AIAA JOURNAL
Vol. 38, No. 5, May 2000

Solution of Two-Dimensional Vorticity Equation on a Lagrangian Mesh

Stephen A. Huyer* and John R. Grant†
U.S. Naval Undersea Warfare Center, Newport, Rhode Island 02841

A novel, vorticity-based solution methodology has been developed to compute unsteady flow past bodies. Vorticity is evolved on a set of points, and the vorticity in the remainder of the field is determined by linear interpolation. Interpolation is accomplished by Delaunay triangularization of the points in the field. Triangulation of the vorticity field provides a basis to integrate the vorticity to compute the velocity. Nodal connectivity from the triangularization also provides a list of the neighboring points that are used to construct a second-order least-squares fit of the vorticity. First- and second-order spatial derivatives can then be computed based on this polynomial fit. Surface vorticity on the body is computed to satisfy the no-slip boundary condition and is introduced into the flow via diffusion. A diffusion transport velocity was derived to account for spatial movement of the vorticity due to viscous diffusion. The points are advected by the sum of the induced velocity (computed from the Biot–Savart integral) and the diffusion velocity. The remaining diffusion term includes a form of the Laplacian and is computed directly. This solution scheme was found to be stable as applied to the problem of impulsively started flow about a circular cylinder and flat plate. Comparisons with experimental and the Blasius boundary-layer solution for a flat plate were used to demonstrate the effectiveness of this method.

Nomenclature

Cd = drag force coefficient, $2F_{\text{Drag}}/\rho U_\infty^2$
c = flat plate chordlength
d = cylinder diameter
l_d = diffusion length scale, $\sqrt{4\nu\Delta t}$
Re = Reynolds number, $U_\infty d/\nu$
t = nondimensional time, $U_\infty t/d$
U_∞ = nondimensional freestream velocity
$\boldsymbol{u}$ = velocity
$\boldsymbol{v}$ = diffusion velocity
γ = vortex sheet strength
Δt = nondimensional time step
ν = kinematic viscosity
σ = nodal source strength
Ω = nodal vorticity amplitude
ω = vorticity

Introduction

INCOMPRESSIBLE, unsteady fluid flow can be characterized by the instantaneous vorticity field. For cases of flow past bodies, the vorticity distribution in the boundary layer and wake determines the characteristics of the entire flowfield. Most vorticity-based methods are inherently unsteady and are a straightforward approach to solving these types of problems. This technique utilizes a velocity–vorticity formulation by taking the curl of the Navier–Stokes momentum equation to solve for these flow variables on a Lagrangian mesh. The solution methodology has distinct advantages over traditional finite difference methods solving the pressure–velocity formulation. The method is grid free, is naturally adaptive to coherent vortex structures, and is subject to little numerical diffusion. The use of a Lagrangian mesh allows for the straightforward treatment of moving surfaces and does not require incorporation of additional terms due to noninertial reference frames. In addition, multiple bodies are included in a straightforward manner due to the Lagrangian nature of the application.

The traditional vortex method[1] describes the vorticity field by means of isotropic elements or blobs that have a strength that depends only on distance from their center; a frequently used strength distribution is the Gaussian. Careful comparison with theoretical and experimental data of two-dimensional calculations using isotropic blobs of uniform size have been reported by Sethian and Ghoniem[2] for a backward-facing step. High-resolution computations for flow past a cylinder have been conducted by Koumoutsakos and Leonard[3] and Subramaniam[4] and represent the standard for using blob methods to compute unsteady flow past surfaces. The original method has the feature that the identity and location of neighboring blobs are not needed to compute the Biot–Savart integral (which determines the velocity), and so the algorithm for this computation is simple. Later work (e.g., Ref. 3) typically employs accelerated methods, for example, the work of Greengard and Rokhlin[5] and Strickland and Baty,[6] to avoid an order N^2 calculation (where N is the number of elements). Even with these elaborations, the simplicity of an approach based on the traditional method remains attractive.

There remain significant difficulties when applying vortex blobs to flows past a surface, however. One is that near the surface, the blob vorticity distribution (a Gaussian) actually penetrates the surface so that a finite value of vorticity is on the inside of the surface. This can cause problems when computing the boundary conditions. Another is that, without sufficient overlap of the blob radii, the computed velocity and associated vorticity field will be extremely noisy. Figure 1 shows the velocity distribution at the surface of a circular cylinder near the top of the cylinder. At an 80-deg cylinder angle, the point is just upstream of the top of the cylinder, and at 100 deg, the point is just downstream. The normal and tangential velocities are typical of what will be computed to satisfy the no-slip, no-flux boundary conditions and include the freestream and field vorticity contributions only. A direct comparison is made between computations using vortex blobs and the present method using triangularized vorticity elements. For the blob cases, radii are set so that the blob overlap is 1.0 or 2.0. As can be seen, the blob method is relatively noisy compared with the smooth velocity distribution of the triangular elements. The fluctuations decrease as overlap is increased to 2.0 (which is considered standard for blob methods) but can still be significant. In our method, which will be presented later, the vorticity in the field is piecewise continuous and is linearly interpolated between points in the field. No overlap of elements is necessary to maintain a smooth velocity and vorticity field. In contrast, blob methods rely on a superposition of the blob functions to determine the field vorticity and subsequent velocity distribution. If sufficient overlap of the functions is not maintained, a noisy velocity distribution can result causing poor satisfaction of the boundary conditions

Received 10 August 1998; revision received 12 October 1999; accepted for publication 12 October 1999.
*Mechanical Engineer, Hydrodynamics Branch. Member AIAA.
†Physicist, Hydrodynamics Branch.

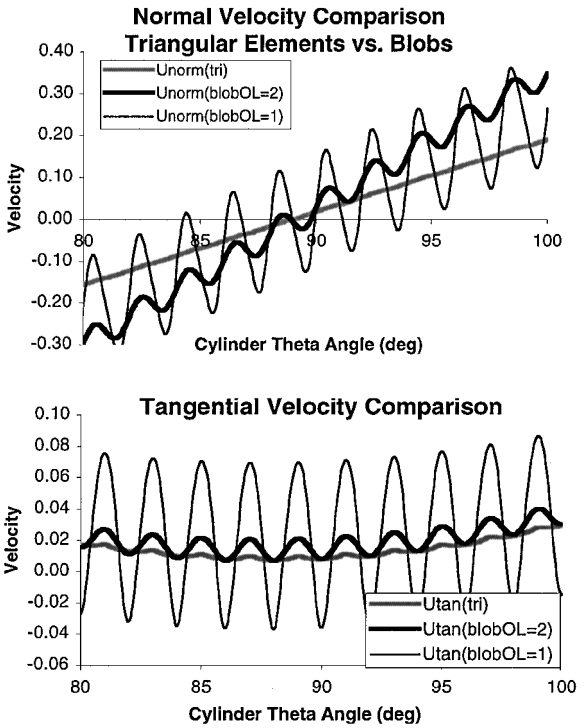


Fig. 1 Normal and tangential surface velocity distribution from cylinder angle = 80–100 deg covering 10 panel locations and 250 points comparing blob vs triangularized vorticity methods and blob overlap of 1.0 and 2.0.

and a diverging solution as seen in Fig. 1. This overlap can be very difficult to maintain, especially near the surface, as the blobs are advected by the flow. For this reason, most blob methods (e.g., Ref. 3) employ periodic interpolation of the blob strength onto a regular grid to maintain the overlap most efficiently in order to avoid noisy boundary conditions. This can lead to artificial numerical diffusion, however, and still does not solve the problem of vorticity penetration into the surface due to the blob function.

Another difficulty with blob methods is the wide range of scales that emerge during the evolution of flows past a surface. An example is the anisotropy of the vorticity distribution in a thin boundary layer. For these cases, the gradient of vorticity in the normal direction is much greater than that in the tangential direction. This presents another compelling motivation for developing an algorithm not based on elements of uniform size but one in which the size and shape of the elements adapt to the local spatial distribution of the vorticity field. Blobs, whatever their size, have a constant radius and are, therefore, isotropic in nature. For thin boundary layers, anisotropic elements are desired.

This point is illustrated by the following calculations for the number of elements required to resolve a two-dimensional laminar flat plate boundary layer for isotropic and anisotropic assumptions applied to the point spacing. Let L be the plate length and let $\delta(x)$ be the boundary-layer thickness,[7] where $\delta(x) = 5\sqrt{(vx/U_\infty)}$. In all of the calculations we let the elements be placed on an approximately rectangular grid covering the area $0 < x < L$, $0 < y < \delta(x)$.

First, suppose the boundary layer is to be resolved by elements of uniform size. To adequately resolve the boundary layer, the point spacing in the normal direction must be the same as in the tangential direction. It can be shown that the number of points required to resolve the boundary layer depends on the total area of the boundary layer. If the resolution in the normal directions is defined by M points, it can be shown that the number of points required to resolve the entire boundary layer is

$$N = \tfrac{2}{375} M^4 Re_L^{\frac{3}{2}}$$

For the very modest values $M = 10$ and $Re_L = 1 \times 10^3$, $N = 1.7 \times 10^6$ elements; for more realistic values of $M = 25$, $Re_L = 1 \times 10^6$, we have $N = 2.1 \times 10^{12}$. This number is large due to the thinness of boundary layer near the leading edge of the plate and the requirement that elements of uniform size be used to resolve the entire boundary layer. This point is further illustrated by the high-resolution runs past a circular cylinder conducted by Koumoutsakos and Leonard.[3] At the end of their calculation, 3×10^5 blobs were present in the flow for $Re = 3 \times 10^3$. For $Re = 9.5 \times 10^3$, well over 1×10^6 blobs were used.

Now, suppose the rows of elements are spaced in the streamwise direction such that the variation of vorticity between neighboring rows is comparable to the variation in vorticity between neighboring elements in the normal direction; this distribution could be the goal of an adaptive distribution of elements for this flow. Again, if M is the number of rows needed to resolve the boundary layer in the normal direction, it can be shown that the total number of points needed to resolve the entire boundary layer is

$$N = M \log\left[(M/5) Re_L^{\frac{1}{2}}\right] \Big/ \log[M/(M-1)]$$

For the parameters $M = 10$, $Re_L = 1 \times 10^3$, we have $N = 400$ elements; for the combination $M = 25$, $Re_L = 1 \times 10^6$, we have $N = 5300$ elements. Adaptive distribution of the elements reduces the number of elements required to resolve the boundary layer to manageable values. The elements in this last example display a strong spatial anisotropy, reflecting that of the flow.

Inclusion of the wake of the plate magnifies the penalty paid by resolving the flow with isotropic elements all of the same size. Although details of the preceding calculations might be adjusted to diminish the disparity in the number of elements required for the respective cases (e.g., beginning the resolution of the boundary layer farther from the leading edge), the qualitative message remains clear. Additionally, the number of isotropic, uniformly sized elements needed to resolve properly three-dimensional flow is correspondingly larger than the two-dimensional example here; we recall the well-known scaling of $Re^{9/4}$ for the number of uniformly spaced points required to resolve the Kolmogorov scales of turbulent flow.

Recognizing the usefulness of anisotropic elements, Marshall and Grant[8,9] introduce an anisotropic blob for use in vortex methods. However, tests in calculations using this form have not been fully satisfactory, due to a difficulty in maintaining a smoothly resolved vorticity field in a fully Lagrangian context. Huyer and Grant[10] and Huyer et al.[11,12] utilize anisotropic rectangular vorticity elements that adapt to the local anisotropy of the flow. This element geometry was shown to be quite useful in solving for unsteady separated flows past circular cylinders and pitching airfoils. However, noisiness in the numerical data due to overlapping elements and stability issues for attached flows appear to limit the engineering and research applications of this element.

The work presented here is motivated by considerations such as those presented concerning the need for anisotropic elements for calculations of complex flows and by the experience gained using the anisotropic elements just mentioned. In the formulation described in this paper, the interpolation needed to perform the Biot–Savart integral for velocity is achieved by a Delaunay triangulation of the Lagrangian points.[13,14] Vorticity is specified at points in the field and is linearly interpolated through triangulation. This approach is similar to the method documented by Russo and Strain,[15] who examined inviscid vorticity fields in the absence of surfaces. It enables highly anisotropic spatial distributions of vorticity to be resolved with a much smaller number of points than would be needed using isotropic elements. Unlike blob methods, there are no concerns regarding overlap of the elements because the triangularization process guarantees a piecewise continuous vorticity field. This has the added benefit that the vorticity is defined at the surface and is zero inside the surface. The triangulation contributes a minimal additional expense to the calculation, but the preceding estimates for the number of elements illustrate the potential increase in efficiency of such a method compared to a traditional vortex method when complex flows are to be investigated.

This technique directly solves the vorticity equation on a Lagrangian mesh and was introduced by Huyer and Grant,[16,17] where unsteady flow past bodies was examined. Computation of the diffusion term in the vorticity equation requires an accurate calculation of the Laplacian of the vorticity. In the formulation presented

here, first and second derivatives are computed by a least-squares fit of a general quadratic to the neighboring points.[8,9] The identity of neighboring points is provided by the triangulation. Thus, the treatment of diffusion further exploits the information provided by the triangulation, helping to reduce the penalty of computing the triangulation. Diffusion into regions of irrotational flow, where initially there are no computational points, is accomplished by rewriting the diffusion term so that an advectionlike term appears. This gives rise to a diffusion velocity (e.g., that of Strickland et al.[18]) that is used, along with the fluid velocity, to transport the points.

Details of the formulation are presented in the next section. Then the method is demonstrated by computing impulsively started flow past a circular cylinder as well as boundary layers over a flat plate. These two disparate test cases provide a stringent test of the method, one involving separated flow, the other fully attached flow. Comparisons with experimental results and Blasius boundary-layer profiles are used to validate this technique.

Methods

Surface and Triangular Mesh Geometry

Representations of various geometries are constructed by connecting line segments from x–y body coordinates. The number of desired points is input to represent the body with the points used to define the surface panels. The line segments on each body form the base of the surface-layer triangles (constructed via Delaunay triangularization) used in the volume of the flow. Additional layers of nodes are placed above the surface and are initially staggered with the first layer above the panel centroid, the second layer above the surface nodes, the third layer again over the panel centroid, and so forth. Each layer is placed above the surface incrementally separated by a distance of $0.5l_d$. These layers contain the same number of points used to represent the surface. As the flow is evolved for subsequent time steps, all of the points above the surface are advected. Figure 2a shows a close-up of the initial triangular mesh distribution constructed by Delaunay triangularization for the cylinder test case. The resolution is much greater in the normal compared with the tangential direction reflecting the steeper vorticity gradients in the normal direction. Consequently, highly anisotropic triangular elements are seen near the surface. As the cylinder flow develops out to $t = 2.0$, Fig. 2b shows the resulting triangular mesh. The triangular elements continue to reflect the gradients in the flow.

In these computations, length was nondimensionalized for a unit cylinder diameter and unit flat plate chord. Flat plate thickness was 0.05, and the leading and trailing edges were rounded using a 3:1 ellipse.

Satisfaction of the Velocity Boundary Conditions

Each panel on the body surface effectively carries two velocity generators: a surface vorticity and a potential source distribution. The sources are needed to ensure that the no-flux boundary conditions are satisfied. Uhlman and Grant[19] showed that as the number of points representing the surface vorticity increases, the strengths of the sources approach zero. The need for the sources is a result of the finite discretization of the surface vorticity distribution.

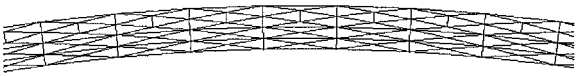


Fig. 2a Initial triangular mesh (close-up of top of cylinder).

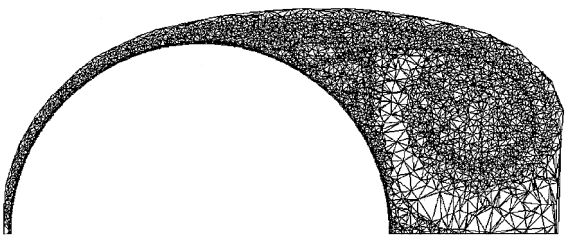


Fig. 2b Developed mesh at $t = 2.0$ for flow past a circular cylinder.

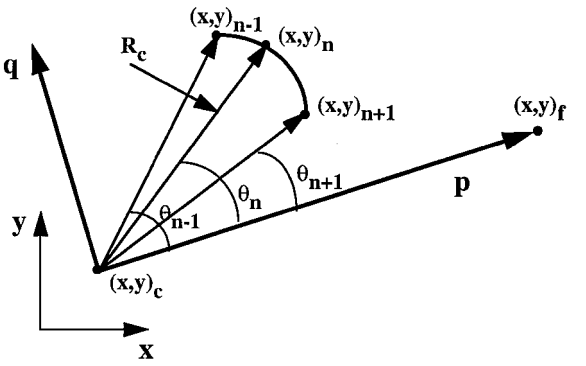


Fig. 3 Panel geometry.

Nonsingular source and vortex panels were developed with the source and vortex sheet strengths computed at the nodal points to enforce the boundary conditions. The panel geometry is shown in Fig. 3. The source and vortex panels are represented by circular arcs that connect three consecutive surface nodes and remain nonsingular. For ease of development, polar coordinates (r, θ) were chosen with radial and tangential velocities computed at the nth surface nodal coordinate. Three points, $n-1$, n, and $n+1$, are required to determine the radius of a circle and the circle centroid, x_c and y_c. We now desire the velocity at some field point, $(x, y)_f$. This requires the integral

$$I(\boldsymbol{x}) = \int_C f(s)\frac{[\boldsymbol{x}_f - \boldsymbol{x}(s)]}{|\boldsymbol{x}_f - \boldsymbol{x}(s)|^2}\,\mathrm{d}s \tag{1}$$

where s is the surface coordinate and f is the surface shape function. Here, f is chosen such that

$$f(s_n) = 1.0, \qquad f(s_{n\pm1}) = 0.0$$

The following function satisfies this in polar coordinates; the integrals are evaluated along θ for

$$f(\theta) = \frac{\sin(\theta_{n+1} - \theta)}{\sin(\theta_{n+1} - \theta_n)}, \qquad \theta_n \le \theta \le \theta_{n+1}$$

$$= \frac{\sin(\theta - \theta_{n-1})}{\sin(\theta_n - \theta_{n-1})}, \qquad \theta_{n-1} \le \theta \le \theta_n \tag{2}$$

This function smoothes the source and surface vorticity distribution along the surface. By defining the shape function in this way, the singularities normally found at the ends of a standard source panel are effectively removed. A local coordinate system (p, q) is defined and shown in Fig. 3 so that the unit vector $\hat{\boldsymbol{p}}$ lies along the $\boldsymbol{x}$–$\boldsymbol{x}_c$ vector. The unit vectors are defined as

$$\hat{\boldsymbol{p}} = \frac{\boldsymbol{x} - \boldsymbol{x}_c}{|\boldsymbol{x} - \boldsymbol{x}_c|}, \qquad \hat{\boldsymbol{q}} = \hat{z} \times \hat{\boldsymbol{p}} \tag{3}$$

By defining the local coordinate system this way, the local normal coordinate of the field point q is zero and consequently drops out of the integral equation. It can then be shown that the integral, expressed in local polar coordinates, is

$$\boldsymbol{I}_\sigma = \hat{\boldsymbol{p}}\boldsymbol{I}_1 + \hat{\boldsymbol{q}}\boldsymbol{I}_2, \qquad \boldsymbol{I}_\gamma = \hat{\boldsymbol{p}}\boldsymbol{I}_2 + \hat{\boldsymbol{q}}\boldsymbol{I}_1$$

where

$$\boldsymbol{I}_1 = R_c \int_{\theta_{n-1}}^{\theta_{n+1}} f(\theta)\frac{(p - R\cos\theta)}{p^2 + R^2 - 2Rp\cos\theta}\,\mathrm{d}\theta$$

$$\boldsymbol{I}_2 = -R_c^2 \int_{\theta_{n-1}}^{\theta_{n+1}} f(\theta)\frac{\sin\theta}{p^2 + R^2 - 2Rp\cos\theta}\,\mathrm{d}\theta \tag{4}$$

R_c, as defined in Fig. 3, is the radius of the local arclength formed by the three consecutive nodal points. When the integral is evaluated at θ_{n+1}, the function $f(\theta)$ becomes zero. This property eliminates the

singularity from the integral expression so that finite velocities are computed at the nodal end points. The integral may be transformed back to the global coordinate system so that

$$\boldsymbol{u}(\boldsymbol{x}) = -\frac{1}{2\pi}\sum_{n=1}^{N}[\sigma_n \boldsymbol{I}_{\sigma n}(x)\hat{n} + \gamma_n \boldsymbol{I}_{\gamma n}(x)\hat{t}] \tag{5}$$

where σ_n and γ_n are the source and vortex amplitudes for node n of a body defined by N points.

There are $2N$ equations produced for the N surface source and vortex sheet values. Requiring the total velocity (that due to the surface source distribution and the field vorticity) to be zero at each surface node produces an equivalent number of equations. Mathematically, it can be shown that this is a nonunique set of equations. Two additional constraints are imposed to make the set of equations unique. Because the integral of vorticity over a bounded volume is zero when the velocity vanishes on the bounding surface (no-slip condition), the integral of the circulation produced by the vortex sheets over the body surface must be zero. Similarly, the integral of the surface potential source over the body surface is zero by continuity. These constraints, integrated over the arclength $R_c\,\mathrm{d}\theta$ may be written as

$$\sum_{n=1}^{N}\sigma_n R_{c_n}(\theta_{n+1} - \theta_{n-1}) = 0, \qquad \sum_{n=1}^{N}\gamma_n R_{c_n}(\theta_{n+1} - \theta_{n-1}) = 0 \tag{6}$$

The matrix solution for the surface strengths is formulated via Lagrange multipliers so that the integral constraints are met exactly and that the $2N$ boundary conditions on velocity are satisfied in a least-squares sense. The deviation from zero of the velocity at the control points has always been found to be slight; maximum values of this deviation for the computations described hereafter are of the order of 10^{-3} of freestream velocity.

Triangular Vorticity Elements

In the present method, nodal vorticity values are known, and a linear variation of vorticity between nodal points is assumed. At each time step, a Delaunay triangularization routine is used to form an unstructured mesh connecting each nodal point, thus forming triangular elements. Delaunay triangulation effectively optimizes the $\mathcal{AR}$ of all triangles constructed from a random distribution of points. For more detail concerning the triangularization method, the reader is referred to the work of Borouchaki and Lo.[14] Consider a single triangular element with vertices located at (x_1, y_1), (x_2, y_2), and (x_3, y_3). Then, in area coordinates,[20]

$$N_1 = (a_1 + b_1 x + c_1 y)/2\Delta, \qquad N_2 = (a_2 + b_2 x + c_2 y)/2\Delta$$
$$N_3 = (a_3 + b_3 x + c_3 y)/2\Delta \tag{7}$$

where

$$\Delta = \text{triangle area}$$
$$a_1 = x_2 y_3 - x_3 y_2, \qquad b_1 = y_2 - y_3, \qquad c_1 = x_3 - x_2 \tag{8}$$

similarly for nodes 2 and 3.

The shape function values are then 1.0 at their respective nodes and 0 at each of the other two nodes. The vorticity over the element may then be expressed as

$$\omega = \Omega_1 N_1 + \Omega_2 N_2 + \Omega_3 N_3 \tag{9}$$

The velocity at a field point f due to an element (from the Biot–Savart integral) is

$$\boldsymbol{u}(\boldsymbol{x}) = \frac{1}{2\pi}\int_A \boldsymbol{\omega}(\boldsymbol{x}) \times \frac{(\boldsymbol{x}_f - \boldsymbol{x})}{|\boldsymbol{x}_f - \boldsymbol{x}|^2}\,\mathrm{d}A \tag{10}$$

This can be rewritten as

$$\begin{aligned}\boldsymbol{u}(\boldsymbol{x}) &= -\frac{1}{2\pi}\int_A \boldsymbol{\omega}(\boldsymbol{x}) \times \nabla\left(\ell n\left[|\boldsymbol{x}_f - \boldsymbol{x}|^2\right]^{\frac{1}{2}}\right)\mathrm{d}A \\ &= -\frac{1}{2\pi}\int_A \{\nabla \times [\boldsymbol{\omega}(\boldsymbol{x})\ell n|\boldsymbol{x}_f - \boldsymbol{x}|] \\ &\quad - [\nabla \times \boldsymbol{\omega}(\boldsymbol{x})]\,\ell n|\boldsymbol{x}_f - \boldsymbol{x}|\}\,\mathrm{d}A\end{aligned} \tag{11}$$

Using the divergence theorem on the first part of the integral, this integral is split into two so that

$$\boldsymbol{u}(\boldsymbol{x}) = -\frac{1}{2\pi}\left(\begin{array}{l}\displaystyle\oint_c \boldsymbol{\omega}(\boldsymbol{x}) \times \hat{n}(\boldsymbol{x})\ell n|\boldsymbol{x}_f - \boldsymbol{x}|\,\mathrm{d}\ell \\ \displaystyle\quad + \int_A [\nabla \times \boldsymbol{\omega}(\boldsymbol{x})]\ell n|\boldsymbol{x}_f - \boldsymbol{x}|\,\mathrm{d}A\end{array}\right) \tag{12}$$

The area integral can be expressed as

$$\boldsymbol{u}_A(\boldsymbol{x}) = \left\{\sum_{k=1}^{3}\frac{|\Omega_k|}{4\pi A}\begin{pmatrix}(x_{k+1} - x_{k-1})\hat{x} \\ +(y_{k+1} - y_{k-1})\hat{y}\end{pmatrix}\right\}E(\boldsymbol{x}, s) \tag{13}$$

where

$$E(\boldsymbol{x}, s) = \int_A \ell n|\boldsymbol{x}_f - \boldsymbol{x}|\,\mathrm{d}A$$

Here Ω_k is the nodal vorticity value at the kth node.

The circuit integral can be evaluated over each leg of the triangle resulting in an integral of the form

$$\boldsymbol{u}_c(\boldsymbol{x}) = \frac{1}{2\pi}\sum_{k=1}^{3}\hat{n}_k \times G(\boldsymbol{x}, \eta) \tag{14}$$

where

$$G(\boldsymbol{x}, \eta) = \int_{\eta_k^-}^{\eta_k^+}\left\{\begin{array}{l}\Omega_{k-1}N_{k-1}(\boldsymbol{x}_f - \boldsymbol{x}) \\ \quad + \Omega_k N_k(\boldsymbol{x}_f - \boldsymbol{x})\end{array}\right\}\ell n|\boldsymbol{x}_f - \boldsymbol{x}|\,\mathrm{d}\eta$$

where η is defined as the segment from the $k-1$ to the kth node.

Construction of Derivatives

A useful way to express first- and second-order spatial derivatives across scattered points is to construct a least-squares solution for all of the triangles that intersect a given node. This is accomplished by expressing the vorticity about a desired node as

$$\omega - \omega_0 = ax + by + cx^2 + dy^2 + exy \tag{15}$$

and determining the constants a, b, c, d, and e by a least-squares fit to the values of ω at points in the neighborhood. Here, x and y are referenced to the local node. For triangularization methods, the elements connected to a given node are known, and, therefore, the nodal points can easily be found. If additional points are needed to maintain accuracy, the elements that neighbor the elements connected to the desired node are found, and their respective nodes are included in the list. The nodal points provide the data with which to construct the least-squared fit of the vorticity field. After the second-order fit of the vorticity is computed, the values of the derivatives are simply computed at a given nodal point ($x = y = 0$):

$$\frac{\partial\omega}{\partial x} = a, \qquad \frac{\partial\omega}{\partial y} = b, \qquad \nabla^2\omega = \frac{\partial^2\omega}{\partial x^2} + \frac{\partial^2\omega}{\partial y^2} = 2.0(c + d) \tag{16}$$

Computation of Surface Vorticity

As mentioned, surface source and vortex sheet strengths are computed to solve for the no-slip, no-flux boundary conditions. At the surface, the vorticity (not the vortex sheet strength) is desired. The appropriate surface vorticity is needed to diffuse the vorticity into the volume. Therefore, a method is required to transfer the vorticity confined in the infinitely thin vortex sheet to actual surface vorticity.

The following technique to accomplish this was devised. After the surface source and vortex sheet strengths are computed, velocities are computed at each point in the flowfield. The vorticity on the surface is then determined by taking the curl of the velocity to obtain $\boldsymbol{\omega} = \nabla \times \boldsymbol{u}$ at the surface. The new assignment of this surface

vorticity will then modify the velocity field, but only the triangular elements connected to the surface will contribute. After the nodal vorticity is computed, the velocities at the surface are recomputed, and vortex sheet and source panel strengths are solved again, ensuring the boundary conditions are met. This iterative process typically requires only two to three iterations until the maximum vortex sheet strength is 10^{-2}. The reason this process converges so rapidly is that fine resolution is maintained near the surface resulting in thin triangular elements. A reduction in velocity contribution from the vortex sheets is directly correlated with an increase in the velocity contribution due to the modified surface vorticity value. In other words, there is effectively no change in the overall velocity in the field. The only change is in the source of the velocity (from vortex sheet to surface vorticity). Thus, the no-slip boundary condition is satisfied as much as possible by the surface vorticity alone, and this vorticity is used to evolve the vorticity field as described in the next section.

Evolution of the Vorticity Field

In inviscid flow, vorticity is transported by the velocity field in the same way as a material element. This type of flow is very well suited to Lagrangian mesh formulations, where the mesh points are transported by the velocity field. When viscosity is included in the formulation, there is physically an additional mode of transport. Viscous fluids transport vorticity via diffusion as well as advection. Rather than introducing new empty points into the mesh onto which vorticity may diffuse, we transport the existing mesh points with the sum of a diffusion velocity and the usual flow velocity. Thus, the mesh points tend to move from regions of larger vorticity magnitude to regions of lesser magnitude, according to the diffusive transport by viscosity.

The diffusion velocity concept has been formulated for vortex blobs by Strickland et al.[18] They begin with the diffusion velocity method proposed by Ogami and Akamatsu[21] and use Fishelov's[22] method to represent the spatial derivatives of vorticity. The vortex blobs advect with the diffusion velocity, and the blob size changes in time according the divergence of the diffusion velocity. In the present method, the diffusion velocity may be developed in the same form as the blob diffusion velocity, but additional terms arise because we are dealing with the vorticity field as a whole instead of individual vortex blobs.

In two dimensions, vorticity is a scalar quantity (z-component only) whose evolution is given by

$$\frac{\partial \omega}{\partial t} + \boldsymbol{u} \cdot \nabla \omega = \nu \nabla^2 \omega \tag{17}$$

The concept of diffusion velocity for scalars (i.e., vorticity that is the dependent variable for two-dimensional flows) is readily developed. A diffusion velocity $\boldsymbol{v}$ such that the transport $\boldsymbol{v}\omega$ is the same as the diffusive transport is defined:

$$\boldsymbol{v}\omega = -\nu \nabla \omega$$

Thus, for scalars, the diffusion velocity is

$$\boldsymbol{v} = -(\nu/\omega)\nabla \omega = -\nu \nabla (\ell n \omega) \tag{18}$$

This is the same form as that developed by Strickland et al.[18] Now, by adding the diffusion velocity advection term to both sides of Eq. (17) we have

$$\frac{\partial \omega}{\partial t} + (\boldsymbol{u} + \boldsymbol{v}) \cdot \nabla \omega = \nu \nabla^2 \omega + \boldsymbol{v} \cdot \nabla \omega \tag{19}$$

Because $\nabla^2(\ell n \omega) = \nabla^2 \omega/\omega - (\nabla \omega)^2/\omega^2$, the final form of the equation can be written as

$$\frac{\partial (\ell n \omega)}{\partial t} + (\boldsymbol{u} + \boldsymbol{v}) \cdot \nabla (\ell n \omega) = \nu \nabla^2 (\ell n \omega) \tag{20}$$

A semi-implicit scheme was devised to evolve the vorticity analogous to a Crank–Nicholson type formulation. Typically, 20 iterations are required to converge the solution. We have found that, for numerical purposes, solving the vorticity equation in terms of the natural log of the vorticity results in a much better behaved and more stable equation, especially where large vorticity gradients are produced. Equation (18) shows that the diffusion velocity becomes very large as the vorticity becomes small. This is handled by setting a cutoff for the vorticity, usually for the magnitude of vorticity equal to 0.001. Another difficulty arises as the vorticity changes sign. As long as the local vorticity field is exclusively positive or negative, there is no problem because the construction of the log of the vorticity magnitude is computed. As the vorticity changes sign, however, it was found that Eq. (19) is needed to take into account positive and negative vorticity and is to be used for these special cases. The magnitude of the vorticity remains relatively small, as do the actual derivatives, and so the calculation remains stable.

The mesh points are now transported according to the equation

$$\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t} = \boldsymbol{u} + \boldsymbol{v} \tag{21}$$

The Lagrangian points were advected according to an Adams–Bashforth method to maintain second-order accuracy in time. For the case of uniform time step Δt and time level n,

$$\boldsymbol{x}^{n+1} = \tfrac{3}{2}(\boldsymbol{u}^n + \boldsymbol{v}^n)\Delta t - \tfrac{1}{2}(\boldsymbol{u}^{n-1} + \boldsymbol{v}^{n-1})\Delta t \tag{22}$$

As the points are continually advected and diffused, the triangles will deform by stretching or compressing. To maintain spatial resolution of the vorticity field, a division algorithm was developed. The vorticity at the centroid of the triangle is computed using both the linear interpolation as well as a second-order approximation. If the difference in vorticity due to the second-order term is 10% of the linear approximation of vorticity, a new nodal point is added at the element centroid. The vorticity at this point is the average of the vorticity amplitudes at each node. To offer additional control on the number of points in the field, an amalgamation routine was written as well. If points become closer than $0.1\ l_d$, the nodal vorticity values are combined in such a way as to conserve total circulation. Amalgamation was found to be necessary for the viscous diffusion calculation to remain stable. Penetration of advecting points inside the body is also not allowed. If a point is advected inside the body, it is reflected to a position just above the point of entry.

Accelerated Calculation

To increase the resolution of the surface as well as the flowfield, it was necessary to compute the interactions of a large set of vorticity elements. The fast multipole method provided a means to compute the flowfield due to the increased number vorticity elements in reasonable CPU times. A description of the calculation used in this paper is given by Huyer et al.[11] and will be not be repeated here. It is based on the method presented by Greengard and Rokhlin.[5] The final expression for each velocity component has the form

$$\Psi = \sum_{n=0}^{\infty} \sum_{m=0}^{n} B_{n-m,m} h^{n-m} k^m \tag{23}$$

An expression for the far-field influence of all vorticity elements in terms of a set of coefficients B and local coordinates relative to the box center h and k for each box in the field was thus formulated. An examination of the errors due to the fast calculation method were computed. Average and maximum errors in the velocity calculation (relative to the freestream velocity) were found to be of the order 10^{-4} and 10^{-3}, respectively.

Effect of Surface Resolution and Time Step

To estimate the convergence associated with the construction of the vorticity field and its derivatives, a vorticity distribution was defined and then discretized and triangulated using the present method.[16,23] The vorticity distribution due to a vortex blob was modeled using the current method and to provide a test for an exact solution. The test function is given by Perlman.[24] For a coarse representation of the vorticity field using 1000 nodes, average errors in velocity were computed to be 0.4%, 1.15% in the first derivative and 1.3% for the Laplacian. Average errors decrease asymptotically as the number of nodal points are increased. For a finer representation of the vorticity field using 10,000 points, average errors in velocity are 0.05%, 0.1% in the first derivative and 0.5% in the second derivative. Defining the typical node spacing h as the area divided

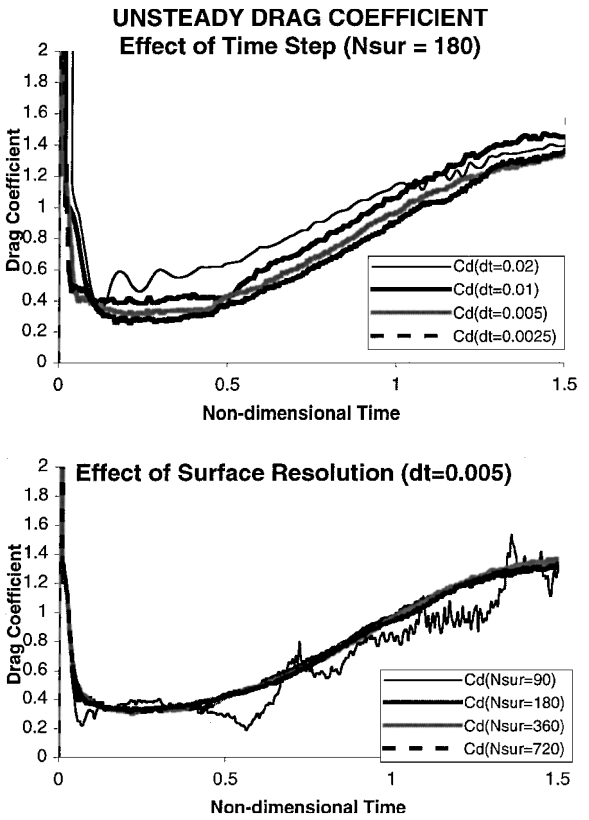


Fig. 4 Effect of surface resolution and time step on the computed drag coefficient for a circular cylinder at $Re = 3 \times 10^3$.

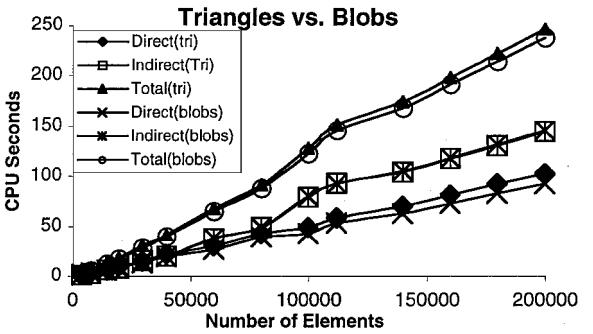


Fig. 5 Timing runs for the field velocity calculation due to the field vorticity only; accelerated calculation compares computational times for the triangular elements and blobs.

by the square root of the number of points and ε as the error, the scheme's order of convergence $= d(\ln \varepsilon)/d(\ln h)$ for the velocity is 1.0; for the first derivative it is 2.0, and it is 1.0 for the Laplacian. This test then gives an indication of the number of points required to represent a spatial region of vorticity.

The effect of surface grid resolution and time step was also examined for flow past a circular cylinder out to $t = 3.0$. This effect was gauged using the drag coefficient. Figure 4 plots the drag coefficient for surface resolution of 90, 180, 360, and 720 surface points for a time stop of 0.005 and a time step of 0.02, 0.01, 0.005, and 0.0025 for a surface resolution of 180 points. A surface resolution of 90 points is insufficient to characterize the initial drag well, and the solution is very noisy. Doubling the surface points to 180 provides a much smoother solution. An even better representation is provided by 360 points, and little improvement is seen for 720 surface points. A time step of 0.02 displays an initially oscillating solution. As Δt is decreased to 0.01, the solution is smoother but is improved for $\Delta t = 0.005$. For $\Delta t = 0.0025$, there is little improvement.

In the cylinder calculations, 360 surface points were used, and for the flat plate calculations, 200 surface points were used. A time step of 0.005 was used in all test cases. The Reynolds number was set to 3×10^3 for the cylinder based on the diameter. Flat plate Reynolds numbers of 1×10^3, 3×10^3, and 1×10^4 were examined based on chord. Initially, 200 surface nodal vorticity values and nine layers of 200 nodes were present at the first time step on each surface for a total of 2000 nodes for the case of the flat plate. The cylinder test case used 360 surface points and nine initial layers for a total of 3600 points on the first time step. Test runs were carried out to a time of 3.0 for the cylinder and 4.0 for the flat plate test cases. At the end of the cylinder calculations, 24,000 points resulting in approximately 48,000 elements were used. This is significantly less that the 300,000 blobs used by Koumoutsakos and Leonard[3] for their calculations.

Timing Considerations

All computations were performed on the Cray C-90 at the Naval Oceanographic Office at Stennis Space Flight Center, Mississippi. Figure 5 shows comparisons of CPU time required to compute the velocity as a function of the number of elements for the direct calculation of nearby elements, indirect calculation involving far-field contributions, and sum of the two. In addition, runs using vortex blobs alone are also shown to provide a comparison with the traditional vortex method and the present method. Because the accelerated calculation is used, CPU time as a function of element number is approximately linear. In addition, above 100,000 elements, the time required for the indirect calculation is approximately 20% greater compared with the direct part of the calculation. What is somewhat surprising is the relatively small penalty paid for computations using triangles vs blobs. The blob calculation required only 4% less time relative to the total computational time required compared with the triangle calculation. The reason for this is that the full Biot–Savart integral over the triangle [Eqs. (12–14)] is only computed at points in close proximity to the element. Four-point Gauss integration is used for elements separated a medium distance to the field point, and one-point Gauss integration is used for the farthest points in the direct calculation. This method significantly reduces the penalty paid for the velocity calculation. It is also admitted that this accelerated method is not optimized compared with other methods reported. The total velocity calculation for 200,000 points requires 237 CPU seconds. By comparison, Koumousatkos and Leonard[3] report that 1-min CPU times are required for 1×10^6 points on a Cray-YMP. A combination of their accelerated calculation and the present method (which requires significantly fewer points) would provide a most efficient algorithm indeed.

CPU time required for the triangulation is relatively small for the present method. For 200,000 triangular elements, the CPU time required to construct the mesh is 13.7 s, which is approximately 5% of the total velocity calculation. Even this is not fully optimized, and faster triangulation schemes have been reported.[14] Construction of the connectivity of the points and formulation of the derivatives for the semi-implicit formulation requires approximately 0.5% of the total calculation for each iteration in the semi-implicit formulation at 15,000 points. Because the cost of computing the derivatives is linear as a function of the number of points (half the number of elements), the relative cost of this computation decreases with increasing number of points.

Results

Flat Plate

Unsteady flow development past a flat plate at zero incidence was examined to validate the method for attached flow test cases. Flow past a flat plate was first computed by Blasius (see Ref. 25), and the solution is well known and has been experimentally verified. For flow past a flat plate, the boundary-layer solution of the flowfield is self-similar and scales as a function of local Reynolds number. Figure 6 shows the unsteady flow development past the flat plate for a Reynolds number of 3×10^3. For this case, the entire flow including the wake is shown. The vorticity field is plotted with black regions representative of negative (clockwise) vorticity and gray areas representing positive vorticity. Plotting the field in this way allows for visualization of the vortex structures. As can be seen, at $t = 0.5$, the initial boundary layer about the flat plate appears to be developed. At $t = 1.0$, initial rollup of the Karman vortex street

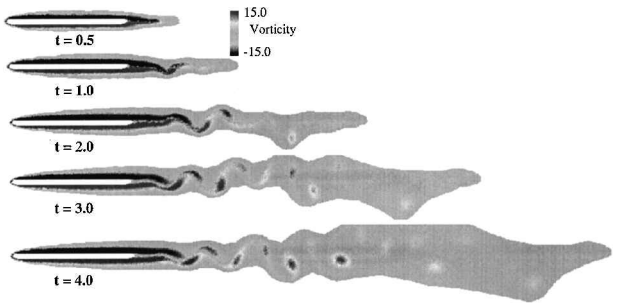


Fig. 6 Unsteady flow development for a 5% thick flat plate for $Re = 3 \times 10^3$ and $t = 0.5$–4.0.

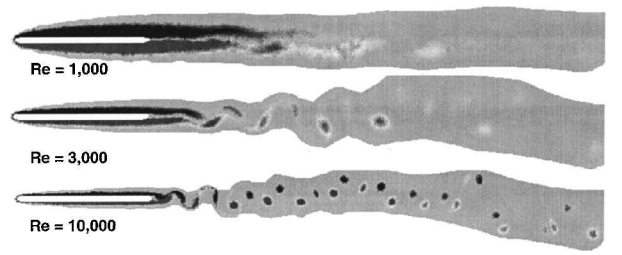


Fig. 7 Reynolds number comparison of the instantaneous vorticity field at $t = 4.0$ for $Re = 1 \times 10^3$, 3×10^3, and 1×10^4.

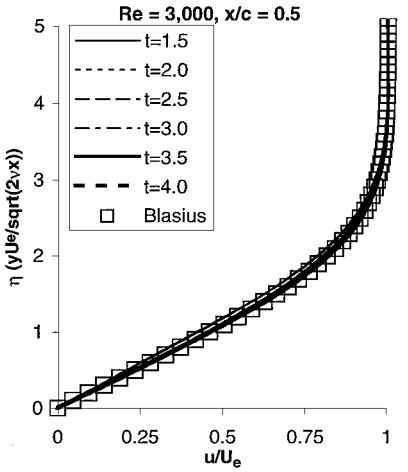


Fig. 8 Flat plate boundary-layer profiles for $Re = 3 \times 10^3$ and $t = 1.5$–4.0.

is seen to occur approximately 0.25 chord lengths downstream. At this point in the calculation, there is little quantitative or qualitative change in the boundary layer over the flat plate as time progresses. In the wake from $t = 2.0$ to 5.0, development of the Karman vortex street can clearly be seen. As the vortex structures are advected downstream, the vorticity contained within the structures begins to diffuse as the vortices expand and begin to lose circulation.

Figure 7 shows a comparative picture of the instantaneous vorticity field at $t = 4.0$ for $Re = 1 \times 10^3$, 3×10^3, and 1×10^4. As the Reynolds number increases, the thickness of the boundary layer decreases. In addition, for $Re = 1 \times 10^3$, there is a very weak rollup of the shed vorticity with little evidence of coherent vortices forming until far downstream. In contrast, there is almost an immediate rollup of the shed vorticity into a well-defined Karman vortex sheet for $Re = 1 \times 10^4$.

Figure 8 shows the boundary-layer profiles for $Re = 3 \times 10^3$ taken at the midchord location for various times ranging between 1.5 and 4.0. These profiles are plotted with the velocity scaled by the edge velocity u/U_e on the abscissa and the nondimensional boundary-layer parameter $\eta(yU_e/\sqrt{2\nu x})$ on the ordinate. These variables give the self-similar boundary-layer solution for laminar boundary layers for all Reynolds numbers. The Blasius solution is also plotted as a reference. This plot shows that the unsteady boundary-layer velocity profiles change very little after $t = 2.0$ and compare favorably with the Blasius solution. In effect, a steady boundary-layer flow has been reached with the only unsteadiness in the flow found in the wake due to the rollup of the shed vorticity.

Table 1 Flat plate boundary-layer average and maximum errors ($\times 10^{-2}$)

	Reynolds number					
	1×10^3		3×10^3		1×10^4	
Chord	Avg	Max	Avg	Max	Avg	Max
0.3	0.45	0.96	0.18	0.29	0.94	1.78
0.5	0.66	1.45	0.12	0.26	0.79	1.38
0.7	1.70	3.04	0.58	0.97	0.58	1.46

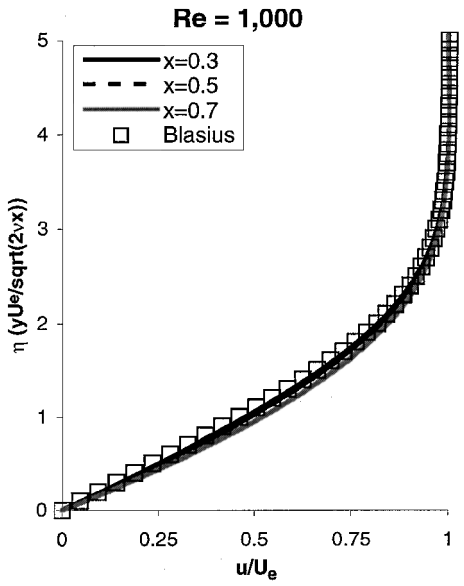
Figure 9 shows the boundary-layer profiles for Reynolds numbers of 1×10^3, 3×10^3, and 1×10^4 at chordwise stations of $0.3c$, $0.5c$, and $0.7c$. As can be seen, all three sets of boundary-layer profiles agree with the Blasius curve, and all appear to be self-similar. Of the three, it appears that $Re = 3 \times 10^3$ has the overall best fit. For both $Re = 1 \times 10^3$ and 1×10^4, the computed boundary layer is qualitatively thinner, exhibiting slightly higher Reynolds number behavior. Table 1 shows the maximum and average error between the computed solution and the Blasius solution. Errors were computed for 200 values of η equally spaced between 0.0 and 3.0 and are shown as the absolute difference between the computed and Blasius solution. Table 1 shows that the average error in velocity is a maximum of 0.0304 for $Re = 1 \times 10^3$ at $x/c = 0.7$. Errors appear minimal for $Re = 3 \times 10^3$ and, overall, appear greatest for $Re = 1 \times 10^3$.

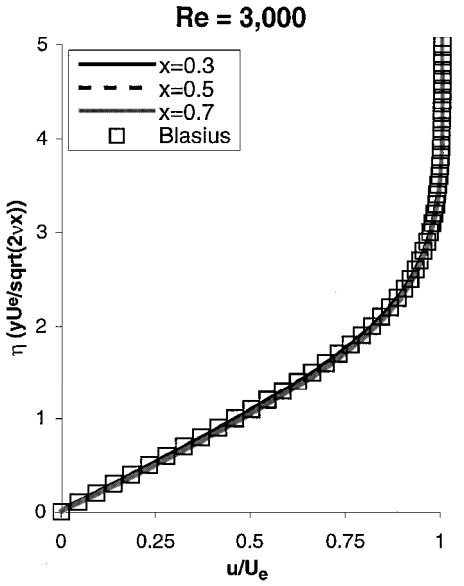
Cylinder

Figure 10 shows the unsteady flow development around a cylinder from time $t = 0.5$ to 4.0. Similar to Figs. 6 and 7, the vorticity field is plotted. Black regions represent negative (clockwise vorticity), and positive (counterclockwise) vorticity is indicated with gray. Very light gray to white regions suggest regions of minimal vorticity. The scale displays vorticity values in the range of ±25. Although maximum vorticity magnitudes in the field are as high as 300.0, it was found that this scale allowed for optimal visualization of the vortex structures. At $t = 0.5$, a thin region of negative vorticity is seen on the upper surface with positive vorticity on the lower surface. Although it cannot be seen in this plot, a very thin layer of opposite-signed vorticity is present from the rear stagnation point, midway up the aft cylinder surface. Initiation of the opposite sign vorticity was seen at approximately $t = 0.22$. By $t = 1.0$, initial rollup of the vorticity sheet is seen. In addition, a secondary vortex of opposite signed vorticity is seen on the upper and lower surfaces. By $t = 2.0$, the flow appears fully developed. By $t = 3.0$, the vortices are extended into the wake, and the feeding sheets of vorticity appear to lose strength. Also, these sheets initially coalesce upstream of the primary vortices. By $t = 4.0$, it appears that the primary vortices have translated slightly upstream. In addition, there appears to be significant asymmetry in the flow on the upper surface compared with that of the lower surface. It is at approximately this point where alternate vortex shedding begins.

Figure 11 shows an enlargement of the unsteady flow at $t = 2.0$. Figure 11 clearly demonstrates the formation of a pair of coherent primary vortices in the wake region of the cylinder. Secondary vortices close to the cylinder wall are formed as well due to the reverse flow velocities of the primary vortices. Tertiary regions of vorticity and fourth regions may be seen as well. Figures 10 and 11 demonstrate how the method spatially resolves the flowfield.

Wake definition parameters of the primary vortices were measured for the length of the separated wake region as well as the location of the vortex structures. These geometrical parameters were experimentally measured by Bouard and Coutanceau.[26] Using their definition, x/D indicates the location of the primary vortex core relative to the cylinder rear stagnation point. L/D is defined as the point where the wake streamlines close and can be approximated as the point where the streamwise velocity changes sign from positive to negative. Note that $y/2D$ is defined as the distance between the two primary vortex cores scaled by twice the cylinder diameter.





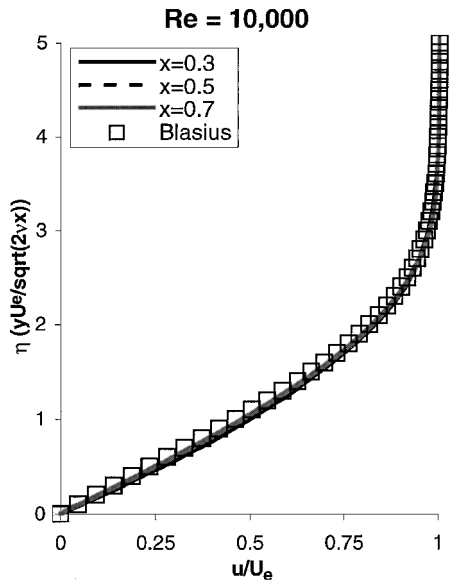


Fig. 9 Flat plate time averaged boundary-layer profiles for $Re = 1 \times 10^3, 3 \times 10^3$, and 1×10^4.

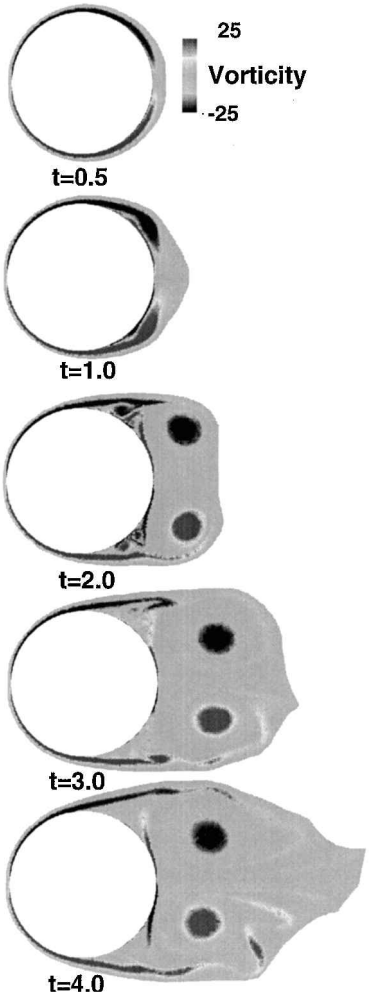


Fig. 10 Unsteady flow development about a cylinder for $Re = 3 \times 10^3$ and $t = 0.5$–4.0.

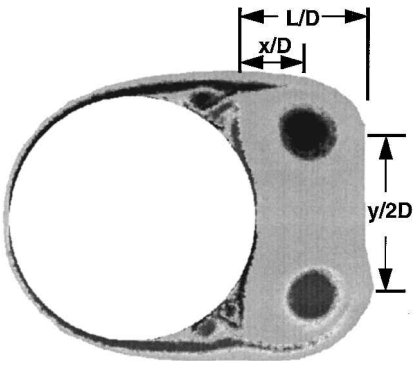


Fig. 11 Instantaneous flow at $t = 2.0$ shown with definition of wake geometry parameters.

Figure 12 shows a plot of these experimental measurements and compares them with the present numerical simulation. Numerical data agree quite well with the experimental results. It does appear that the calculation slightly underestimates the streamwise position of the vortex core as well as the length of the wake region during the initial flow development from $t = 1.0$ to 2.0. From $t = 2.0$ to 3.0, it appears that the flow development that previously lagged comes into full development, and better agreement is seen between numerical and experimental results. During the later stages of flow development, it appears that the vertical positions of the vortices are less for the numerical simulation.

Figure 13 shows a plot of the streamwise velocity along the cylinder centerline from the rear stagnation point to 2.5 cylinder diameters (referenced from the aft cylinder stagnation point)

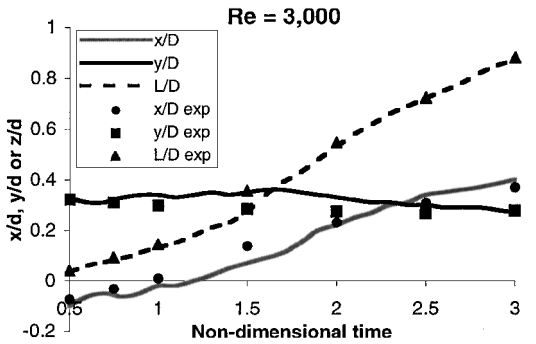


Fig. 12 Wake geometry parameters for vortex position and extent of wake; symbols display experiments by Bouard and Coutanceau.[26]

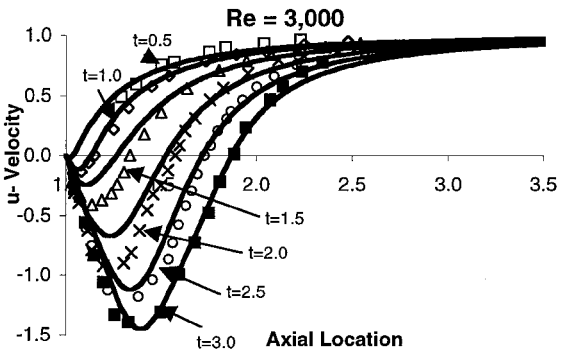


Fig. 13 Unsteady wake velocities along cylinder centerline for $Re = 3 \times 10^3$ and $t = 0.5$–3.0; symbols display experiments by Bouard and Coutanceau.[26]

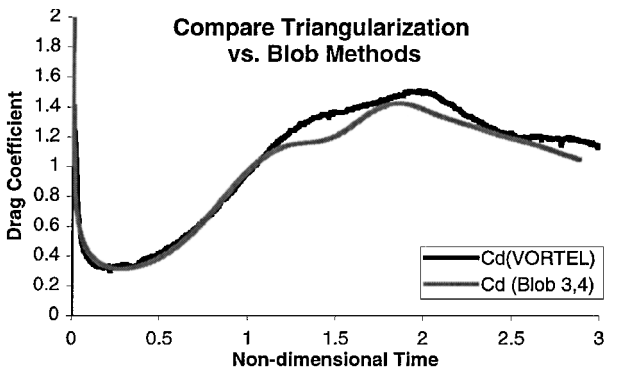


Fig. 14 Unsteady cylinder drag coefficient for $Re = 3 \times 10^3$ and comparison with the vortex blob methods of Koumoutsakos and Leonard[3] and Subramaniam.[4]

downstream. Streamwise velocity was computed to demonstrate the magnitudes of the reverse flow region aft of the cylinder and provide comparisons with experimental data. Symbols display the experimental measurements of Bouard and Coutanceau,[26] whereas the solid lines display the numerical results. Agreement appears satisfactory for times of 0.5 and 1.0. At $t = 1.5$ and 2.0, the magnitude of the reverse flow appears to be underestimated by the numerical solution. For $t = 2.5$ and 3.0, it appears that increased circulation results in increased reverse flow velocities with better agreement seen with the experimental data.

Surface pressure was computed using the expression derived by Uhlman.[27] The integrated surface pressure and viscous skin friction yielded the drag coefficient that is plotted in Fig. 14. To provide a comparison, the drag coefficient computed by Koumoutsakos and Leonard[3] is presented. Subramaniam[4] obtained essentially identical results in Cd. In their[3,4] and other traditional vortex blob methods, drag coefficient is computed from the momentum defect caused by the time rate of change of the complex moment of circulation. There is excellent agreement during the initial flow startup between 0.0 and 1.0, especially for the initial drag well seen at $t = 0.3$. The slope of the increase in Cd beginning at $t = 0.5$ correlate quite well. Between $t = 1.0$ and 2.0, our method predicts an initial hump at $t = 1.5$ and maximum Cd value of 1.5 at $t = 1.95$ where the traditional method predicts a small hump at $t = 1.25$ and maximum of 1.42 at $t = 1.85$. Both methods agree relatively well especially considering the radically different computational methods used.

Discussion and Conclusions

A solution methodology based on direct solution of the vorticity equation on a Lagrangian mesh has been presented. It is inherently unsteady and able to automatically adapt to the local vorticity field resolving coherent vortex structures where necessary. A surface geometry is specified using surface source and vortex sheets. The surface vorticity is then calculated as the curl of the velocity field, and an iteration scheme is used to minimize the strength of the vortex sheets so that the boundary conditions are satisfied by the surface vorticity alone. Viscous diffusion transfers the vorticity off the surface into the flow. Vorticity is specified at points, and the field is triangularized. Integration of the vorticity over the triangles determines the velocity field. Derivatives were computed using a second-order least-squares fit of the local field variables. This numerical scheme was then validated by examining the unsteady flow development past a flat plate and circular cylinder.

The calculation was able to faithfully reproduce the laminar flow over a flat plate for the Reynolds numbers examined (up to 1×10^4). Accuracy was gauged with comparison to the standard Blasius laminar boundary-layer profile, which has been verified extensively through experiments over the years. The computed boundary layers were self-similar in nature and scaled with chordwise location as well as Reynolds number. Agreement was best for $Re = 3 \times 10^3$, where maximum relative error in local velocity was 1.4%. Interestingly, mismatch of boundary-layer profiles from computation to theory showed that the boundary layers were skewed toward higher Reynolds number behavior. For computations, one might expect the profiles to be skewed toward a lower Reynolds number due to possible artificial diffusion effects. This suggests that the Lagrangian vorticity method produces little artificial numerical diffusion due to the vorticity being moved with the local flow, with no need to explicitly compute the advection term.

Flow past a circular cylinder is inherently more complex due to the separation of the surface vorticity. The separated vortical structures produce secondary, tertiary, and even quaternary vortices due to the flows proximal to the surface. The computational algorithm was able to reproduce the finer details of the unsteady flow. Although not shown, qualitative comparisons of the unsteady flow development shown in Figs. 10 and 11 were consistent with the experimental flow visualization conducted by Bouard and Coutanceau.[26] This included qualitative agreement with the finer structures produced. Experiments and computations both showed the tertiary vortices formed as the primary vortex spawned the secondary vortex that produced the tertiary vortex.

Quantitative comparisons were made between vortex core and wake geometries and the centerline streamwise wake velocities. In both cases, good agreement was found. Comparisons between the calculation and experimental values showed similar flow development up to $t = 1.0$. By $t = 1.5$, the computed streamwise vortex position and length of the vortex wake lagged behind those produced experimentally. By $t = 2.0$, these flow quantities were recovered, and good agreement between calculation and experiment was found. The effect of lagging vortex position at $t = 1.5$ and 2.0 is demonstrated by the centerline velocity plots. Here, maximum centerline reverse flow velocities were underpredicted by the code. This would be a consequence of immature vortex development. As time progresses, the vortex pair builds up circulation so that by $t = 2.5$ and 3.0 good agreement between experimental and computational flow velocities is reached.

Subramaniam[4] compared his numerical results of the centerline streamwise wake velocities with experiments as well as several numerical vortex methods. In these comparisons, the numerical results underpredicted the maximum reverse flow velocities at $t = 1.5$ and 2.0 but recovered by $t = 2.5$. The reason for the consistent discrepancy between two-dimensional numerical calculations and experiments is not clear.

Numerical results from the present method compare very well with results obtained from Koumoutsakos and Leonard[3] and Subramaniam.[4] These researchers used high-resolution vortex blob methods to obtain their results. For flow past a cylinder at $Re = 3 \times 10^3$, Koumoutsakos and Leonard used on the order of 300,000 points. Our calculation effectively performed the same job using only 24,000 points at $t = 3.0$. This demonstrates the ability of our method to account for the inherent flow anisotropy, especially near the surface. The penalty we pay is for a more complex integral formulation that only slightly increases CPU time per point. Additional penalties are Delaunay triangularization of the vorticity field and construction of derivatives for the semi-implicit scheme. These penalties sum to less than 10% of the total computational effort and are small compared with the velocity calculation. By contrast, the diffusion scheme developed by Subramaniam[4] required as much CPU time as the velocity calculation.

Baseline calculations of the flow past a cylinder and flow past a flat plate demonstrate that the code can compute both separated and attached flows. It was unnecessary to change the scheme or approach to compute these two disparate test cases. The ability to compute a wide range of flows is a major objective of this research. Although not presented, the code also has the ability to compute flows past multiple bodies and to compute unsteady flows for bodies moving relative to one another.[23] A significant advantage of this formulation is its ready extension to three dimensions. Results of some initial results have been reported for inviscid vortex ring problems as well as viscous unsteady flow produced by unmanned undersea vehicle bow thrusters.

Acknowledgments

This research was supported by Office of Naval Research Contract 98WX30168, James A. Fein, Program Manager, and by the U.S. Naval Undersea Warfare Center Internal Research Program, Stuart Dickinson, Program Manager.

References

[1]Chorin, A. J., "Numerical Study of Slightly Viscous Flow," *Journal of Fluid Mechanics*, Vol. 57, No. 4, 1973, pp. 785–796.

[2]Sethian, J. A., and Ghoniem, A. F., "Validation Study of Vortex Methods," *Journal of Computational Physics*, Vol. 74, No. 2, 1988, pp. 283–317.

[3]Koumoutsakos, P., and Leonard, A., "High-Resolution Simulations of the Flow Around an Impulsively Started Cyliinder Using Vortex Methods," *Journal of Fluid Mechanics*, Vol. 296, No. 1, 1995, pp. 1–38.

[4]Subramaniam, S., "A New Mesh-Free Vortex Method," Ph.D. Dissertation, Dept. of Mechanical Engineering, Florida State Univ., Tallahassee, FL, Dec. 1996.

[5]Greengard, L., and Rokhlin, V., "A Fast Algorithm for Particle Simulations," *Journal of Computational Physics*, Vol. 73, No. 11, 1987, pp. 325–348.

[6]Strickland, J. H., and Baty, R. S., "An Overview of Fast Multipole Methods," Sandia National Lab., Rept. SAND95-2405, Albuquerque, NM, Nov. 1995.

[7]Schlichting, H., *Boundary-Layer Theory*, 7th ed., McGraw–Hill, New York, 1987, p. 140.

[8]Marshall, J. S., and Grant, J. R., "Penetration of a Blade into a Vortex Core: Vortex Response and Unsteady Blade Forces," *Journal of Fluid Mechanics*, Vol. 306, No. 1, 1996, pp. 83–109.

[9]Marshall, J. S., and Grant, J. R., "A Method for Determining the Velocity Induced by Highly Anisotropic Vorticity Blobs," *Journal of Computational Physics*, Vol. 126, No. 138, 1996, pp. 286–298.

[10]Huyer, S. A., and Grant, J. R., "Computation of Unsteady Separated Flow Fields Using Anisotropic Vorticity Elements," *Journal of Fluids Engineering*, Vol. 118, Dec. 1996, pp. 839–849.

[11]Huyer, S. A., Grant, J. R., and Uhlman, J. S., "A Vortex Element Representation of Two-Dimensional Unsteady Separated Flows," AIAA Paper 94-0075, Jan. 1994.

[12]Huyer, S. A., Grant, J. R., and Uhlman, J. S., "Computation of Unsteady Separated Flowfields Past an Oscillating Airfoil Using Discrete Vortex Elements," AIAA Paper 94-2257, June 1994.

[13]de Berg, M., van Kreveld, M., Overmars, M., and Schwarzkopf, O., *Computational Geometry*, Springer–Verlag, Heidelberg, Germany, 1997, pp. 200–210.

[14]Borouchaki, H., and Lo, S. H., "Fast Delaunay Triangulation in Three Dimensions," *Computational Methods in Applied Mechanics and Engineering*, Vol. 128, 1995, pp. 153–167.

[15]Russo, G., and Strain, J. A., "Fast Triangulated Vortex Methods for the 2D Euler Equations," *Journal of Computational Physics*, Vol. 111, 1994, pp. 291–323.

[16]Huyer, S. A., and Grant, J. R., "Incorporation of Boundaries for 2D Triangular Vorticity Element Methods," *Forum on Vortex Methods for Engineering Applications*, Sandia National Lab., Albuquerque, NM, 1995, pp. 211–225.

[17]Huyer, S. A., and Grant, J. R., "Computation of Incipient Separation Via Solution of the Vorticity Equation on a Lagrangian Mesh," *Vortex Flows and Related Numerical Methods II*, edited by Y. Gagnon, G.-H. Cottet, D. G. Dritschel, A. F. Ghoniem, and E. Meiburg, European Series in Applied and Industrial Mathematics, 1996.

[18]Strickland, J. H., Kempka, S. N., and Wolfe, W. P., "Viscous Diffusion of Vorticity Using the Diffusion Velocity Concept," *Vortex Flows and Related Numerical Methods II*, edited by Y. Gagnon, G.-H. Cottet, D. G. Dritschel, A. F. Ghoniem, and E. Meiburg, European Series in Applied and Industrial Mathematics, 1996, pp. 135–151.

[19]Uhlman, J. S., and Grant, J. R., "A New Method for the Implementation of Boundary Conditions in the Discrete Vortex Element Method," ASME 1993 Fluids Engineering Spring Meeting, American Society of Mechanical Engineers, Washington, DC, June 1993.

[20]Zienkiewicz, O. C., *The Finite Element Method*, 3rd ed., McGraw–Hill, New York, 1977, pp. 165–169.

[21]Ogami, Y., and Akamatsu, T., "Viscous Flow Simulation Using the Discrete Vortex Model—The Diffusion Velocity Method," *Computers and Fluids Journal*, Vol. 19, 1991, pp. 433–441.

[22]Fishelov, D., "A New Vortex Scheme for Viscous Flows," *Journal of Computational Physics*, Vol. 86, No. 1, 1990, pp. 200–223.

[23]Huyer, S. A., and Grant, J. R., "Examination of Unsteady Flow Past Multiple Bodies by Solution of the Vorticity Equation on a Lagrangian Mesh," AIAA Paper 97-0661, Jan. 1997.

[24]Perlman, M., "On the Accuracy of Vortex Methods," *Journal of Computational Physics*, Vol. 59, No. 2, 1985, pp. 200–223.

[25]Rosenhead, L., *Laminar Boundary Layers*, Dover, New York, 1963, pp. 222–229.

[26]Bouard, R., and Coutanceau, M., "The Early Stage of Development of the Wake Behind an Impulsively Started Cylinder for $40 < Re < 10^4$," *Journal of Fluid Mechanics*, Vol. 101, No. 583, 1980, pp. 583–607.

[27]Uhlman, J. S., "An Integral Equation Formulation of the Equations of Motion of an Incompressible Fluid," U.S. Naval Undersea Warfare Center, TR 10,086, Newport, RI, July 1992.

P. Givi
Associate Editor