run a virtual plate is inserted into an equilibrium turbulent flow that had been created between two ordinary no-slip surfaces. The flow is then allowed to re-equilibrate for 520 time steps before time averaged data is taken for about seven large eddy turnover times ($\approx h/(u_{\mathrm{rms}})_{\max}$). In Fig. 12 we see that the mean velocity profile with a modeled flat plate located one grid site off the shear free boundary (at $y^+ \equiv y/l^* = 0.151$) is virtually the same as that obtained by Handler *et al.* [15], both for the bulk of the flow and for the flow close to the wall. The root mean squared velocity and Reynolds stress profiles (Fig. 13, 14) also agree well; the small differences are likely to be due to the lack of sufficient time averaging for the present calculations. It is worth emphasizing that, even though there is some delay in the force response of the virtual wall owing to its integral nature ($\alpha = -200$, $\beta = -1.5$, $\Delta t = 0.03$, $\Delta t\, u^{*2}/\nu = 0.135$, and the reverse force is $-160G$), the fluctuating quantity profiles do not suffer significantly and are essentially the same as above a true no-slip surface. The natural frequency of the forcing (thus, the choice of α) was chosen to be above the frequency having the greatest energy in a frequency spectrum of the fluctuating velocities near the wall. On the virtual plane itself, the spectra show two to three orders of magnitude less energy at moderate to low frequencies than at a location a few viscous units above the plane. This indicates that the forcing effectively brings even the fluctuating velocities to rest. Even though the force field responds slower than the fastest time scales in the flow, there seems to be no net effect

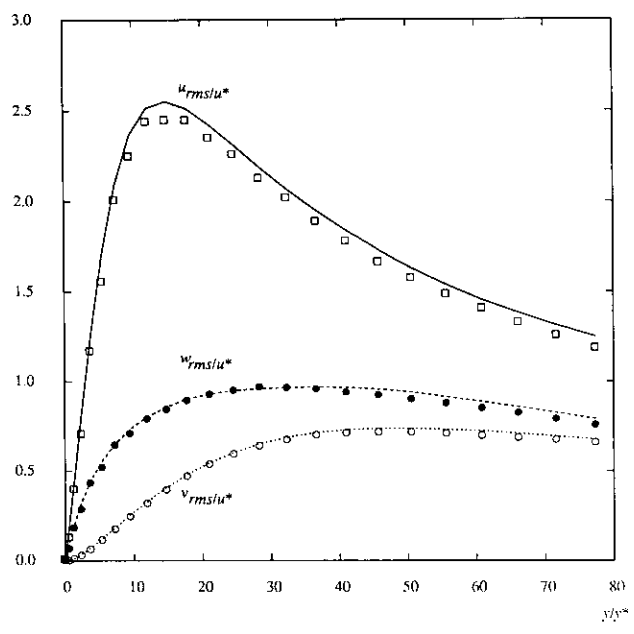

FIG. 13. Fluctuating velocity profiles: Lines—Handler *et al.* (Solid—u_{rms}/u^*, dashes—w_{rms}/u^*, dots—v_{rms}/u^*); symbols—present results.

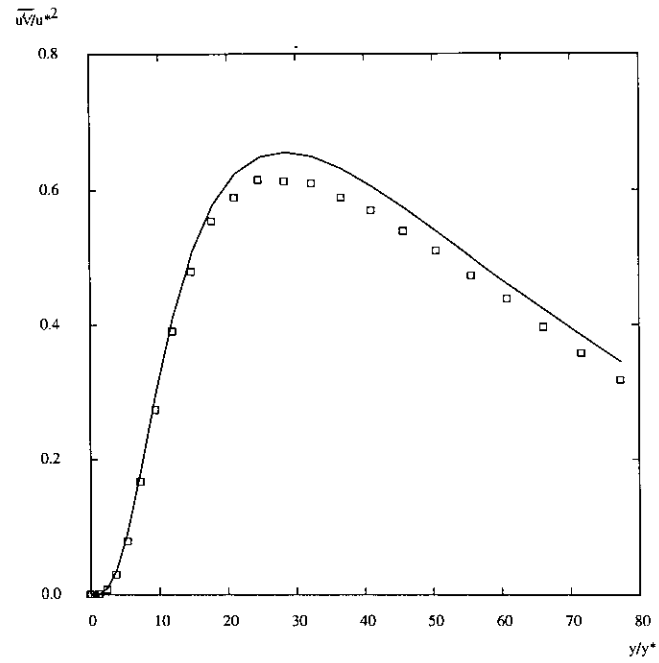

FIG. 14. Reynolds stress profiles: Lines—Handler *et al.*, symbols—present results.

on the mean profiles. This, we believe, is because there is negligible energy at the highest frequencies so the error is minimal. The gains can still be made larger with the same time step before the method becomes unstable. (With $\alpha = -500$, $\beta = -2.5$ the peak fluctuating velocity energy on the virtual plane drops by another order of magnitude.) We also note that we do not pay a penalty for maintaining the stability of the forcing scheme; the time step is only limited by the Courant condition. It is, of course, possible to reduce the time step considerably and raise the gains to have a force that responds faster than the fastest flow time scales but this is unnecessary.

In a third example of three-dimensional channel flow we simulate the effect of streamwise grooves along the virtual boundary. Such grooves, or riblets, have been determined experimentally to reduce the turbulent drag. Riblets having a triangular or cusped cross section, a peak to peak spacing of 10 to 20 viscous units, and a height of 5 to 15 viscous units are most effective (Walsh [36]) in reducing drag. The mechanism for this drag reduction still lacks a complete explanation and others as well [5–8] are applying direct numerical simulation to the problem. We model a channel of dimensions 1250:250:375 viscous units, somewhat more narrow than in the example above but still sufficiently large to include a few boundary layer low-speed streaks which are spaced by roughly 100 viscous units (Jimenez and Moin [16]). The spatial resolution is $48 \times 64 \times 128$. The ribs stretch in the x direction for the full length of the channel. The y locations, measured in grid sites, where the velocity is brought to rest are, at increasing z locations, $22456765422\cdots$. These points lie on the expanded $\frac{3}{2}$'s grid which has the usual cosine grid spacing in the y direction ($y_i = h - \cos[\pi(i-1)/N_y]$). The peak to peak spacing of the ribs is nine expanded grid sites or 18 viscous units and the ribs are six regular grid sites (five viscous units) high. In this case there are 21 ribs along the channel width. Below the ribs we apply a back force that is small ($4G$) below the rib peaks and large ($800G$) below the valleys. The back force profile was chosen by trial and error to minimize the energy in the highest u mode in the mean flow over each portion of the ribs. Once the trial and error procedure was complete and the oscillations were very small, we found that the moderate spectral smoothing described above was sufficient to eliminate the remaining spatial oscillations.

The results are seen in Fig. 15 and 16 which give views looking down the channel in the x direction. The lines are contours of constant streamwise velocity and are separated by $\Delta u = 0.05$ and 0.02 in Figs. 15 and 16, respectively. The arrows indicate the velocity in the $y-z$ plane. We observe what appears to be ordinary turbulent channel flow in the bulk of the domain. There are streamwise vortices which are seen to draw slow-moving fluid off of the walls and bring energetic fluid close to the walls. If we look closely at an interesting area in the riblet region (Fig. 16) we see that the ribs seem to affect only the region close to the wall. As expected, near the rib peaks the contours are closely spaced and the velocity gradient (hence drag) is high, while in the rib valleys the gradient is low. We particularly note that the ribs, essentially defined by the zero velocity contour line, are of a consistently uniform shape even though each is experiencing a different local flow. None are substantially distorted and each has a shape that corresponds well with the forced grid points. The somewhat rounded peak and the artificial smoothing applied may account for the relatively poor drag reduction; this particular configuration produced a drag reduction (drag on ribbed surface/drag on flat top surface during a single run) of $2.2 \pm 1.9\%$. The error bars indicate a 90% confidence interval, assuming a normal distribution for our 23 independent realizations. Experiments report drag reductions for riblets with these dimensions of as much as 5%. As the ribs are smoothed further they become less distinct from an ordinary flat surface and the drag reduction should disappear. Animations made of this

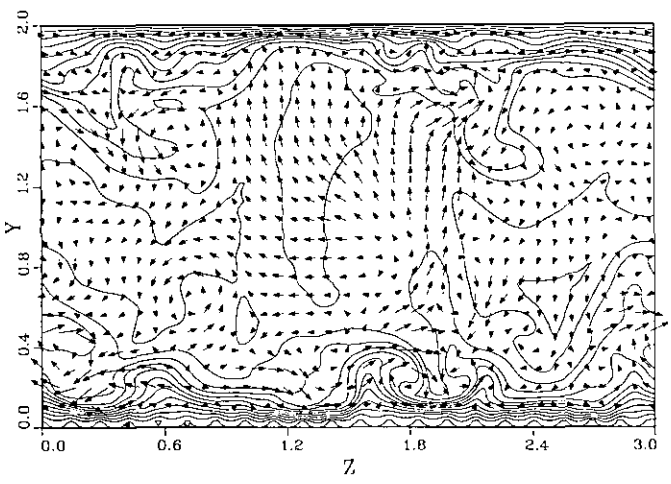

FIG. 15. View looking down the channel in the x direction showing contours of constant streamwise (u) velocity and selected v, w velocity vectors. Top boundary is smooth no-slip; bottom boundary has riblets.

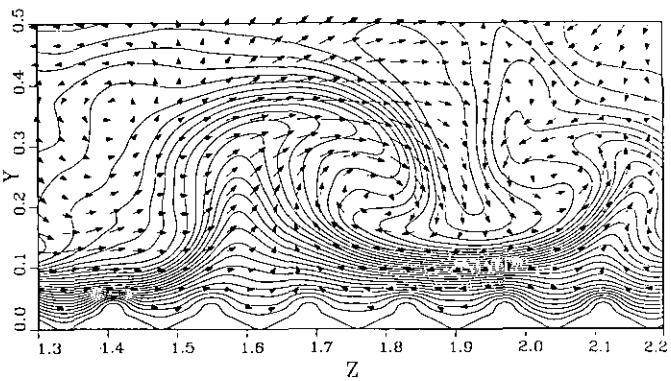

FIG. 16. Close-up view of a portion of Fig. 15.

simulation do not yet reveal any apparent mechanism for the drag reduction. We, as have others (Choi *et al.* [5]), find a few percent drag increase for similarly shaped riblets if the flow is laminar. This increase is related to the exposed surface area of the rib. A more detailed investigation of the riblet simulation is now in progress.

5. DISCUSSION

The feedback forcing method described here permits simulations with complex boundary geometries within a spectral approach, which framework is ordinarily thought to be unsuited for complex geometries. Neither domain decomposition nor mapping methods are required in this approach. Computer memory overhead for the solid surfaces was also minimal; vectors containing the cell indices and velocity time integral for the boundary points are all that is required. For the case of the riblet surface, the simulation ran 5% slower than the non-riblet channel flow on a Cray YMP. It is difficult to compare the efficiency of the cylinder flow calculations with those of Pepin since different computers were used (JPL Mark 4 parallel processor for Pepin and Cray YMP in the present study). However, our simulations took about 10 min on a single processor of the Cray, whereas Pepin's calculation required roughly 10 h on the Mark 4. We note that Pepin's method has since been made much more efficient.

The present method was shown by example to model adequately both simple flows and several that were not so simple. The results appear promising. The simulation of riblets in particular appears unique in that a code already validated for turbulent flow in an ordinary rectangular channel was modified with relative ease to simulate flow over a complex structure on one of the boundaries. It should be emphasized that the method of virtual boundaries does not yield an exact simulation of solid boundaries but rather provides a quantitatively accurate model calculation. The method has several drawbacks as well. The boundary location is not exactly defined both because the force may not respond sufficiently quickly to a changing flow and because, even if it does respond, the velocity is only zero at the point where it is measured and may not be precisely zero between boundary points. It also may be difficult to exactly measure the surface forces; the force field itself will not necessarily provide a good measure because the necessary smoothing techniques can distort the results. Determining the pressure and viscous forces from the velocity field right at the surface can also be difficult since the surface location is not sufficiently well defined. Specification of a control volume away from the surface may be advantageous for this purpose. Finally, unless a surface lies along a smooth mesh line, a curved boundary may appear rough on the smallest scales unless smoothing sufficiently blurs any step-like structures. Of course, if the laminar boundary layer over the surface is substantially thicker than the roughness, the roughness effect should be small.

Certain refinements of the technique may yet meet some of these objections. Nonetheless, it appears that the technique may be useful for obtaining a reasonably accurate representation of a surface in a flow in which the near surface activity is not the primary interest. As with any other computational technique where surface details *are* of interest, it is necessary to have a non-uniform mesh that concentrates grid points near the surface. For example, in the riblet calculations of Choi *et al.* [5, 6], Chu *et al.* [7], and Chu and Karniadakis [8] mesh points could be concentrated right along the rib surface. In the present calculations the Chebyshev cosine grid did enable us to concentrate grid points near the riblets, although not in an optimal way. The fact, however, that the spanwise grid spacing was everywhere constant (128 modes in the riblet computation) was costly in that the increased z resolution was not needed away from the ribbed surface. This feature (resolution in z at mid-channel equal to that near the ribs) was present in the calculations of Chu *et al.* [7], Chu and Karniadakis [8], and Choi *et al.* [5, 6] as well.

The force boundary conditions described above are also adequate for compressible and/or non-isothermal flows. As it stands, the force field creates an adiabatic wall; it simply pushes the flow without changing its temperature. The procedure must be modified to provide a non-adiabatic wall. In this case one can change the internal energy by using local integral heating to maintain the temperature at a desired value. It is also possible to create a shear free boundary. The regular force field can be used to maintain no through flow and a second force field may be applied at interior points to force no shear on the surface. Similar ideas can be applied to other surface quantities as well. Problems involving rapid temporal change will naturally place strong restrictions on the time resolution of the feedback mechanism.

ACKNOWLEDGMENTS

The work presented here was in part supported by ONR under award N00014-91-J-1588 and a gift from ORMAT Industries. The computations were performed at the Pittsburgh Supercomputing Center. One of us (D.G.) is grateful to D. Gottlieb for helpful conversations.

REFERENCES

1. M. Bar-Lev and H. T. Yang, *J. Fluid Mech.* **72**, 625 (1975).
2. Deleted in proof.
3. G. A. Bird, *Molecular Gas Dynamics* (Clarendon Press, Oxford, 1976).
4. R. Bouard and M. Coutanceau, *J. Fluid Mech.* **101**, 583 (1980).
5. H. Choi, P. Moin, and J. Kim, *Phys. Fluids* **3**, No. 8 (1991).
6. H. Choi, P. Moin, and J. Kim, in *Proceedings, Turbulence Structures and Control Conference, April 1–3, 1991, Columbus, OH.*

7. D. Chu, R. Henderson, and G. Karniadakis, *Theor. Comput. Fluid Dyn.*, **3** (1992).
8. D. Chu, and G. Karniadakis, *J. Fluid Mech.*, in press.
9. R. C. Dorf, *Modern Control Systems* (Addison–Wesley, Reading, MA, 1983).
10. L. J. Fauci and C. S. Peskin, *J. Comput. Phys.* **77**, No. 1 (1988).
11. L. J. Fauci, Armstrong Lab Report AL-TR-1991-0045, May 1991 (unpublished).
12. A. L. Fogelson and C. S. Peskin, *J. Comput. Phys.* **79** (1988).
13. D. Gottlieb, L. Lustman, and C. Streett, in *Spectral Methods for Partial Differential Equations*, edited by R. G. Voigt, D. Gottlieb and M. Y. Hussaini (SIAM, Philadelphia, 1984).
14. D. Gottlieb, M. Y. Hussaini, and S. Orszag, in *Spectral Methods for Partial Differential Equations*, edited by R. G. Voigt, D. Gottlieb, and M. Y. Hussaini (SIAM, Philadelphia, 1984).
15. R. A. Handler, E. W. Hendricks, and R. I. Leighton, NRL Memorandum Report 6410, 1989 (unpublished).
16. J. Jimenez and P. Moin, *J. Fluid Mech.* **225**, 213 (1991).
17. J. Kim, P. Moin, and R. Moser, *J. Fluid Mech.* **177**, 133 (1987).
18. J. Koplik, J. R. Banavar, and J. F. Willemsen, *Phys. Fluids A* **1**, No. 5 (1989).
19. A. Leonard, *J. Comput. Phys.* **37**, 523 (1980).
20. D. M. McQueen and C. S. Peskin, *J. Comput. Phys.* **82** (1989).
21. F. Pepin, Ph.D. thesis, Calif. Inst. of Technology, 1990 (unpublished).
22. C. S. Peskin, *J. Comput. Phys.* **10** (1972).
23. C. S. Peskin, *J. Comput. Phys.* **25** (1977).
24. C. S. Peskin and D. M. McQueen, *J. Comput. Phys.* **81** (1989).
25. C. S. Peskin and D. M. McQueen, *J. Comput. Phys.* **37** (1980).
26. C. S. Peskin, "Two Examples of Mathematics and Computing in the Biological Sciences: Blood Flow in the Heart and Molecular Dynamics, in *100 Years of American Mathematics*, (Am. Math. Soc., Providence, RI, in press).
27. P. J. Roache, *Computational Fluid Dynamics* (Hermosa, Albuquerque, NM, 1972).
28. E. P. Salathe and L. Sirovich, *Phys. Fluids* **10**, No. 7 (1967).
29. L. Sirovich, *Phys. Fluids* **10**, No. 1 (1967).
30. L. Sirovich, *Phys. Fluids* **11**, No. 7 (1968).
31. D. Sulsky and J. U. Brackbill, *J. Comput. Phys.* **96**, 339 (1991).
32. C. Tu and C. S. Peskin, *SISSC*, in press.
33. S. O. Unverdi and G. Tryggvason, *J. Comput. Phys.* **100** (1992).
34. J. A. Viecelli, *J. Comput. Phys.* **4** (1969).
35. J. A. Viecelli, *J. Comput. Phys.* **8** (1971).
36. M. J. Walsh, "Progress in Astronautics and Aeronautics," in *Viscous Drag Reduction in Boundary Layers*, edited by D. Bushnell and J. Hefner, (AIAA, Washington, DC, 1990), Vol. 123.