JOURNAL OF COMPUTATIONAL PHYSICS **134,** 75–95 (1997)
ARTICLE NO. CP975667

Three-Dimensional Vortex Simulation of Time Dependent Incompressible Internal Viscous Flows

Adrin Gharakhani and Ahmed F. Ghoniem

Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139

Received March 24, 1995; revised December 6, 1996

A hybrid random vortex-boundary element method is developed for the solution of time-dependent incompressible three-dimensional internal flow problems. The numerical scheme is grid-free within the flow domain and is based on a combination of the Lagrangian vortex method to capture the convection and stretch of the vortical field, the random walk method to describe the diffusion process, and the boundary element method to superimpose a potential flow on the vortical field such that the normal flux boundary condition is satisfied. The no-slip boundary condition is satisfied by generating vorticity tiles on solid boundaries, which are subsequently diffused and convected into the flow interior. Additionally, a boundary condition is devised for the application of fully developed flow properties at the exit plane. In this paper, the formulation and the numerical scheme are presented, followed by a parametric study of the accuracy of the method using the model problem of the flow in a duct with square cross section at $R_e = 100$. We show that the method converges to the analytical solution of the problem as the resolution of the time integration and the discretization are improved, and we discuss the impact of each resolution parameter on the accuracy. In addition, selected results from the simulation of an impulsively started flow over a cube at $R_e = 100$ are presented. We use the results of this test case to demonstrate that the method captures the effect of sharp edges, parallel and normal to the streamwise flow direction, on the flow dynamics. © 1997 Academic Press

1. INTRODUCTION

Discrete vortex methods are approximation techniques for the simulation of the unsteady, incompressible, high Reynolds number Navier–Stokes equations in unbounded or wall-bounded domains. In this approach, the governing equations are expressed in the vorticity transport form. The vorticity field is represented by a collection of vortex elements and its evolution is evaluated by tracing the trajectory of the elements, as well as their vorticity vectors, in the Lagrangian frame of reference.

Vortex methods offer some significant advantages. To begin with, they are grid-free and thus eliminate the often tedious task of volumetric meshing of complex three-dimensional domains. This advantage is even more pronounced in situations where moving boundaries are encountered, such as in the case of an internal combustion engine with moving valves and a piston, where the meshing process must be repeated each timestep. In addition, due to the Lagrangian nature of the method, convection is approximated with minimal numerical diffusion, making the scheme an excellent tool for analyzing high Reynolds number flows. Vortex methods are also self-adaptive and, hence, are capable of dynamically concentrating computational elements where significant velocity gradients evolve, such as in regions with high recirculation or in shear layers. Furthermore, vortex methods readily facilitate an intuitive and quantitative tool for interrogating "three-dimensionality effects" in the flow by monitoring vorticity stretch in the field. This is an immediate consequence of representing the Navier–Stokes equations in the velocity–vorticity formulation, where the three-dimensional equations are distinguished from their two-dimensional counterpart by the extra vorticity stretch term. Finally, vortex methods are naturally adaptable to massively parallel computing, which can be exploited to solve large-scale problems efficiently.

Currently, three-dimensional vortex methods display a few computational difficulties which need to be resolved with more vigor. First, the computational cost for evaluating the vortex element velocities using traditional summation techniques grows quadratically with the number of elements. This imposes a severe limit on the maximum number of elements that can be used in a simulation. Fortunately, fast multipole expansion techniques are now available which achieve near-linear CPU dependence on the number of elements [10, 32]. Nevertheless, physically correct and numerically accurate vortex element merging algorithms may still be necessary to limit the proliferation of the number of elements. Second, more general vorticity boundary conditions need to be devised to allow for geometries with curved, time-varying surfaces. Finally, the use of vortex monopoles to discretize the three-dimensional vorticity field in standard vortex element methods does not implicitly guarantee the solenoidality of the discretized vorticity in space and time. One attempt to correct this problem is the periodic redistribution or regridding of the field vorticity over the vortex elements, such that some

0021-9991/97 $25.00

measure of the divergence of vorticity is imposed [60]. Although this method appears to stabilize the solution, especially of the Euler equations, it does not represent a fundamental remedy to the problem. To this end, a new derivation has recently been proposed which is based on the canonical Hamiltonian formulation for the incompressible flow [11, 49]. The Hamiltonian structure of the flow is described in terms of a variable that is analogous to a vortex dipole [17] and preserves all invariants of the three-dimensional Euler equations, as well as maintains the solenoidality of the velocity and vorticity [11, 49]. However, the singularity in the velocity kernel of a vortex dipole is one order higher than that of a vortex monopole, which imposes a more severe condition on the stability of the method. Furthermore, the extension of the method to wall-bounded flows is nontrivial.

In an unbounded domain, the velocity at a point is expressed in terms of the positions and strengths of all vortices according to the familiar Biot-Savart law [3]. However, the velocity kernel is singular and must be regularized to avoid numerical blowup as the vortex elements approach each other. Chorin introduced the concept of a "vortex blob" by removing the velocity singularity at the center. This was later interpreted as the convolution of the singular vortices with a smoothing function, or the use of finite area elements whose vorticity remains concentrated within a finite size core radius and decays rapidly outside it [13]. The convergence and the accuracy properties of the two- and three-dimensional vortex blob method have been obtained for unbounded domains [4, 6–8, 31, 33, 44]. In addition, Beale and Majda proposed a methodology for constructing core functions with arbitrarily high spatial order [9]. A rigorous numerical verification was also provided for conditions under which the proposed core functions produce accurate solutions. Recently, Hou [35] and Hou *et al.* [36] presented stable solutions with point vortex methods by introducing a novel desingularization technique for the velocity kernel.

In the context of Lagrangian vortex methods, the diffusion process may be approximated stochastically or using a deterministic approach. In the stochastic approach, the similarity between the Green function for the diffusion equation and the Gaussian probability distribution function is exploited to approximate viscous diffusion by the random walk method, in which a Wiener process is added to the motion of each vortex element [13, 15]. This method is widely used due to its simplicity and ease of implementation in flows with solid boundaries. However, as verified by various numerical experiments, the obtained solutions may exhibit relatively high levels of noise and necessitate the use of a large number of elements and a small timestep [30, 41, 46, 53]. Recently, a new smoothing technique with superior accuracy and convergence properties has been proposed for the random walk method; however, further tests are necessary to verify its success [23]. Deterministic techniques that maintain the grid-free nature of the vortex method have been proposed as well. Degond and Mas-Gallic [19] introduced a vorticity exchange mechanism between the elements by replacing the Laplacian in the diffusion process with a smoothed integral and discretizing the latter over the elements. The accuracy of the method has been verified numerically and its stability proven for positive kernels. One disadvantage of the method is that higher order kernels may not be positive everywhere in the domain and the solution may become unstable [21]. Fishelov proposed an alternative approximation for the Laplacian of the diffusion equations by directly differentiating the vorticity cutoff function and by convolving the latter with the vorticity distribution [21]. A mathematical proof for the consistency and stability of this method was also given, but further numerical verification is needed. A different school of thought treats the diffusion of vorticity as the convective transport of the vorticity gradients, whereby the vortex elements travel grid-free by a "diffusion velocity," in addition to the regular convection velocity [20, 37, 48]. One disadvantage of this approach is the need to evaluate the vorticity gradients accurately.

The random vortex method was first applied to two-dimensional wall-bounded flows by Chorin [12]. In this case, the vortical component of the velocity in the domain is modified by a potential flow such that the proper normal flux is admitted at the boundary. To solve the potential flow problem, the method of images and Schwarz Christoffel transformations have been implemented for simple geometries [27, 28, 47], whereas finite element methods have been applied for complex geometries [45]. The no-slip boundary condition is imposed by continuously generating vortex sheets at the wall with sufficient strength to zero out the tangenital velocities [15, 28, 42, 45, 50]. Within a thin layer near the boundary, where normal gradients are stronger than those in the wall direction, the sheets are convected and diffused according to the Prandtl approximation of the Navier–Stokes equations. Beyond the layer the sheets are converted into vortex elements. A rigorous mathematical proof for the consistency of the vorticity generation scheme is not available yet; however, the consistency of the approximation of the heat equation with boundaries using a similar approach has been established by Hald [34]. Furthermore, numerous computational experiments verify the robustness of the method in solving the Prandtl equations [12, 14, 50]. Additionally, a comprehensive set of comparisons with experimental data have been compiled for flow in planar geometries where available [28, 29, 47, 56]. More recently, an extended vortex-finite element method was used to successfully emulate a flow visualization experiment studying the piston-induced intake processes in an axi-symmetric port-cylinder geometry with a valve [26].

Three-dimensional inviscid flows were first simulated using a vortex filament approach [43], which is an excellent candidate for solving the Euler equations. In this model, a collection of filaments, each forming a closed vortex line and segmented along the vorticity vector, is used to approximate the initial vorticity field. By construction, Kelvin's theorem for the conservation of circulation is implicitly satisfied in this scheme. Beale and Majda proposed extending the notion of a two-dimensional vortex element to a three-dimensional element with spherical core function and provided mathematical proof for the existence of solutions for short times [6]. However, the method was handicapped by the fact that velocity gradients, used to compute vorticity stretch, were obtained by a finite difference approximation—thus contradicting the spirit of grid-free Lagrangian techniques. Anderson and Greengard alleviated this problem by differentiating the regularized kernel for the velocity in the Biot–Savart integral directly to obtain the regularized kernel for the velocity gradients [1]. The convergence of the latter was subsequently proved by Beale [4]. This approach and its variants have been used quite extensively in three-dimensional unbounded domains [38–40, 59, 60], and simulation results have been compared with exact solutions or experiments wherever possible. (See [60] for a review of the current state of the three-dimensional vortex methods in unbounded domains.)

The two-dimensional sheet generation algorithm was first extended to three dimensions by Chorin, where he combined the latter with a vortex stick method to simulate three-dimensional high Reynolds number flow over a flat plate [14]. Later, Fishelov combined Chorin's tile generation scheme [14] with the vortex method of Anderson and Greengard [1] to study the flow over a flat plate and transition to turbulence [22]. Both Chorin and Fishelov applied the random walk method to approximate diffusion, and the method of images to impose the zero flux condition through the wall.

In this paper, expanding upon the ideas introduced above, we present a hybrid random vortex-boundary element method for the grid-free simulation of the time-dependent incompressible Navier–Stokes equations in three-dimensional internal flow configurations. In this approach, the field vorticity is discretized using a collection of spherical vortex elements with second-order core function. Convection is evaluated by tracking the trajectory of the elements in the Lagrangian frame of reference. Diffusion is obtained by the random walk method. Additionally, the elemental vorticity vectors are adjusted to account for the stretch of vorticity, which is caused by the interaction between the local vorticity vector and the velocity gradients. The velocity and its gradients are expressed as a superposition of a vortical component evaluated by the Biot–Savart law, using the method of Anderson and Greengard [1]; and a potential component obtained from the solution of a three-dimensional Neumann problem over the domain, such that the correct normal flux boundary condition is imposed on the domain boundary.

The Neumann problem is solved by the boundary element method, assuming piecewise-quadratic variation of the potential and its normal flux across each boundary element. The no-slip boundary condition is satisfied by generating vortex tiles at the boundaries [42]. Within a thin prespecified region near the boundary, the tiles convect and diffuse in the Lagrangian frame of reference according to a boundary-layer-like approximation [14, 22]. Beyond this region, the tiles are converted to spherical vortex elements such that the volumetric vorticity is conserved. (Note: throughout this paper, by the "Prandtl equations" we imply the equations governing the flow very close to solid walls using a boundary-layer-like approximation of the vorticity transport formulation and simplified by the assumption that vorticity stretch in this layer is negligible.) At present the solid walls are discretized by planar rectangular vortex tiles and boundary elements; the extension to the more general quadrilateral elements will be presented in future applications. To improve the accuracy and flexibility of the proposed method, we have developed a novel technique to remove a spike in the boundary potential that arises from the solution of the ill-posed internal Neumann problem [25]. We have also developed a regularized formulation for the accurate evaluation of the potential velocity gradients at the vortex elements [25]. In addition, we have formulated an extended Prandtl solution algorithm that allows the interaction among unequally sized tiles. This is imperative for tile solutions that involve expanding or shrinking boundaries, such as the cylinder of an engine, where the size of the tile generated at one timestep is invariably different from that at the next timestep. We have also developed a new formulation for imposing the fully developed flow condition at the exit plane.

We conduct a parametric study to examine the accuracy of the vortex-boundary element method, using the example of flow in a duct with square cross section at $R_e = 100$, and we compare the results with the exact solution at the exit plane. We also provide selected results from an example of impulsively started flow over a cube at $R_e = 100$ and compare them with available data.

2. SCHEMATIC DESCRIPTION

The random vortex-boundary element method may best be introduced using a conceptual description of the flow dynamics near solid boundaries. The computational field is decomposed into an "interior domain," where the Navier–Stokes equations are applied, and a thin user-specified region in the vicinity of the boundary surface called the "numerical boundary layer," where Prandtl-like approximations are utilized.

Initially, there is no vorticity in the field and a potential flow is imposed on the geometry such that the normal flux through the solid boundaries is zero. This in turn induces a slip velocity at the walls, which is annulled by creating vortex tiles on the walls with surface vorticity equal to the slip velocity. The diffusion of these tiles towards the domain interior marks the end of the first timestep. If a tile traverses the numerical boundary layer, it is converted into a spherical vortex element that has a core radius and volumetric vorticity vector equal to the length and volumetric vorticity vector of the tile, respectively. In subsequent timesteps, the vortex elements in the interior are convected, stretched, and diffused, according to the Navier–Stokes equations—subject to the proper normal flux boundary condition. At the same time, the tiles are convected in the numerical boundary layer according to the Prandtl equations. As a result of the motion of the vortex elements and tiles, a net slip velocity is induced on the solid boundaries, which is canceled by generating vortex tiles there. Finally, the conclusion of each cycle is marked by the diffusion of all vortex tiles—old and new—into the field.

During the course of the computation, the tiles may jump out of the domain due to the random walk, in which case they are simply reflected back into the field. On the other hand, should the vortex elements jump out of the domain or into the numerical boundary layer, they are eliminated altogether. Stationary state is achieved when the total number of vortex elements is statistically invariant, i.e., when the total number of vortex tiles converted into vortex elements is balanced by the total number of vortex elements leaving the computational domain.

It must be emphasized here that the ‘‘numerical boundary layer’’ is not the numerical solution to the physical boundary layer problem; it is a very thin layer within which vorticity is better approximated by vortex tiles than by elements.

3. NUMERICAL FORMULATION

The equations of motion of an incompressible fluid within a three-dimensional wall-bounded domain D and with boundary surfaces ∂D are expressed in the vorticity transport form of the Navier–Stokes equations as

$$\frac{\partial \boldsymbol{\omega}}{\partial t} + \boldsymbol{u}\cdot\nabla\boldsymbol{\omega} = \boldsymbol{\omega}\cdot\nabla\boldsymbol{u} + \frac{1}{\mathrm{R_e}}\nabla^2\boldsymbol{\omega}, \quad \boldsymbol{x}\in D, \tag{3.1}$$

$$\nabla\cdot\boldsymbol{u} = 0, \quad \boldsymbol{x}\in D, \tag{3.2}$$

$$\boldsymbol{\omega}(\boldsymbol{x}, t) = \nabla^\wedge\boldsymbol{u}, \quad \boldsymbol{x}\in D, \tag{3.3}$$

$$\boldsymbol{u}(\boldsymbol{x}, t) = (\boldsymbol{u}\cdot\boldsymbol{\tau}, \boldsymbol{u}\cdot\boldsymbol{\rho}, \boldsymbol{u}\cdot\boldsymbol{n}), \quad \text{prescribed}, \boldsymbol{x}\in\partial D, \tag{3.4}$$

$$\boldsymbol{u}(\boldsymbol{x}, t=0) = \boldsymbol{u}_o(\boldsymbol{x}), \quad \text{prescribed}, \boldsymbol{x}\in D, \tag{3.5}$$

where $\boldsymbol{x} = (x, y, z)$ is the position vector in Cartesian coordinates normalized by a reference length, L; $\boldsymbol{u}(\boldsymbol{x}, t) = (u, v, w)$ is the velocity vector normalized by a characteristic speed, U; t is the time normalized by L/U; $\boldsymbol{\omega}(\boldsymbol{x}, t) = (\omega_x, \omega_y, \omega_z)$ is the vorticity vector normalized by U/L; $\mathrm{R_e} = UL/\nu$ is the Reynolds number, and ν is the kinematic viscosity. At the boundary surfaces, the velocity is expressed in terms of the local orthogonal coordinate system $\boldsymbol{\tau}$-$\boldsymbol{\rho}$-$\boldsymbol{n}$, where $\boldsymbol{n} = (n_x, n_y, n_z)$ is the unit outward normal, and $\boldsymbol{\tau} = (\tau_x, \tau_y, \tau_z)$ and $\boldsymbol{\rho} = (\rho_x, \rho_y, \rho_z)$ are the unit tangents to the boundary. Note that, as in most time dependent problems, the velocity in Eq. (3.4) is not well determined on the exit boundary. This issue will be addressed in more detail later.

In what follows, the solution of the equations above will be obtained by decomposing the computational field into an interior domain and a boundary domain. The interior domain is discussed in Section 3.1 and involves the solution of Eqs. (3.1)–(3.3) in conjunction with the normal flux boundary condition. Since the tangential velocity boundary condition is not strictly enforced for the interior domain, the walls experience slip velocities. The no-slip boundary condition is applied in the context of the boundary domain, which is discussed in Section 3.2. In this section, the process of vorticity generation at the wall and its evolution within the thin numerical boundary layer are explained.

3.1. Interior Domain

Equations (3.1) are decomposed using a viscous splitting scheme whereby, within each timestep, the evolution of vorticity is first obtained from the Euler equations and is then modified according to the diffusion equation [5]. Consequently, the Navier–Stokes equations for the interior domain are approximated using

$$\frac{\partial \boldsymbol{\omega}}{\partial t} + \boldsymbol{u}\cdot\nabla\boldsymbol{\omega} = \boldsymbol{\omega}\cdot\nabla\boldsymbol{u}, \quad \boldsymbol{x}\in D, \tag{3.1.1}$$

$$\frac{\partial \boldsymbol{\omega}}{\partial t} = \frac{1}{\mathrm{R_e}}\nabla^2\boldsymbol{\omega}, \quad \boldsymbol{x}\in D, \tag{3.1.2}$$

$$\nabla\cdot\boldsymbol{u} = 0, \quad \boldsymbol{x}\in D, \tag{3.1.3}$$

$$\boldsymbol{\omega}(\boldsymbol{x}, t) = \nabla^\wedge\boldsymbol{u}, \quad \boldsymbol{x}\in D, \tag{3.1.4}$$

$$\boldsymbol{u}(\boldsymbol{x}, t) = \boldsymbol{u}\cdot\boldsymbol{n}, \quad \text{prescribed}, \boldsymbol{x}\in\partial D, \tag{3.1.5}$$

$$\boldsymbol{u}(\boldsymbol{x}, t=0) = \boldsymbol{u}_o(\boldsymbol{x}), \quad \text{prescribed}, \boldsymbol{x}\in D. \tag{3.1.6}$$

For unbounded domains, solutions obtained by Eqs. (3.1.1)–(3.1.4) have been proved to converge, in space and time, to those of the Navier–Stokes and convergence is shown to improve at a rate that is proportional to the Reynolds number [5]. (The tangential boundary condition in Eq. (3.4) is not enforced explicitly in the solution of the interior domain. However, it is enforced explicitly in the boundary domain and the vorticity exchange across the boundary between the interior and the wall regions is used to link the two solutions.)

In the equations above, the velocity (and its gradients) are decomposed into a vortical field, $\boldsymbol{u}_\omega(\nabla \boldsymbol{u}_\omega)$, and a potential component, $\boldsymbol{u}_P(\nabla \boldsymbol{u}_P)$, according to the Helmholtz decomposition rule:

$$\boldsymbol{u} = \boldsymbol{u}_\omega + \boldsymbol{u}_P. \tag{3.1.7}$$

Each velocity component can then be constructed based purely on vector identities and kinematic relationships, such that the continuity equation (3.1.3) and the normal flux boundary condition (3.1.5) are satisfied implicitly. The velocity gradients are subsequently evaluated by differentiating the expressions for the velocities.

The grid-free solution of the resulting Euler equations is discussed in Section 3.1.1, and the solution of the remaining diffusion equation is explained in Section 3.1.2. The formulations for obtaining the vortical and the potential velocities and their gradients are described in Sections 3.1.3 and 3.1.4, respectively.

3.1.1. *Euler Equation*

The Euler equations, as defined by (3.1.1) and (3.1.3)–(3.1.6), in conjunction with (3.1.7), can be expressed in their equivalent Lagrangian formulation as

$$\frac{d\boldsymbol{\chi}}{dt}(\boldsymbol{x}, t) = \boldsymbol{u}(\boldsymbol{\chi}, t), \qquad \boldsymbol{\chi}(\boldsymbol{x}, t_o) = \boldsymbol{\chi}_o, \tag{3.1.8}$$

$$\frac{d\boldsymbol{\omega}}{dt}(\boldsymbol{\chi}, t) = \boldsymbol{\omega}(\boldsymbol{\chi}, t)\cdot\nabla\boldsymbol{u}(\boldsymbol{\chi}, t), \quad \boldsymbol{\omega}(\boldsymbol{\chi}, t_o) = \boldsymbol{\omega}_o, \tag{3.1.9}$$

where $\boldsymbol{\chi}$ prescribes the trajectory of a fluid element initially at $\boldsymbol{\chi}_o$ and with vorticity $\boldsymbol{\omega}_o$. Equation (3.1.8) prescribes the transport of vorticity associated with a fluid element along its trajectory, while Eq. (3.1.9) governs the vorticity stretch.

We begin the solution of (3.1.8) and (3.1.9) by discretizing the continuous vorticity field into a collection of N_V vortex elements, each centered at $\boldsymbol{x}_j$ with element volume ΔV_j and vorticity vector $\boldsymbol{\omega}_j$,

$$\boldsymbol{\omega}(\boldsymbol{x}, t) = \sum_{j=1}^{N_V} \tilde{\boldsymbol{\Gamma}}_j(t)\delta(\boldsymbol{x} - \boldsymbol{x}_j), \tag{3.1.10}$$

where $\tilde{\boldsymbol{\Gamma}}_j(t) = \boldsymbol{\omega}_j(t)\,\Delta V_j$ is the volumetric vorticity and $\delta(\cdot)$ is the Dirac delta function. Combining the discretized vorticity distribution (3.1.10) with (3.1.8) and (3.1.9) yields the grid-free solution of the Euler equations as a pair of vectorial ordinary differential equations for the N_V elements:

$$\frac{d\boldsymbol{\chi}_i}{dt}(\boldsymbol{x}, t) = \boldsymbol{u}_i(\boldsymbol{\chi}_i, t), \qquad \boldsymbol{\chi}_i(\boldsymbol{x}, t_o) = \boldsymbol{\chi}_{i,o}, \tag{3.1.11}$$

$$\frac{d\tilde{\boldsymbol{\Gamma}}_i}{dt}(\boldsymbol{\chi}_i, t) = \tilde{\boldsymbol{\Gamma}}_i(\boldsymbol{\chi}_i, t)\cdot\nabla\boldsymbol{u}_i(\boldsymbol{\chi}_i, t), \quad \tilde{\boldsymbol{\Gamma}}_i(\boldsymbol{\chi}_i, t_o) = \tilde{\boldsymbol{\Gamma}}_{i,o}, \quad i = 1, \ldots, N_V, \tag{3.1.12}$$

where $\boldsymbol{\chi}_i$ describes the trajectory of the ith vortex element and t_o is the time at which it is introduced into the domain. Note that t_o is not necessarily zero since vortex elements may be added to the field at any time during the flow evolution as a result of vorticity generation. Equations (3.1.11) and (3.1.12) are integrated in time to any order of accuracy to obtain the flow evolution

$$\boldsymbol{\chi}_i^*(\boldsymbol{x}, t_{k+1}) = \boldsymbol{\chi}_i(\boldsymbol{x}, t_k) + F[\boldsymbol{u}_i(\boldsymbol{\chi}_i, t_k)]\,\Delta t, \tag{3.1.13}$$

$$\begin{aligned}\tilde{\boldsymbol{\Gamma}}_i(\boldsymbol{\chi}_i, t_{k+1}) &= \tilde{\boldsymbol{\Gamma}}_i(\boldsymbol{\chi}_i, t_k)\\ &\quad + F[\tilde{\boldsymbol{\Gamma}}_i(\boldsymbol{\chi}_i, t_k)\cdot\nabla\boldsymbol{u}_i(\boldsymbol{\chi}_i, t_k)]\,\Delta t,\\ &\quad k = 0, 1, \ldots; i = 1, \ldots, N_V,\end{aligned} \tag{3.1.14}$$

where $\Delta t = (t_{k+1} - t_k)$ is the integration timestep and $F[\cdot]$ represents the time integration scheme. Note that the asterisk in Eq. (3.1.13) implies the intermediate step in the viscous splitting algorithm and is not related to the $F[\cdot]$ integration scheme. We experimented with the second-order modified Euler and the fourth-order Runge–Kutta methods and selected the former for its economy and acceptable accuracy.

3.1.2. *Diffusion Equation*

The Green function for the diffusion equation (3.1.2) is given by $G_t(\boldsymbol{x}, t) = (4\pi t/\mathrm{R_e})^{-3/2} \exp(-|\boldsymbol{x}|^2/(4t/\mathrm{R_e}))$, which is also the three-dimensional Gaussian probability distribution function with zero mean and variance equal to $2t/\mathrm{R_e}$. Accordingly, the diffusion process within each timestep, Δt, can be approximated stochastically if each of the vortex elements is repositioned by a random displacement as

$$\begin{aligned}\boldsymbol{\chi}_i(\boldsymbol{x}, t_{k+1}) &= \boldsymbol{\chi}_i^*(\boldsymbol{x}, t_{k+1}) + \boldsymbol{\eta}_i(\Delta t),\\ &\quad k = 0, 1, \ldots; i = 1, \ldots, N_V,\end{aligned} \tag{3.1.15}$$

where $\boldsymbol{\eta}_i = (\eta_x, \eta_y, \eta_z)_i$ are random variables in each of the three coordinate directions, selected independently from a Gaussian distribution with zero mean and variance equal to $2\Delta t/\mathrm{R_e}$. The rate of convergence of the random walk method is proportional to $1/\sqrt{N_V \mathrm{R_e}}$ [46, 53] and the solution, in terms of the vorticity, can be rather noisy. However, since the random walk approximates the diffusion of vorticity, its integral—the velocity distribution—is expected to be smooth.

3.1.3. *Vortical Component*

The vortical velocity is expressed in terms of the curl of a solenoidal vector stream function as

$$\boldsymbol{u}_\omega = \nabla^\wedge \boldsymbol{\Psi}. \tag{3.1.16}$$

Taking the curl of (3.1.16), substituting in (3.3), and applying the assumption that $\boldsymbol{\Psi}$ is divergence-free yields the Poisson equation

$$\nabla^2 \boldsymbol{\Psi} = -\boldsymbol{\omega}. \tag{3.1.17}$$

The solution of Eq. (3.1.17) in free space is given by

$$\boldsymbol{\Psi}(\boldsymbol{x}, t) = \int G(\boldsymbol{x} - \boldsymbol{x}')\boldsymbol{\omega}(\boldsymbol{x}', t)\, dV(\boldsymbol{x}'), \tag{3.1.18}$$

where $\boldsymbol{x}'$ is the location of the volume element $dV(\boldsymbol{x}')$, $G(r) = 1/4\pi r$ is the three-dimensional Green function for the Poisson equation, and $r = |\boldsymbol{x} - \boldsymbol{x}'|$. Combining (3.1.16) and (3.1.18) yields the Biot–Savart law for the vortical velocity, $\boldsymbol{u}_\omega$,

$$\boldsymbol{u}_\omega(\boldsymbol{x}, t) = \int K(\boldsymbol{x} - \boldsymbol{x}')^\wedge \boldsymbol{\omega}(\boldsymbol{x}', t)\, dV(\boldsymbol{x}'), \tag{3.1.19}$$

where $K(\boldsymbol{x}) = -\boldsymbol{x}/4\pi|\boldsymbol{x}|^3$ is the three-dimensional velocity kernel. Given the vorticity distribution (3.1.10), the singularity in the velocity kernel may result in numerical instability as elements approach each other. To remove the singularity, we smooth out the vorticity associated with each element by implementing the vortex "blob" regularization technique, where the delta function in (3.1.10) is replaced by a spherical core function $g_\sigma(\boldsymbol{x}) = (1/\sigma^3)g(|\boldsymbol{x}|/\sigma)$ with core radius σ. The criteria for constructing proper core functions with arbitrary spatial order, as well as the necessary conditions for maintaining convergence and stability are available in the literature [4, 6, 7, 9]. In our computations we applied the second-order core function $g(\boldsymbol{x}) = (3/4\pi)\ [1 - \tanh^2(|\boldsymbol{x}|^3)]$—selected from a list of functions made available by Beale and Majda [9]. We also experimented with $g(\boldsymbol{x}) = (3/4\pi)\exp(-|\boldsymbol{x}|^3)$ from the same list and found the distribution to be spurious in the vicinity of the core center, due to machine precision, which often led to numerical instability.

The discrete, regularized vortical velocity is obtained by substituting the smooth vorticity distribution into Eq. (3.1.19),

$$\boldsymbol{u}_\omega(\boldsymbol{x}, t) = \sum_{j=1}^{N_V} K_\sigma(\boldsymbol{x} - \boldsymbol{x}_j)^\wedge \tilde{\boldsymbol{\Gamma}}_j(t), \tag{3.1.20}$$

where $K_\sigma(\boldsymbol{x}) = K(\boldsymbol{x})f(|\boldsymbol{x}|/\sigma)$, $f(r) = 4\pi \int_0^r g(r')r'^2\, dr' = \tanh(r^3)$, and $K_\sigma(0) = 0$.

Equation (3.1.20) can now be differentiated analytically in Eulerian coordinates to obtain the vortical velocity gradients [1],

$$\nabla \boldsymbol{u}_\omega(\boldsymbol{x}, t) = \sum_{j=1}^{N_V} \nabla K_\sigma(\boldsymbol{x} - \boldsymbol{x}_j)^\wedge \tilde{\boldsymbol{\Gamma}}_j(t), \tag{3.1.21}$$

$$\nabla K_\sigma(\boldsymbol{x}) = \left(\frac{\nabla \boldsymbol{x}}{\boldsymbol{x}} - 3\frac{\nabla|\boldsymbol{x}|}{|\boldsymbol{x}|}\right) K_\sigma(\boldsymbol{x}) - \boldsymbol{x}\frac{\nabla|\boldsymbol{x}|}{|\boldsymbol{x}|} g_\sigma(\boldsymbol{x}),$$

$$\nabla K_\sigma(0) = -\frac{\nabla \boldsymbol{x}}{3\sigma^3} g(0).$$

3.1.4. *Potential Component*

The potential velocity at any point in the domain is expressed in terms of the gradient of a scalar function as

$$\boldsymbol{u}_P = -\nabla\Phi. \tag{3.1.22}$$

Substituting (3.1.7) and (3.1.22) in the continuity equation yields the internal Neumann problem,

$$\nabla^2\Phi(\boldsymbol{x}) = 0, \qquad \boldsymbol{x} \in D, \tag{3.1.23}$$

$$\begin{aligned} q(\boldsymbol{x}_o) &= -\boldsymbol{u}_P(\boldsymbol{x}_o)\cdot \boldsymbol{n} \\ &= -(\boldsymbol{u}(\boldsymbol{x}_o) - \boldsymbol{u}_\omega(\boldsymbol{x}_o))\cdot \boldsymbol{n}, \quad \boldsymbol{x}_o \in \partial D, \end{aligned} \tag{3.1.24}$$

$$\int_{\partial D} q(\boldsymbol{x}_o)\, dS(\boldsymbol{x}_o) = 0, \qquad \boldsymbol{x}_o \in \partial D, \tag{3.1.25}$$

where $q(\boldsymbol{x}_o) = \nabla\Phi(\boldsymbol{x}_o)\cdot \boldsymbol{n}$ is the normal flux at the boundary and $S(\boldsymbol{x}_o)$ is the surface area of the boundary.

The boundary condition (3.1.24) incorporates the effect of the vortical velocity component on the total flux at the boundaries, and constraint (3.1.25) ensures that the net flux into the domain is zero. Note that the solution for the internal Neumann problem is unique only up to a constant in Φ; nevertheless, the velocity and its gradients, which are of interest to us, are defined uniquely. The nonunique nature of the potential distribution is analogous to that of the pressure distribution in the primitive variable formulation where the pressure gradient, and not the pressure, is of relevance to the computation.

To preserve the grid-free nature of the vortex element method, the Neumann problem is solved by the direct boundary element method which converts the differential equation (3.1.23) within the volume into an integral equation on the surface. We begin the solution of the Neumann problem by decomposing the boundary surface into a union of rectangular elements and by assigning piecewise-quadratic variation of the potential and its normal flux across each element. Since the normal flux is multivalued on the boundary edges and corners, we implement two separate global node numbering systems for Φ and q. The first is used to discretize the potential distribution over N_Φ global nodes in the regular sense, while the second utilizes a multiple-node discretization approach assigning as many nodes to a point on an edge (or a corner) as there are surfaces sharing the edge (or the corner)—to describe the normal flux over N_q global nodes [25]. The discrete boundary integral equation yields [2]

$$\begin{aligned}\alpha\Phi(\boldsymbol{x}) = \sum_{k=1}^{M}\Bigg[&-\int_{\partial D_k}\Phi_k(\boldsymbol{x}_o)\nabla G(\boldsymbol{x}_o,\boldsymbol{x})\cdot\boldsymbol{n}\,dS_k(\boldsymbol{x}_o)\\ &+\int_{\partial D_k} q_k(\boldsymbol{x}_o)G(\boldsymbol{x}_o,\boldsymbol{x})\,dS_k(\boldsymbol{x}_o)\Bigg], \qquad (3.1.26)\\ &\boldsymbol{x}_o\in\partial D, \boldsymbol{x}\in D,\end{aligned}$$

where M is the number of rectangular elements, G is the three-dimensional Green function as defined earlier, and α is the normalized solid angle,

$$\alpha = \begin{cases} 1, & \boldsymbol{x}\in(D\backslash\partial D),\\ -\displaystyle\sum_{k=1}^{M}\int_{\partial D_k}\nabla G(\boldsymbol{x}_o,\boldsymbol{x})\cdot\boldsymbol{n}\,dS_k(\boldsymbol{x}_o), & \boldsymbol{x}\in\partial D,\\ 0, & \boldsymbol{x}\notin D,\end{cases}$$

while $\Phi_k(\boldsymbol{x}_o) = \sum_{i=1}^{8} I_k^i(\xi,\zeta)\Phi_{p_k^\Phi(i)}(\boldsymbol{x}_o)$ and $q_k(\boldsymbol{x}_o) = \sum_{i=1}^{8} I_k^i(\xi,\zeta)q_{p_k^q(i)}(\boldsymbol{x}_o)$. $I(\cdot)$ is the quadratic interpolation function, and $\xi\equiv\xi_k = \boldsymbol{\tau}^k\cdot(\boldsymbol{x}_o^k - \boldsymbol{x})$ and $\zeta\equiv\zeta_k = \boldsymbol{\rho}^k\cdot(\boldsymbol{x}_o^k - \boldsymbol{x})$ are orthogonal coordinates local to element k; $p_k^\Phi(i)$ and $p_k^q(i)$ represent pointers from the local node i of element k to the corresponding global node numbers for Φ and q, respectively. Coalescing $\boldsymbol{x}$ with each of the N_Φ collocation nodes and carrying out the singular integrations in (3.1.26) sets up a linear system of equations for the unknown potential at the N_Φ nodes. The matrix representing the linear system is singular and reflects the nonuniqueness of the potential distribution in discrete form. To obtain a unique solution, the potential at an arbitrary collocation point m may be set to zero and the matrix inverted by the method of elimination. However, since collocation techniques are sensitive to grid distribution, the resulting boundary potential suffers from a highly localized discretization error—manifesting itself as a spike at m—which deteriorates the accuracy of the potential solution in the interior substantially. We alleviate this problem by adding a pseudo-Lagrange multiplier to the system of equations, which equilibrates the imbalance between the left and right sides of the equations by scaling and redistributing the residual errors on an area-weighted basis [25].

Subsequent to the solution of the linear system of equations, the potential velocity and its gradients may be evaluated anywhere in the domain by directly differentiating Eq. (3.1.26). However, the integrals contain $O(r^{-3})$ and $O(r^{-4})$ singularities, respectively, and the accuracy of the integral evaluations deteriorates significantly as $\boldsymbol{x}$ approaches the boundary. The accurate evaluation of the potential velocity and its gradients is especially critical near the boundary, because vorticity is generated at the walls and its evolution is influenced by the potential component substantially.

A regularization technique is available which reduces the order of the integrand singularity for the velocity [57, 58] and its gradients [25] by one and two, respectively, in exchange for the evaluation of the tangential derivatives of the boundary potential and its normal flux,

$$\begin{aligned}u_{P_j}(\boldsymbol{x}) = \sum_{k=1}^{M}\Bigg[\int_{\partial D_k}\{&(\xi_k n_j - \eta_k\tau_j)\Phi^k_{,\xi}(\xi,\zeta)\\ &+(\zeta_k n_j - \eta_k\rho_j)\Phi^k_{,\zeta}(\xi,\zeta)\\ &-(\xi_k\tau_j + \zeta_k\rho_j + \eta_k n_j)q^k(\xi,\zeta)\}\frac{dS_k}{4\pi r_k^3}\Bigg], \qquad (3.1.27)\end{aligned}$$

$$\begin{aligned}\frac{\partial u_{P_j}(\boldsymbol{x})}{\partial x_l} = \sum_{k=1}^{M}\Bigg[\int_{\partial D_k}\{&(\xi_k(n_j\tau_l + \tau_j n_l)\\ &+\zeta_k\rho_j n_l + \eta_k(n_j n_l - \tau_j\tau_l))\Phi^k_{,\xi\xi}(\xi,\zeta)\\ &+(\zeta_k(n_j\rho_l + \rho_j n_l) + \xi_k\tau_j n_l + \eta_k(n_j n_l - \rho_j\rho_l))\Phi^k_{,\zeta\zeta}(\xi,\zeta)\\ &+(\xi_k n_j\rho_l + \zeta_k n_j\tau_l - \eta_k(\tau_j\rho_l + \rho_j\tau_l))\Phi^k_{,\xi\zeta}(\xi,\zeta)\\ &+(\xi_k(n_j n_l - \tau_j\tau_l) - \zeta_k\rho_j\tau_l - \eta_k(n_j\tau_l + \tau_j n_l))q^k_{,\xi}(\xi,\zeta)\\ &+(\zeta_k(n_j n_l - \rho_j\rho_l) - \xi_k\tau_j\rho_l\\ &-\eta_k(n_j\rho_l + \rho_j n_l))q^k_{,\zeta}(\xi,\zeta)\}\frac{dS_k}{4\pi r_k^3}\Bigg], \quad (j,l) = 1,2,3,\end{aligned} \qquad (3.1.28)$$

where j and l indices indicate direction with respect to the global coordinate system and follow the Einstein rule, $(\cdot)_{,r}$ represents differentiation with respect to $\boldsymbol{x}_o$ in the rth direction, and $\eta_k = \boldsymbol{n}^k\cdot(\boldsymbol{x}_o^k - \boldsymbol{x})$—not to be confused with the random walk displacement. The first- and second-degree tangential derivatives of the normal flux and the potential are obtained by directly differentiating the corresponding interpolation functions. For example, the first derivative of the potential in the local ξ direction is given by $\Phi^k_{,\xi}(\boldsymbol{x}_o) = \sum_{i=1}^{8}(\partial I_k^i(\xi,\zeta)/\partial\xi)\Phi_{p_k^\Phi(i)}(\boldsymbol{x}_o)$, while the derivative of the latter in the local ζ direction is given by $\Phi^k_{,\xi\zeta}(\boldsymbol{x}_o) = \sum_{i=1}^{8}(\partial^2 I_k^i(\xi,\zeta)/\partial\xi\partial\zeta)\Phi_{p_k^\Phi(i)}(\boldsymbol{x}_o)$.

3.2. Boundary Domain

In this section the processes of vorticity generation and its evolution within the numerical boundary layer extending over a flat rectangular wall are described. The algorithm is applied to all solid walls simultaneously. Fishelov's [22] implementation using tiles of equal size is herein extended to the case of tiles with arbitrary size, which is significant in problems dealing with expanding or shrinking boundaries. In what follows, all variables are defined with respect to the local coordinate system, z is assigned normal to the plate and into the flow interior, and $z = 0$ represents the wall surface.

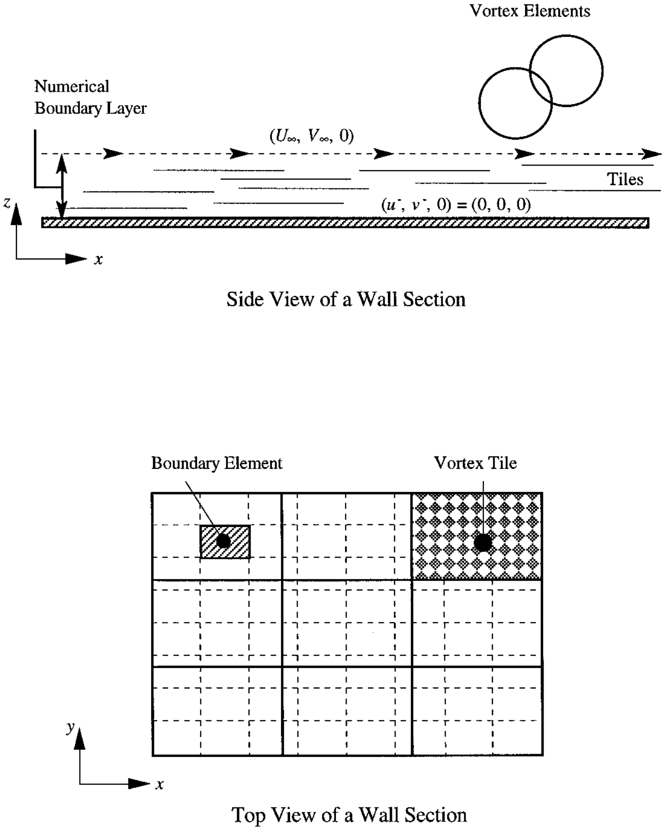
Vortex Elements
Numerical Boundary Layer
$(U_\infty, V_\infty, 0)$
Tiles
$(u^-, v^-, 0) = (0, 0, 0)$
z
x
Side View of a Wall Section
Boundary Element
Vortex Tile
y
x
Top View of a Wall Section

FIG. 1. Schematic diagram of a typical boundary domain.

3.2.1. *No-Slip at the Wall*

The distribution of vorticity in the interior and the application of the normal flux boundary condition at the walls induce a tangential velocity (u^+, v^+), which is generally different from the prescribed boundary condition (u^-, v^-)—being (0, 0) for a stationary wall. Such a velocity jump across the wall is equivalent to an infinitely large vorticity along an infinitesimally thin surface, centered at the wall,

$$\hat{\boldsymbol{\gamma}}(\boldsymbol{x}, t) = (\hat{\gamma}_x, \hat{\gamma}_y, \hat{\gamma}_z) = (-(v^+ - v^-), (u^+ - u^-), 0), \quad (3.2.1)$$

where $\lim_{\Delta z \to 0,\ \omega \to \infty}[\boldsymbol{\omega}(\boldsymbol{x}, t)\Delta z] \to \hat{\boldsymbol{\gamma}}(\boldsymbol{x}, t)$ [42]. In what follows, for the lack of a better terminology, we will call $\hat{\boldsymbol{\gamma}}(\boldsymbol{x}, t)$ and its discretized form "surface vorticity." Equation (3.2.1) defines the mechanism for satisfying the no-slip condition at the wall. $\hat{\boldsymbol{\gamma}}(\boldsymbol{x}, t)$ is now discretized by carpeting the wall with a set of rectangular vortex tiles with sides $h^t_{x_i}$ and $h^t_{y_i}$ and with surface vorticity $\overline{\boldsymbol{\gamma}}(x_i, y_i, 0, t)$ at the center, $(x_i, y_i, 0)$. $\overline{\boldsymbol{\gamma}}(x_i, y_i, 0, t)$ is in turn linked to the velocity jumps at the center of the boundary elements, which are themselves retrieved from the Navier–Stokes solution. The tile distribution is allowed to be different from that of the boundary elements, so that a tile may lie entirely within one or "shadow" over multiple boundary elements. (Figure 1 is the schematic of a typical solid boundary domain.) The surface vorticity of a tile is then obtained by summing the area-weighted jump contributions from all boundary elements that are shadowed by the tile,

$$\overline{\gamma}_x(x_i, y_i, 0, t) = -\sum_{j=1}^{M_B} \overline{(v^+ - v^-)}_j \Lambda(x_i, x_j, h^t_{x_i}, h^b_{x_j}) \Lambda(y_i, y_j, h^t_{y_i}, h^b_{y_j}), \quad (3.2.2)$$

$$\overline{\gamma}_y(x_i, y_i, 0, t) = \sum_{j=1}^{M_B} \overline{(u^+ - u^-)}_j \Lambda(x_i, x_j, h^t_{x_i}, h^b_{x_j}) \Lambda(y_i, y_j, h^t_{y_i}, h^b_{y_j}), \quad (3.2.3)$$

where

$$\Lambda(x_i, x_j, h^t_{x_i}, h^b_{x_j}) = 1 - \frac{\operatorname{Max}\left(\frac{|h^t_{x_i} - h^b_{x_j}|}{2}, \operatorname{Min}\left(|x_i - x_j|, \frac{h^t_{x_i} + h^b_{x_j}}{2}\right)\right) + \frac{h^t_{x_i} - h^b_{x_j}}{2}}{h^t_{x_i}}$$

and M_B is the number of boundary elements on the wall, each centered at (x_j, y_j, 0) with sides $h^b_{x_j}$ and $h^b_{y_j}$. The velocity overbars denote the area-averaged values over the boundary elements. Note that for a uniform distribution of tiles, $\Lambda(x_i, x_j, h^t_{x_i}, h^b_{x_j})$ is reduced to the familiar form of $\Lambda(x_i, x_j, h_x) = 1 - \operatorname{Min}(|x_i - x_j|, h_x)/h_x$ [22, 50].

To obtain a finer discretization of the flow in the direction normal to the wall, each tile is split into $N_{S,i} = (|\overline{\boldsymbol{\gamma}}(x_i, y_i, 0, t)|/\gamma_{\max} + 0.5)$ stacks of tiles if $\overline{\boldsymbol{\gamma}}(x_i, y_i, 0, t)$ exceeds a predefined maximum surface vorticity, $\gamma_{\max}$ [22, 28, 29, 50]. The geometric attributes of the new tiles are identical to those of the original one, while their surface vorticity values are given by $\overline{\boldsymbol{\gamma}}(x_i, y_i, 0, t) = \overline{\boldsymbol{\gamma}}(x_i, y_i, 0, t)/N_{S,i}$. In the present work, we assign the inverse of $\gamma_{\max}$ as the user-specified parameter and call it the "boundary layer resolution parameter," *BLRP*, owing to the fact that higher values of *BLRP* are expected to improve the resolution in the normal direction to the wall. The choice of *BLRP* depends on how complicated the flow near the boundary is. However, since a higher value of *BLRP* implies a larger number of vortex elements in the field, computational cost is also a factor in selecting a suitable value.

Finally, the discrete no-slip boundary condition is satisfied by generating the discrete vorticity at the wall,

$$\tilde{\boldsymbol{\gamma}}(x, y, 0, t) = \sum_{i=1}^{N_T} \tilde{\boldsymbol{\gamma}}(x_i, y_i, 0, t)\beta(x, x_i, h^t_{x_i})\beta(y, y_i, h^t_{y_i}), \qquad (3.2.4)$$

where $\beta(x, x_i, h_{x_i}) = [s(x - (x_i - h_{x_i}/2)) - s(x - (x_i + h_{x_i}/2))]$, s is the Heaviside step function, and N_T is the total number of vortex tiles.

3.2.2. *Prandtl Equation*

Once vorticity is generated at the wall, its evolution within the numerical boundary layer, D_b, is approximated by the equations [14, 22]

$$\frac{\partial \boldsymbol{\omega}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{\omega} = \frac{1}{\mathrm{R_e}} \frac{\partial^2 \boldsymbol{\omega}}{\partial z^2}, \qquad \boldsymbol{x} \in D_b, \qquad (3.2.5)$$

$$\nabla \cdot \boldsymbol{u} = 0, \qquad \boldsymbol{x} \in D_b, \qquad (3.2.6)$$

$$\boldsymbol{\omega}(\boldsymbol{x}, t) = (\omega_x, \omega_y, \omega_z) \cong \left(-\frac{\partial v}{\partial z}, \frac{\partial u}{\partial z}, 0\right), \qquad \boldsymbol{x} \in D_b, \qquad (3.2.7)$$

$$\boldsymbol{u}(\boldsymbol{x}, t = 0) = \boldsymbol{u}_o(\boldsymbol{x}), \qquad \boldsymbol{x} \in D_b,$$

$$\boldsymbol{u}(x, y, z = 0, t) = (0, 0, 0), \quad \boldsymbol{u}(x, y, z = b, t) = (U_\infty, V_\infty, 0),$$

where b is a user-specified numerical boundary layer thickness to be discussed later, and $(U_\infty, V_\infty, 0)$ is the velocity at the edge of the boundary layer as seen by the wall. Note that vorticity stretch is assumed negligible within the boundary layer and is omitted from the equations. This assumption may be questionable, since the stretch term is in the same order as the convective term. We have, however, experimented with equations that contain the stretch term and have found the solution to be numerically unstable. (Our experience corroborated Chorin's findings [16].) We suspect that poor or noisy tile discretization of the velocity gradients causes an unphysical growth of the vorticity in the boundary layer, which eventually leads to blowup. Since b is in practice small and the tiles jump into the interior in a few timesteps, it seems reasonable to "postpone" the effect of stretch until the tiles are converted to vortex elements. Therefore, we neglect the stretch term to maintain numerical stability.

Application of viscous splitting to the Lagrangian equivalent of (3.2.5) yields

$$\frac{d\boldsymbol{\chi}}{dt}(\boldsymbol{x}, t) = \boldsymbol{u}(\boldsymbol{\chi}, t), \qquad \boldsymbol{\chi}(\boldsymbol{x}, t_o) = (x, y, 0, t_o), \qquad (3.2.8)$$

$$\frac{d\boldsymbol{\omega}}{dt}(\boldsymbol{\chi}, t) = \frac{1}{\mathrm{R_e}} \frac{\partial^2 \boldsymbol{\omega}}{\partial z^2}(\boldsymbol{\chi}, t), \quad \boldsymbol{\omega}(\boldsymbol{\chi}, t_o) = \boldsymbol{\omega}(x, y, 0, t_o). \quad (3.2.9)$$

We begin the solution of Eqs. (3.2.8) and (3.2.9) by discretizing the continuous vorticity within the layer into a collection of N_T vortex tiles, centered at $\boldsymbol{x}_i$ and with surface vorticity $\tilde{\boldsymbol{\gamma}}(\boldsymbol{x}_i, t)$:

$$\boldsymbol{\omega}(\boldsymbol{x}, t) = \sum_{i=1}^{N_T} \tilde{\boldsymbol{\gamma}}(\boldsymbol{x}_i, t)\beta(x, x_i, h^t_{x_i})\beta(y, y_i, h^t_{y_i})\delta(z - z_i). \qquad (3.2.10)$$

Combining (3.2.10) with (3.2.8) and (3.2.9) yields the grid-free tile solution within the numerical boundary layer,

$$\boldsymbol{x}_i^*(t_{k+1}) = \boldsymbol{x}_i(t_k) + \boldsymbol{u}(\boldsymbol{x}_i, t_k)\,\Delta t, \qquad (3.2.11)$$

$$z_i(t_{k+1}) = |z_i^*(t_{k+1}) + \eta_i(\Delta t)|, \quad k = 0, 1, \ldots; i = 1, \ldots, N_T, \qquad (3.2.12)$$

where $\boldsymbol{x}_i = (x_i, y_i, z_i)$ denotes the tile trajectory $\boldsymbol{\chi}_i$ in the current context and is predicted by applying the first-order Euler time integration, which has been demonstrated to be sufficiently accurate in this case [50]. The asterisk pertains only to the z direction, implying that viscous diffusion is in the direction normal to the wall. Furthermore, to simulate diffusion from the wall into the domain interior, any tile that jumps below the wall is reflected back into the field [50].

For a given vorticity distribution within the layer, the corresponding u and v velocity components at a point are derived by directly integrating the approximate definitions for ω_y and ω_x, respectively. In this formulation, the boundary conditions are satisfied by construction. The w component is obtained by satisfying the continuity equation (3.2.6):

$$u(\boldsymbol{x}, t) = U_\infty(x, y, b, t) - \int_z^b \omega_y(x, y, z', t)\, dz,' \qquad (3.2.13)$$

$$v(\boldsymbol{x}, t) = V_\infty(x, y, b, t) + \int_z^b \omega_x(x, y, z', t)\, dz', \qquad (3.2.14)$$

$$w(\boldsymbol{x}, t) = -\partial_x \int_0^z u(x, y, z', t)\, dz' - \partial_y \int_0^z v(x, y, z', t)\, dz'. \qquad (3.2.15)$$

The discrete tangential components of the tile convective velocities in (3.2.11) are obtained by substituting the discrete vorticity distribution (3.2.10) in Eqs. (3.2.13) and (3.2.14), and area-averaging the results at the tile centers (x_i, y_i, z_i):

$$u(\boldsymbol{x}_i, t) = U_\infty(x_i, y_i, b, t) - \frac{1}{2}\tilde{\gamma}_y(\boldsymbol{x}_i, t) - \sum_{\substack{j=1 \\ j\neq i}}^{N_T} \tilde{\gamma}_y(\boldsymbol{x}_j, t)\varphi_j(x_i, y_i)s(z_j - z_i), \tag{3.2.16a}$$

$$v(\boldsymbol{x}_i, t) = V_\infty(x_i, y_i, b, t) + \frac{1}{2}\tilde{\gamma}_x(\boldsymbol{x}_i, t) + \sum_{\substack{j=1 \\ j\neq i}}^{N_T} \tilde{\gamma}_x(\boldsymbol{x}_j, t)\varphi_j(x_i, y_i)s(z_j - z_i), \quad i = 1, ..., N_T, \tag{3.2.16b}$$

where $\varphi_j(x_i, y_i) = \Lambda(x_i, x_j, h^t_{x_i}, h^t_{x_j})\Lambda(y_i, y_j, h^t_{y_i}, h^t_{y_j})$. Similarly, the normal velocity component is obtained from Eq. (3.2.15),

$$w(\boldsymbol{x}_i, t) = -\frac{(I_x^+ - I_x^-)}{h^t_{x_i}} - \frac{(I_y^+ - I_y^-)}{h^t_{y_i}}, \quad i = 1, ..., N_T, \tag{3.2.16c}$$

where the derivatives are approximated by the divided difference rule, and

$$I_x^\pm = U_\infty\left(x_i \pm \frac{h^t_{x_i}}{2}, y_i, b, t\right)z_i - \sum_{j=1}^{N_T} \tilde{\gamma}_y(\boldsymbol{x}_j, t)\varphi_j\left(x_i \pm \frac{h^t_{x_i}}{2}, y_i\right)\mathrm{Min}(z_i, z_j), \tag{3.2.16d}$$

$$I_y^\pm = V_\infty\left(x_i, y_i \pm \frac{h^t_{y_i}}{2}, b, t\right)z_i + \sum_{j=1}^{N_T} \tilde{\gamma}_x(\boldsymbol{x}_j, t)\varphi_j\left(x_i, y_i \pm \frac{h^t_{y_i}}{2}\right)\mathrm{Min}(z_i, z_j). \tag{3.2.16e}$$

The numerical algorithm proceeds as follows: During each timestep, vortex tiles within the layer are first convected according to (3.2.11) and using (3.2.16), thus establishing the velocity link between the interior and the boundary domains. Note that this step is ignored if there are no tiles within the numerical layer, including the very first timestep. Subsequent to the convection process, new vortex tiles are created on the boundary according to (3.2.4) to satisfy the no-slip boundary condition. The conclusion of this process is coincident with the end of the diffusion step within the interior domain, and the simultaneous satisfaction of both the normal flux and the no-slip boundary conditions. Finally all tiles, old and new, are diffused inward and normal to the walls according to (3.2.12)—signifying the end of the computational timestep. Once a tile jumps into the flow interior, it is converted into a spherical vortex element with volumetric vorticity $\tilde{\boldsymbol{\Gamma}}(\boldsymbol{x}, t) = \tilde{\boldsymbol{\gamma}}(\boldsymbol{x}, t)h^t_x h^t_y$ and with core radius $\sigma = \mathrm{Max}(h^t_x, h^t_y)$—linking the vorticity between the interior and the boundary domains.

The criterion for selecting the numerical boundary layer thickness b is artibrary but is normally taken to be some factor of the standard deviation of the random walk, $b = BLTC\sqrt{2\Delta t/\mathrm{R_e}}$, so that the tiles will jump into the flow interior in a few timesteps with relatively high probability [22, 45]. Note that $\mathrm{R_e}$ is the global Reynolds number as defined earlier and must not be confused with the local Reynolds number, which is based on the tile dimensions. $BLTC$ is a user-defined "boundary layer thickness coefficient," which is recommended to be in the range of $1.0 \leq BLTC \leq 3.0$ and is usually set at 1.5. We will show in our numerical experiments that the solution is sensitive to the choice of $BLTC$—at least at lower Reynolds numbers.

Within the numerical boundary layer, Δt is constrained by the requirement that all tiles reside within their own region of influence for at least one timestep: $\Delta t \leq (h/U)_{\min}$. Note that although the inequality has the appearance of the familiar CFL condition, it is only an accuracy condition for properly resolving the timescales in the flow. On the other hand, within the interior domain, Δt must be checked against the inter-element distances to ensure numerical stability by maintaining core overlap everywhere and at all times. Since the core radius is set equal to the tile side, it is reasonable to retain $\Delta t \leq (h/U)_{\min}$ as the stability criterion in the domain interior.

3.3. Exit Boundary Condition

Specification of a suitable exit boundary condition for incompressible wall-bounded flows is a nontrivial task in any numerical method, because of the elliptic nature of the equations of motion and the effect the arbitrary choice of the boundary condition can have on the rest of the flow field. Fortunately, for convection dominated flows the exit boundary effects remain relatively local and the error signals do not penetrate deeper than one or two characteristic length scales [29]. To date, the most commonly applied boundary condition in vortex methods is the specification of a uniform potential flux at the exit such that potential continuity is satisfied on the boundary:

$$(\boldsymbol{u}_P(\boldsymbol{x}_o) \cdot \boldsymbol{n})_{\text{exit}} = \left(\frac{-1}{S_{\text{exit}}}\right)\int_{\partial D\backslash\partial D_{\text{exit}}} \boldsymbol{u}_P(\boldsymbol{x}_o) \cdot \boldsymbol{n}\, dS(\boldsymbol{x}_o), \quad \boldsymbol{x}_o \in \partial D. \tag{3.3.1}$$

For high Reynolds number flow in a duct, for example, where the fully developed velocity profile is nearly flat, the above condition produces satisfactory results if the duct length is extended by an extra duct height; for lower Reynolds numbers an extension of two duct heights is safer. However, there are two obvious disadvantages to this boundary condition: (1) the extra duct length implies unnecessarily extra computations; (2) a flat potential flux almost guarantees a slip velocity at the walls circumscribing the exit surface, which is inconsistent with the applied no-slip condition there.

We introduce here the more physically correct concept of a fully developed velocity profile at the exit, $(\partial \boldsymbol{u} \cdot \boldsymbol{n}/\partial n)_{\text{exit}} = 0$. In what follows, the subscript "exit" is dropped from all variables for brevity. Decomposing the exit velocity into its potential and vortical components yields $\partial \boldsymbol{u}_P \cdot \boldsymbol{n}/\partial n = -\partial \boldsymbol{u}_\omega \cdot \boldsymbol{n}/\partial n$. Furthermore, the three-dimensional Laplace equation may be rearranged at the exit plane, in terms of the local coordinate system, into

$$\frac{\partial^2 \Phi}{\partial \xi^2} + \frac{\partial^2 \Phi}{\partial \zeta^2} = -\frac{\partial}{\partial n}\left(\frac{\partial \Phi}{\partial n}\right) = \frac{\partial \boldsymbol{u}_P \cdot \boldsymbol{n}}{\partial n}.$$

Combining the last two equations leads to a two-dimensional Poisson equation at the exit plane with a Neumann boundary condition:

$$\nabla^2 \Phi(\boldsymbol{x}_o) = -\frac{\partial \boldsymbol{u}_\omega(\boldsymbol{x}_o) \cdot \boldsymbol{n}}{\partial n}, \quad \boldsymbol{x}_o \in \partial D, \tag{3.3.2}$$

$$\begin{aligned} \nabla \Phi(\boldsymbol{x}'_o) \cdot \hat{\boldsymbol{n}} &= -\boldsymbol{u}_P(\boldsymbol{x}'_o) \cdot \hat{\boldsymbol{n}} \\ &= -(\boldsymbol{u}(\boldsymbol{x}'_o) - \boldsymbol{u}_\omega(\boldsymbol{x}'_o)) \cdot \hat{\boldsymbol{n}}, \quad \boldsymbol{x}'_o \in \partial(\partial D), \end{aligned} \tag{3.3.3}$$

$$\begin{aligned} \int_{\partial(\partial D)} \nabla \Phi(\boldsymbol{x}'_o) \cdot \hat{\boldsymbol{n}}\, dP(\boldsymbol{x}'_o) \\ = -\int_{\partial D} \frac{\partial \boldsymbol{u}_\omega(\boldsymbol{x}_o) \cdot \boldsymbol{n}}{\partial n}\, dS(\boldsymbol{x}_o), \quad \boldsymbol{x}'_o \in \partial(\partial D), \end{aligned} \tag{3.3.4}$$

where ∂D is the exit plane, $\partial(\partial D)$ is its boundary, and $P(\boldsymbol{x}'_o)$ represents the contour enclosing the exit plane; $\hat{\boldsymbol{n}}$ is the unit outward normal to the boundary of, and in, the exit plane.

The Poisson equation can now be solved by a standard finite element method using the same grid distribution that is used in the boundary element section of the solution. Note that the equation above does not have a unique solution in the potential and the assembled matrix is singular. However, the singularity may be removed by the standard penalty method and the arbitrary assignment of the potential at a point on the exit surface. Given the potential distribution at the exit, the three-dimensional Laplace equation (3.1.23) admits a mixed Dirichlet–Neumann boundary condition, which has a unique solution.

4. RESULTS

Results presented in the next two subsections are intended to test the convergence of the method, to show the impact of different numerical parameters on the solution accuracy, and to demonstrate the ability of the scheme to capture complex vortical structures. For this purpose, we chose a rather simple flow which possesses an analytical solution, and a more complex flow which has been studied before using a different numerical approach. We emphasize here that being the first test for the method, the flows were carefully selected such that a true comparison can be conducted between the simulation results and other solutions.

4.1. Flow in a Duct

We present here results from a set of parametric studies for the case of flow in a duct with square cross section. A total of nine cases were examined. We chose this particular geometry because of the availability of an exact solution for the fully developed velocity profile [54], against which we verified our results. The Reynolds number based on the duct height H and uniform inlet velocity U_{in} was set at 100. The ratio of the duct length L_c to its height must satisfy $L_c/H \geq \mathrm{R_e}/20$ to obtain a fully developed velocity profile at the exit [55]. Given this condition and the discussion in Section 3.3, the duct dimensions were set at 1 : 1 : 7 to minimize the effect of the exit boundary condition on the solution. The Reynolds number was chosen so as to ensure that the flow remains laminar and stable, even under the weak numerical perturbations caused by the random walk process.

Table I shows the list of all the relevant parameters in this study. The tile and boundary element resolutions in the table refer to the number of rectangular elements per duct wall. The origin is set at a corner of the duct inlet, and the z coordinate points toward the exit surface. In all cases, the no-slip boundary condition is satisfied on the walls and the inlet velocity is fixed at $U_{\text{in}} = 1.0$. The exit boundary condition is satisfied either by the standard or the fully developed exit velocity formulation—as assigned in the table. The initial conditions are set as follows: At $t = 0^-$ the velocity and the vorticity are zero throughout the domain. At $t = 0^+$ a potential flow field with a uniform velocity, $U_{\text{in}} = 1.0$, is applied to the duct. This sets the slip velocity at the duct walls which generates the vorticity field within the numerical boundary layer.

Figure 2 depicts the rear and side views of five typical timeslices from the evolution of the vortex elements and their velocity vectors, emanating from one of the duct walls in Case 9. (The sticks represent the velocity vectors and the solid circles depict their origin at the center of the vortex elements.) The transient boundary layer development between the flow startup at $t = 0^+$ and the onset of stationary state is captured in the first four time frames. The flow in the last frame is fully developed and is at stationary state, which is confirmed by the familiar bound-

TABLE I

Case	*BLRP*	*BLTC*	Δt	Number of timesteps	BEM resolution	Tile resolution	Boundary condition
1	4	1.5	0.1	300	4 × 24	3 × 21	Std.
2	2	1.5	0.1	300	4 × 24	3 × 21	Std.
3	8	1.5	0.1	300	4 × 24	3 × 21	Std.
4	4	1.0	0.1	300	4 × 24	3 × 21	Std.
5	4	0.5	0.1	600	4 × 24	3 × 21	Std.
6	4	1.5	0.1	300	4 × 24	3 × 21	Ful. Dev.
7	4	0.5	0.1	600	4 × 24	3 × 21	Ful. Dev.
8	4	1.5	0.01	3000	4 × 24	3 × 21	Std.
9	4	1.5	0.1	300	8 × 35	7 × 32	Std.

ary layer envelope. The vorticity vectors for the displayed elements point into the paper, and those from the opposite wall point out of the paper. Similarly, the vectors from the side walls are directed such that they form loops with the former pairs. The tiles in the numerical boundary layer, occupying the blank space between the elements and the lower wall, are not displayed to avoid unnecessary clutter. The distinction between the physical boundary layer, which grows in time and space to fill one quarter of the duct at the exit, and the thin numerical boundary layer, which is fixed, is readily evident in the figure.

Cases 1 through 3 represent a parametric study of the convergence properties of the numerical scheme as a function of *BLRP*, whereas Cases 1, 4, and 5 represent a similar study as a function of *BLTC*. (Case 1 is the baseline case.) Figures 3a and 3b depict the respective time histories for

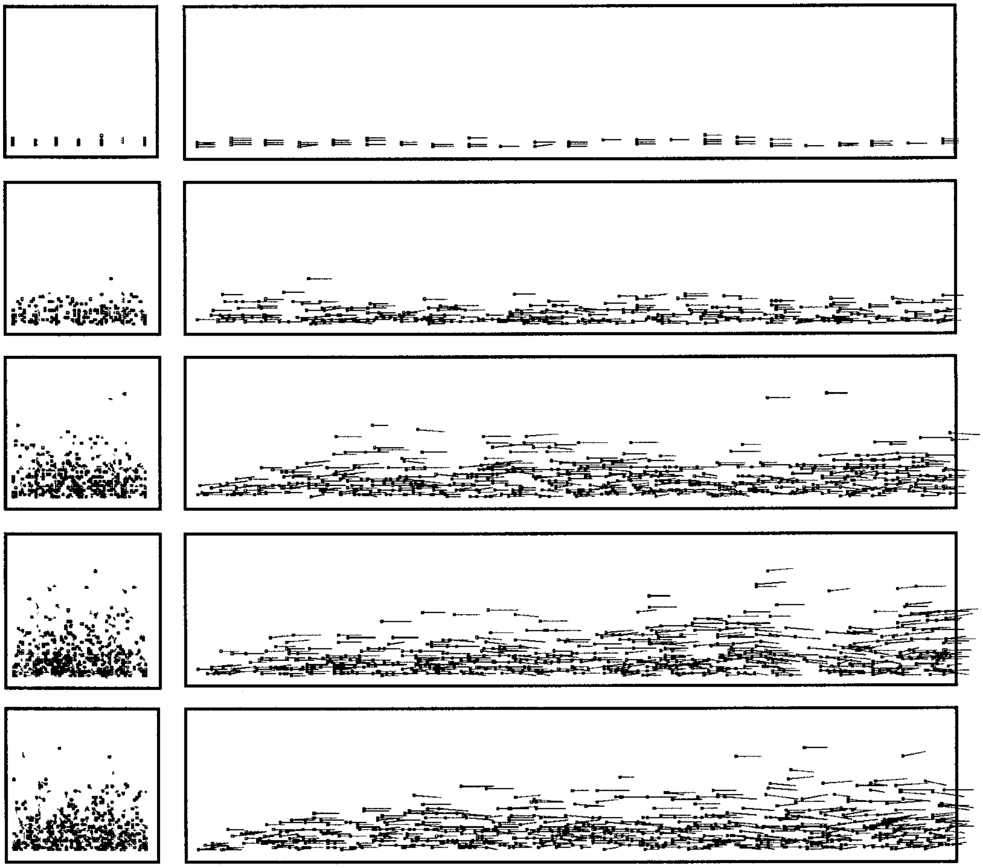

FIG. 2. Rear and side views of vortex element trajectories originated at the bottom wall. From top to bottom: t = 0.1, 0.5, 2.0, 10.5, 21.0.

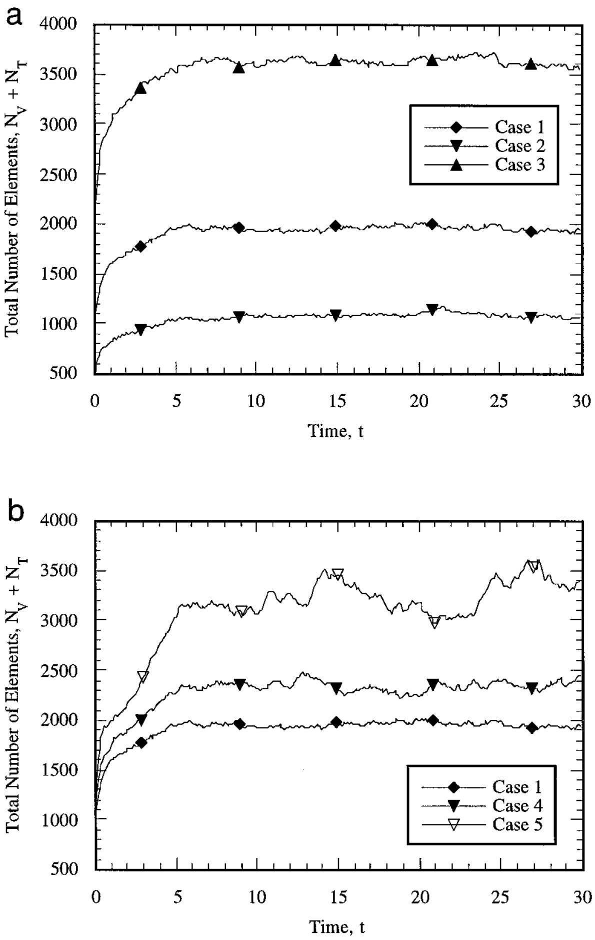

FIG. 3. Total number of vortex elements and tiles versus time.

the total number of vortex elements and tiles. In all cases the number of elements reach their plateau at $t < 10$, signifying the end of the transient between the flow startup and the onset of stationary state. Also note that the transition period to steady state increases with the increase in *BLRP*; however, it is independent of *BLTC* and stays unchanged at $t \approx 6$. It is not obvious why the plots in Fig. 3b display such strong oscillations as *BLTC* is reduced; however, their existence suggests that longer runs are necessary to filter out the effect of the oscillations from the steady state solution. Figure 4 is the log–log plot of the average number of vortex elements and tiles at steady state versus *BLRP* and *BLTC*, where averaging was performed beyond $t = 10$. Note that a doubling of *BLRP* increases the total number of elements in the field by $2^{0.87}$ only, which is roughly equal to $2^{\sqrt{3}/2}$. Similarly, reducing *BLTC* by a factor of 2 increases the number of elements by $2^{0.46}$ or roughly $\sqrt{2}$.

Figure 5 depicts the steady state streamwise velocity profile as a function of *BLRP* at four locations on the $z = 4.5$ surface. The steady state flow field is obtained by averaging the instantaneous solutions over several timesteps. For the current comparison, the results from times 20.0 through 30.0 were sampled at five timestep intervals, and averaged over a total of 20 samples. Recall that increasing the value of *BLRP* is analogous to concentrating more grid points in the vicinity of the walls, and is thus expected to improve the field resolution there. However, increasing *BLRP* from 2 to 8 does not improve the velocity field within the domain appreciably and, as the figure shows, the profiles are almost indistinguishable at $z = 4.5$. Previous numerical experiments with two- and three-dimensional boundary layer studies suggest that *BLRP* values as large as 20 may be necessary to obtain sufficiently converged solutions [22, 50]. However, considering the commensurate increase in the number of elements and tiles, which is approximately 8100 according to the results in Fig. 4, further increase in *BLRP* is not computationally justifiable. It must be added here that the velocity profiles obtained in Case 9 are also very close to those of Case 1, and increasing the tile and boundary element resolutions improved the duct flow field only slightly.

Figure 6 depicts the convergence as a function of *BLTC* for the steady state streamwise velocity profile, at four locations on the $z = 4.5$ surface. Time averaging of the velocity profiles is performed from 20.0 through 30.0 for Case 4, and from 20.0 through 50.0 for Case 5. It is clear from the figure that *BLTC* has a significant influence on convergence. Indeed, for the duct problem, it is the most effective parameter in Table I for obtaining fast convergence. Interestingly, all previous parametric studies have concentrated on the variation of the tile resolution, *BLRP* and the timestep [22, 50], setting *BLTC* equal to 1.5 based on statistical reasoning. In particular, the reduction in the timestep has traditionally been assumed to improve the tile solution by (1) increasing the time integration accuracy during the convection and diffusion steps, and (2) decreas-

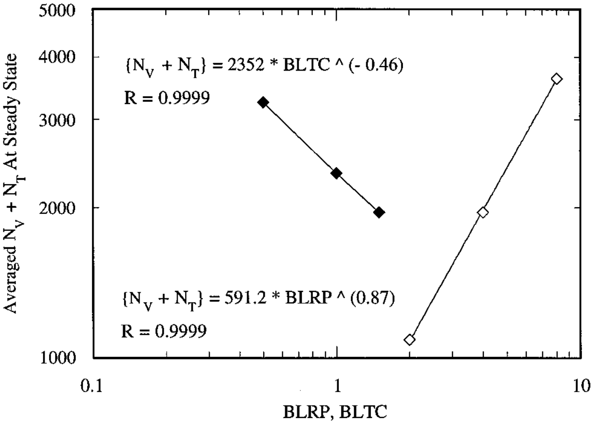

FIG. 4. Stationary state number of vortex elements and tiles versus *BLRP* and *BLTC*.

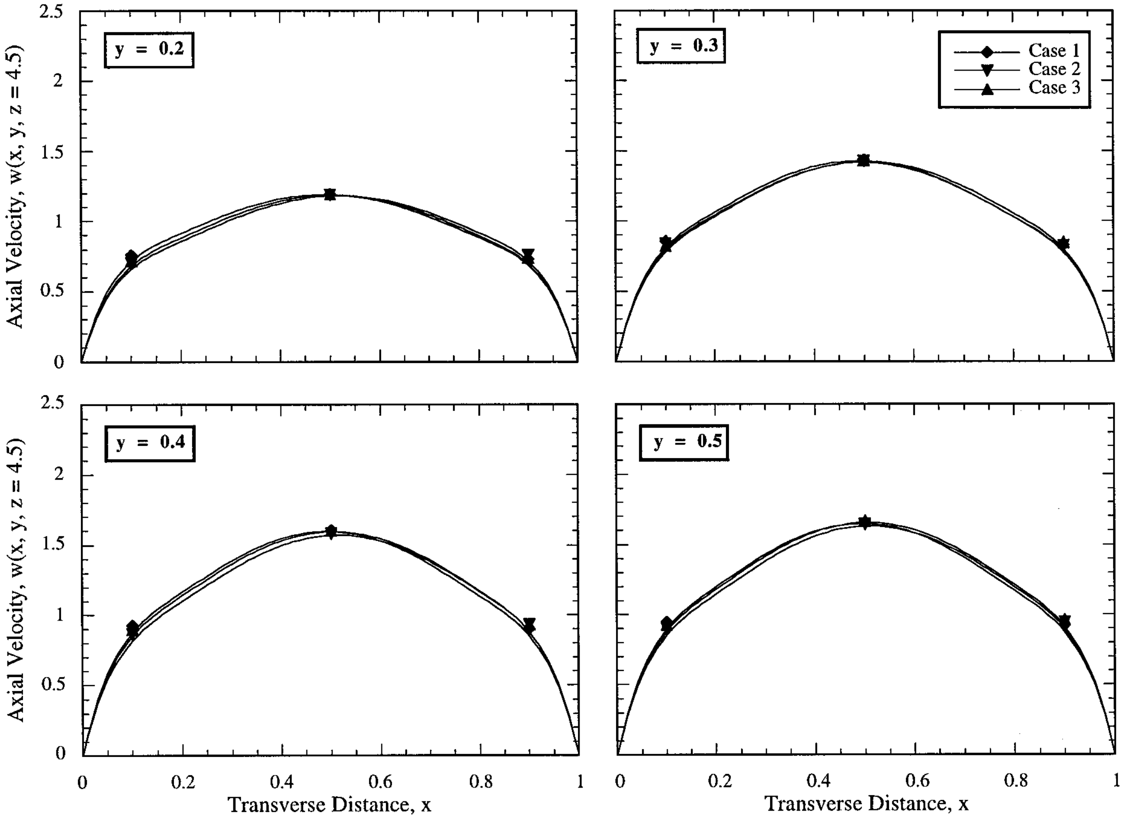

FIG. 5. Effect of *BLRP* on the streamwise velocity at $z = 4.5$.

ing the noise level due to the random walk simulation of diffusion. However, while these assessments are correct, our experiments with *BLTC* suggest that the primary contribution of the timestep reduction to the solution accuracy is the reduction in the numerical boundary layer thickness, which in turn improves the validity of the assumptions behind the Prandtl approximation.

To verify our conclusion above, we devised Case 8 such that its numerical boundary layer thickness is almost equal to that of Case 5. Note that, compared to the baseline case, the timestep in Case 5 is kept unchanged while its *BLTC* is reduced; whereas the timestep in Case 8 is reduced as its *BLTC* is kept unchanged. Figure 7 shows the total number of vortex elements and tiles as a function of time, for Cases 5, 7, and 8. The statistically good match between the element histories of Cases 5 and 8 is a first-order indication of the match between the two flow fields. Figure 8 depicts the steady state streamwise velocity profiles at four locations on the $z = 4.5$ surface, obtained by Cases 5 and 8 and the exact solution. The time averaging of the velocity profiles for Case 8 is performed from 20.0 through 30.0. First, note that the simulation results compare very well with the exact solution, considering that only three tiles are used per width of the duct to satisfy the no-slip boundary condition, and only four boundary elements per width to satisfy the no-flux condition. Second, note that results from Cases 5 and 8 are similar, proving our suggestion that, all else being equal, convergence is primarily controlled by the numerical boundary layer thickness and not the timestep or *BLTC*, individually. Also note that as a domain decomposition approach, in which essentially different equations are integrated in different domains—the Navier–Stokes equations in the interior and the Prandtl approximation near the solid walls—it is important to establish the bound on the domain within which the latter can replace the former. Figure 5 shows that no matter how refined the resolution of the vorticity in the near wall zone is, if the extent of this zone normal to the wall is "large," the error cannot be forced below a certain rather large bound. On the other hand, Fig. 6 shows that even with relatively coarse discretization of the near wall vorticity, the error is reduced substantially as the numerical boundary layer (or the domain within which the Navier–Stokes equations are approximated using the Prandtl-type equations) is made smaller! This is also in agreement with an error analysis of the discretization which would be used to show consistency. The total error in the approximation is a function of the boundary layer thickness (*BLTC*) and the vorticity resolu-

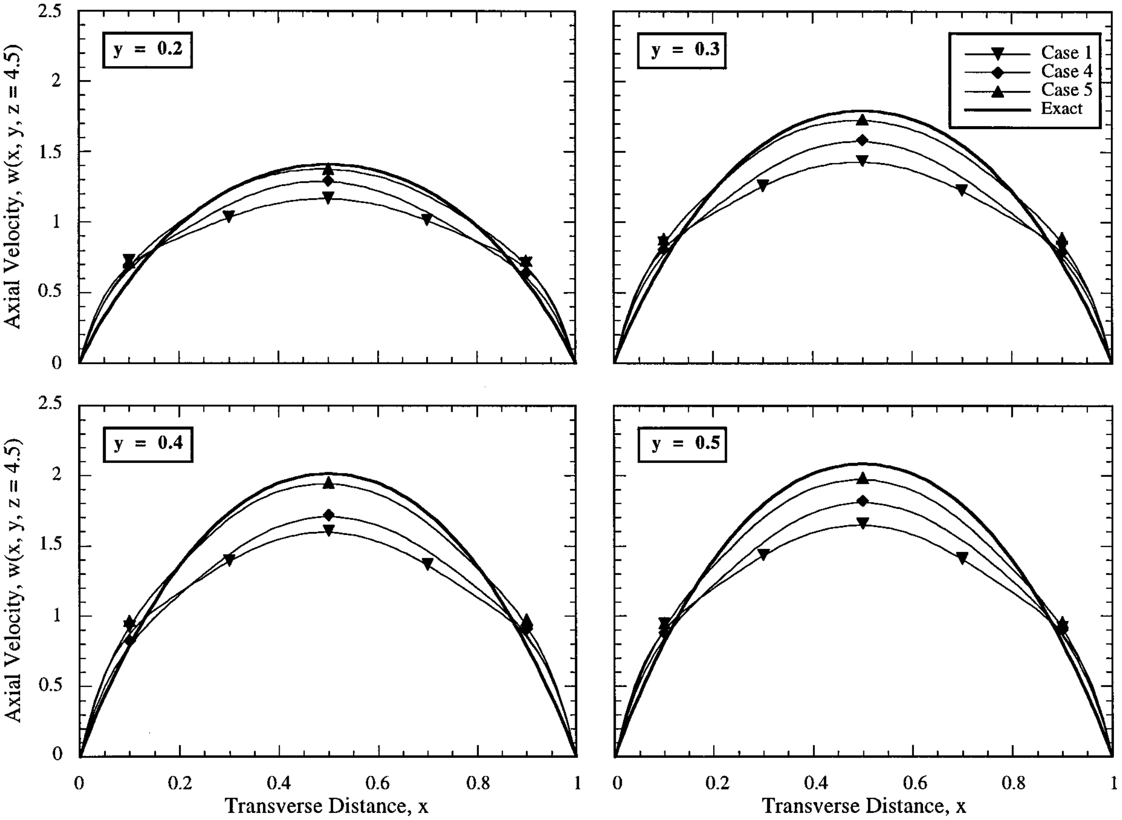

FIG. 6. Effect of *BLTC* on the streamwise velocity at $z = 4.5$.

tion parameter (*BLRP*). As the second error is reduced by increasing *BLRP*, the total error would be bounded by the first which is controlled by *BLTC*.

Figure 9 represents the effect of sample size on the steady state velocity profile at four locations on the $z = 4.5$ surface for Case 5. For all three time periods—10, 20, and 30—the sampling was performed at every five timesteps. Note that, as the sample size is increased, the velocity profiles converge monotonically toward a steady state solution. Since symmetry is not enforced at the vorticity evolution level, some asymmetry is observed in the profiles which improves with increasing sample size. The absence of an explicit method to guarantee the solenoidality of vorticity may be another potential source of asymmetry in the velocity profile.

Cases 6 and 7 were devised to compare the proposed fully developed exit boundary condition with the standard boundary condition applied in Cases 1 and 5, respectively. First, note that, as shown in Fig. 7, the time histories of the number of particles and tiles are in good statistical agreement for Cases 5 and 7, which is a first-order indication that the velocity fields are similar in the two cases. A survey of the velocity profiles at selected streamwise stations verifies the similarity of the flow field up to $z = 5.0$, beyond which the two solutions diverge from each other. Figure 10 shows the effect of the exit boundary condition on the exit velocity profiles obtained from Cases 5, 7, and 8. The velocity profiles from Cases 5 and 8 indicate that reducing the timestep improves the exit velocity only slightly. On the other hand, the relative accuracy of the

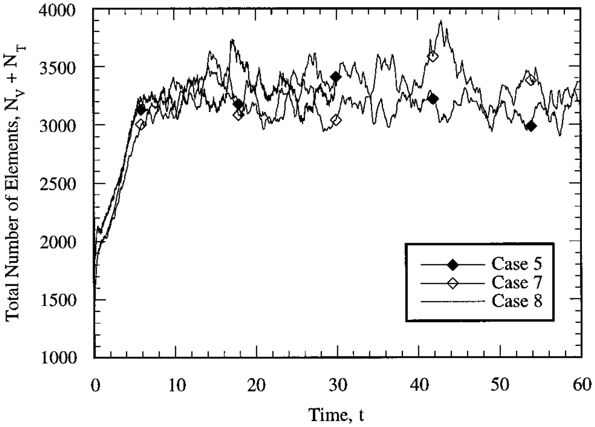

FIG. 7. Effect of the numerical boundary layer thickness on the number of vortex elements and tiles.

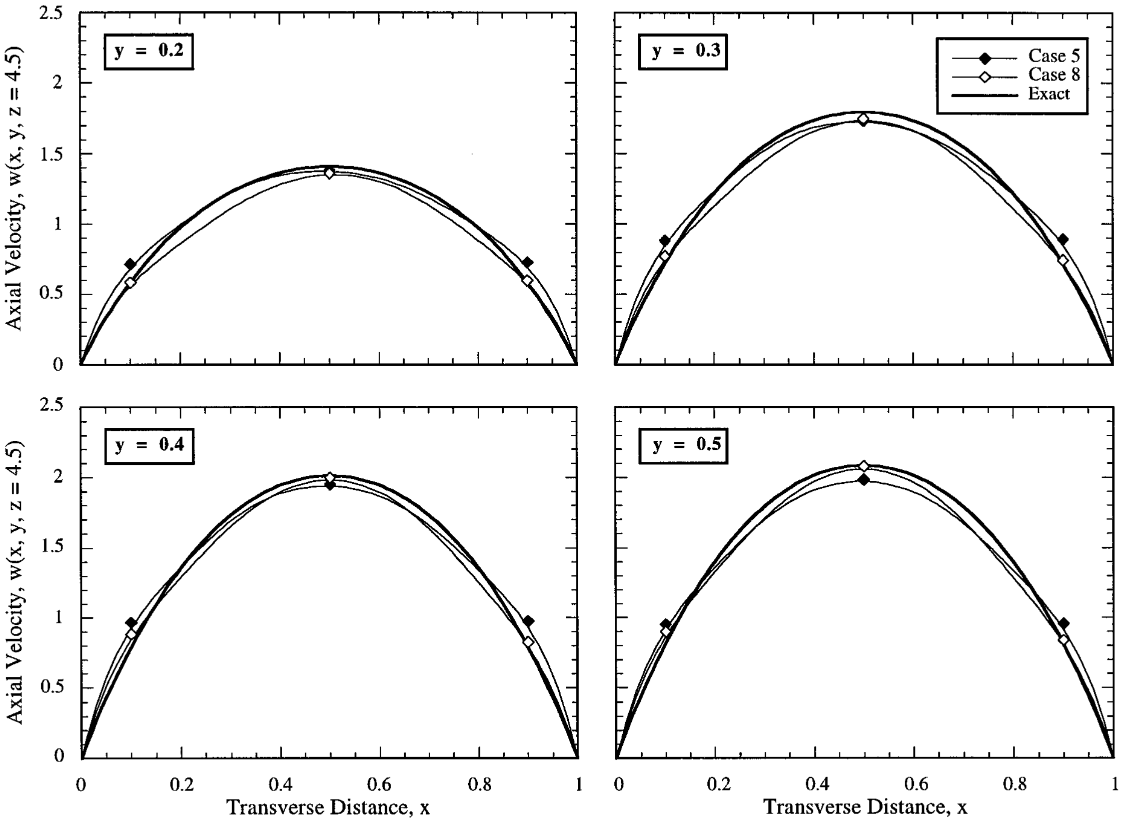

FIG. 8. Comparison of the streamwise velocity profiles obtained by the current method and the exact solution.

fully developed exit boundary condition is clearly evident in the figure. Consequently, the benefit of applying our proposed exit condition is a 30% reduction in the length of the computational domain for this particular geometry, and a correspondingly reduced computational cost. (Results obtained from the comparison of Cases 1 and 6 led to the same conclusions and will not be presented in this paper.)

The test results presented so far establish the numerical convergence of the method in the case of a rather simple, but well defined flow. They show the impact of the time-step, the resolution of the wall discretization, and the resolution of the discretized vorticity within the numerical boundary layer on the accuracy of the solution. This is significant, given that for more complex three-dimensional high Reynolds number flows it is rather difficult to establish such convergence. In the next section, we demonstrate the performance of the method in a more complex flow.

4.2. Flow over a Cube

In this section, we present results from an example of impulsively started flow over a cube confined in a duct with square cross section. The detailed analysis of the simulation of this flow is beyond the scope of this paper; here we demonstrate some of the salient features of the vortex-boundary element method and its ability to capture three-dimensional vortical flow structures that evolve due to separation at sharp edges. The Reynolds number based on the cube side, a, and uniform inlet velocity, U_{in}, was set at 100. The length and velocity scales were normalized by a and U_{in}, respectively. The cube and the duct were aligned concentrically, and their walls were positioned normal to each of the three coordinate directions. The origin was set at the cube center, with the z coordinate pointing in the streamwise direction toward the exit surface. The dimensions of the confining duct in the $x:y:z$ directions were set at 16:16:8, respectively, and the inlet plane was placed at $z = -2.5$. To discretize the potential flow, 4×4 and 10×10 uniformly distributed boundary elements were used on each of the six faces of the confining duct and the cube, respectively; 8×8 uniformly distributed tiles were used on each face of the cube to satisfy the no-slip boundary condition. For computational simplicity, we considered free-slip boundary condition for the confining duct; therefore, no tiles were generated on the duct walls. Additionally, we set $\Delta t = 0.05$, $BLTC = 1.5$, and $BLRP = 3$. The inlet velocity boundary condition was maintained at $U_{\text{in}} = 1.0$. The standard exit boundary condi-

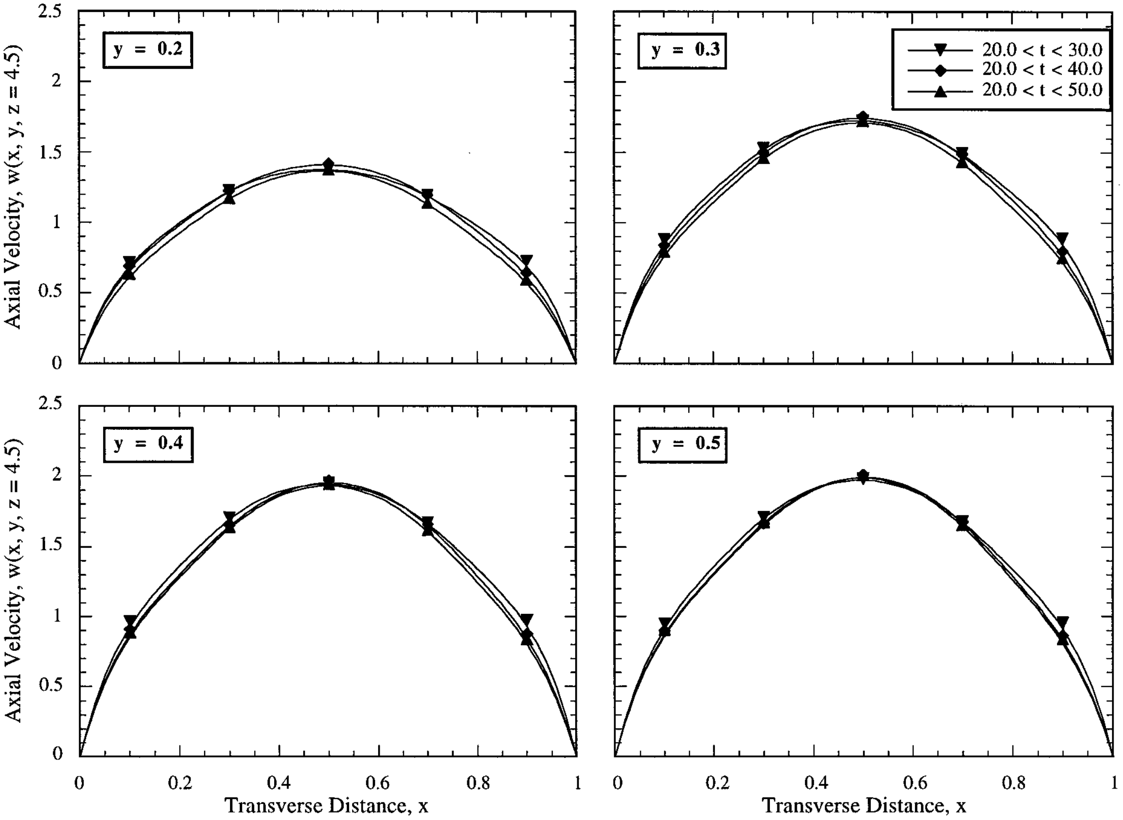

FIG. 9. Effect of sample size on the stationary state velocity profile at $z = 4.5$.

tion was applied for this problem. The no-flux boundary condition was enforced on all cube walls, as well as the side walls of the duct. At $t = 0^+$, a uniform velocity, $U_{\text{in}} = 1.0$, was applied throughout the field to start the impulsive motion.

Figure 11 depicts a representative sequence from the initial stages of flow development within a slice of volume with 0.25 thickness around the plane of symmetry at $x = 0.0$—the plane of symmetry of the cube. The solid circles denote the location of the vortex elements and the line segments originating from them represent the velocity of the elements, normalized by the instantaneous maximum speed in the field. Using the instantaneous maximum in the normalization helps identify the *relative* strength of various flow regions and processes in the domain in *each* time frame. Notice once again that the vortex elements near the cube surfaces are separated from them by fixed-size thin blank spaces. These are the *numerical* boundary layers where the Prandtl approximation is used to evaluate the evolution of the wall-generated vortex tiles. (The tiles are not shown in the figure to avoid clutter.) On the other hand, the envelopes of the vortex elements spanning over the upper and lower cube surfaces demonstrate the initiation and growth of the *physical* boundary layer. The latter reaches a quasi-stationary state by $t = 2.0$ and is symmetric with respect to the plane of symmetry of the cube. Note that the sharp edges of the cube do not cause flow separation at the upstream side of the cube. This is verified by a high resolution finite-difference simulation of flow over a cube [51, 52], as well as a finite-volume simulation of flow over a square rod [18], at the same Reynolds number.

The formation of a very weak near-field wake behind the cube is evident in Fig. 11. The length of the aft-end recirculation bubble at $t = 2.0$ is approximately equal to half the cube width and corroborates well with the value of 0.52 obtained by Raul *et al.* [52]. Figure 12 depicts the corresponding mean centerline axial velocity behind the cube, obtained by the present method and the finite-difference simulation by Raul and Bernard [51]. Considering the relatively low resolution of the present simulation, the two profiles compare very well. In Fig. 11, notice the presence of a mild asymmetry in the wake, especially at $t = 2.0$. This is caused by the random walk perturbation, which generates a random noise-level disturbance in the field. Although it is not shown here, beyond $t = 2.0$, the inherent instability of the wake gives rise to further growth of the observed asymmetry of the aft-end bubbles and develops an unsteady wake with an approximate period of 7 [24].

The predicted stationary state drag coefficient of 2.14

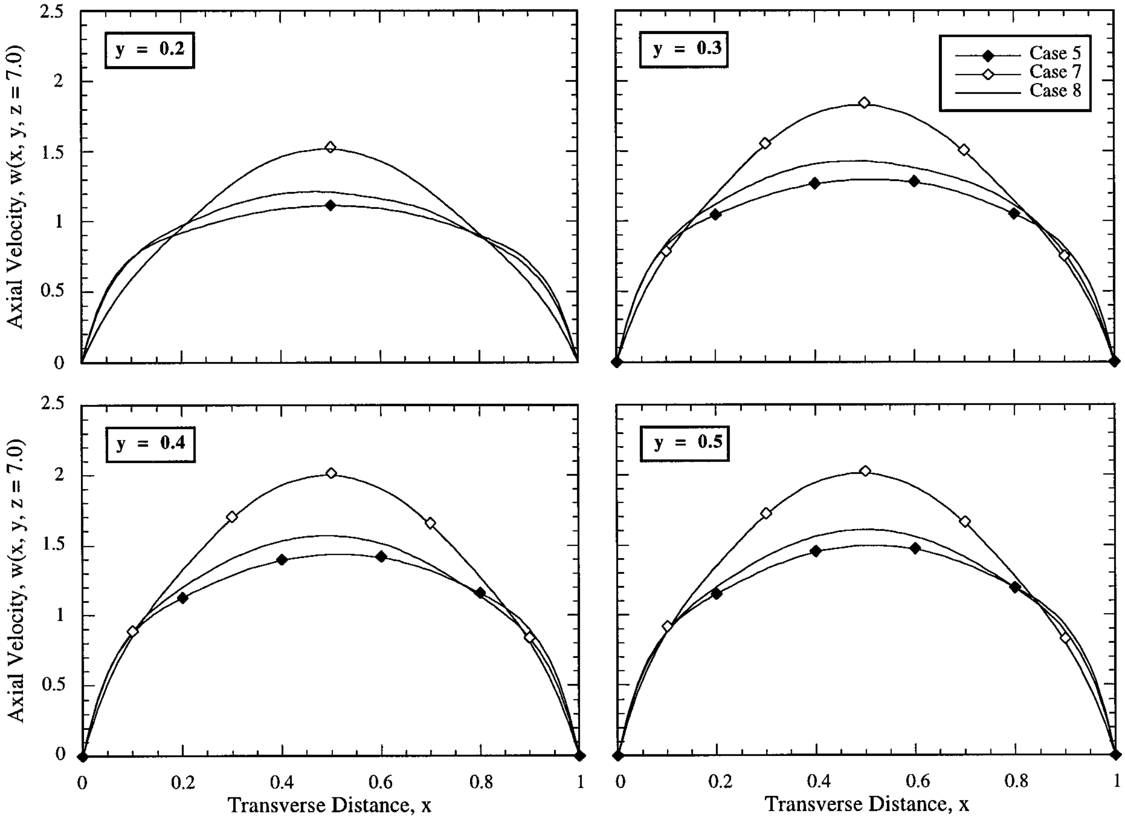

FIG. 10. Effect of the exit boundary condition on the velocity profile at the exit surface.

compares well with the value of 1.95 obtained by Raul *et al.* [52]. We obtained the drag by applying the integral form of the momentum balance between the inlet and the $z = 2.5$ surfaces. The pressures at the two stations were assumed to be equal in this computation, since, as shown in Fig. 12, the velocity behind the cube is fairly well recovered at $z = 2.5$. Details of these computations will be published elsewhere.

Figure 13 depicts the two-dimensional, stationary state velocity vector projections on selected xy planes in the vicinity of the cube, viewed from the rear of the cube into the negative z direction. The flow in the upstream half of the cube, $z < 0$, is essentially in the streamwise direction. In the downstream half of the cube, $z > 0$, small and weak vortical flow structures begin to form at both edges of each face and grow in size and strength in the positive z direction toward the trailing edge of the cube. The development of the two counterrotating vortical flow structures near each edge of the cube is a manifestation of the existence of a pair of oppositely signed streamwise vortices there. Notice, by comparing the scale of the vortical flow vectors with that at the upstream face of the cube, that the rotations are very weak and that the transverse velocity components are an order of magnitude smaller than the streamwise component. Raul *et al.* [52] do not provide information on the structure of the flow near the cube edges at $R_e = 100$. However, Raul and Bernard [51] report the existence of three-dimensional flow structures in the downstream half of the cube edges at $R_e = 2000$, which are in qualitative agreement with our prediction.

5. CONCLUSION

A hybrid random vortex-boundary element method has been developed for the grid-free simulation of unsteady incompressible viscous flow inside three-dimensional domains. In this approach, the velocity–vorticity formulation of the Navier–Stokes equations is adopted. The Lagrangian vortex method is utilized to evaluate the convection and stretch of the vorticity in an unbounded domain, and the random walk method is used to simulate its diffusion. To account for the boundary effects, a potential flow is imposed on the latter such that the proper normal flux boundary condition is satisfied. The boundary element method is used to solve the Neumann problem that defines this potential flow. The no-slip boundary condition is satisfied by generating vortex tiles at the solid walls. Within a thin user-specified region near the boundary, the evolution

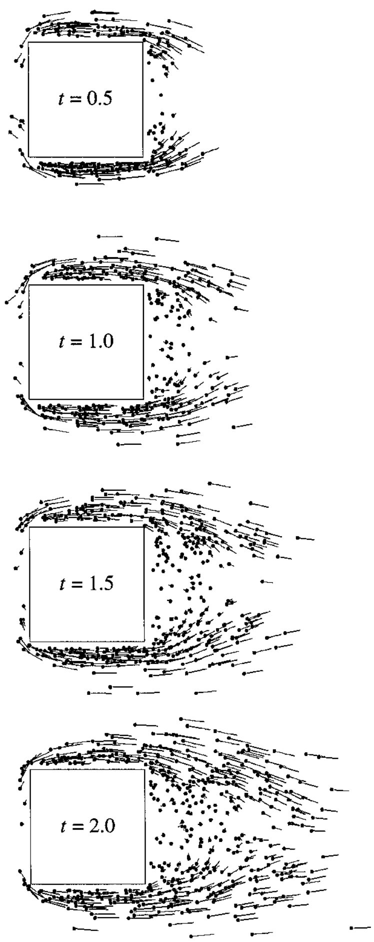

FIG. 11. The initial stages of wake development within a volume of thickness 0.25, with its plane of symmetry lying on the $x = 0$ plane.

of the vortex tiles is prescribed by simplified Prandtl equations. Beyond this region, the tiles are converted to spherical vortex elements.

The accurate evaluation of the potential velocity used to impose the normal flux boundary condition, and its gradients near the walls is very critical to the success of three-dimensional vortex simulations. To this end, we have developed a boundary element method to obtain the solution of the ill-posed internal Neumann problem, as well as to evaluate the potential velocity and its gradients near the boundary, accurately. In addition, the solution algorithm for the Prandtl equation was extended to allow the interaction among unequally sized tiles. Furthermore, a new formulation was developed for imposing the fully developed flow condition at the exit plane.

The numerical convergence and the accuracy of the method were demonstrated using the example of flow in a duct with square cross section at $R_e = 100$, by comparing the predictions with the exact solution. For the range of

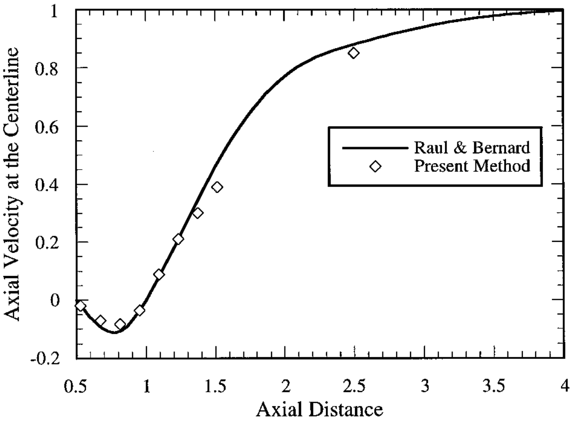

FIG. 12. The axial velocity behind and on the centerline of the cube.

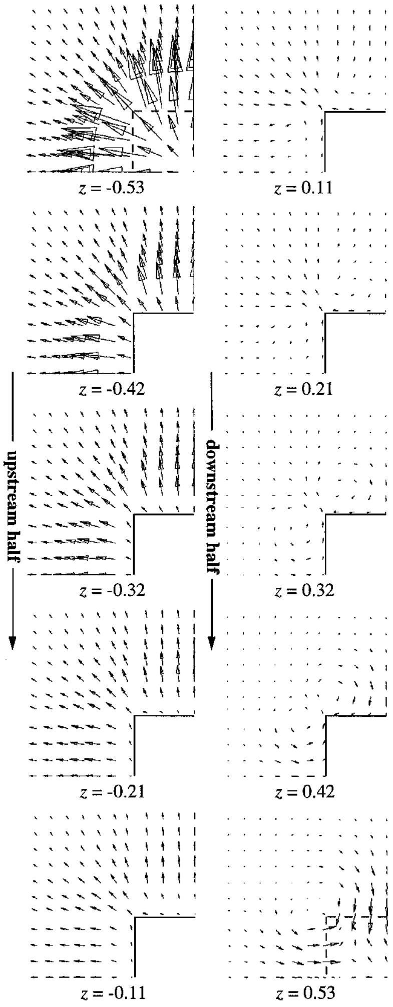

FIG. 13. Stationary state velocity projections on selected xy planes.

parameters tested, the fully developed velocity profiles near the exit were shown to be insensitive to the tile resolution in the direction normal to the wall. However, the accuracy was shown to improve dramatically as the numerical boundary layer thickness was reduced. This suggests the need to conduct more rigorous tests to optimize the utility of the Prandtl approximations near the boundary. The proposed fully developed exit boundary condition was shown to yield substantially more accurate velocity profiles at the exit as compared to the standard approach. As a result, computations can now be performed without the need to unnecessarily extend the numerical domain.

In addition to the duct flow case, selected results from the simulation of an impulsively started cube at $R_e = 100$ were presented. The predicted centerline axial velocity behind the cube and the stationary state drag coefficient were shown to agree well with an available high resolution finite-difference simulation. This establishes the success of the method in handling flow separation and the accompanying dynamics around sharp corners. Moreover, the results show the formation of weak streamwise vortices with opposite signs around the streamwise corners of the cube.

ACKNOWLEDGMENTS

This project was funded by Ford Motor Company. The numerical experiments were performed on the Cray-YMP at the National Center for Supercomputing Applications, and on the Cray-C90 at the Pittsburgh Supercomputing Center.

REFERENCES

1. C. R. Anderson and C. Greengard, On vortex methods, *SIAM J. Numer. Anal.* **22**(3), 413 (1985).
2. P. K. Banerjee and R. Butterfield, *Boundary Element Methods in Engineering Science* (McGraw–Hill, New York, 1981).
3. G. K. Batchelor, *An Introduction to Fluid Dynamics*, (Cambridge Univ. Press, London, 1967).
4. J. T. Beale, A convergent 3-D vortex method with grid-free stretching, *Math. Comput.* **46**(174), 401 (1986).
5. J. T. Beale and A. Majda, Rates of convergence for viscous splitting of the Navier–Stokes equations, *Math. Comput.* **37**(156), 243 (1981).
6. J. T. Beale and A. Majda, Vortex methods. I. Convergence in three dimensions, *Math. Comput.* **39**(159), 1 (1982).
7. J. T. Beale and A. Majda, Vortex methods. II. Higher order accuracy in two and three dimensions, *Math. Comput.* **39**(159), 29 (1982).
8. J. T. Beale and A. Majda, Vortex methods for fluid flow in two or three dimensions, *Contemporary Mathematics, Amer. Math. Soc.* **28,** 221 (1984).
9. J. T. Beale and A. Majda, High order vortex methods with explicit velocity kernels, *J. Comput. Phys.* **58,** 188 (1985).
10. T. F. Buttke, Fast vortex methods in three dimensions, in *Vortex Dynamics and Vortex Methods, Lectures in Applied Mathematics*, Vol. 28, p. 51 edited by C. R. Anderson and C. Greengard, (Amer. Math. Soc., Providence, RI, 1990).
11. T. F. Buttke, Velicity methods: Lagrangian numerical methods which preserve the Hamiltonian structure of incompressible fluid flow, in *Proceedings of the NATO Advanced Research Workshop on Vortex Flows and Related Numerical Methods, Grenoble, France, 1992*, Vol. 395, p. 39, edited by J. T. Beale, G.-H. Cottet, and S. Huberson (Kluwer, Boston/London).
12. A. J. Chorin, Vortex sheet approximation of boundary layers, *J. Comput. Phys.* **27,** 428 (1978).
13. A. J. Chorin, Numerical study of slightly viscous flow, *J. Fluid Mech.* **57,** 785 (1973).
14. A. J. Chorin, Vortex models and boundary layer instability, *SIAM J. Sci. Stat. Comput.* **1**(1), 1 (1980).
15. A. J. Chorin and J. E. Marsden, A *Mathematical Introduction to Fluid Mechanics* (Springer-Verlag, New York/Berlin, 1979).
16. A. J. Chorin, private communication, 1996.
17. R. Cortez, Impulse-based methods for fluid flow, Ph.D. thesis, Dept. of Math., University of California, Berkeley, CA, 1995.
18. R. W. Davis, E. F. Moore, and L. P. Purtell, A numerical-experimental study of confined flow around rectangular cylinders, *Phys. Fluids* **27**(1), 46 (1984).
19. P. Degond and S. Mas-Gallic, The weighted particle method for convection-diffusion equations, Part I: The case of an isotropic viscosity. Part II: The anisotropic case, *Math. Comput.* **53,** 485 (1989).
20. P. Degond and F.-J. Mustieles, A deterministic approximation of diffusion equations using particles, *SIAM J. Sci. Stat. Comput.* **11**(2), 293 (1990).
21. D. Fishelov, A new vortex scheme for viscous flows, *J. Comput. Phys.* **86,** 211 (1990).
22. D. Fishelov, Vortex methods for slightly viscous three-dimensional flow, *SIAM J. Sci. Stat. Comput.* **11**(3), 399 (1990).
23. A. L. Fogelson and R. H. Dillon, Optimal smoothing in function-transport particle methods for diffusion problems, *J. Comput. Phys.* **109,** 155 (1993).
24. A. Gharakhani, 3-D vortex-boundary element method for the simulation of unsteady, high Reynolds number flows, Sc.D. thesis, Dept. of Mech. Engrg., MIT, Cambridge, MA, 1995.
25. A. Gharakhani and A. F. Ghoniem, BEM solution of the 3D internal Neumann problem and a regularized formulation for the potential velocity gradients, *Int. J. Numer. Methods Fluids* **24**(1), 81 (1997).
26. A. Gharakhani and A. F. Ghoniem, Vortex simulation of intake flow in axi-symmetric reciprocating valve/cylinder configurations, in preparation.
27. A. F. Ghoniem, A. J. Chorin, and A. K. Oppenheim, Numerical modelling of turbulent flow in a combustion tunnel, *Phil. Trans. R. Soc. Lond. A* **304,** 303 (1982).
28. A. F. Ghoniem and Y. Gagnon, Vortex simulation of laminar recirculating flow, *J. Comput. Phys.* **68,** 346 (1987).
29. A. F. Ghoniem and K. K. Ng, Numerical study of the dynamics of a forced shear layer, *Phys. Fluids* **30**(3), 706 (1987).
30. A. F. Ghoniem and F. S. Sherman, Grid-free simulation of diffusion using random walk methods, *J. Comput. Phys.* **61,** 1 (1985).
31. J. Goodman, Convergence of the random vortex method, *Commun. Pure Appl. Math.* **40**(2), 189 (1987).
32. L. Greengard and V. Rokhlin, A fast algorithm for particle simulations, *J. Comput. Phys.* **73,** 325 (1987).
33. O. Hald, Convergence of vortex methods for Euler's equations, II, *SIAM J. Numer. Anal.* **16**(5), 726 (1979).
34. O. Hald, Convergence of random methods for a reaction-diffusion equation, *SIAM J. Sci. Stat. Comput.* **2**(1), 85 (1981).
35. T. Y. Hou, A new desingularization for vortex methods, *Math. Comput.* **58**(197), 103 (1992).

36. T. Y. Hou, J. Lowengrub, and M. J. Shelley, The convergence of an exact desingularization for vortex methods, *SIAM J. Sci. Comput.* **14**(1), 1 (1993).

37. S. N. Kempka and J. H. Strickland, A method to simulate viscous diffusion of vorticity by convective transport of vortices at a non-solenoidal velocity, Sandia Report, SAND93-1763, UC-700 (1993).

38. O. M. Knio and A. F. Ghoniem, Numerical study of a three-dimensional vortex method, *J. Comput. Phys.* **86,** 75 (1990).

39. O. M. Knio and A. F. Ghoniem, Three-dimensional vortex simulation of rollup and entrainment in a shear layer, *J. Comput. Phys.* **97,** 172 (1991).

40. O. M. Knio and A. F. Ghoniem, The three-dimensional structure of periodic vorticity layers under non-symmetric conditions, *J. Fluid Mech.* **243,** 353 (1992).

41. P. D. Koumoutsakos, Direct numerical simulations of unsteady separated flows using vortex methods, Ph.D. thesis, Dept. of Mech. Engrg., Caltech, Pasadena, CA, 1993.

42. Sir H. Lamb, *Hydrodynamics*, (Dover, New York, 1945).

43. A. Leonard, Vortex methods for flow simulation, *J. Comput. Phys.* **37,** 289 (1980).

44. D.-G. Long, Convergence of the random vortex method in two dimensions, *J. Amer. Math. Soc.* **1**(4), 779 (1988).

45. L.-F. Martins and A. F. Ghoniem, Vortex simulation of the intake flow in a planar piston-chamber device, *Int. J. Numer. Methods Fluids* **12,** 237 (1991).

46. F. Milinazzo and P. G. Saffman, The calculation of large Reynolds number fluid flow using discrete vortices with random walk, *J. Comput. Phys.* **23,** 380 (1977).

47. H. N. Najm and A. F. Ghoniem, Numerical simulation of the convective instability in a dump combustor, *AIAA J.* **29**(6), 911 (1991).

48. Y. Ogami and T. Akamatsu, Viscous flow simulation using the discrete vortex model—The diffusion velocity method, *Computers & Fluids* **19**(3/4), 433 (1991).

49. V. I. Oseledets, On a new way of writing the Navier–Stokes equation: The Hamiltonian formalism, *Commun. Moscow Math. Soc.* **44,** 210 (1989).

50. E. G. Puckett, A study of the vortex sheet method and its rate of convergence, *SIAM J. Sci. Stat. Comput.* **10**(2), 298 (1989).

51. R. Raul and P. S. Bernard, A numerical investigation of the turbulent flow field generated by a stationary cube, *J. Fluids Engrg.* **113,** 216 (1991).

52. R. Raul, P. S. Bernard, and F. T. Buckley, Jr., An application of the vorticity-vector potential method to laminar cube flow, *Int. J. Numer. Methods Fluids* **10,** 875 (1990).

53. S. G. Roberts, Accuracy of the random vortex method for a problem with non-smooth initial conditions, *J. Comput. Phys.* **58,** 29 (1985).

54. H. Rouse, *Advanced Mechanics of Fluids*, (John Wiley & Sons, Inc., New York/London/Sydney, 1965).

55. H. Schlichting, *Boundary Layer Theory*, (McGraw-Hill, New York, 1968).

56. J. A. Sethian and A. F. Ghoniem, Validation of the vortex method, *J. Comput. Phys.* **74,** 283 (1988).

57. V. Sladek and J. Sladek, Non-singular boundary integral representation of potential field gradients, *Int. J. Numer. Methods Eng.* **33,** 1181 (1992).

58. V. Sladek and J. Sladek, Non-singular boundary integral representation of stresses, *Int. J. Numer. Methods Eng.* **33,** 1481 (1992).

59. G. S. Winckelmans and A. Leonard, Improved vortex methods for three-dimensional flows, in *Proceedings of the SIAM Workshop on Mathematical Aspects of Vortex Dynamics, Leesburg, Virginia, 1988*, edited by R. E. Caflisch (SIAM, Philadelphia, 1989).

60. G. S. Winckelmans and A. Leonard, Contributions to vortex particle methods for the computation of three-dimensional incompressible unsteady flows, *J. Comput. Phys.* **109,** 247 (1993).