



A cartesian grid method for modeling multiple moving objects in 2D incompressible viscous flow

David Russell *, Z. Jane Wang

Department of Theoretical and Applied Mechanics, Cornell University, 212 Kimball Hall, Ithaca, NY 14853, USA

Received 18 October 2002; received in revised form 26 May 2003; accepted 28 May 2003

Abstract

We present an efficient method for solving 2D incompressible viscous flows around multiple moving objects. Our method employs an underlying regular Cartesian grid to solve the system using a streamfunction–vorticity formulation and with discontinuities representing the embedded objects. The no-penetration condition for the moving geometry is satisfied by superposing a homogenous solution to the Poisson’s equation for the streamfunction. The no-slip condition is satisfied by generating vorticity on the surfaces of the objects. Both the initial Poisson solution and the evaluation of the homogenous solution require embedding irregular discontinuities in a fast Poisson solver. Computation time is dictated by the time required to do a fast Poisson solution plus solve an integral form of Laplace’s equation. There is no significant increase in computational cost if the geometry of the embedded objects is variable and moving relative to the underlying grid. We test the method against the canonical example of flow past a cylinder, and obtained new results on the flow and forces of two cylinders moving relative to each other.

© 2003 Elsevier B.V. All rights reserved.

Keywords: Moving boundary in Cartesian grid; Embedded discontinuity in Poisson solver; Flow past two cylinders; Computational fluid dynamics; Variable geometry; Flow past multiple bodies

1. Introduction

The method described in this paper results from our desire to model multiple objects moving relative to each other in a 2D incompressible viscous fluid in an efficient and accurate way. Examples include sedimentation of many particles and the interaction of the fore and hind wings of a dragonfly.

Flow around a *single* moving object can be efficiently calculated by using a single grid body fitted grid in the non-inertial reference frame. This approach does not work, at least for a single static discretization, for multiple objects moving relative to each other. It is possible to use a grid mapping that varies with time as

* Corresponding author. Tel.: +1-607-254-8593; fax: +1-607-255-2011.

E-mail addresses: dbr8@cornell.edu (D. Russell), jane.wang@cornell.edu (Z. Jane Wang).

the objects move, see for example [14]. This works well for simple geometries of fixed prescribed motions, but is computationally expensive for general motions.

An alternative is to employ a fixed Cartesian grid method which has the advantage of enabling the use of fast solvers. One drawback of using such a method is that there is no grid refinement in the area of greatest interest, the boundary of the objects. In fact, in general the boundaries are where the errors associated with the method are usually the largest. Since we are interested primarily in low to moderate Reynolds number ranges (about 10–1000), it is possible for us to afford the computational cost of ensuring that there is sufficient boundary discretization. Another difficulty lies in how to incorporate the boundary conditions on the immersed surfaces in an accurate way.

Many existing methods have followed this philosophy, the immersed boundary method [24,25] is one example. In this case the object in the flow is replaced by forcing terms which represent the forces imposed on the fluid by the presence of the object. Originally, the force calculation was first-order accurate at the boundary. Recent refinements, for example Minion and Cortez [5] and Lai and Peskin [17], increase the accuracy to second-order and above at the boundary. However such a method runs into difficulty when the object is rigid. Mohd–Yusof [32] has devised an alternate means of calculating the forcing terms, which has subsequently been refined and applied [9,29]. This method appears to work very well for rigid bodies in prescribed motion, but it is not clear how to apply it to a streamfunction–vorticity formulation.

A number of other Cartesian methods have been applied to problem of irregular geometry [11,15]. Udaykumar et al. [28,31] has presented a finite-volume Cartesian method, using local geometry and quadratic interpolation functions to calculate flux across the elements that intersect the immersed boundary. The resulting linear system is solved using a multi-grid method and has been shown to be applicable to moving geometry problems.

Recently, Calhoun [2] put forward a method of modeling irregular shapes in a Cartesian grid using a streamfunction–vorticity formulation. The irregular boundary is represented on the underlying flow solver using discontinuity conditions. This is an extension of a method developed by Mayo and McKenney [23] to quickly solve a Poisson’s equation on an irregular region using a regular Cartesian underlying grid. To calculate the values of the discontinuities, the method requires a solution to a rather large linear system which is eventually condensed into a manageable matrix. For a static geometry, once this system has been LU decomposed, the discontinuity values for each step may be found via backsubstitution. For a moving geometry, the linear system must be regenerated every timestep. The computational cost associated with this is prohibitive; as a result, this method is very efficient for static geometry but not practical for moving geometry. Here we develop a method of calculating discontinuity values that does not suffer such a penalty if the geometry is moving with respect to the regular grid.

In the method presented here, the underlying flow solver is also based on a streamfunction–vorticity formulation calculated on a regular Cartesian grid. Our method also corrects for the presence of the irregular moving objects by using embedded discontinuities in the streamfunction. Instead of using a linear system to couple all the variables involved, we calculate the values of the discontinuities in separate steps. Streamfunction discontinuity values associated with the no-penetration condition are calculated using a superposed homogenous solution. Boundary vorticity is generated to satisfy the no-slip condition by using an interpretation of Thom’s formula. The net result is a method that asymptotes to a computational cost of $O(N \ln(N) + M)$ where N is the number of nodes in the regular grid and M is the total surface discretization of all immersed objects.

Following this section details of the algorithm are presented as well as initial examples of its use. Section 2 briefly describes the streamfunction–vorticity formulation of the 2D Navier–Stokes equations for incompressible fluid. Section 3 introduces the details of the full method, starting with a motivating overview and moving on into the details of each step. Section 4 provides the details of the process of solving a discrete Poisson equation with embedded discontinuities, which is done in a slightly different way than has previously been published. Section 5 provides examples of the algorithm, primarily against the problem of 2D

incompressible flow past a circular cylinder. We also provide an example of multiple moving objects, two cylinders moving past one another. Finally the last section contains our conclusions and possible future work.

2. Streamfunction–vorticity formulation of the Navier–Stokes equations

In a streamfunction–vorticity formulation, the equations governing incompressible 2D flow can be written as

$$\frac{\partial \omega}{\partial t} + \mathbf{u} \cdot \nabla \omega = \nu \Delta \omega, \quad (1)$$

$$\Delta \psi = \omega, \quad (2)$$

$$\mathbf{u} = \nabla^\perp \psi = \left(-\frac{\partial \psi}{\partial y}, \frac{\partial \psi}{\partial x} \right), \quad (3)$$

$$\omega = \nabla \times \mathbf{u}, \quad (4)$$

where ω is the vorticity (a scalar in 2D flow), \mathbf{u} is the velocity, ν is the specific viscosity, and ψ is the streamfunction of the velocity field.

The equation governing vorticity evolution (1) is derived from the full Navier–Stokes equations by assuming incompressibility and taking the curl. The incompressibility assumption ($\nabla \cdot \mathbf{u} = 0$) is implicitly satisfied by the use of a streamfunction to determine the velocity, as can be easily seen by taking the divergence of Eq. (3). Note that some authors use $\Delta \psi = -\omega$ for equation (2) and use the negative of our choice for the perpendicular gradient operator ∇^\perp . The effective difference is a change in the sign of calculated ψ .

One interesting aspect of this formulation is the nature of the boundary conditions. For an object moving in the flow, the boundary conditions applied are derived from velocity conditions. Namely, that there is no penetration (normal velocity) and no slip (tangential velocity). This translates into conditions on $\frac{\partial \psi}{\partial n}$ and $\frac{\partial \psi}{\partial s}$ where n represents a direction normal to the boundary and s represents a direction tangential to the boundary.

While other boundary conditions are generally used for the far field, in this paper they will be presented as also using velocity-prescribed conditions for simplicity. On the far field, the $\frac{\partial \psi}{\partial s}$ condition may be integrated around the boundary since the streamfunction can have an arbitrary constant added without affecting the solution. This results in both Dirichlet and Neumann boundary conditions for the Poisson's equation for velocity (2) and none for the vorticity evolution equation (1). It is important to note that, for the full system of *both* equations, the proper number of boundary conditions exist to make this a well-posed problem.

3. The numerical method

3.1. Overview

One of the major difficulties that must be overcome is solving for a velocity field that is both consistent with the vorticity distribution and satisfies the boundary conditions. Lighthill [19] introduced a line of

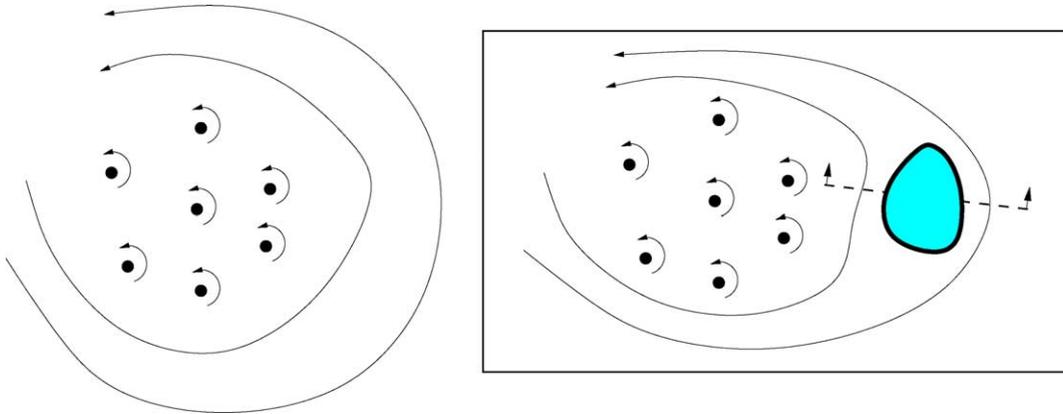


Fig. 1. Determination of velocity field by vorticity distribution and boundary conditions.

reasoning that addresses this issue. Consider a particular vorticity distribution that specifies a velocity field via the Biot–Savart law. A cartoon of a velocity distribution resulting from a simple vorticity distribution is shown in the left side of Fig. 1. The resulting velocity field is not unique in the sense that any irrotational velocity field may be superposed upon it without changing the vorticity distribution. The irrotational field can be determined by the kinematic constraint representing the no-penetration boundary condition. The superposition of these fields uniquely determines the velocity field for the problem subject to the no-penetration condition, as shown in the right side of Fig. 1.

However in general the resulting velocity field will have non-zero tangential velocities at the solid boundaries. In order to satisfy no-slip conditions, Lighthill proposes the existence of a discontinuity in the velocity field. This discontinuity would manifest itself as a singular distribution of vorticity along the boundary. The vorticity created by this “vortex sheet” would then be diffused and convected into the surrounding fluid.

In some sense the presence of an impenetrable object is communicated instantly throughout the whole of the fluid and introduces a *global* adjustment to the flow field. This adjustment does not create any additional vorticity in the system. Satisfying the no-slip condition, on the other hand, is a *local* adjustment to the flow field involving the introduction of vorticity into the system along the boundaries.

In terms of differential equations, the particular solution is given by the distribution of the right-hand-side field while a homogenous solution is given by the Dirichlet boundary conditions. The difference between the resulting streamfunction normal gradient at the boundary and the desired Neumann boundary condition implies a discontinuity in the normal gradient, which can be enforced by a singular distribution of vorticity along the boundary. A much simplified cartoon of the sequence we follow is presented in Fig. 2, where $f(s)$ represents the assigned Dirichlet boundary conditions.

Following this sequence allows us to satisfy the no-penetration and no-slip conditions in separate distinct adjustments, rather than being coupled in one large system.

To summarize, the sequence is as follows:

- Solve Poisson’s equation for velocity including the discontinuity represented by the body (vorticity zero on the inside) (ψ_p).
- Solve a homogeneous inviscid problem using boundary methods in such a way that when it is superposed upon the previous solution the no-penetration condition is satisfied (ψ_L).
- Distribute vorticity around the boundaries to satisfy the no-slip condition.
- Integrate the vorticity equation in time, including the effects of those singular sources.

Following are the details of the implementation of each step in this sequence.

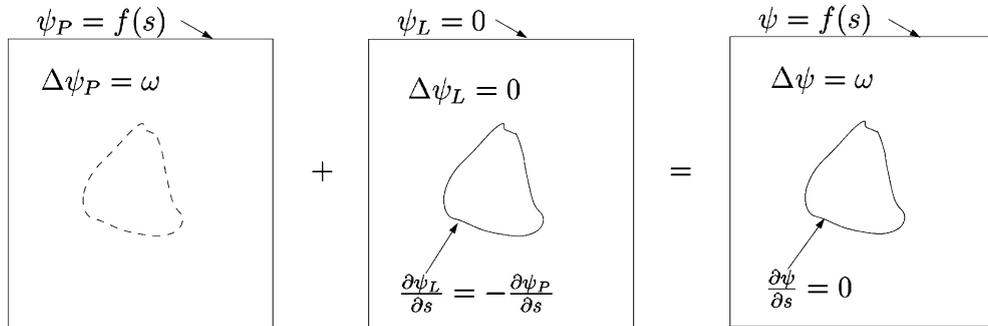


Fig. 2. Simplified cartoon of the process for finding the velocity streamfunction for a fixed geometry.

3.2. Underlying flow solver on the regular grid

For the underlying flow solver on the regular grid, we use a method recently put forward by E and Liu [8] in a paper on vorticity boundary conditions. A complete discussion of the method is not appropriate here and we refer the reader to [8], but a brief description and a highlight of some points that affect our additions for irregular geometries follows.

The method uses second-order centered finite differencing that evolves the vorticity equation *explicitly* in time using fourth-order Runge–Kutta integration. The no-slip condition on the exterior boundary is satisfied by generating vorticity determined by Thom’s formula, a process which is second-order accurate [13]. Also, note that the use of fourth-order Runge–Kutte integration in time relaxes somewhat the time-step restrictions associated with stability in an explicit method.

The Poisson’s equation can be solved using any available “black box” solver, here, and where it appears elsewhere in this paper. In our case we used an FFT solver, which takes $O(N \ln(N))$ time.

3.3. Solving Poisson’s equation

The domain of our solution will always be a regular rectangular region, chosen around the full range of motion of the objects immersed in the fluid. The underlying grid will always be regular but the discretization need not be the same in the x and y directions (see Fig. 3).

Let us assume for present purposes that we are imposing velocity boundary conditions on both the far field boundaries and on the boundaries of the immersed object. We can then describe the boundary conditions as being in both $\frac{\partial \psi}{\partial s}$ and $\frac{\partial \psi}{\partial n}$. We can integrate $\frac{\partial \psi}{\partial s}$ along the far field boundary and choose an arbitrary value (generally 0) for the integration constant, producing standard Dirichlet boundary conditions for the far field.

Let us also assume that we start with a known vorticity distribution in the fluid, as well as known values of the vorticity on the immersed object boundaries. For the first timestep both are generally zero. For subsequent steps, the vorticity in the regular grid outside the immersed object is given by integrating the vorticity evolution equation in time, and the vorticity on the boundary of the object is calculated to satisfy the no-slip condition as described in detail later.

We would like to calculate the velocity field created by that vorticity distribution by solving the equation

$$\Delta \psi_P = \omega, \tag{5}$$

where $\omega = 0$ inside the object if it has a zero angular velocity.

We can now use the discontinuity method presented in Section 4 to solve the Poisson’s equation to second-order accuracy including the effects of the discontinuity in ω along the irregular boundary. We call the streamfunction that results ψ_P .

The ψ_p field is interpolated onto the irregular boundary to calculate $\frac{\partial \psi_p}{\partial s}$ on the object boundaries. Here, and for the rest of the paper unless otherwise specified, we use fourth-order polynomial interpolation that is aware of the discontinuities at the boundaries. Specifically, the interpolation algorithm takes the presence of the discontinuities into account in order to provide a smooth interpolation function near the discontinuous boundaries.

Fig. 4 illustrates the solution found for ψ_p in the case of flow past an impulsively started circular cylinder at time 0^+ . Since there is no vorticity, either in the fluid or the boundary in the initial state, the solution found in this case is simply a uniform flow to the right.

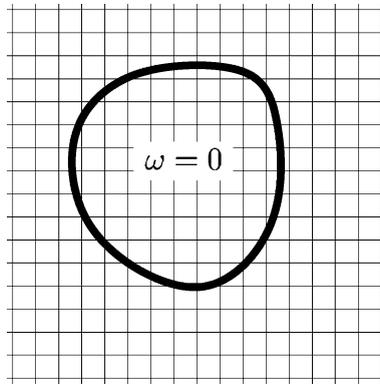


Fig. 3. Irregular geometry overlaid on regular grid, causing a discontinuity in ω .

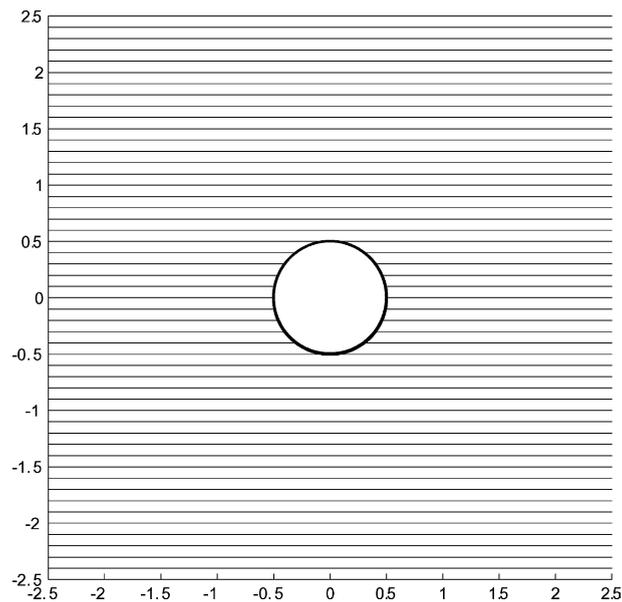


Fig. 4. Contour of ψ_p for impulsively started cylinder.

3.4. Satisfying no-penetration

In Section 3.3 we calculated values of $\frac{\partial \psi_P}{\partial s}$ on the immersed object boundaries. Note that the velocity field specified by ψ_P does not satisfy either the no-penetration or no-slip condition on the boundaries of the immersed objects. We know that ultimately we need our solution to match a given $\frac{\partial \psi}{\partial s}$ which is determined by the no-penetration condition. We solve for a value $\frac{\partial \psi_L}{\partial s}$ such that

$$\frac{\partial \psi_P}{\partial s} + \frac{\partial \psi_L}{\partial s} = \frac{\partial \psi}{\partial s}. \tag{6}$$

Now we can integrate $\frac{\partial \psi_L}{\partial s}$ to find Dirichlet conditions for the immersed object boundary and solve the system

$$\Delta \psi_L = 0 \tag{7}$$

with far-field boundary conditions

$$\psi_L = 0 \tag{8}$$

and boundary conditions on the object of

$$\psi_L(s) = \int_0^s \frac{\partial \psi_L}{\partial s} ds + C, \tag{9}$$

where C is a constant of integration.

To determine C , we appeal to Lighthill’s interpretation of the streamfunction–vorticity formulation. One can describe the homogenous correction we are seeking as an inviscid correction to the flow field. Any circulation created around an object in the flow by this correction implies a source of vorticity created somewhere within the object. We assert that this correction should not add any vorticity to the system.

This gives us a method for determining C for each object immersed in the flow. We determine what value of C will result in zero circulation added in the ψ_L correction. To do this, we require that

$$\oint \frac{\partial \psi_L}{\partial n} ds = 0. \tag{10}$$

This adds one equation and one unknown for each object immersed in the flow.

To find the full ψ_L field, we first solve the Dirichlet–Neumann map. Meaning, using the Dirichlet boundary values of ψ_L , we can solve an integral relation resulting in unique values of $\frac{\partial \psi_L}{\partial n}$. For ease of development, we currently use a simple boundary element method modified to solve for the extra unknown and constraint for each object. The underlying integral equation used is

$$\Theta(\vec{p})\psi_L(\vec{p}) = \int_S \psi_L(\vec{x}) \frac{\cos(\theta(\vec{p}, \vec{x}))}{r(\vec{p}, \vec{x})} dS - \int_S \frac{\partial \psi_L(\vec{x})}{\partial n} \ln(r(\vec{p}, \vec{x})) dS, \tag{11}$$

where $\Theta(\vec{p}) = 2\pi$ on the interior of the region and represents the interior angle if \vec{p} falls on the boundary, $\theta(\vec{p}, \vec{x})$ is the angle between the surface normal and the vector from the surface to \vec{p} , and $r(\vec{p}, \vec{x})$ is the distance from the surface point to \vec{p} . This equation is discretized, converted into a linear system on the boundary points, and solved using LU decomposition and back-substitution. This results in values of $\frac{\partial \psi_L}{\partial n}$ on the discrete boundary points.

It is possible to use a multipole method to solve this system in order M time where M represents the number of boundary elements. See for example [3,12]. We intend to implement this alternate solver at a future date. Note that the iterative solution process will be extremely efficient, as there will only be small

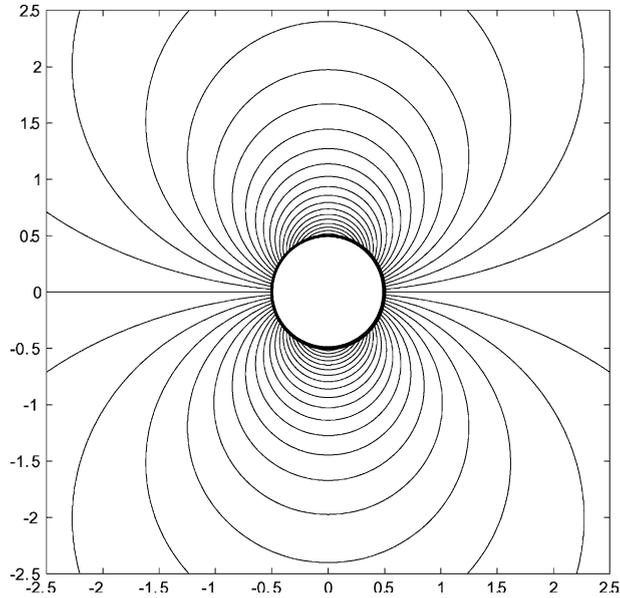


Fig. 5. Contour of ψ_L for impulsively started cylinder.

changes from timestep to timestep allowing the use of the previous steps solution as the initial conditions in the solver. The ψ_L field that results from such a computation is displayed in Fig. 5.

We now have the boundary values of ψ_L and $\frac{\partial \psi_L}{\partial n}$, which can be used in the discrete form of equation (11) to fully describe the interior ψ_L field. Actually evaluating the boundary integral equations for every point in the field would be computationally expensive, even with a multipole evaluator. However, we can once again use the concept of embedded discontinuities presented in the next section. We treat the boundary values of ψ_L and $\frac{\partial \psi_L}{\partial n}$ as discontinuities in the ψ_P problem. Now we include the effects of discontinuities in all three of $[\psi]$, $[\frac{\partial \psi}{\partial n}]$, and $[\frac{\partial^2 \psi}{\partial n^2}]$ in a discrete Poisson's equation problem. This has the effect of solving for the ψ_L field and superposing it on the ψ_P solution at the same time. The result is the discrete solution of ψ which now conforms to both the far field and the immersed object Dirichlet boundary conditions.

An example of the solution (to this point) for the impulsively started circular cylinder is shown in Fig. 6.

3.5. Satisfying no-slip

At this point, the no-penetration condition on the far field and immersed object boundaries have been satisfied. It remains to satisfy the no-slip condition. On the far field boundary, we can simply use Thom's formula exactly like the pure cavity flow case described in [8]. It seems logical, then, to seek a generalization of Thom's formula for the irregular immersed boundaries. First, however, we explore Thom's formula and pose some interpretations of it.

In a previous section, we discussed Lighthill's view of the no-slip condition being enforced by the existence of vortex sheets. These sheets represent discontinuities in the velocity field from some finite relative tangential velocity with the surface to no-slip. For consistency, if nothing else, we would like to be able to relate these vortex sheets with Thom's formula.

Thom's formula for the left boundary can be written as

$$\omega_{0,j} = \frac{2}{\Delta x^2} (\psi_{1,j} - \psi_{0,j} + \Delta x V_{0,j}) + D_{yy}[\psi_{0,j}], \quad (12)$$

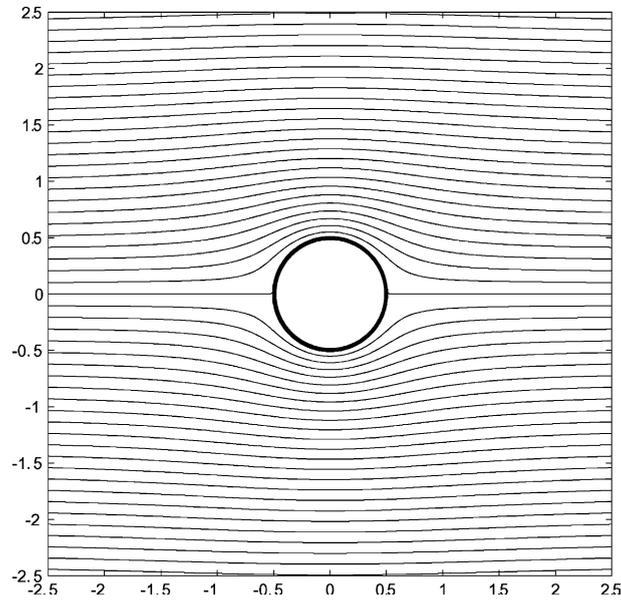


Fig. 6. Contour of ψ for impulsively started cylinder.

where V now represents the desired tangential velocity at the surface point $(0, j)$, and D_{yy} is a standard second-order centered finite difference formula. The left side of Fig. 7 shows a geometric interpretation of the derivation of Thom’s formula using a “ghost point” outside the computational domain. The “ghost point” $(-1, j)$ is positioned in such a way that the slope between it and the point $(1, j)$ matches the desired

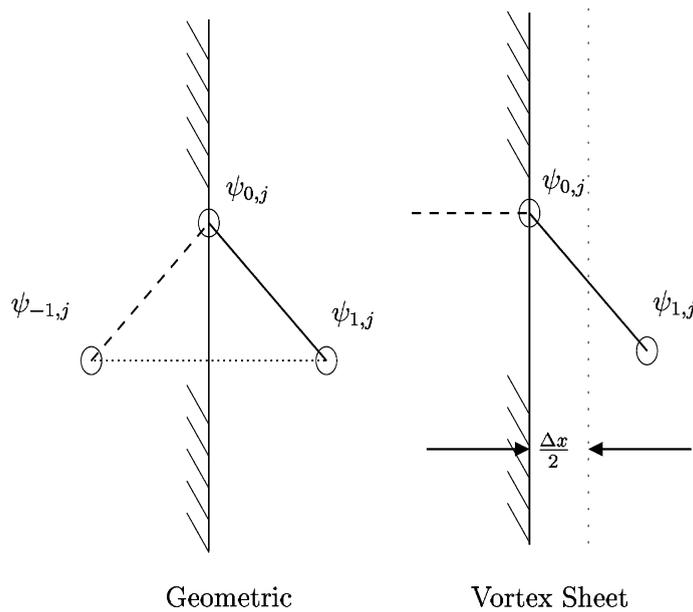


Fig. 7. Interpretations of Thom’s formula.

Neumann boundary conditions. This determines the difference in slope between $(-1, j)$ to $(0, j)$ and $(0, j)$ to $(1, j)$, which in turn determines the effective boundary vorticity.

There is another available interpretation of Thom's formula, one which can in fact be used to derive it. If we desire a discontinuity in tangential velocity of $[V]$, the vorticity sheet strength γ necessary to cause that discontinuity is simply $\gamma = -[V]$. Looking at Eq. (12), we see that the $\psi_{1,j} - \psi_{0,j}$ can be written as a first-order forward finite-difference approximation of the slope at the boundary. We can now rewrite equation (12) as

$$\omega_{0,j} = -\frac{2}{\Delta x}[V], \quad (13)$$

where $[V]$ now represents the difference between desired and actual tangential velocity.

We can see that the formula for boundary vorticity is now different from the vortex sheet strength by a factor of $2/\Delta x$. It seems that Thom's formula represents the value of vorticity a boundary node should have to represent a sheet strength of $-[V]$. It is pointless to use the value of vorticity at the sheet location, since that is infinite by virtue of it being a singular source. However, in a discrete system, the vorticity at a node can be thought of (in a very approximate sense) as representing a volume average of the total vorticity in a region it occupies. If we consider the occupying region of a boundary node to be half the distance to the next node inwards (since the other direction is outside the computational domain) we arrive at exactly Thom's formula.

It is troubling that the value of vorticity at the boundary is grid-dependant in the sense that it contains a Δx term. This brings up the issue of how to generalize Thom's formula for an irregular boundary, since there is no clear consistent Δx . It is also important to realize that any arbitrary distance will not suffice; whatever Δx is chosen at a point on the irregular boundary must be consistent with the finite difference equations used to evolve the vorticity.

The choice of Δx and its integration into the vorticity evolution equation is left to the next section. For now, assuming we *have* a value of Δx , we still need to calculate an effective boundary vorticity. Since the desired tangential velocity is known, we need to calculate an accurate value of actual tangential velocity or, equivalently, the normal slope of the potential field. We do this by calculating $\frac{\partial \psi_p}{\partial n}$ at the same time as $\frac{\partial \psi_p}{\partial s}$ is calculated in the previous section and in the same manner. Since $\frac{\partial \psi_p}{\partial n}$ is directly returned by the Dirichlet-Neumann map solved on the boundary, the superposition gives us $\frac{\partial \psi}{\partial n}$ or the tangential velocity at the surface.

We observed that for a moving geometry the vorticity could show unstable behavior unless a small amount of under-relaxation was used in the generation of the vorticity on the surface. It is generally necessary to apply a fraction of the correction to insure stability.

It turns out that reducing the generated vorticity in this way has a surprisingly small effect on the steady-state result. Even using 50% of the correction produces less than a 7% difference in the calculated coefficient of drag for the circular cylinder case at $Re = 40$, for instance. It does, however, impact the transient results. Experimentation has shown that applying 90% of the correction provides a more than adequate margin of safety, but this is somewhat problem dependent. Most of the results presented below used values of either 100% or 95%.

3.6. Integrating vorticity

We have now shown how to solve the Poisson's equation to determine the velocity field and how to generate vorticity on the boundaries to satisfy the no-slip condition. We have not yet shown how to introduce the singular vorticity and the discontinuities in the flow field into the discrete vorticity evolution equation

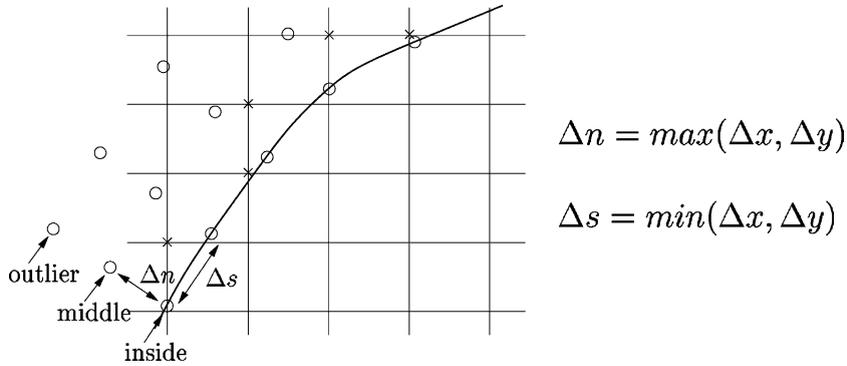


Fig. 8. Local overset grid.

$$\begin{aligned} \frac{\partial \omega_{i,j}}{\partial t} = & -\frac{u}{2\Delta x}(\omega_{i+1,j} - \omega_{i-1,j}) - \frac{v}{2\Delta y}(\omega_{i,j+1} - \omega_{i,j-1}) + \frac{v}{\Delta x^2}(\omega_{i-1,j} - 2\omega_{i,j} + \omega_{i+1,j}) \\ & + \frac{v}{\Delta y^2}(\omega_{i,j-1} - 2\omega_{i,j} + \omega_{i,j+1}). \end{aligned} \tag{14}$$

The problem lies in how to handle the exterior irregular points, marked as \times in Fig. 8. These are points whose values must be calculated each timestep but at which the finite-difference stencil crosses the discontinuity. Our goal is to find a way to calculate values for these points at each step without drastically reducing our stable time step.

One possibility would be to make use of the same discontinuity tools used in previous sections. However, to do this with second-order accuracy would require knowing the values of $\left[\frac{\partial \omega}{\partial n}\right]$ on the boundary (and preferably $\left[\frac{\partial^2 \omega}{\partial n^2}\right]$ as well). We do not have a method for determining this without performing finite differencing on the surrounding values of vorticity, which would introduce serious stability problems.

Gibou [11] is an example of a method of introducing irregular Dirichlet boundaries into a regular cartesian grid solving a diffusion equation, in this particular case a transient heat equation. However, the method requires the use of an implicit solver to avoid drastic timestep decreases. Our method of finding boundary vorticity currently requires the use of an explicit method.

Instead we turn to a thin grid (three nodes deep) which conforms to the surface of the object, as seen in Fig. 8. We refer to this as the “local” grid. The outside row of nodes, which we call the *outlier* nodes, have their vorticity values calculated by interpolating on the surrounding grid. The inner row of nodes, which are attached to the surface of the object, have their values determined by Thom’s formula as described in the previous section. The middle row of nodes have their values calculated by applying the vorticity evolution equations to the *local* grid. The $\Delta \omega$ term is now being calculated in a local curvature system, so a local curvature term is added

$$\Delta \omega = \frac{\partial^2 \omega}{\partial n^2} + \kappa \frac{\partial \omega}{\partial n} + \frac{\partial^2 \omega}{\partial s^2}. \tag{15}$$

These middle values must be maintained and evolved in a manner consistent with Runge–Kutta time integration. Notice that this system, which is essentially a minimalist use of patched or overset grids (sometimes called a chimera grid), allows a consistent Δx to be used in Thom’s formula at the boundary. We choose the spacing of this patched grid in such a way that the critical timestep is not affected, and yet all exterior irregular points are guaranteed to fall between the first two rows of the local grid. Irregular points are defined as points whose five point centered finite-difference stencil crosses the immersed objects boundary.

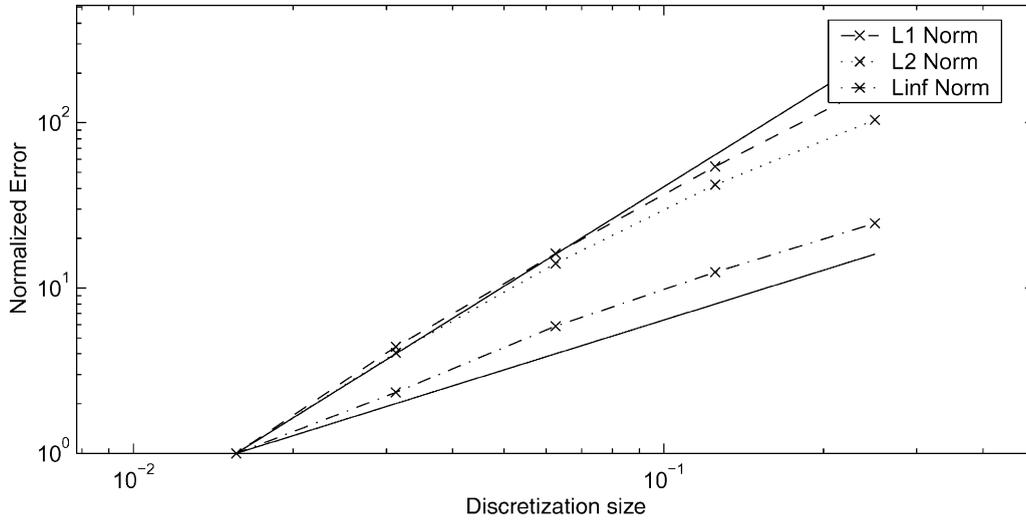


Fig. 9. Convergence test for $n = 64, 128, 256, 512, 1024$ compared to $n = 2048$ case. Flow around cylinder at $Re = 40$ and time = 16. Curves were normalized so that their smallest values were 1.0 for comparison.

The exterior irregular points are the only points which are affected by the presence of the immersed body. Irregular points on the interior of the body have their vorticity determined by the body’s angular velocity. The vorticity at exterior irregular points is determined by interpolating on the *local* grid, using fourth-order interpolation in the tangential direction and third-order interpolation in the radial direction.

We note that the use of interpolation is unfortunate in that it introduces error in the region that is of the highest interest and that is most sensitive to small perturbations. It is hoped that the use of fourth-order interpolation wherever possible (and third-order where it is not) keeps the error tolerable.

To probe the effect of this error, we performed a convergence test for flow past a cylinder at $Re = 40$ run to a time of 16. The grid discretization was increased by powers of two, starting at 64×64 and ending at 2048×2048 . Fig. 9 shows the results for three norms of the error. Shown for comparison are the lines indicating first-order and second-order convergence. As expected, overall convergence is of second-order. The infinity norm, however, predicts only slightly higher than first-order accuracy. Since the largest error always falls on or near the boundary, this effectively indicates the order of accuracy of the local grid approximation.

The above process provides us with three things: a method of determining boundary vorticity that is consistent with the finite-difference system being used, a method of determining vorticity values at exterior irregular points and, finally, a method that does not provoke any penalties in stable timestep. It is, however, the source of the largest errors introduced into the system. We are actively pursuing in other more optimal methods of including the effects of the immersed boundary in the evolution equations.

3.7. Calculating forces

The viscous drag (skin friction) and pressure drag for a general body immersed in a fluid can be calculated by

$$\mathbf{F}_v = \nu \oint_{\Gamma} \omega \hat{\mathbf{t}} ds, \tag{16}$$

$$\mathbf{F}_p = -\nu \oint_{\Gamma} \int_0^s \frac{\partial \omega}{\partial n} \hat{\mathbf{n}} ds' ds + A \mathbf{a}, \tag{17}$$

where it is assumed unit density ($\rho = 1$), A is the area of the immersed body, and \mathbf{a} is the acceleration of the center of mass.

The distribution of ω on the surface is given directly by our method. The value of $\frac{\partial\omega}{\partial n}$ is not, unlike Calhoun's [2] method. Values of $\frac{\partial\omega}{\partial n}$ must be calculated in some manner from available data. This proves to be somewhat difficult.

Finite differencing, taking advantage of the local grid, uses information at some distance from the object. If this distance is on the same order of magnitude as the boundary layer, the values obtained are unreliable. We would also suspect that the results would show poor resolution in time, as changes in the vorticity distribution would need to be distributed some distance away from the surface before they would be reflected in the results.

Another method has been tried; it is based on a means of relating normal gradient of vorticity to vorticity sheet strength at the surface as described in a paper by Koumotsakos et al. [16]. Currently, in practice the errors introduced by the local grid get amplified and dominate the results. It shows promise however, and has much improved transient behavior. Work is still ongoing in this direction.

For the results shown in this paper, we use first-order finite differencing on the local grid to calculate $\frac{\partial\omega}{\partial n}$. This helps to minimize the distance from the surface of the object used to estimate the gradient.

3.8. Summary and sequence

The following is a brief description of the steps taken in this method every Runge–Kutta substep, followed by the order of operations required. For order calculations, N is the number of grid nodes and M is the number of elements in the boundary of the objects. The order values given assumes that the length of boundary element is the same order as the grid spacing to avoid complexity involving the ratios of those two lengths.

- Explicitly evolve vorticity in time on local and global grids, $O(N + M)$.
- Determine geometric information if the geometry is moving, $O(M)$.
- Interpolate values of vorticity on exterior irregular nodes from the local grid, $O(M)$.
- Calculate an initial streamfunction (ψ_p) by solving Poisson's equation with one discontinuity type, $O(N \ln(N) + M)$.
- Calculate Dirichlet boundary conditions for the inviscid correction (ψ_L) to within a free constant for each object, $O(M)$.
- Solve Dirichlet–Neumann map, resulting in discontinuity values on the boundary of immersed objects. Currently $O(M^3)$, possible in $O(M)$.
- Calculate the final streamfunction (ψ) by solving Poisson's equation with three discontinuity types, $O(N \ln(N) + M)$.
- Determine boundary vorticity via Thom's formula, $O(M)$.
- Interpolate values of vorticity on exterior irregular nodes from the local grid, $O(M)$.
- Calculate velocity field from streamfunction, $O(N)$.

Note that, assuming a multipole solver is implemented, the dominant terms in this process are $O(N \ln(N))$ as N becomes large. Also note that, if the geometry is not moving relative to the regular grid, the geometric information need not be calculated, and even the relatively inefficient boundary element method we are currently using reduces to a back-substitution each timestep.

4. Incorporating discontinuities into a Poissons equation solver

We dedicate this section to a full description of the method used to incorporate discontinuity conditions into a Poisson's equation on a regular grid. This tool is essential to the method being described. Using a regular underlying grid, we overlay an irregular boundary with known discontinuities in $[\psi]$, $[\frac{\partial\psi}{\partial n}]$, and $[\frac{\partial^2\psi}{\partial n^2}]$.

The effects of the presence of this boundary on the discrete vorticity distribution can be found with second-order accuracy, allowing a subsequent Poisson’s equation solution to be second-order accurate.

Mayo [21–23] put forward a method of solving the Poisson’s equation numerically on an irregular region using a regular underlying grid. The presence of the irregular boundary was included with second-order accuracy by treating the boundary as a discontinuity in the values of vorticity at those locations, resulting in a discontinuity in the second derivative of the streamfunction, as well.

Leveque [18] later generalized the approach, showing a method of maintaining second-order accuracy for equations of the form $\nabla \cdot (\beta \nabla \psi) + \kappa \psi = f$, including discontinuities in β , κ , and f . Leveque’s method requires the solution of a 6×6 system of equations for each irregular point on the boundary.

In our particular case, there is no need for a discontinuity in β , so we can automatically simplify Leveque’s formulation somewhat. In practice, we use a method that allows an explicit correction for the presence of the discontinuity. There is no need to solve a system of equations.

4.1. A 1D example

To better understand the concepts involved, it is instructive to look at a simple 1D system. Specifically, consider the problem

$$\frac{\partial^2 \psi}{\partial x^2} = \omega, \tag{18}$$

$$\omega = \delta(x^*) \left[\frac{\partial \psi}{\partial x} \right], \tag{19}$$

$$\psi(0) = 0, \tag{20}$$

$$\psi(L) = 0. \tag{21}$$

The solution to this problem is a sequence of two lines joined at x^* whose difference in slope is $\left[\frac{\partial \psi}{\partial x} \right]$, as seen in Fig. 10.

A center-difference-scheme formulation of this problem could be given as

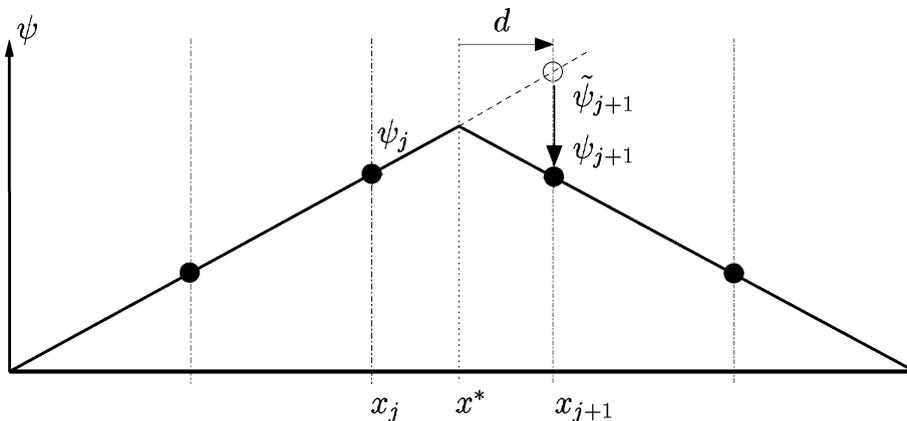


Fig. 10. Analytical and discrete solution to example 1D problem.

$$\frac{\psi_{i+1} - 2\psi_i + \psi_{i-1}}{\Delta x^2} = \omega_i, \quad (22)$$

$$\psi_0 = 0, \quad (23)$$

$$\psi_N = 0. \quad (24)$$

Thus, given ψ_i , the above difference scheme gives zero ω_i 's everywhere except for those locations whose stencil crosses the discontinuity at x^* , which we call x_j and x_{j+1} . Specifically for point x_j ,

$$\omega_j = \frac{d \left[\frac{\partial \psi}{\partial x} \right]}{\Delta x^2}. \quad (25)$$

This immediately suggests that when solving the inverse problem, in order to recover ψ_i correctly using the same finite difference scheme, ω_j has to be modified from its continuous value by the amount specified in Eq. (25). The vorticity at point x_{j+1} must also be adjusted in a similar manner.

This result can be generalized to discontinuities in ψ and any of its derivatives, resulting in a Taylor-like expansion

$$\omega_j = \tilde{\omega}_j + \frac{[\psi] + d \left[\frac{\partial \psi}{\partial x} \right] + \frac{1}{2} d^2 \left[\frac{\partial^2 \psi}{\partial x^2} \right] + \dots}{\Delta x^2}. \quad (26)$$

It is worth noting that an alternative method for handling the discontinuity is expressing it terms of a delta function and approximating the delta function by a distribution function of a finite width, as is done, for example, in Peskin's immersed boundary method. A difficulty there is to find an appropriate distribution, which, in general, is unknown a priori. In a sense, our method uses the finite difference scheme to determine the distribution function, which turns out to be a hat function for the centered-difference scheme. Also, note that the above approach can be extended in a straightforward way into multiple dimensions (as is shown in the next section), while finding the appropriate multi-dimensional distribution functions can be cumbersome.

4.2. An efficient 2D implementation

The points affected by the presence of the boundary are illustrated in Fig. 11. Points whose standard five-point finite difference stencil crosses the boundary are called *irregular points*, following the convention introduced by Mayo. It is these points whose values of discrete vorticity must be modified.

Referring to Fig. 11, we would like to find the correction to the value of ψ_Δ that a $D_{xx}[\psi_\diamond] + D_{yy}[\psi_\diamond]$ operation would see *if the discontinuity were not present*. In other words, if the values of ψ^+ , $\frac{\partial \psi^+}{\partial n}$, and $\frac{\partial^2 \psi^+}{\partial n^2}$ continued on as they were without interruption on the \ominus side. Armed with this correction to ψ_Δ , which we will call ψ_Δ^c , we can calculate the effective ω_\diamond using standard finite-difference formulas.

In our approach we find the nearest normal to the boundary for every irregular point (the point marked as \bullet in Fig. 11). In some sense this is the most accurate location to choose, since we know the discontinuities in terms of the normal derivative. Also, it simplifies expressions for ψ_Δ^c since cross terms in the s direction disappear. The final correction formula is a simple Taylor series expansion

$$\psi_\Delta^c = [\psi_\bullet] + d \left[\frac{\partial \psi_\bullet}{\partial n} \right] + \frac{1}{2} d^2 \left[\frac{\partial^2 \psi_\bullet}{\partial n^2} \right], \quad (27)$$

where d is the distance to the nearest normal point. Note that very careful attention must be paid to signs, and to which direction the normal is defined. Our sign conventions are that $[\psi] = \psi^+ - \psi^-$, where

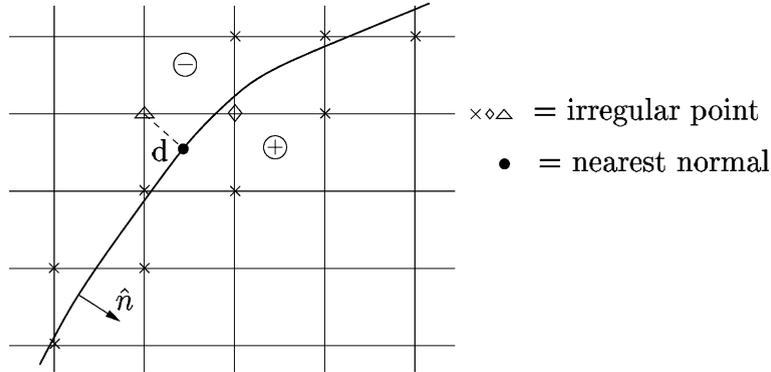


Fig. 11. 2D geometry of an irregular discontinuity.

+ represents the positive normal side. Also, the distance to the interface d is positive for points on the positive normal side and negative for points on the other side. In our example in Fig. 11 the distance d has a negative value.

For our particular implementation we do not directly know $[\frac{\partial^2 \psi}{\partial n^2}]$, instead we know $[\omega]$. If we assume a local curvature coordinate system we can write

$$[\omega] = \left[\frac{\partial^2 \psi}{\partial n^2} \right] + \kappa \left[\frac{\partial \psi}{\partial n} \right] + \left[\frac{\partial^2 \psi}{\partial s^2} \right], \tag{28}$$

where κ is the local curvature at that point. And since $[\frac{\partial^2 \psi}{\partial s^2}]$ is uniquely determined by the distribution of $[\psi]$ we can ultimately write

$$\psi_{\Delta}^c = [\psi_{\bullet}] + d \left[\frac{\partial \psi_{\bullet}}{\partial n} \right] + \frac{1}{2} d^2 \left([\omega] - \kappa \left[\frac{\partial \psi_{\bullet}}{\partial n} \right] - \left[\frac{\partial^2 [\psi_{\bullet}]}{\partial s^2} \right] \right). \tag{29}$$

The final correction to the vorticity is then

$$\omega_{\diamond} = \omega_{\diamond} - \frac{\psi_{\Delta}^c}{\Delta x^2}. \tag{30}$$

In practice we use the following sequence. We first calculate and store ψ^c for all irregular nodes using Eq. (29). We then calculate the correction to ω for each irregular point using the values of ψ^c for those nodes in its stencil that cross the boundary.

The geometric information required can be found in $O(M)$ time, where M is the number of points discretizing the boundary. This assumes that the length of a boundary segment is the same order of magnitude as the grid spacing.

4.3. Convergence test

In the description of his method for calculating discontinuity correction values, Leveque shows second-order accuracy analytically. This calculation has not been done for our alternate method, but we have performed tests on the same example problems performed by Leveque. Fig. 12 shows the geometry used for all the examples, a circle inside a simple rectangular region. Different equations for ψ are assumed to exist on the inside and the outside of the cylinder, resulting in discontinuities along the circular boundary.

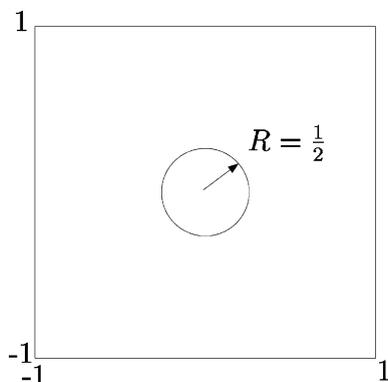


Fig. 12. Geometry of the 2D discontinuity tests.

Table 1 summarizes the results for those problems that were tested by Leveque as well. The first column displays the equation for ψ used on the inside of the circle, and the second that used on the outside. The third column shows the resulting discontinuity conditions on the circular boundary for that problem. The problems were calculated using an $n \times n$ grid, where n is listed in the fourth column. The fifth column presents the errors calculated from the analytical solutions on the inside and outside, with Leveque’s results shown for comparison. The error norm used was $\|E_n\|_\infty = \max |u(x_i, y_i) - u_{i,j}|$, or the maximum absolute error over any gridpoint. We clearly show second-order convergence, with our errors being either slightly better or slightly worse than Leveque depending on the problem. For a problem that can be represented by a quadratic polynomial we show convergence to numerical limits for all discretizations, as expected.

Table 1
Results of 2D discontinuity tests

Inside	Outside	Discontinuities	n	$\ E_n\ _\infty$		
				Leveque	Present	Ratio
1	$1 + \ln(2r)$	$[\frac{\partial\psi}{\partial n}, \frac{\partial^2\psi}{\partial n^2}]$	20	2.3908×10^{-3}	6.1254×10^{-3}	–
			40	8.3461×10^{-4}	1.2600×10^{-3}	4.87
			80	2.4451×10^{-4}	3.1379×10^{-4}	4.01
			160	6.6856×10^{-5}	7.7495×10^{-5}	4.05
			320	1.5672×10^{-5}	1.8041×10^{-5}	4.21
$e^x \cos y$	0	$[\psi], \frac{\partial\psi}{\partial n}$	20	4.37883×10^{-4}	2.3669×10^{-4}	–
			40	1.07887×10^{-4}	6.6066×10^{-5}	3.98
			80	2.77752×10^{-5}	1.5251×10^{-5}	3.95
			160	7.49907×10^{-6}	3.8565×10^{-6}	4.09
			320	1.74001×10^{-7}	9.4390×10^{-7}	3.97
$x^2 - y^2$	0	$[\psi], \frac{\partial\psi}{\partial n}$	20	–	2.5535×10^{-15}	–
			40	–	2.7478×10^{-15}	–
			80	–	1.1546×10^{-14}	–
			160	–	1.2812×10^{-14}	–
			320	–	2.6298×10^{-14}	–

5. Results

5.1. Flow past a stationary circular cylinder

As a first test of the complete system, we modeled flow past a stationary circular cylinder. This allowed us to test the solution process without the complication of the immersed object moving relative to the regular grid. The geometry used for this test is shown in Fig. 13. The boundary conditions used were:

- $\psi = f(s), \frac{\partial \psi}{\partial n} = g(s)$ on the left boundary,
- $\psi = f(s), \omega = 0$ on the top and the bottom,
- $\frac{\partial \psi}{\partial n} = 0, \frac{\partial \omega}{\partial n} = 0$ on the right,

where $f(s)$ and $g(s)$ are determined by the analytical solution to the problem of potential flow past a circular cylinder.

The flow was calculated for several different Reynold’s numbers, controlled by altering the value of specific viscosity. The far field stream velocity was held constant at $U_\infty = 1$. The base calculations were done on a 640×320 grid, then run again on a 1280×640 grid by both extending the far-field twice as far away and by doubling the discretization density. The 640×320 discretizations resulted in a grid spacing of 0.05, or 20 nodes across the cylinder. All output was non-dimensionalized by $\hat{\omega} = \omega r / U_\infty, \hat{\psi} = \psi / r U_\infty,$ and $\hat{t} = U_\infty t / r.$

Fig. 14 shows flow details around a particular example: flow at $Re = 20$. The flow was run to a time of 128, more than sufficient to reach a steady state. Values of nodes on the interior near the surface have been

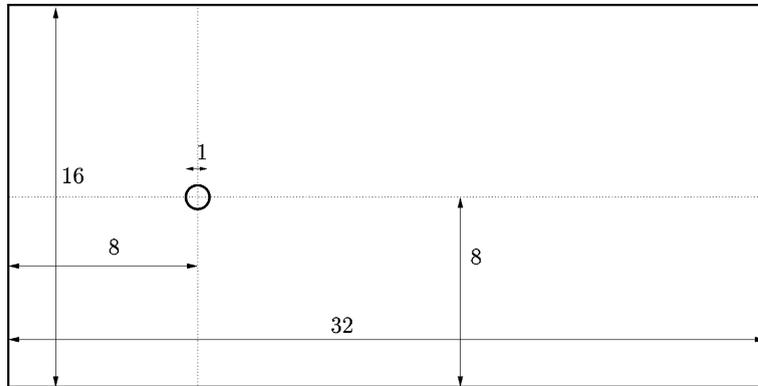


Fig. 13. Geometry used for stationary cylinder examples.

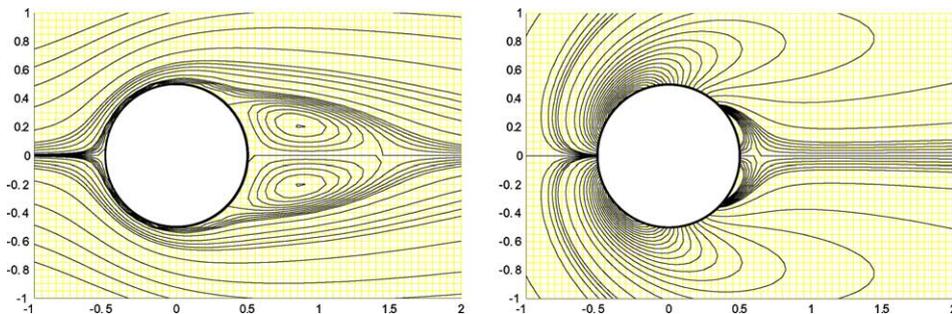


Fig. 14. Streamlines and vorticity contours for $Re = 20$ cylinder. Stream contour values are $-1:0.2:1, -0.1:0.02:0.1,$ and $-0.01:0.002:0.01.$ Vorticity contour values are $-4:0.2:4$ and $-0.1:0.01:0.1.$

extrapolated from discontinuity values at the surface in order to provide smoother contours. As can be seen, the method gives very good results for flow given the limits of the underlying discretization.

Figs. 15 and 16 provide a wider view of the flow around a cylinder at $Re = 40$.

Table 2 shows a summary of results for the cases with steady-state solution. As Calhoun [2] did in her recent paper, we compared length of the trailing bubble, angle of separation, and coefficient of drag. The geometric results compare very favorably with both experimental and previous computational results, although for the base case the values of coefficient of drag calculated were slightly higher than previous numerical studies using non-Cartesian methods.

We also ran two different cases to test possible causes of the artificially high drag calculated. First, we ran a case with twice the discretization (four times the number of nodes), which, surprisingly, resulted in a slight increase in the calculated steady state drag. This is an indication that our method of calculating the surface vorticity gradient is slightly underpredicting the numeric results. Secondly, we ran a case with the same discretization but with twice the distance to the outer boundaries (and again four times the number of nodes). In this case the calculated drag was reduced considerably, indicating that our simplification of the far field boundary conditions is a source of increased drag at this Reynold's number range.

Between $Re = 40$ and $Re = 50$ we expect to see a transition to instability. At $Re = 50$ the system still maintains symmetry at a time of 128, so we artificially perturbed the system to observe the response. As can

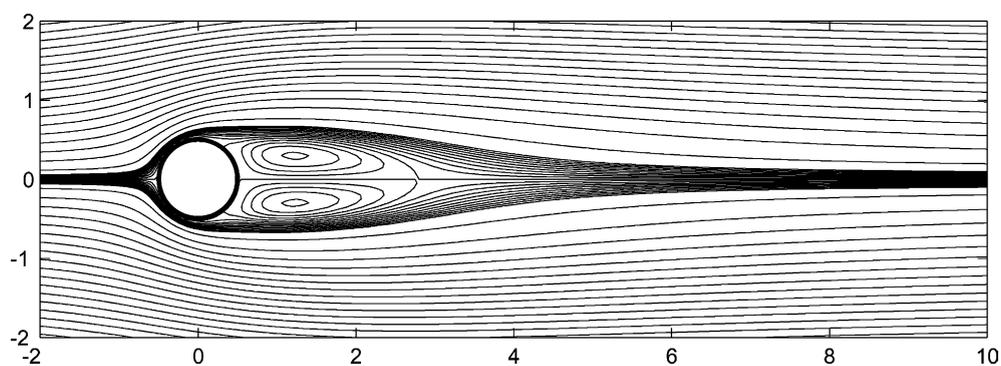


Fig. 15. Streamlines for $Re = 40$ cylinder. Contour values are $-4:0.2:4$ and $-0.1:0.01:0.1$.

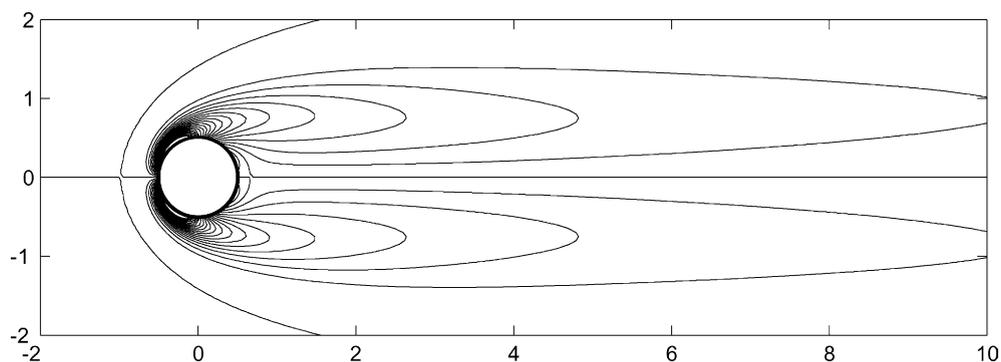


Fig. 16. Vorticity for $Re = 40$ cylinder. Contour values are $-4:0.2:4$.

Table 2
Summary of results for $Re = 20$ and $Re = 40$

	$Re = 20$			$Re = 40$		
	L	θ	C_D	L	θ	C_D
Tritton [27]	–	–	2.22	–	–	1.48
Coutanceau and Bouard [6]	0.73	42.3°	–	1.89	52.8°	–
Fornberg [10]	0.91	–	2.00	2.24	–	1.50
Dennis and Chang [7]	0.94	43.7°	2.05	2.35	53.8°	1.52
Calhoun [2]	0.91	45.5°	2.19	2.18	54.2°	1.62
Present base case	0.94	43.3°	2.13	2.29	53.1°	1.60
2 × Disc.	0.93	43.9°	2.17	2.25	53.8°	1.63
2 × Far field	0.93	43.2°	2.02	2.32	53.1°	1.52

Base case results are compared against the results of doubling the discretization density and doubling the far field distance.

be seen in Fig. 17 we successfully predict the onset of instability between those two Reynolds numbers. At $Re = 40$ the perturbation oscillations in lift damp out and at $Re = 50$ they amplify.

Fig. 18 shows the expected trailing Von Karman vortex street at a $Re = 100$. This particular simulation was run to a time of 696 to ensure full development of the quasi-steady-state condition. Fig. 19 shows the evolution in coefficient of lift and drag versus time. The behavior is as expected, with the period of low drag

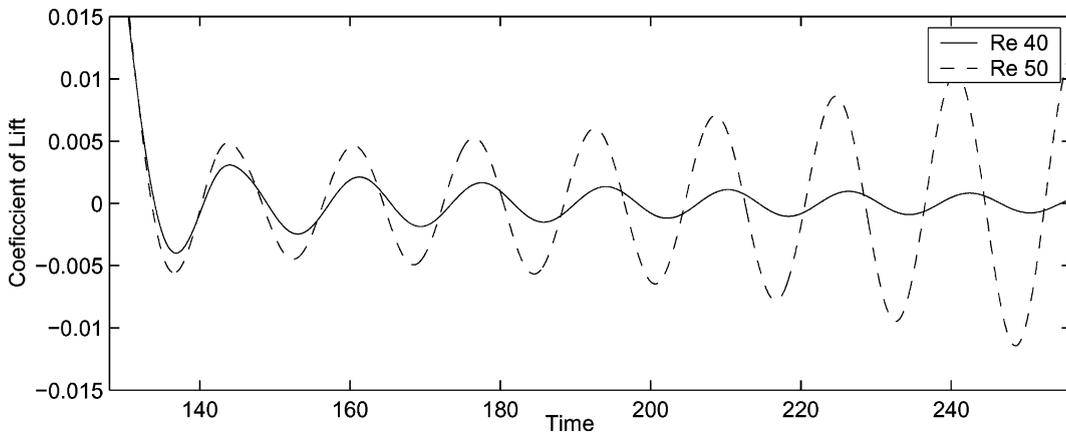


Fig. 17. Coefficient of lift vs time after a perturbation for $Re = 40$ and $Re = 50$, showing transition to instability.

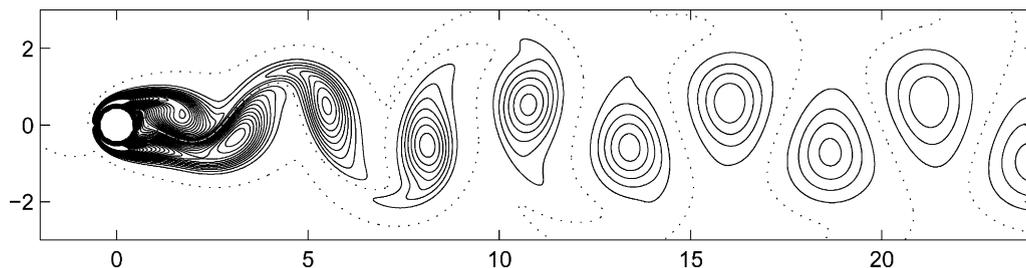


Fig. 18. Vortex street behind cylinder at $Re = 100$. Contour values are $-2.5:0.1:2.5$. Dotted line is zero contour.

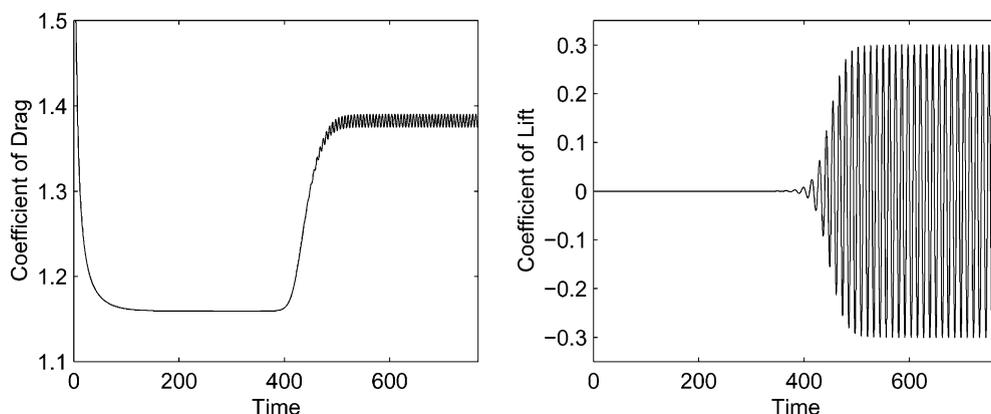


Fig. 19. Coefficient of lift and drag vs time for cylinder at $Re = 100$. Oscillations in lift build exponentially from the start till they reach the final amplitude.

before the onset of instability determined by the amount of time it takes for the amplification in oscillations about the coefficient of lift to build up.

Fig. 20 examines the evolution of vorticity around the surface of the cylinder at $Re = 100$. We directly compare against results obtained by Collins [4].

Table 3 provides a summary of results for the higher Reynold’s number cases studied. In general the magnitudes of the oscillations in C_D are underpredicted, and at $Re = 200$ the oscillations in C_L are underpredicted as well. We suspect that this is due to our method of calculating $\frac{\partial \omega}{\partial n}$ at the object surface, and that a more reliable method of calculating the pressure forces on the objects would help alleviate this. An increase in the far-field distance appears to have much less of an impact on the results, which is expected as the Reynold’s number increases.

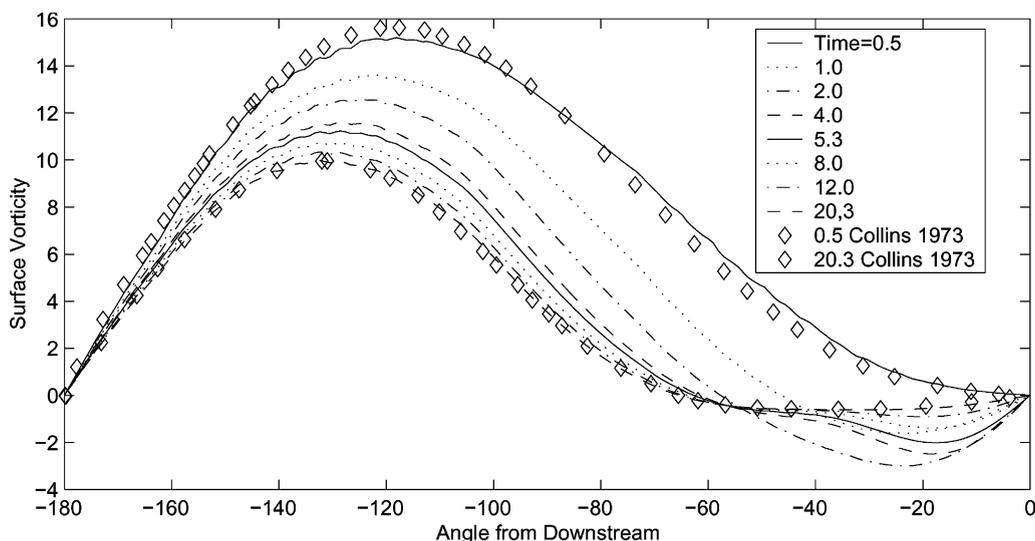


Fig. 20. Distribution of vorticity on cylinder surface at $Re = 100$ and various times.

Table 3
Summary of results for $Re = 100$ and $Re = 200$

	$Re = 100$			$Re = 200$		
	C_D	C_L	St	C_D	C_L	St
Rosenfeld et al. [26]	–	–	–	1.31 ± 0.04	± 0.65	0.20
Wright et al. [30]	–	–	–	1.33 ± 0.04	± 0.68	0.196
Braza et al. [1]	1.36 ± 0.015	± 0.25	–	1.40 ± 0.05	± 0.75	–
Liu et al. [20]	1.35 ± 0.012	± 0.339	0.164	1.31 ± 0.049	± 0.69	0.192
Calhoun [2]	1.33 ± 0.014	± 0.298	0.175	1.17 ± 0.058	± 0.67	0.202
Present base case	1.38 ± 0.007	± 0.300	0.169	1.29 ± 0.022	± 0.50	0.195
2 × Disc.	1.43 ± 0.009	± 0.322	0.172	1.45 ± 0.036	± 0.63	0.201
2 × Far field	1.34 ± 0.007	± 0.276	0.165	1.26 ± 0.026	± 0.47	0.192

Base case results are compared against the results of doubling the discretization density and doubling the far field distance.

More disturbing is the increase in drag as the discretization density is increased. Once again the method of calculating $\frac{\partial \omega}{\partial n}$ is probably to blame, slightly underpredicting pressure forces.

5.2. Flow past a moving circular cylinder

The true test of the value added to our method lies in modelling objects which move relative to the underlying grid. As a first test of this, we again model cylindrical flow, but for a cylinder moving to the left at a velocity of 1. We used the same geometry as in the previous example except the cylinder starts out 16 units to the right. The boundary conditions used are:

- $\psi = 0, \frac{\partial \psi}{\partial n} = 0$ on the left boundary.
- $\psi = 0, \omega = 0$ on the top and the bottom.
- $\frac{\partial \psi}{\partial n} = 0, \frac{\partial \omega}{\partial n} = 0$ on the right.

These boundary conditions are chosen to be approximately equivalent to those used in the previous example, adjusting for the new frame of reference.

Figs. 21 and 22 show the results after a time of 32, when the cylinder has arrived in the same position as the cylinder in the fixed case.

We can adjust for the different frame of reference by adding a linear gradient onto the streamfunction to directly compare against the fixed case. The results of such an adjustment are shown in Fig. 23.

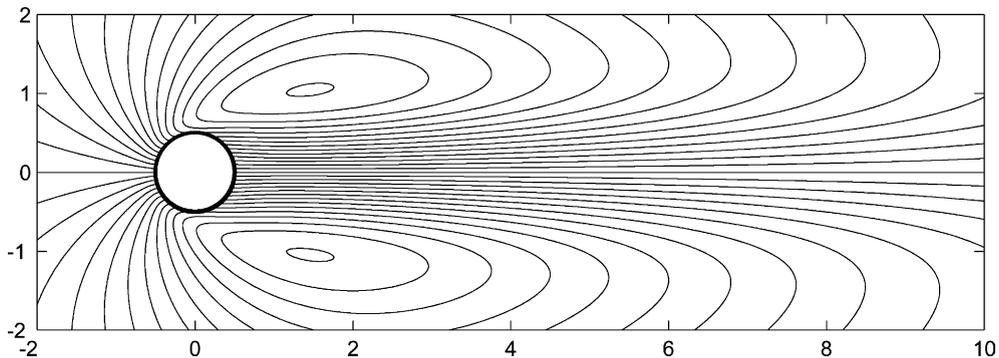


Fig. 21. Streamlines for $Re = 40$ moving cylinder at a time of 32. Contour values are $-1.4:0.1:1.4$.

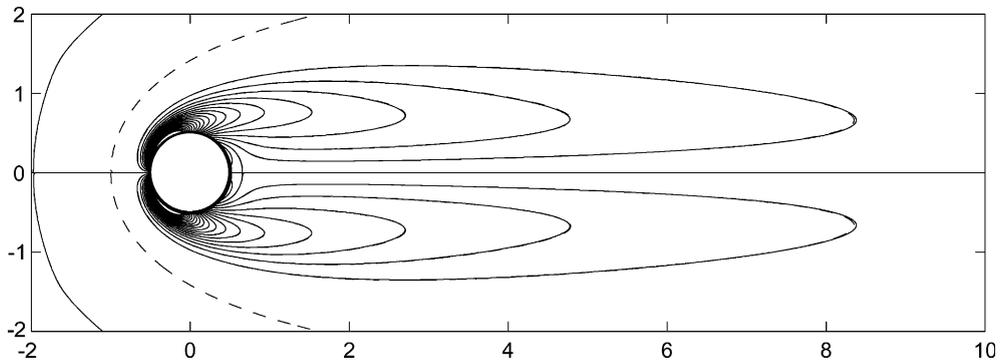


Fig. 22. Vorticity for $Re = 40$ moving cylinder at a time of 32. Contour values are $-4:0.2:4$. Dashed lines are the contours for the fixed case at the same time.

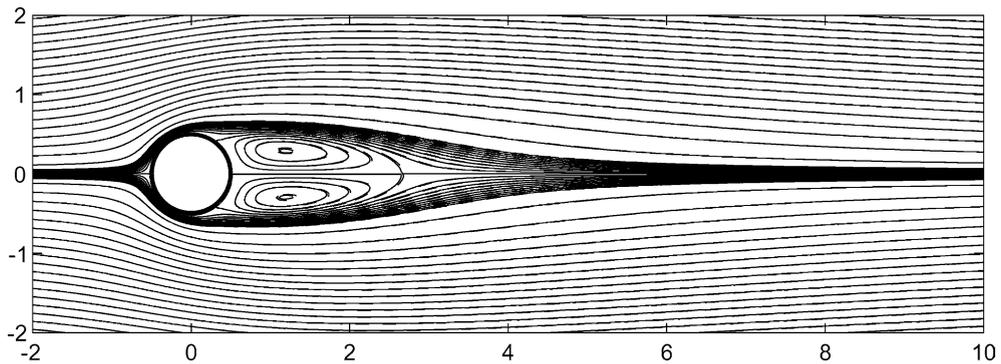


Fig. 23. Adjusted streamlines for $Re = 40$ moving cylinder. Contour values are $-4:0.2:4$ and $-0.1:0.01:0.1$. Dashed lines are the contours for the fixed case at the same time.

Table 4 summarizes the results for the $Re = 40$ case run, which are compared against the results for the fixed case at the same time. As the cylinder passes through the grid, the specifics of the geometric interaction change. This causes “noise” in some of the results. The magnitude of this noise is reflected in the error terms for the coefficient of drag.

Fig. 24 shows a comparison between coefficient of drag histories for the moving and the fixed cylinder case. One can clearly see the “noise” caused by changing error terms as the cylinder moves through the grid.

Table 4
Summary of results for $Re = 40$ moving cylinder, compared against fixed case at the same time

	L	θ	C_D
Fixed base case	2.21	52.9°	1.62
2 × Disc.	2.18	53.6°	1.65
Moving base case	2.17	52.0°	1.59 ± 0.01
2 × Disc.	2.15	53.2°	1.66 ± 0.01

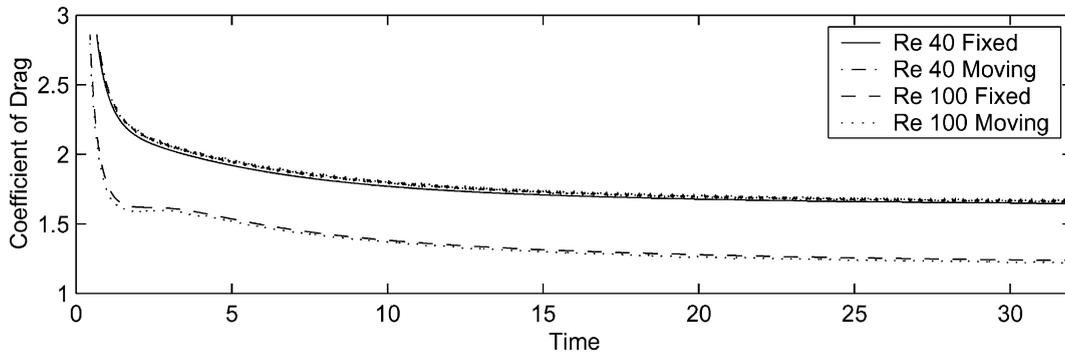


Fig. 24. Comparison of drag histories for moving and fixed cylinder at $Re = 40$ and $Re = 100$.

The slight difference in values are probably caused by the difference in effective boundary conditions between the two cases.

5.3. Flow past two cylinders moving with respect to each other

This example provides our first demonstration of flow past multiple moving objects; in this case, flow past two cylinders moving with respect to one another. The initial geometry for this test is shown in Fig. 25. Both cylinders move at a velocity of 1. A grid of 640×320 was used for this test, which took approximately 1.4 h to run to a time of 32 on a Pentium III 930 MHz. Of this time, approximately 64% was taken solving the boundary element system, which would be significantly reduced under a multipole solver. The boundary conditions used were zero velocity on all four edges, or cavity flow.

Figs. 26 and 27 show vorticity and streamline contours at a time of 16, just as the two cylinders are closest to each other.

Fig. 28 shows the streamline contours at time 32. Note the influence of the boundary conditions on the flow field. Fig. 29 shows the evolution of coefficient of lift and drag for one of the cylinders versus time. We see an expected increase in drag as the cylinders approach each other, but an unexpected (to us initially)

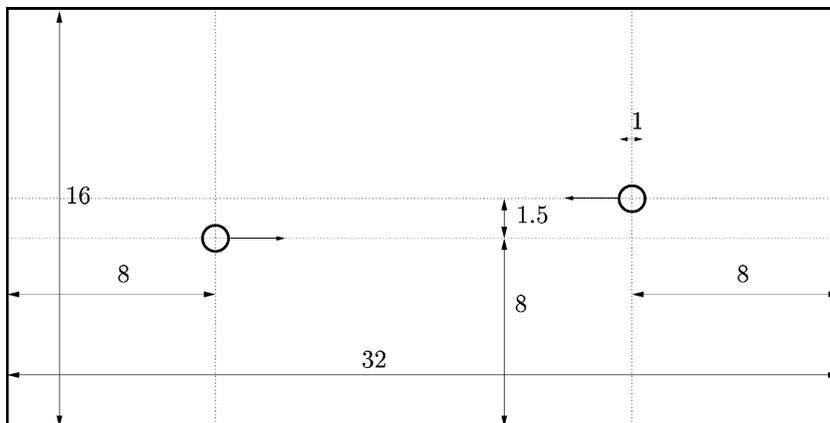


Fig. 25. Geometry for two moving cylinder test.

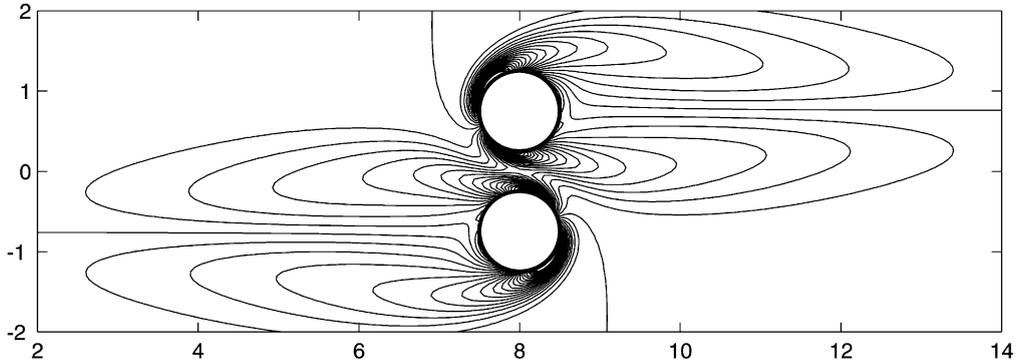


Fig. 26. Vorticity for $\hat{t} = 16$. Contour values are $-4:0.2:4$.

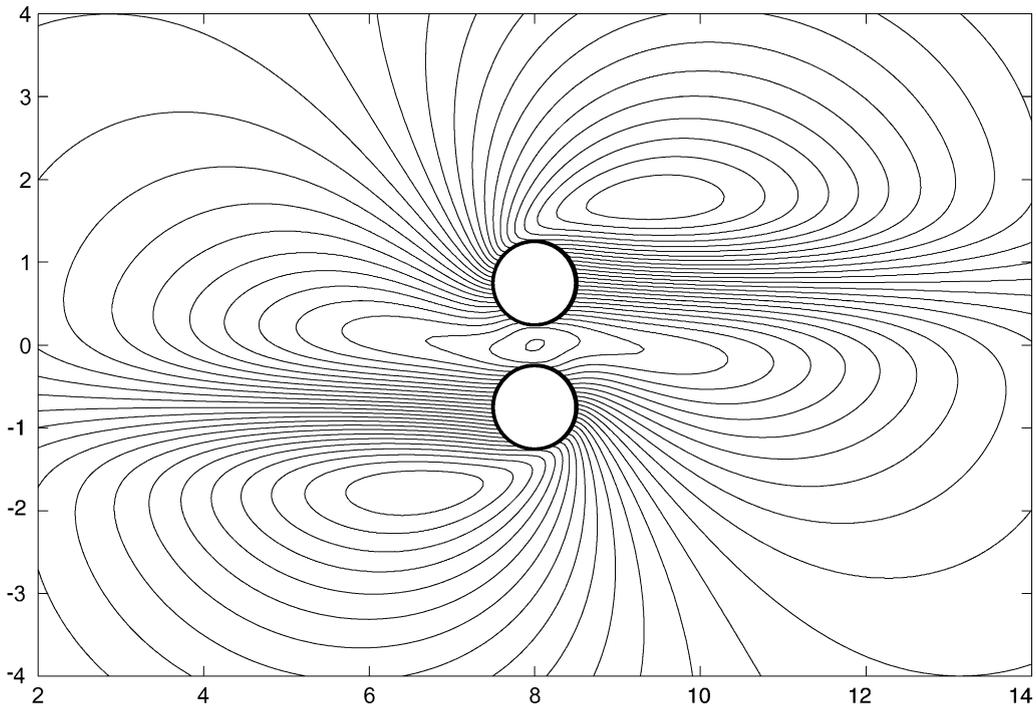


Fig. 27. Streamlines for $\hat{t} = 16$. Contour values are $-1.4:0.2:1.4$.

decrease in drag as they pass in close proximity. The drag then rises back and begins to approximate the profile of a single cylinder, as expected.

Fig. 30 shows the change in the stagnation angle of flow about the cylinder versus time for the lower cylinder. As the two cylinders pass, the stagnation point moves upward to a maximum of about 24.5° .

Fig. 31 shows the pressure distribution around the lower cylinder at two representative times. At a time of 12 the cylinder is still basically behaving as if it were alone. The time of 15.6 was chosen because that is when the stagnation angle is at its highest. Note that the pressure distributions can both be modified by an arbitrary unknown constant; they make sense as a relative distribution only.

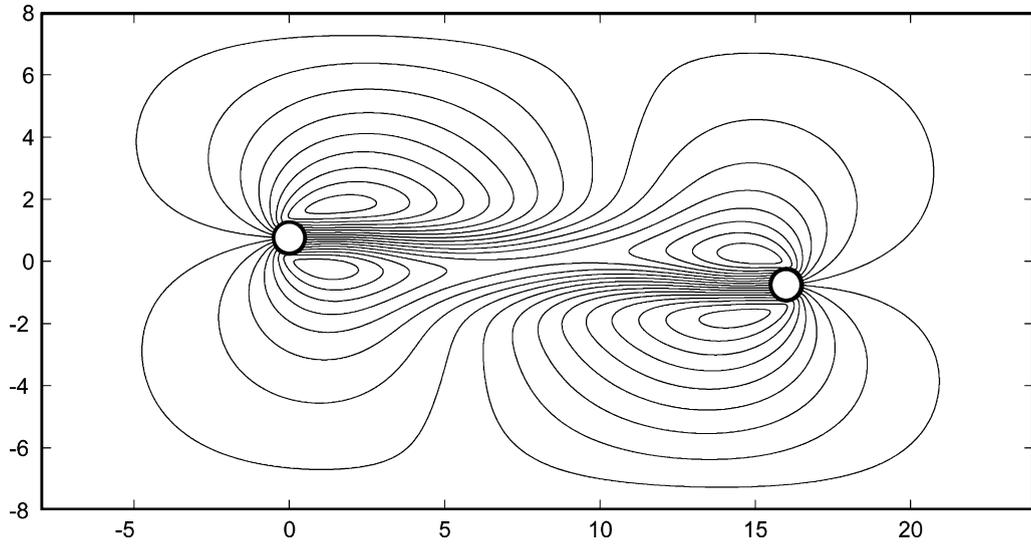


Fig. 28. Streamlines for $\hat{t} = 32$. Contour values are $-1.4:0.2:1.4$.

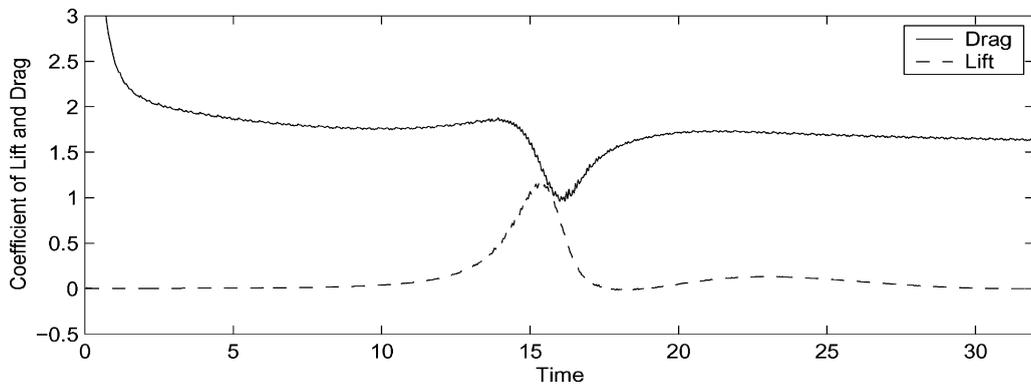


Fig. 29. Lift and drag vs time for upper cylinder. Closest proximity occurs at $\hat{t} = 16$.

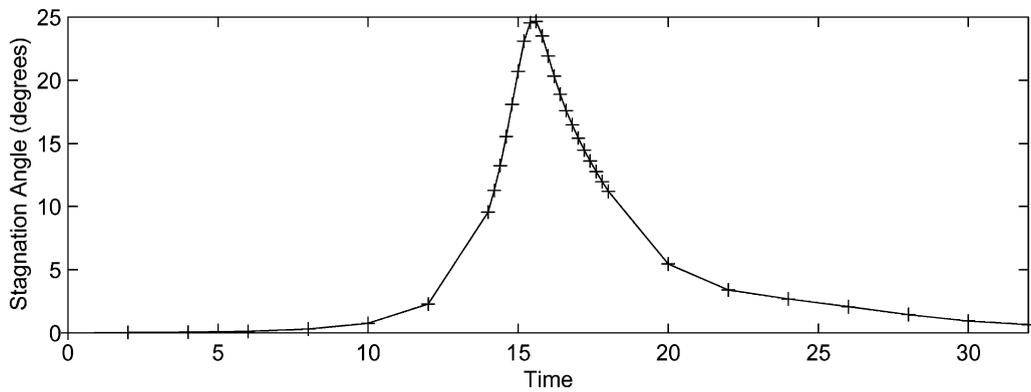


Fig. 30. Stagnation angle vs time for lower cylinder.

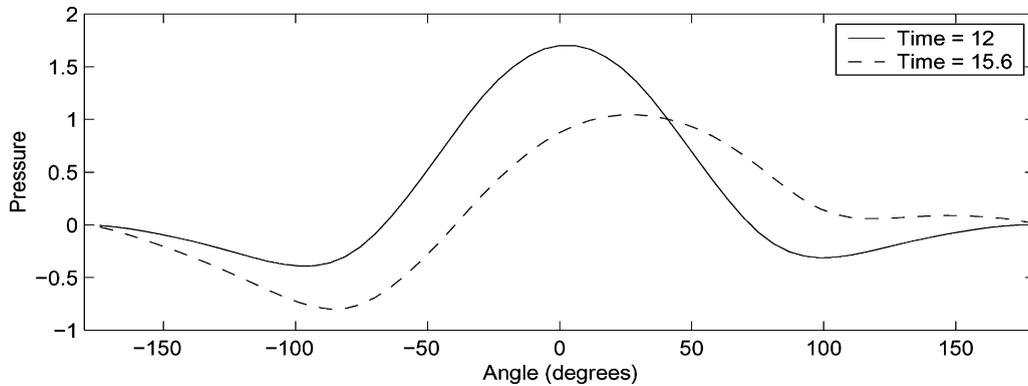


Fig. 31. Pressure distribution around cylinder at $\hat{t} = 12$ and $\hat{t} = 15.6$.

6. Conclusions and future work

We have presented a Cartesian method for the analysis of moving geometry problems using a streamfunction–vorticity formulation. For a given vorticity distribution, the velocity field can be calculated to second-order accuracy including the effects of the irregular moving geometry. This is done using the idea of embedded discontinuities in the streamfunction. Evolving the vorticity field in time is accomplished with the aid of a small overset grid, chosen to allow the calculation of vorticity on nodes near the boundary which would otherwise impose a severe timestep restriction. The resulting vorticity evolution is second-order accurate overall, but only slightly higher than first-order near the boundary.

The method as currently implemented can calculate a given step in $O(N \ln(N) + M^3)$, where N is the number of nodes in the grid and M is the number of elements in the irregular boundary discretization. Using a multipole method to solve the integral equation involved can reduce this to $O(N \ln(N) + M)$. The timestep restrictions associated with an explicit method are somewhat mitigated by the use of fourth-order Runge–Kutta time integration. If we choose to limit the movement of the geometry relative to the underlying grid in a single step (for example to one-tenth the discretization distance), this limit is generally more restrictive than the stable flow solver timestep.

Since the original submission of this paper work has progressed on the creation of an alternate method of handling the discontinuity in the vorticity evolution equation, without introducing the local grid. The new formulation was designed to improve accuracy near the boundary and eliminate errors that made it difficult to simulate objects undergoing rotation. We hope to provide the results of this modification soon.

Acknowledgements

This research is supported by the National Science Foundation under Contract DMS-0075510. The authors are grateful for the advice and comments of D. Caughey, T. Healy, and S. Kulkarni.

References

- [1] M. Braza, P. Chassaing, H. Ha Minh, Numerical study and physical analysis of the pressure and velocity fields in the near wake of a circular cylinder, *Journal of Fluid Mechanics* 165 (1986) 79–130.

- [2] D. Calhoun, A cartesian grid method for solving the two-dimensional streamfunction–vorticity equations in irregular regions, *Journal of Computational Physics* 176 (2) (2002) 231–275.
- [3] J. Carrier, L. Greengard, V. Rokhlin, A fast adaptive multipole algorithm for particle simulations, *SIAM Journal of Scientific and Statistical Computing* 9 (4) (1988) 669–685.
- [4] W.M. Collins, S.C.R. Dennis, Flow past an impulsively started circular cylinder, *Journal of Fluid Mechanics* 60 (1) (1973) 105–127.
- [5] R. Cortez, M. Minion, The Blob projection method for immersed boundary problems, *Journal of Computational Physics* 161 (2000) 428–453.
- [6] M. Coutanceau, R. Bouard, Experimental determination of the main features of the viscous flow in the wake of a circular cylinder in uniform translation. Part 1. Steady flow, *Journal of Fluid Mechanics* 79 (2) (1977) 231–256.
- [7] S.C.R. Dennis, Gau-Zu Chang, Numerical solutions for steady flow past a circular cylinder at Reynolds numbers up to 100, *Journal of Fluid Mechanics* 42 (3) (1970) 471–489.
- [8] E. Weinan, Jian-Guo Liu, Vorticity boundary condition and related issues for finite difference schemes, *Journal of Computational Physics* 124 (1996) 368–382.
- [9] E.A. Fadlun, R. Verzicco, P. Orlandi, J. Mohd-Yusof, Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations, *Journal of Computational Physics* 161 (2000) 35–60.
- [10] B. Fornberg, A numerical study of steady viscous flow past a circular cylinder, *Journal of Fluid Mechanics* 98 (4) (1980) 819–855.
- [11] F. Gibou, R.P. Redkiw, Li-Tien Cheng, M. Kang, A second-order-accurate symmetric discretization of the poisson equation on irregular domains, *Journal of Computational Physics* 176 (2002) 205–227.
- [12] A. Greenbaum, L. Greengard, G.B. McFadden, Laplace’s equation and the Dirichlet–Neumann map in multiply connected domains, *Journal of Computational Physics* 105 (1993) 267–278.
- [13] Huaxiong Huang, B.R. Wetton, Discrete compatibility in finite difference methods for viscous incompressible fluid flow, *Journal of Computational Physics* 126 (1996) 468–478.
- [14] Weizhang Huang, Practical aspects of formulation and solution of moving mesh partial differential equations, *Journal of Computational Physics* 171 (2001) 753–775.
- [15] H. Johansen, P. Colella, A cartesian grid embedded boundary method for poisson’s equation on irregular domains, *Journal of Computational Physics* 147 (1) (1998) 60–85.
- [16] P. Koumoutsakos, A. Leonard, F. Pepin, Boundary conditions for viscous vortex methods, *Journal of Computational Physics* 113 (1994) 52–61.
- [17] Ming-Chih Lai, C.S. Peskin, An immersed boundary method with formal second-order accuracy and reduced numerical viscosity, *Journal of Computational Physics* 160 (2000) 705–719.
- [18] R.J. Leveque, Zhilin Li, The immersed interface method for elliptic equations with discontinuous coefficients and singular sources, *SIAM Journal on Numerical Analysis* 31 (4) (1994) 1019–1044.
- [19] M.J. Lighthill, in: J. Rosenhead (Ed.), *Introduction. Boundary Layer Theory*, Oxford University Press, New York, 1963, p. 54.
- [20] C. Liu, X. Sheng, C.H. Sung, Preconditioned multigrid methods for unsteady incompressible flows, *Journal of Computational Physics* 139 (1998) 35–57.
- [21] A. Mayo, The fast solution of Poisson’s and the biharmonic equations on irregular regions, *SIAM Journal on Numerical Analysis* 21 (2) (1984) 285–299.
- [22] A. Mayo, The rapid evaluation of volume integrals of potential theory on general regions, *Journal of Computational Physics* 100 (1992) 236–245.
- [23] A. McKenney, L. Greengard, A. Mayo, A fast Poisson solver for complex geometries, *Journal of Computational Physics* 118 (1995) 348–355.
- [24] C.S. Peskin, D.M. McQueen, A general method of the computer simulation of biological systems interacting with fluids, *The Society for Experimental Biology* (1995) 265–275.
- [25] C.S. Peskin, D.M. McQueen, Fluid dynamics of the heart and its valves, *Case Studies in Mathematical Modeling – Ecology, Physiology, and Cell Biology* 14 (1996) 309–337.
- [26] M. Rosenfeld, D. Kwak, M. Vinokur, A fractional step solution method for the unsteady incompressible Navier–Stokes equations in generalized coordinate systems, *Journal of Computational Physics* 94 (1991) 102–137.
- [27] D.J. Tritton, Experiments on the flow past a circular cylinder at low Reynolds numbers, *Journal of Fluid Mechanics* 6 (4) (1959) 547–567.
- [28] H.S. Udaykumar, R. Mittal, P. Rampungoon, A. Khanna, A sharp interface cartesian grid method for simulating flows with complex moving boundaries, *Journal of Computational Physics* 174 (2001) 345–380.
- [29] R. Verzicco, J. Mohd-Yusof, P. Orlandi, D. Haworth, Large Eddy simulation in complex geometric configurations using boundary body forces, *AIAA Journal* 38 (3) (2000) 427–433.
- [30] J.A. Wright, R.W. Smith, An edge-based method for the incompressible Navier–Stokes equations on polygonal meshes, *Journal of Computational Physics* 169 (2001) 24–43.

- [31] T. Ye, R. Mittal, H.S. Udaykumar, W. Shyy, An accurate Cartesian grid method for viscous incompressible flows with complex immersed boundaries, *Journal of Computational Physics* 156 (1999) 209–240.
- [32] J. Mohd. Yusof, Interaction of massive particles with turbulence, Ph.D. Thesis (Cornell University), 1996.