

11-10-75

Dr-1700

UCRL-51894

Numerical Solution of the Navier-Stokes Equations at High Reynolds Numbers

Aleksei Ilyich Shestakov
(Ph. D. Thesis)

Prepared for U.S. Energy Research & Development
Administration under contract No. W-7405-Eng-48



MASTER



DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

NOTICE

"This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Energy Research & Development Administration, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately-owned rights."

Printed in the United States of America
Available from
National Technical Information Service
U. S. Department of Commerce
5285 Port Royal Road
Springfield, Virginia 22151
Price: Printed Copy \$ *; Microfiche \$2.25

<u>*Pages</u>	<u>NTIS Selling Price</u>
1-50	\$4.00
51-150	\$5.45
151-325	\$7.60
326-500	\$10.60
501-1000	\$13.60



LAWRENCE LIVERMORE LABORATORY
University of California, Livermore, California, 94550

UCRL-51894

Numerical Solution of the Navier-Stokes Equations at High Reynolds Numbers

Aleksei Ilyich Shestakov

(Ph. D. Thesis)

— NOTICE —

The report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Energy Research and Development Administration, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

Km

Acknowledgement

I should principally like to thank my thesis advisor, Professor Alexandre Chorin, for the unselfish aid he has given me during the writing and research of this work. Without his invaluable inspiration, guidance and support, I would never have been able to complete this dissertation.

Thanks should be given to Dr. Paul Concus and the Lawrence Berkeley Laboratory for the support and provision of computer facilities during the early research phase for this paper, and I would also like to thank Drs. Berni Alder, John Killeen and Julius Chang and the Lawrence Livermore Laboratory for the generous support they gave me during this past year.

I would like to thank Professors Oscar Buneman and Gene Golub of Stanford University for their helpful suggestions and references.

Dr. Ted Einwohner and Mary Ann Mansigh at the Livermore Laboratory were very helpful to me in dealing with the complex laboratory computer system. Without their time-saving suggestions, the completion of this work would have been postponed for several months; hence, many thanks for their help.

My colleague, Michel McCoy, served as a useful companion and listener. Our discussions were very helpful to me and I certainly appreciate his aid.

I would like to thank Raul Mendez for the useful discussions and study sessions we had when we were beginning our research efforts.

Lastly, I would like to express thanks to my wife Lea for her generous patience and encouragement.

Alekseĭ Ilyich Shestakov

Table of Contents

0.	Abstract	P1
1.	Introduction	1
2.	Equations of interest and description of problem	6
3.	Principle of the method of solution	8
4.	Chorin's vortex scheme	11
5.	Finite difference method	16
	1. Difference operators	
	2. Time splitting scheme	
	3. Evaluation of velocity fields	
	4. Boundary conditions for difference scheme	
	5. Stability criteria for matrix equations	
6.	Domain interaction	26
	1. ϵ_1 flow from D_1 into D_2	
	2. ϵ_2 flow from D_2 into D_1	
7.	Numerical details	32
	1. Numerical evaluation of velocity fields	
	2. Approximation via biquadratic spline	
	a. One-dimensional spline	
	b. Two-dimensional spline	
	c. Linear equations for the spline coefficients	
	d. Solution of matrix equation	
	3. Timing tests for the linear equations	
	4. Tridiagonal matrix inversion	
8.	Numerical experiments and results	52
	1. Interpolation tests	
	2. Circulating flow in a cavity	
	3. Comparison of results	
9.	Discussion of Results	63
10.	References	58
11.	Figures	71

NUMERICAL SOLUTION OF NAVIER - STOKES
EQUATIONS AT HIGH REYNOLDS NUMBERS

by

Aleksei Ilyich Shestakov

ABSTRACT

A numerical method is presented which is designed to solve the Navier-Stokes equations for two-dimensional, incompressible flow. The method is intended for use on problems with high Reynolds numbers for which calculations via finite difference methods have been unattainable or unreliable. The proposed scheme is a hybrid utilizing a time-splitting finite difference method in areas away from the boundaries. In areas neighboring the boundaries, the equations of motion are solved by the newly proposed vortex method by Chorin. The major accomplishment of the new scheme is that it contains a simple way for merging the two methods at the interface of the two subdomains. The proposed algorithm is designed for use on the time dependent equations but can be used on steady state problems as well. The method is tested on the popular, time-independent, square cavity problem, an example of a separated flow with closed streamlines. Numerical results are presented for a Reynolds number of 10^3 .

1. Introduction

Presently much of the numerical work on the Navier-Stokes equations is done by approximating the partial derivatives in the equations of motion by finite differences. Since the interesting phenomena occurring in fluids of low viscosity initially appear in regions of small area, it has been very difficult to produce reliable results for problems with high Reynolds numbers.

Analysis implies that at least several grid points must fall within the boundary layer whose thickness is $O(R^{-1/2})$; thus, an upper bound on the Reynolds number seems to be imposed, since one always has a finite amount of space available on the computer. This problem is especially important in cases dealing with flows in wakes or separated flows when the initial boundary layer may not be visible to a coarse finite difference grid. Another drawback of finite differences is that in areas near the boundaries sharp gradients may give rise to large truncation errors which swamp the original approximation. The problem is further complicated by the truncation errors which may cause a numerical viscosity to form which is greater than the viscosity of interest, Dorodnicyn [11]. If care is not exercised, it is possible to generate the right solution to the wrong problem. In fact, as discussed by Fox, Herring, et al. [13], it is often difficult to predict when difference methods go awry. They can easily give incorrect solutions without signalling that further resolution is required. All of these drawbacks have led some researchers to construct other numerical schemes suitable for use

in flows with high Reynolds numbers.

A relatively new scheme was proposed by Chorin [8]. The scheme is grid-free and seems to simulate the physics encountered in fluids. In Chorin's scheme, the vorticity in the fluid and in the boundary layers is subdivided into blobs of small but finite support, and the equations of motion are solved by following the blobs throughout the fluid. The scheme has given some reliable results (Chorin [8], Davari [10], Shestakov [22]), however, it sometimes suffers the drawback of requiring long running times. Its speed, or lack of it, is due to the fact that if n vortex blobs are present in the fluid, then $O(n^2)$ interactions must be computed per time step, since each vortex influences all of the others. Furthermore, the blobs move according to two components, one of which is a random displacement. This displacement leads to a partially random distribution of vorticity which becomes more random as the number of blobs present in the fluid decreases. Hence, one must sometimes resort to unsatisfying methods to get accurate results: use many vortices, or average over ensembles (Shestakov [22]).

Since only at the boundary or whenever blobs are near one another, is it essential to keep track of individual vortices, it is natural to consider a "grouping" scheme which would replace vortex blobs far from an object by "larger" blobs. This approach has been tried (Davari [10]) but is also unsatisfying since large vortices still lead to $O(n^2)$ operations per time step. It is well known that in two-dimensional flow the all-important effect of viscosity is confined to regions near the boundaries of solid bodies. The effect

of viscosity is to create the vorticity making up the boundary layer which later separates into the fluid. Once away from the boundary, the effect of viscosity is negligible. On the other hand, as previously noted, it is this thin viscous layer which is often invisible to finite difference methods and which is responsible for most of the interesting phenomena. Assuming that vorticity was already introduced into the fluid, finite difference methods should work very well in regions away from boundaries. It is thus natural to try to combine these two methods of solution.

The following describes an algorithm to do just that. In areas close to the boundary, the vorticity is evaluated and subdivided into vortex blobs of small support. Away from the body, the Navier-Stokes equations are solved by a time-splitting finite difference method. The proposed hybrid method is tested on the popular problem of computing the steady-state incompressible flow inside a square cavity when one of the sides of the cavity slides in its own plane with constant unit velocity.

The cavity flow problem has been very popular over the years and enjoyed an extensive bibliography, [1], [3], [5], [14], [17], [19], and [21]. It is part of a larger class of steady, separated flows with closed streamlines studied by Batchelor [2]. The velocity field depends on the Reynolds number and has been evaluated numerically and experimentally by several researchers. As the Reynolds number, R , tends to zero the so-called creeping flow case arises which yields a symmetric velocity field and one large vortex filling out most of the cavity with its center near the sliding edge.

As R increases, two things have been observed. First, the center of the large vortex moves downstream parallel to the sliding edge, and secondly, the small counter-rotating vortex in the upstream stationary corner begins to grow in size. The growth of this vortex continues until $R \approx 500$, at which point the corner vortex begins to shrink back into its corner.

The position of the center of the large interior vortex exhibits a similar peculiar behavior. As mentioned above, as R grows away from zero, the center of the vortex moves downstream. Above 100, however, the center of the vortex turns the corner and begins to move toward the center of the cavity. As proposed by Batchelor, and as verified by others, the large center vortex becomes inviscid as R increases and the vorticity in the interior approaches a uniform limiting value. The viscous effects are then confined to increasingly thin shear layers along the three stationary walls.

Several researchers have obtained results for high Reynolds numbers. Burgraff, [5], did an extensive study and was able to get numerical solutions for R as high as 500. Pan and Acrivos [1], computed the creeping flow case and performed laboratory experiments for higher R . They were able to make visual observations for R as large as 4000 at which point instabilities began to affect the flow. Greenspan [14] computed numerical solutions for $R = 2000$, 1.5×10^4 using a mesh size of $1/20$, and for $R = 10^4$, 10^5 using a mesh size of $1/40$. However, his iterative technique does only a local check on the iterate and the computation is stopped when the values do not change appreciably from one iteration to the next. Greenspan also

did not observe any counter-rotating vortices nor did he exhibit any velocity profiles through the vortex center. Using a non-uniform grid, and a new method for evaluating the wall vorticity, Runchal, et al., [21], obtained results for $R = 10^4$. They were also not able to visualize the corner vortices nor did they display any velocity profiles. Testing four different differencing techniques, Bozeman and Dalton [3] were able to get two to converge at $R = 10^3$. They then discarded one of them as giving an obviously incorrect result. All of the above numerical calculations were performed by iterating on the steady-state Navier-Stokes equations and as noted by Bozeman and Dalton, this technique may produce solutions which don't change appreciably from one time step to another rather than the true steady-state solution.

All of the above numerical work was done by approximating the derivatives by finite differences. Since, on occasion, different difference schemes yield different results, all of the above computations should be carefully analyzed before pronounced correct. I hope that the present work will serve a twofold purpose: Serve as another contribution to the literature of a square cavity problem, and more importantly, introduce a numerical technique that may be useful in problems with high Reynolds numbers.

2. Equations of interest and description of problem

To avoid confusion, scalars and scalar functions will be denoted by simple letters, e.g. h , ϕ , x , and the underline will be reserved to distinguish vectors, and vector functions, e.g. \underline{r} , \underline{U} , $\nabla\phi$.

The equations of interest are the non-dimensionalized Navier-Stokes equations for two-dimensional, incompressible flow, written in the vorticity transport form

$$\partial_t \xi + (\underline{U} \cdot \nabla) \xi = \frac{1}{R} \Delta \xi, \quad (2.1)$$

$$(\nabla \cdot \underline{U}) = 0. \quad (2.2)$$

Definition $\underline{U} = (U, V) \equiv$ velocity vector ,

$R \equiv$ Reynolds number ,

$\xi = \nabla \times \underline{U} \equiv$ vorticity .

Since \underline{U} is two dimensional, ξ is considered to be a scalar.

Equations (2.1) and (2.2) are to be solved in a domain D with the imposed "no-slip" boundary condition:

$$\underline{U} \text{ (at the boundary) } = \text{velocity of the boundary,} \quad (2.3a)$$

or if the boundary is at rest,

$$\underline{U} = 0 \text{ at the boundary.} \quad (2.3b)$$

There will also be a specified initial condition,

$$\underline{U} (t = 0) = \underline{f}_0, \text{ a given function.} \quad (2.4)$$

Since the flow is incompressible and two-dimensional, there exists a scalar function ψ which can be used to define the velocity:

$$\underline{U} = (U, V) = (\partial_y \psi, -\partial_x \psi). \quad (2.5)$$

Definition $\psi =$ stream function.

Using (2.5) and the definition of ξ as the curl of \underline{U} , it follows

that:

$$\Delta\psi = -\xi . \quad (2.6)$$

Equation (2.5) determines ψ up to an additive constant. If the boundary condition (2.3b) is to hold, then ψ satisfies,

$$\psi = \text{constant}$$

at the boundary.

Consider, in particular, the problem of computing the steady-state velocity field inside a unit-square cavity (Figure 1).

Initially the fluid is assumed to be at rest and at $t = 0$, the bottom edge has impulsively begun to slide in its own plane with unit velocity.

3. Principle of the method of solution

Equations (2.1) and (2.2) will be solved by integrating them forward in time. Given ξ^m , the vorticity at the m^{th} time step, the problem at hand is to calculate ξ^{m+1} .

Let $D = D_1 \cup D_2$, such that

$$D_1 \cap D_2 = \emptyset, \text{ and}$$

$$\text{dist}(D_2, \partial D) = \delta > 0.$$

Let

$$\xi_1^m = \xi_1^m + \xi_2^m, \quad (3.1a)$$

such that,

$$\text{support}(\xi_i^m) \subseteq D_i. \quad (3.1b)$$

Since ξ_1^m is to be a collection of vortex blobs,

$$\xi_1^m(\underline{r}) = \sum_j \xi_j \xi_0(\underline{r} - \underline{r}_j^m). \quad (3.2)$$

ξ_2^m will denote the vorticity given on an interior grid; thus, define

$$\{z_{1j}^m\} = \xi_2^m.$$

Let \underline{u}_1^m , \underline{u}_2^m , and ψ_1^m , ψ_2^m correspond to the velocities and stream functions induced by the two vorticities ξ_1^m , and ξ_2^m respectively. Then, it follows,

$$\underline{u}_1^m(\underline{r}) = \sum_j z_j \underline{u}_0(\underline{r} - \underline{r}_j^m), \quad (3.3)$$

and

$$\psi_1^m(\underline{r}) = \sum_j \xi_j \psi_0(\underline{r} - \underline{r}_j^m). \quad (3.4)$$

$\xi_0(\underline{r})$ is the vorticity function of one vortex located at the origin, $\underline{u}_0(\underline{r})$ is the velocity field induced by that vortex, and $\psi_0(\underline{r})$ is the induced stream function (see §4).

Calculate \underline{U}_2^m by solving

$$\Delta \psi_2^m = -\xi_2^m \quad \text{in } D \quad (3.5)$$

with the boundary values,

$$\psi_2^m(s) = -\psi_1^m(s) \quad \text{along } \partial D. \quad (3.6)$$

Knowing ψ_2^m , let

$$\underline{U}_2^m \equiv (\partial_y \psi_2^m, -\partial_x \psi_2^m). \quad (3.7)$$

If $\psi^m \equiv \psi_1^m + \psi_2^m$, then (3.6) implies

$$\psi^m = 0 \quad \text{on } \partial D, \quad (3.8)$$

and if $\underline{U}^m \equiv \underline{U}_1^m + \underline{U}_2^m$, then (3.8) implies

$$\underline{U}^m \cdot \underline{n} = 0 \quad \text{on } \partial D,$$

where \underline{n} is the normal vector along ∂D . Thus \underline{U}^m is a velocity field which has no component normal to the boundary, or, in other words, there is no "leakage" or "seepage" out of or into D .

The updating of the field ξ^m is done by two distinct methods.

In the domain D_1 , the one adjacent to the boundary, the vorticity is represented by ξ_1^m , a collection of vortices (see (3.2)). The field ξ_1^m is advanced in time by Chorin's vortex scheme [8] (see §4). In the interior subdomain, D_2 , the vorticity is given by ξ_2^m , a function known on a grid over the domain. The equations of motion, (2.1), and (2.2) are solved by approximating the partial derivatives by finite differences and by using a time-splitting scheme to advance ξ_2^m .

After ξ_1^m, ξ_2^m are advanced, it is clear that (3.1b) may be violated; that is, some of the vortices which make up ξ_1^m may now lie outside D or inside D_2 , and similarly some vorticity stored in D_2 on the grid may have convected or diffused into D_1 or outside D . The vortices

which now lie outside D are eliminated, while the ones lying in D_2 are interpolated onto the grid. By choosing the time step carefully, it can be shown that vorticity originally in D_2 cannot travel more than a specified distance, which is taken to be less than δ , the distance from D_2 to ∂D . The vorticity in D_2 which has been convected or which has diffused into D_1 is "remembered" on an "expanded" mesh. This vorticity is then coagulated into new vortices. The above process thus gives a new $\xi_1^{m+1} \subseteq D_1$ and a new $\xi_2^{m+1} \subseteq D_2$. This completes the evaluation of ξ^{m+1} .

4. Chorin's vortex scheme

In this section, a sketch of Chorin's vortex scheme [8] is presented as it pertains to this problem.

Consider the problem of computing the evolution of vorticity ξ_1 according to equations (2.1), (2.2), (2.5), and (2.6) subject to the boundary conditions (2.3a) or (2.3b). Chorin assumes that this vorticity in D_1 can be written as a sum of blobs, or vortices,

$$\xi_1(\underline{r}) = \sum_j \xi_{1,j}(\underline{r} - \underline{r}_j)$$

where each $\xi_{1,j}$ is a radially symmetric function of small support,

and $|\underline{r}| = \sqrt{x^2 + y^2}$. Assuming that the $\xi_{1,j}$ differ only in strength and not in shape, by assigning a strength ζ_j to the j^{th} blob, rewrite ξ_1 as:

$$\xi_1(\underline{r}) = \sum_j \zeta_j \zeta_0(\underline{r} - \underline{r}_j), \quad (4.1)$$

where

$$\zeta_0(\underline{r}) = \begin{cases} \frac{1}{2\pi\sigma r} & , r < \sigma \\ 0 & , r \geq \sigma \end{cases} \quad (4.2)$$

The constant σ is the cut-off length, to be determined shortly and $r = |\underline{r}|$. Although $\zeta_0(\underline{r})$ is singular at the origin, its shape becomes convincing when one considers the integral of the vorticity,

$$\iint_D \xi_1(x,y) dx dy = \iint_D \sum_j \zeta_j \zeta_0(\underline{r} - \underline{r}_j) d\underline{r} = \sum_j \zeta_j$$

where the integrals have been evaluated using polar coordinates.

Equation (4.2) will look even more convincing after the generation

of vorticity at the boundary has been discussed.

The basic blob $\zeta_0(\underline{r})$ generates its own blob stream function, ψ_0 , which is computed by solving

$$\Delta\psi_0 = -\zeta_0.$$

Neglecting boundary values for the moment, it follows that

$$\psi_0(\underline{r}) = \begin{cases} -\frac{1}{2\pi} \log r & , r \geq \sigma \\ -\frac{1}{2\pi\sigma} r & , r < \sigma . \end{cases} \quad (4.3)$$

Using (2.5), (4.1), and (4.3), it is possible to generate the velocity field induced by ζ_1 :

$$\underline{u}_1(x,y) = \sum_j \zeta_j \underline{u}_0(x - x_j, y - y_j) . \quad (4.4)$$

$\underline{u}_0(x,y)$ is then the velocity induced by a single blob located at the origin, and (x_j, y_j) is the location of the j^{th} vortex. \underline{u}_0 can be calculated by differentiating (4.3);

$$\underline{u}_0(x,y) = (\partial_y \psi_0, -\partial_x \psi_0) = \begin{cases} \frac{1}{2\pi r^2} (-y, x) & , r \geq \sigma \\ \frac{1}{2\pi\sigma} (-y, x) & , r < \sigma . \end{cases}$$

The velocity \underline{u}_0 , and therefore \underline{u}_1 , is a continuous velocity field obtained from a discontinuous, in fact singular, distribution of vorticity.

It's possible (Davari [10]), to generate a continuous stream fn., ψ_0 , by the addition of a constant, which yields

$$\psi_0(\underline{r}) = \begin{cases} -\frac{1}{2\pi} \log r & , r \geq \sigma \\ -\frac{1}{2\pi} (1 - \log \sigma - \frac{r}{\sigma}) & , r < \sigma . \end{cases}$$

and it is this expression which is used by the computer program.

The computation of the evolution of ξ_1 according to (2.1) is accomplished by a first order correct in time, differencing algorithm which uses no spatial grid but merely keeps track of the centers of the vortices. The field \underline{u}_1 does not satisfy any boundary conditions; sections 5 and 7 explain how to generate a velocity field \underline{u}_2 with $\nabla \times \underline{u}_2 = 0$ in D_1 and which cancels the normal boundary component of \underline{u}_1 . Then defining $\underline{u} \equiv \underline{u}_1 + \underline{u}_2$, it follows that,

$$\underline{u} \cdot \underline{n} = \underline{u}_1 \cdot \underline{n} + \underline{u}_2 \cdot \underline{n} = 0 \text{ along } \partial D.$$

Thus, \underline{u} is a velocity field which satisfies the normal boundary condition.

Momentarily neglecting the second boundary condition, $\underline{u} \cdot \underline{s} = 0$, solve the equation,

$$\partial_t \xi_1 + (\underline{u} \cdot \nabla) \xi_1 = \frac{1}{R} \Delta \xi_1 \quad (4.5)$$

in D_1 in two steps. Equation (4.5) states that the vortices move according to two components. The first, the convective one, is Euler's equation,

$$\partial_t \xi_1 + (\underline{u} \cdot \nabla) \xi_1 = 0 \quad , \quad (4.6)$$

or if we follow the fluid particles

$$\frac{D\xi_1}{Dt} = 0 \quad .$$

The second component is the diffusive one,

$$\partial_t \xi_1 = \frac{1}{R} \Delta \xi_1 \quad . \quad (4.7)$$

Equation (4.6) is solved by keeping track of the locations of the vortices. If (x_i, y_i) denote the location of the i^{th} vortex blob then,

$$\frac{d}{dt} (x_i, y_i) = \sum_{j \neq i} \zeta_j \underline{u}_0 (x_i - x_j, y_i - y_j) + \underline{u}_2 (x_i, y_i),$$

or if (x_i^m, y_i^m) is the location of the i^{th} vortex at the m^{th} time step, then

$$(x_i^{m+1}, y_i^{m+1}) = (x_i^m, y_i^m) + k \left[\sum_{j \neq i} \zeta_j \underline{u}_0 (x_i^m - x_j^m, y_i^m - y_j^m) + \underline{u}_2 (x_i^m, y_i^m) \right] \quad (4.8)$$

where k is the time step. This is simply Euler's method for the solution of o.d.e's.

Equation (4.7) is solved by the random walk method developed by Chorin [8], whose idea was described by Courant, et al. [9] in their classic paper on difference schemes. If $\underline{\eta} = (\eta_1, \eta_2)$ is a vector whose components are gaussianly distributed random variables, with mean zero and variance $2k/R$, then (4.7) is approximated by

$$(x_i^{m+1}, y_i^{m+1}) = (x_i^m, y_i^m) + (\eta_1, \eta_2) \quad (4.9)$$

(see [8] for an easy algorithm for constructing η_1, η_2).

Denoting the term in the [.] brackets in (4.8) by \underline{u}^m , then (4.8) and (4.9) are combined to yield,

$$(x_i^{m+1}, y_i^{m+1}) = (x_i^m, y_i^m) + k \underline{u}^m + (\eta_1, \eta_2)$$

as the approximation to (4.5).

Consider now the generation of vorticity at the boundary. As previously noted, the field \underline{u} does not satisfy the tangential boundary condition,

$\underline{u} \cdot \underline{s} = 0$ or $\underline{u} \cdot \underline{s} = \text{tangential velocity of the boundary}$, where \underline{s} is the unit tangent vector to ∂D . To correct this deficiency in \underline{u} , imagine the existence of a thin viscous boundary layer which cancels the tangential component of \underline{u} . To evaluate the vorticity necessary to

do this, integrate $\nabla \times \underline{u}$ along the boundary layer. If the boundary is divided into segments of length h , and the midpoint rule is used as an approximation to the integral, the vorticity in a boundary layer of thickness v and length h is

$$\int_{-h/2}^{h/2} \int_0^v (\nabla \times \underline{u}) \, dy dx = \int_{-h/2}^{h/2} \int_0^v (-\partial_y u) \, dy dx \approx -U(0,v) h.$$

$U(0,v)$ is the free stream velocity at the edge of the boundary layer and it is set equal to $U(0,0)$, the tangential velocity component of \underline{u} which needs to be cancelled (see Fig. 2). Note that the above calculation was done using a local coordinate system. This vorticity is then coagulated into a new vortex blob. The newly created blob is allowed to diffuse, and it is its shape, inside the cut-off length σ which exerts a constant velocity at the boundary equal to $-\underline{u} \cdot \underline{s}$. This accomplishes the cancellation of $\underline{u} \cdot \underline{s}$ and gives a value for σ , namely $\sigma = h/2\pi$. The new blobs diffuse into D and become part of ξ_1 . The entire process is then reiterated.

5. Finite difference method.

1. Difference operators

In the interior subdomain, D_2 , the vorticity field ξ_2 is advanced in time by approximating the equation of motion:

$$\begin{aligned} \partial_t \xi_2 + (\underline{U} \cdot \nabla) \xi_2 &= \frac{1}{R} \Delta \xi_2, \\ (\nabla \cdot \underline{U}) &= 0 \end{aligned} \quad (5.1)$$

by a finite difference scheme.

Subdivide D_2 , which in the problem of interest is a square domain, with a square mesh of fixed mesh size d . Assume that ∂D_2 always lies the same distance δ from ∂D . Define

$$\zeta_{i,j}^m \equiv \xi_2(\delta + id, \delta + jd, mk),$$

i.e. $\zeta_{i,j}^m$ represents the vorticity at the point $x = \delta + id$, $y = \delta + jd$, $t = mk$. Assume that D_2 is covered by a mesh with $M + 1$ grid points in each coordinate spatial direction, hence

$$\xi_2^m = \{\zeta_{i,j}^m\}_{i,j=0}^M.$$

Define the shift operators

$$S_{\pm 1} \zeta_{i,j} \equiv \zeta_{i \pm 1, j}, \quad S_{\pm 2} \zeta_{i,j} \equiv \zeta_{i, j \pm 1}, \quad I \zeta_{i,j} \equiv \zeta_{i,j};$$

define the one sided difference operators,

$$D_{\pm 1} \equiv \pm (S_{\pm 1} - I)/d, \quad D_{\pm 2} \equiv \pm (S_{\pm 2} - I)/d,$$

and define the central difference operator

$$D_{01} \equiv \frac{1}{2}(D_{+1} + D_{-1}), \quad D_{02} \equiv \frac{1}{2}(D_{+2} - D_{-2}).$$

2. Time-splitting scheme

Using these definitions, (5.1) is solved by a time-splitting method, whereby an intermediate field $\zeta^{m+1/2}$ is calculated. Then using U^m ,

and $\zeta^{m+1/2}$, the vorticity at the next time step, ζ^{m+1} , is evaluated.

Symbolically,

$$\zeta^{m+1/2} - \zeta^m = \frac{k}{2} \left[-U^m D_{01} \zeta^{m+1/2} - v^m D_{02} \zeta^m + \frac{1}{R_{+1} D_{-1}} \zeta^{m+1/2} + \frac{1}{R_{+2} D_{-2}} \zeta^m \right] \quad (5.2a)$$

$$\zeta^{m+1} - \zeta^{m+1/2} = \frac{k}{2} \left[-U^m D_{01} \zeta^{m+1/2} - v^m D_{02} \zeta^{m+1} + \frac{1}{R_{+1} D_{-1}} \zeta^{m+1/2} + \frac{1}{R_{+2} D_{-2}} \zeta^{m+1} \right]. \quad (5.2b)$$

If the unknowns are combined, the equations can be rewritten as

$$\left(I + \frac{k}{2} U^m D_{01} - \frac{k}{2R_{+1} D_{-1}} \right) \zeta^{m+1/2} = \left(I - \frac{k}{2} v^m D_{02} + \frac{k}{2R_{+2} D_{-2}} \right) \zeta^m, \quad (5.3a)$$

$$\left(I + \frac{k}{2} v^m D_{02} - \frac{k}{2R_{+2} D_{-2}} \right) \zeta^{m+1} = \left(I - \frac{k}{2} U^m D_{01} + \frac{k}{2R_{+1} D_{-1}} \right) \zeta^{m+1/2}. \quad (5.3b)$$

It can be proved that the analogue of (5.3) is unconditionally stable when equation (5.1) is linear and has constant coefficients.

Using a Von Neuman Stability analysis, let

$$\zeta_{i,j}^m = \rho^m e^{\sqrt{-1} (i\theta_1 + j\theta_2)}. \quad (5.4)$$

To prove stability it is necessary to show that $\rho \leq 1$.

Define $c_1 \equiv \frac{Uk}{2d}$, $c_2 \equiv \frac{Vk}{2d}$, $b \equiv \frac{Vk}{Rd^2}$,

and substitute (5.4) into (5.3). After cancelling powers of e , and substituting (5.3a) into (5.3b) obtain

$$\rho = \prod_{j=1}^2 \left| \frac{1 - b(1 - \cos\theta_j) - \sqrt{-1c_j} \sin\theta_j}{1 + b(1 - \cos\theta_j) + \sqrt{-1c_j} \sin\theta_j} \right| .$$

ρ is necessarily ≤ 1 , if each one of the factors in the product is ≤ 1 . However, since $b > 0$, and $1 - \cos\theta_j > 0$, $j = 1, 2$, it is evident that the numerator of each factor is less in modulus than the denominator. Therefore, $\rho < 1$ and the scheme is unconditionally stable.

3. Evaluation of velocity fields

At the beginning of each time step, the locations and strengths of each vortex blob are stored in an array, thereby specifying the field ξ_1^m . Correspondingly, the vorticity ξ_2^m is given on a grid covering D_2 ,

$$\xi_2^m = \left\{ \xi_{1,j}^m \right\}_{1,j=0}^M .$$

In order to advance ξ_2 in D_2 , the velocity field \underline{u} must be evaluated on the same grid. The velocity \underline{u} is a sum of the velocities induced by the two functions ξ_1 and ξ_2 . However, (4.4) is not used to calculate \underline{u}_1 on D_2 , as this would entail summing over all the blobs for each grid point. Instead, ψ_1 and \underline{u}_1 are computed on the mesh points constituting ∂D_2 . Using these calculated values for ψ_1 on ∂D_2 as boundary data, the finite difference analogue of

$$\Delta\psi_1 = 0 \text{ on } D_2 \quad (5.5)$$

is solved. The number of points per side for this problem is always odd (see below) and may often be of the form $4L - 1$, L a positive integer. The solution of (5.5) is then obtained using the FACR algorithm of Hockney [15]. After (5.5) is solved, the field \underline{u}_1 is approximated by

central differences of the computed stream function values for ψ_1 .

The field \underline{u}_2 is evaluated in D and ∂D by solving (3.5) with the boundary conditions (3.6), and using central differences of the result.

Because the mesh will "expand" by two mesh widths over one time step (see 5.4), the field \underline{u} will need to be tabulated at the grid points falling outside D_2 . At these points \underline{u}_1 is computed via (4.4), and u_2 is evaluated by differentiating the bi-quadratic spline interpolator to ψ_2 in D_1 (see 57). After the velocities have been evaluated, the field \underline{u} is specified at the $(M+5)^2$ points making up D_2 , ∂D_2 and two mesh widths surrounding D_2 , or symbolically,

$$\underline{u}^m = \left\{ \begin{matrix} \underline{u}_{ij}^m \\ i, j = -2 \end{matrix} \right\}^{M+2}$$

These values are then stored in an array in memory.

4. Boundary conditions for difference scheme

The finite difference scheme is used in the domain D_2 which is a distance δ away from the real boundary ∂D . Hence, the grid on which ξ_2 is updated does not reach any solid boundaries of the fluid. After examining (5.3a), it is clear that if (5.1) is discretized at the point $(\delta + id, \delta + jd)$, then (5.3a) gives a linear equation for the 3 unknowns

$$\zeta_{i-1,j}^{m+1/2}, \zeta_{i,j}^{m+1/2}, \zeta_{i+1,j}^{m+1/2}.$$

Hence, to compute $\zeta_{0,j}^{m+1/2}$, $\zeta_{-1,j}^{m+1/2}$ is needed; $\zeta_{-1,j}^{m+1/2}$ is the vorticity which may be carried from D_2 to the point $(x,y) = (\delta - d, \delta + jd)$.

Discretizing (5.3a) at the $M+1$ points $(x_i, y_j) = (\delta + id, \delta + jd)$, where $i = 0, 1, \dots, M$, gives $M+1$ linear equations for the $M+3$ unknowns $\zeta_{i,j_0}^{m+1/2}$, $i = -1, 0, 1, \dots, M+1$. Hence, in order to solve the system,

two more equations must be added for the unknowns $\zeta_{-1,j_0}^{m+1/2}$ and $\zeta_{M+1,j_0}^{m+1/2}$.

Discretizing (5.3a) at the point $(x,y) = (\delta - d, \delta + j_0 d)$ will yield a linear equation for $\zeta_{-2,j_0}^{m+1/2}$, $\zeta_{-1,j_0}^{m+1/2}$, $\zeta_{0,j_0}^{m+1/2}$. Setting $\zeta_{-2,j_0}^{m+1/2} = 0$ means that no vorticity from D_2 can travel beyond one mesh width in a time step of $k/2$.

Vorticity from D_2 can reach the point in question in only two ways, by convection or by diffusion. It can be assured that ξ_2 is not convected beyond one mesh width as long as

$$|U|_{\max} \frac{k}{2} \leq d. \quad (5.6)$$

Diffusion, as discussed by Chorin [8] and Courant [9] is equivalent to a random walk. Tschebysheff's theorem (Lamperti [18]) states that gaussianly distributed random variables with mean zero will rarely be greater in magnitude than three standard deviations. During a time step of duration k , vorticity will diffuse with a standard deviation of $\sqrt{2k/R}$. Hence, in a time step of length $k/2$, ξ_2 will not likely travel beyond one mesh width if

$$3 \sqrt{k/R} \leq d. \quad (5.7)$$

If (5.6) and (5.7) are satisfied, $\zeta_{-2,j_0}^{m+1/2}$ and $\zeta_{M+2,j_0}^{m+1/2}$ can both be set to zero and thereby get $M + 3$ equations for $M + 3$ unknowns.

After the completion of the implicit differencing in the x-direction, support (ξ_2) has expanded beyond D_2 by one mesh width.

Hence, it follows that

$$\xi_2^{m+1/2} = \left\{ \zeta_{1,j}^{m+1/2} \right\}_{i,j=-1}^{M+1}. \quad (5.8)$$

The solution of (5.3b) is obtained in a similar fashion. The equations are implicitly discretized in the y-direction and along each column solve a system of $M + 5$ equations for the $M + 5$ unknowns

$$\zeta_{i,j}^{m+1}, \quad j = -2, -1, 0, \dots, M + 2 \quad .$$

The system is closed by virtue of setting $\zeta_{i,-3}^{m+1}$ and $\zeta_{i,M+3}^{m+1}$ equal to zero by the same reasoning as above. Hence, at the completion of these 2 half-steps, support (ξ_2) has spread over 2 mesh widths beyond ∂D_2 and the vorticity ξ_2 is tabulated as:

$$\xi_2^{m+1} = \left\{ \zeta_{i,j}^{m+1} \right\}_{i,j=-2}^{M+2} \quad . \quad (5.9)$$

5. Stability criteria for matrix equations

The heart of the time-splitting scheme involves solving many tridiagonal systems. Although the Von-Neuman stability analysis showed that the difference scheme is stable, the algorithm used to invert the tridiagonal systems gives some stability conditions guaranteeing that the inversion is numerically stable.

Using the definitions of §5.1, (5.3a) can be rewritten as

$$\left[I + \frac{k}{4d} U_{i,j}^m (S_{+1} - S_{-1}) - \frac{k}{2Rd^2} (S_{+1} - 2I + S_{-1}) \right] \zeta_{i,j}^{m+1/2} = \left[I - \frac{k}{4d} V_{i,j}^m (S_{+2} - S_{-2}) + \frac{k}{2Rd^2} (S_{+2} - 2I + S_{-2}) \right] \zeta_{i,j}^m \quad . \quad (5.10)$$

Definition

$$\alpha \equiv \frac{k}{2(Rd^2+k)} \quad , \quad \beta \equiv \frac{Rkd}{4(Rd^2+k)} \quad , \quad \gamma \equiv \frac{Rd^2-k}{Rd^2+k} \quad . \quad (5.11)$$

If (5.11) is now substituted in (5.10), obtain:

Definition $(\|U\|_{\max}^m)^2 = \max_{i,j} [(U_{i,j}^m)^2 + (V_{i,j}^m)^2]$.

Recalling (5.11), condition (5.16a) is guaranteed if

$$Rdk \|U\|_{\max}^m < 2k + 4Rd^2. \quad (5.17)$$

Condition (5.16b) requires:

$$2 \max_i (\alpha, |C_i|) < 1.$$

If

$$Rd \|U\|_{\max}^m < 2, \quad (5.18)$$

then for $i = 0, \dots, M$,

$$\alpha > |C_i|,$$

and in this case (5.16b) requires:

$$1 > 2\alpha = \frac{k}{Rd^2+k} \text{ or } Rd^2 + k > k,$$

which is always true. Therefore, if (5.18) holds then (5.16b) is fulfilled. It is also evident that if (5.18) is true, that (5.17) holds; hence, (5.16a) holds as well. It has thus been proven:

Lemma 1. If $Rd \|U\|_{\max}^m < 2$, then the tridiagonal matrix solver is stable.

If, however, (5.18) does not hold, i.e. if for some (i_0, j_0) ,

$Rd U_{i_0, j_0}^m \geq 2$, then along some row j_0 , $\alpha \leq C_{i_0}$. The matrix A of

equation (5.15) may still be invertible as long as (5.16b) holds. In this case (5.16b) requires that

$$1 > 2|C_{i_0}| = \frac{Rdk}{2(Rd^2+k)} \|U\|_{\max}^m,$$

or rather,

$$2(Rd^2 + k) > Rdk \left\| \left| U_i \right|_{\max}^m \right. \quad (5.19)$$

If (5.19) holds, then (5.17) is also true, hence condition (5.16a) would be satisfied. Moreover, (5.19) will certainly be satisfied whenever:

$$0 < 2k + Rd(2d - \left\| \left| U \right|_{\max}^m k \right. \right) \quad (5.20)$$

Inequality (5.20) will be always satisfied if

$$\left\| \left| U \right|_{\max}^m k < 2d \right. , \quad (5.21)$$

which means that in a time step k , vorticity cannot be convected more than 2 mesh widths. The above has proved:

Lemma 2. If $\left\| \left| U \right|_{\max}^m k < 2d \right.$, then the tridiagonal matrix solver is stable.

Once $\xi_2^{m+1/2}$ has been computed, a similar computation will yield ξ_2^{m+1} . In this case, since $\xi_2^{m+1/2}$ has spread 1 mesh width beyond ∂D_2 , ξ_2^m will spread 2 mesh widths beyond ∂D_2 . Equation (5.3b) takes the form.

$$\left[I + \frac{k}{4d} V_{1,j}^m (S_{+2} - S_{-2}) - \frac{k}{2Rd^2} (S_{+2} - 2I + S_{-2}) \right] \zeta^{m+1} = \left[I - \frac{k}{4d} U_{1,j}^m (S_{+1} - S_{-1}) + \frac{k}{2Rd^2} (S_{+1} - 2I + S_{-1}) \right] \zeta^{m+1}, \quad (5.22)$$

The system (5.22) is now solved for fixed x , e.g. for fixed column

1. Using the definitions (5.11) and defining

$$C_j \equiv \beta U_{1,j}^m, \quad D_j \equiv \beta V_{1,j}^m,$$

(5.22) takes the form:

$$\begin{aligned}
& - (\alpha + D_j) \zeta_{i,j-1}^{m+1} + \zeta_{i,j}^{m+1} - (\alpha - D_j) \zeta_{i,j+1}^{m+1} = \\
& (\alpha + C_j) \zeta_{i-1,j}^{m+1/2} + \gamma \zeta_{i,j}^{m+1/2} + (\alpha - C_j) \zeta_{i+1,j}^{m+1/2} .
\end{aligned}$$

The resulting tridiagonal systems are similar to those encountered in the solution for $\zeta^{m+1/2}$, except that the dimension has increased by two, due to the "expansion of the mesh", e.g. for fixed i , (5.22) holds for $j = -2, -1, \dots, M+2$ where $\zeta_{i,-3}^{m+1} = \zeta_{i,M+3}^{m+1} = 0$. The resulting analysis regarding the invertability of the matrices is the same as before.

6. Domain interaction

1. ξ_1 flow from D_1 into D_2

After the field ξ_1 has been updated via Chorin's vortex method, equation (3.1b) may no longer hold. Some of the vortex blobs may have escaped through the boundary ∂D , and some may have been displaced inside the subdomain D_2 . The blobs which have escaped through ∂D are simply discarded from the computation, but the ones inside D_2 must be interpolated onto the finite difference mesh.

The blobs which cross into D_2 are interpolated onto the mesh by a simple linear interpolating scheme. The scheme is known to plasma physicists as "area weighting" or "charge sharing". There are two important considerations to keep in mind when interpolating. First, it is essential that no vorticity be created or destroyed. Secondly, the blobs crossing into D_2 will situate themselves at locations not coincident with the mesh points, and the interpolating scheme will replace these blobs with vorticity distributed on a grid. Hence, although the total amount of vorticity may be unchanged, it will be distributed differently. This different distribution will exert a change in the induced velocity, and it is important to minimize this effect.

Recall that ζ_j is not the value of the vorticity, but, is instead the vorticity integrated over the entire domain. In other words, if ξ_1 consisted of only 1 vortex blob,

$$\xi_1^m(\underline{r}) = \zeta_1 \xi_0(\underline{r} - \underline{r}_1^m), \text{ then } \iint_D \xi_1^m(\underline{r}) d\underline{r} = \zeta_1.$$

Assume that one vortex with strength ζ_1 has crossed into D_2 (see Fig. 3). Draw horizontal and vertical lines through the vortex center and calculate the ratio of the areas of the 4 resulting rectangles to the area of a mesh square:

$$A_1 = (d - x)(d - y)/d^2; \quad A_2 = x(d - y)/d^2 \quad ;$$

$$A_3 = xy/d^2; \quad A_4 = y(d - x)/d^2 \quad .$$

Compute the grid vorticities using the A_j 's:

$$\zeta_{1,j} = \zeta_1 A_1/d^2, \quad \zeta_{1+1,j} = \zeta_1 A_2/d^2, \quad .$$

$$\zeta_{1+1,j+1} = \zeta_1 A_3/d^2, \quad \zeta_{1,j+1} = \zeta_1 A_4/d^2.$$

The area weighting is done to minimize the "self-force" phenomenon known to plasma physicists. This phenomenon is a numerical error caused by the presence of a grid that has the vortex blobs exerting forces on itself. The hope is that by distributing the vorticity due to 1 vortex over 4 grid points, surrounding fluid particles do not realize that the vorticity carried by a vortex blob has been redistributed onto the grid. By means of the trapezoidal rule for integration, it can be shown that no vorticity was created or destroyed:

$$\iint_D \zeta_2 dx dy = \sum \zeta_{1,j} d^2 = \zeta_1 \sum_{j=1}^4 A_j = \zeta_1 .$$

It is interesting to note that the above method of interpolating is equivalent to Peskin's " δ " function technique [16]. In this case the support of the discrete " δ " function is 4 mesh squares instead of 16. Peskin needed 16 because he had 4 different "chains" of points which were linked by the boundary conditions. The 4 "chains" needed to

feel the forces simultaneously hence, the "large support" of the " δ " function.

The above interpolating scheme is also equivalent to assuming that the vortex blobs have a square base and that the basic blob stream function has the form

$$\epsilon_c(x,y) = \begin{cases} \frac{1}{d^4}(d - |x|)(d - |y|) , & |x|, |y| < d \\ 0 & , \text{ otherwise .} \end{cases}$$

To perform the interpolation with this blob, one simply evaluates the blob function at the grid points.

2. ϵ_2 flow from D_2 into D_1

The completion of the time-splitting scheme results in an "expanded" mesh with vorticity ϵ_2 deposited on mesh points outside ∂D_2 . This vorticity must then be rearranged into the form of vortex blobs to satisfy (3.1b). A simple method, albeit expensive, would be to replace every nonzero vorticity value of ϵ_2 outside D_2 with a blob centered on a mesh point. This new blob would have the strength $\tau_e = d^2 \zeta_{1,j}$. This procedure is expensive since it generates many new vortex blobs per time step.

To be able to keep down the number of newly reconstructed vortex blobs, the following simple scheme was devised.

Assume that only the four neighboring values $\zeta_{-2,j}$, $\zeta_{-1,j}$, $\zeta_{-2,j+1}$ and $\zeta_{-1,j+1}$ (see Fig. 4) need to be reconverted into a vortex blob. The total volume of vorticity that needs to be transferred is $\zeta_e = (\zeta_{-2,j} + \zeta_{-1,j} + \zeta_{-2,j+1} + \zeta_{-1,j+1})d^2$. These four grid values are combined into one blob centered at the center of mass of the four vorticity values.

$$\text{Define } |\zeta_e| \equiv |\zeta_{-2,j}| + |\zeta_{-1,j}| + |\zeta_{-1,j+1}| + |\zeta_{-2,j+1}| .$$

Then the center of the blob (x_e, y_e) is

$$\begin{aligned} x_e &= \delta - 2d + d(|\zeta_{-1,j+1}| + |\zeta_{-1,j}|)/|\zeta_e| , \\ y_e &= \delta + jd + d(|\zeta_{-2,j+1}| + |\zeta_{-1,j+1}|)/|\zeta_e| . \end{aligned} \quad (6.1)$$

The scheme (6.1) places the center of the blob inside the mesh square of interest. It is interesting to note that under some particular distributions of the mesh vorticity, scheme (6.1) is the inverse of the interpolating scheme described in the previous section.

Before discussing how scheme (6.1) is implemented to transfer the vorticity ξ_2 located in D_1 into vortex blobs, recall that at the beginning of each time step, ξ_2 is distributed on an $(M+1) \times (M+1)$ mesh contained in D_2 , e.g.

$$\xi_2^m = \left\{ \zeta_{i,j}^m \right\}_{i,j=0}^M .$$

After the time-splitting step, ξ_2^m has been updated and distributed on the expanded mesh; denoting this updated vorticity by $\xi_2^{m+1,*}$,

it follows

$$\xi_2^{m+1,*} = \left\{ \zeta_{i,j}^{m+1} \right\}_{i,j=-2}^{M+2} .$$

The values of $\xi_2^{m+1,*}$ that must be recombined into vortex blobs, compose the set

$$\left\{ \zeta_{i,j}^{m+1} \right\}^*$$

$$\forall i,j \in \{(i,j) \mid -2 \leq i,j \leq M+2\} - \{(i,j) \mid 0 \leq i,j \leq M\}.$$

Consider the bottom edge of D_2 , i.e. the vorticity values

$$\left\{ \zeta_{i,j}^{m+1} \right\} \text{ for } i = -2, -1, \dots, M+2, j = -1, -2.$$

Scheme (6.1) will generate $M+4$ vortex blobs lying between $y = \delta - 2d$ and $y = \delta - d$ where $\delta = \text{dist}(\partial D_2, \partial \mathcal{D})$.

Some care must be exercised to ensure that no new vorticity is created or destroyed and that the apportionment of vorticity among the blobs is similar to its previous apportionment on the grid. In particular, 2 blobs will be generated by the 6 values:

$$\zeta_{i-1,-2}^{m+1}, \zeta_{i-1,-1}^{m+1}, \zeta_{i,-1}^{m+1}, \zeta_{i,-2}^{m+1}, \zeta_{i+1,-2}^{m+1}, \zeta_{i+1,-1}^{m+1},$$

where $1 \leq i \leq M-1$. Scheme (6.1) will generate one blob inside the square with vertices $(i-1, -2)$, $(i-1, -1)$, $(i, -2)$, $(i, -1)$.

This blob will have a total vorticity specified by $\frac{1}{2}\zeta_{i-1,-2}^{m+1} + \frac{1}{2}\zeta_{i-1,-1}^{m+1} + \frac{1}{2}\zeta_{i,-2}^{m+1} + \frac{1}{2}\zeta_{i,-1}^{m+1}$. The remaining $\frac{1}{2}\zeta_{i,j}^{m+1}$ will be used to generate a blob in the neighboring square.

In a corner, a different algorithm is used. Consider the 8 values

$$\zeta_{-2,-2}^{m+1}, \zeta_{-1,-2}^{m+1}, \zeta_{0,-2}^{m+1}, \zeta_{-2,-1}^{m+1}, \zeta_{-1,-1}^{m+1}, \zeta_{0,-1}^{m+1}, \zeta_{-2,0}^{m+1}, \zeta_{-1,0}^{m+1}.$$

The corner blob is generated using the 4 grid values

$$\zeta_{-2,-2}^{m+1}, \frac{1}{2}\zeta_{-1,-2}^{m+1}, \frac{1}{2}\zeta_{-2,-1}^{m+1}, \frac{1}{3}\zeta_{-1,-1}^{m+1}.$$

The blob immediately to the right is generated using the values

$$\frac{1}{2}\xi_{-1,-2}^{m+1}, \frac{1}{2}\xi_{0,-2}^{m+1}, \frac{1}{3}\xi_{-1,-1}^{m+1}, \frac{1}{2}\xi_{0,-1}^{m+1}.$$

Finally, the blob above the corner is generated using the values

$$\frac{1}{2}\xi_{-2,-1}^{m+1}, \frac{1}{3}\xi_{-1,-1}^{m+1}, \frac{1}{2}\xi_{-2,0}^{m+1}, \frac{1}{2}\xi_{-1,0}^{m+1}.$$

In this way, all of the vorticity ξ_2^{m+1} lying outside D_2 has been transferred into blobs and the location of the blobs within each square assures that fluid particles several mesh points away do not realize that a transformation has occurred.

7. Numerical details

1. Numerical evaluation of velocity fields

At the beginning of each time step, evaluate a velocity \underline{u}_2 satisfying:

$$\nabla \times \underline{u}_2 = \xi_2 \quad \text{in } D, \text{ and} \quad (7.1)$$

$$\underline{u}_2 \cdot \underline{n} = -\underline{u}_1 \cdot \underline{n} \quad \text{along } \partial D. \quad (7.2)$$

The field \underline{u}_2 can be calculated by differentiating a stream function ψ_2 which satisfies:

$$\Delta \psi_2 = -\xi_2 \quad \text{in } D, \text{ and} \quad (7.3)$$

$$\psi_2(s) = -\psi_1(s) \quad \text{along } \partial D. \quad (7.4)$$

Thus, if

$$\underline{u}_2 = (\partial_y \psi_2, -\partial_x \psi_2),$$

equations (7.1) and (7.2) are satisfied.

To calculate ψ_2 , subdivide D into $L+1$ points in each direction and thereby define d as:

$$d \equiv 1/(L+1).$$

Let $\psi_{i,j}$ denote the approximation to $\psi_2(id, jd)$. Thus, using (7.3),

$$\psi_{i,j-1} + \psi_{i-1,j} - 4\psi_{i,j} + \psi_{i+1,j} + \psi_{i,j+1} = d^2 \xi_2[i,j] + o(d^2) \quad (7.5)$$

where now,

$$\xi_2[i,j] \equiv \xi_2(id, jd)$$

and $\xi_2[i,j] \equiv 0$, for all values $(id, jd) \notin D_2$. At points $\psi_{i,j}$ coinciding

with ∂D , the value of $\psi_{i,j}$ is calculated from (7.4) and the formula

$$\psi_1(\underline{r}) = \sum_j \epsilon_j \psi_0(\underline{r} - \underline{r}_j),$$

where ψ_0 is defined by (4.3).

If equation (7.5) is to be satisfied at every interior point, the following large linear system must be solved:

$$E \underline{\psi} = \underline{y},$$

where E is block tridiagonal

$$E = \begin{bmatrix} A & I & & & \\ & I & A & & \\ & & & I & A \cdot \\ & & & & \cdot \cdot \\ & & & & & A \end{bmatrix}, \quad A = \begin{bmatrix} -4 & 1 & & & \\ & 1 & -4 & 1 & \\ & & & 1 & -4 \cdot \\ & & & & \cdot \cdot & 1 \\ & & & & & & 1 & -4 \end{bmatrix},$$

dimension $(A) = (L \times L)$, and E is made up of L block rows and L block columns. The number L is set to be of the form $2^e - 1$, where e is a positive integer. The problem is thus placed in ready-made form to utilize the cyclic odd-even reduction algorithm [6].

At the completion of the evaluation of ψ_2 , there will be L^2 interior values and $4(L+1)$ boundary values (from (7.4)) of ψ_2 stored in an array. Denote these grid values by:

$$\psi_2 = \{\psi_{i,j}\}_{i,j=0}^{L+1}.$$

The velocity field \underline{u}_2 is then approximated by two different methods in the subdomains D_1 and D_2 . The use of the finite difference scheme in D_2 requires knowing \underline{u}_2 on only the mesh points. Hence, \underline{u}_2 is approximated there by centered differences of ψ_2 , e.g.

$$\underline{u}_2 = (u_2, v_2)_{1,j} = \frac{1}{2d}(\psi_{1,j+1} - \psi_{1,j-1} + \psi_{1-1,j} - \psi_{1+1,j}).$$

However, the vortex scheme has no such neat arrangement. The velocity \underline{u}_2 will need to be evaluated at the centers of the blobs which need not coincide with the mesh points. Consequently, it will be necessary to evaluate \underline{u}_2 anywhere inside D_1 . This is accomplished by approximating ψ_2 by a bi-quadratic spline in D , which is then analytically differentiated to yield a continuous approximation to \underline{u}_2 in D_1 .

The coefficients of the time-splitting scheme are functions of the velocity \underline{u} , a sum of the two fields \underline{u}_1 and \underline{u}_2 . At grid points on the "expanded mesh" in D_1 and on ∂D_2 , \underline{u}_1 is evaluated by (4.4) while \underline{u}_2 is computed by differentiating the spline approximator to ψ_2 .

At mesh points in D_2 , \underline{u}_2 is evaluated by centered differences as described above. It would, however, be too expensive to evaluate \underline{u}_1 on the mesh points inside D_2 by means of (4.4). In fact, the use of (4.4) in D_2 would defeat the purpose of the entire method, since a finite difference method was introduced to avoid the countless individual vortex calculations. Since, however, D_2 is specifically designed to be away from the boundary and since $\text{support}(\epsilon_1) \subseteq D_1$, it is possible to make a decent approximation to the field \underline{u}_1 in D_2 by numerically solving

$$\Delta \psi_1 = 0 \text{ in } D_2 \quad (7.6)$$

with the boundary values $\psi_1(s)$ specified on ∂D_2 . The function $\psi_1(s)$ is computed on ∂D_1 using

$$\psi_1(\underline{r}) = \sum_j \epsilon_j \psi_0(\underline{r} - \underline{r}_j). \quad (7.7)$$

Thus, the vortex formulas, (7.7) and (4.4) are used to calculate only the field \underline{u}_1 and the stream function on the boundary of D_2 , and (4.4) is used to calculate \underline{u}_1 on the "expanded" mesh.

The numerical calculation of (7.6) proceeds as above. The same grid used in the computation of $\psi_{1,j}$ is now used to compute $\phi_{i,j}$, where we define

$$\phi_{i,j} \equiv \psi_1(\delta + id, \delta + jd), \quad \delta \equiv \text{dist}(\partial D_2, \partial D),$$

$$1 \leq i, j \leq M - 1.$$

As above, (7.6) is approximated by

$$\phi_{i,j-1} + \phi_{i-1,j} - 4\phi_{i,j} + \phi_{i+1,j} + \phi_{i,j+1} = 0, \quad (7.8)$$

with the boundary values evaluated by (7.7). Equation (7.8) leads to the linear system

$$F \underline{\phi} = \underline{y}, \quad (7.9)$$

where F has the same form as the matrix E in the previously described system, but with block dimension $(M - 1) \times (M - 1)$.

The integer $M - 1$, unfortunately does not have the simple form of $2^p - 1$. In the problem of interest, $M - 1$ was always odd, sometimes of the form $4N - 1$, where N is a positive integer. The system (7.9) was solved by a method proposed by Hockney [13], where the number of cyclic reductions used was either one or two, depending on whether M was a multiple of two or four respectively. The FFT was not used, since the number was no longer highly composite. However, at the start of the calculations the required e-vectors and e-values were computed and stored in memory.

2. Approximation via b¹-quadratic spline

a. One-dimensional spline

Let $\{x_j\}$ be a monotonically increasing sequence of equally spaced points on the real line, and let $\{y_j\}$ denote the values that a function takes at those points, i.e. $y_j = y(x_j)$. Assume that $d = x_{j+1} - x_j$, for all j .

For some particular j and for $|x - x_j| \leq d/2$, define the spline interpolator,

$$f(x) = g_j + x D_0 g_j + \frac{1}{2} \left(x^2 + \frac{d^2}{4} \right) D_+ D_- g_j.$$

It is assumed above that $x_j = 0$, if not, substitute $x - x_j$ for x .

The operator D_0 , $D_+ D_-$ are the first and second central difference operators on the spline coefficients $\{g_j\}$, i.e.

$$D_0 g_j = \frac{1}{2d}(g_{j+1} - g_{j-1}), \text{ and } D_+ D_- g_j = \frac{1}{d^2}(g_{j+1} - 2g_j + g_{j-1}).$$

Thus, f may be rewritten as:

$$\begin{aligned} f(x) &= \frac{1}{8}(g_{j-1} + 6g_j + g_{j+1}) + \frac{x}{2d}(g_{j+1} - g_{j-1}) \\ &\quad + \frac{1}{2} \frac{x^2}{d^2} (g_{j+1} - 2g_j + g_{j-1}). \end{aligned}$$

Relationships for the spline coefficients are derived by requiring that the spline agree with the given values at the points x_j , i.e.

$$y_j = f(x_j) = \frac{1}{8}(g_{j-1} + 6g_j + g_{j+1}). \quad (7.10)$$

Equation (7.10) becomes a tridiagonal system for the spline coefficients $\{g_j\}$. Once the $\{g_j\}$ are known, the evaluation of the spline at a point necessitates the look up of three spline coefficients in a table,

namely g_{j+1} and g_j .

b. Two-dimensional spline

Assume that a square mesh of width d is superimposed on D and that $\psi_{i,j}$ is an approximation to $\psi_2(\text{id}, \text{jd})$, the stream function due to the vorticity ε_2 . There are a total of L points in each direction, hence a total of L^2 interior values of $\psi_{i,j}$. The problem at hand is to find continuous functions approximating the derivatives of ψ_2 which can be evaluated anywhere in D .

Assume for simplicity that $|x - x_j|, |y - y_j| \leq d/2$ and $(x_j, y_j) = (0, 0)$. Define the two-dimensional spline interpolator,

$$f(x, y) \equiv g_i^{(j)}(y) + x D_{01} g_i^{(j)}(y) + \frac{1}{2}(x^2 + \frac{d^2}{4}) D_{10} D_{-1} g_i^{(j)}(y) \quad (7.11)$$

where

$$D_{01} g_i^{(j)}(y) = \frac{1}{2d} [g_{i+1}^{(j)}(y) - g_{i-1}^{(j)}(y)] ,$$

and

$$D_{+1} D_{-1} g_i^{(j)}(y) = \frac{1}{d^2} [g_{i+1}^{(j)}(y) - 2g_i^{(j)}(y) + g_{i-1}^{(j)}(y)]$$

are the first and second central differences of $g_i^{(j)}(y)$ on the index i . The function $g_i^{(j)}(y)$ is the one-dimensional spline in the vertical (y) direction around $y = y_i$ at the point $x = x_j$:

$$g_i^{(j)}(y) = \frac{1}{8}(f_{i,j+1} + 6f_{i,j} + f_{i,j-1}) + \frac{y}{2d}(f_{i,j+1} - f_{i,j-1}) + \frac{1}{2d^2} y^2 (f_{i,j+1} - 2f_{i,j} + f_{i,j-1}) . \quad (7.12)$$

Once the coefficients are computed, the evaluation of the spline at any point necessitates the look up of the nine spline coefficients

$f_{1,j}, f_{1,j+1}, f_{1+1,j}, f_{1+1,j+1}$, which are stored in a table. This table can be stored over the old function values $\psi_{1,j}$ since they are now superfluous.

There are, however, more spline coefficients than original function values. Initially there were L^2 interior values $\psi_{i,j}$ and $4(L+1)$ boundary values, giving a total of $(L+2)^2$ values. Using the previous nomenclature these grid values are stored in the array $\{\psi_{i,j}\}_{i,j=0}^{L+1}$. As previously noted, the evaluation of the spline necessitates the look up of 9 spline coefficients. If the coefficient $f_{1,j}$ is stored in the same location as $\psi_{1,j}$, and if the spline is evaluated "near" the boundary location $(x,y) = (1\delta, 0)$, then the nine values $f_{1+1,1+1}, f_{1+1,1}, f_{1+1,0}$, and $f_{1,0}$ must be looked up. If the spline is evaluated "near" the corner $(x,y) = (0,0)$, then the nine coefficients $f_{1+1,1+1}, f_{1+1,0}, f_{0,1+1}, f_{0,0}$ must be looked up. Thus, a "border" of spline coefficients surrounds the original function values, and it is necessary to determine $(L+4)^2$ spline coefficients: $\left\{ f_{i,j} \right\}_{i,j=-1}^{L+2}$.

c. Linear equations for the spline coefficients

The $(L+4)^2$ spline coefficients will be determined by solving a system of $(L+4)^2$ linear equations that they must satisfy.

To begin, $(L+2)^2$ linear equations can be immediately found by requiring that the spline agree with the function at the L^2 interior and $4(L+1)$ boundary points. For example, if the spline must be correct at $(0,0)$, then using (7.12) in (7.11) obtain

$$\begin{aligned} \psi_2(0,0) &= \psi_{0,0} = g_0^{(0)}(0) + \frac{1}{8} d^2 D_{+1} D_{-1} g_{(0)}^{(0)}(0) \\ &= \frac{1}{64} \left[36f_{0,0} + 6(f_{0,1} + f_{1,0} + f_{0,-1} + f_{-1,0}) \right. \\ &\quad \left. + f_{1,1} + f_{1,-1} + f_{-1,1} + f_{-1,-1} \right] \end{aligned}$$

or

$$\begin{aligned} 64\psi_{0,0} &= f_{-1,-1} + 6f_{0,-1} + f_{1,-1} + 6(f_{-1,0} + 6f_{0,0} + f_{1,0}) \\ &\quad + f_{-1,1} + 6f_{0,1} + f_{1,1} . \end{aligned}$$

Similar equations will hold at the remaining $(L+2)^2 - 1$ remaining mesh points.

However, $(L+4)^2$ coefficients have been used in the $(L+2)^2$ equations, hence $(L+4)^2 - (L+2)^2 = 4L + 12$ more equations are needed to have the number of unknowns match the number of equations. After some examination, it is evident that the normal derivative of the spline at the boundary can be arbitrarily prescribed. Letting $f = f(x,y)$ denote the spline, it follows that $\frac{\partial f}{\partial n} = \underline{u}_2 \cdot \underline{s}$. Since, $\frac{\partial f}{\partial s} = -\underline{u}_2 \cdot \underline{n} = \underline{u}_1 \cdot \underline{n}$, by construction, the spline generates a velocity which nullifies the normal velocity at the boundary due to the vortices. It is tempting to prescribe $\frac{\partial f}{\partial n} = -\underline{u}_1 \cdot \underline{s}$ and thereby construct a velocity field satisfying both boundary conditions. However, I felt that this would give rise to an unreasonably non-smooth flow \underline{u}_2 which is not irrotational near the boundary, hence the temptation was resisted. I assumed that the flow \underline{u}_2 is smooth enough that (7.3) still held at the boundary of D . Equation (7.5) was hence assumed to hold on the boundary with the r.h.s. set equal to zero. Thus, it was possible to express the fictitious point outside the boundary in terms of its

image inside the domain and three points on the boundary (see Fig. 5).

Thus,

$$\psi_{i,-1} = 4\psi_{i,0} - \psi_{i,1} - \psi_{i-1,0} - \psi_{i+1,0}$$

Using this point outside the boundary, an approximation to the normal derivative can be obtained by letting,

$$\frac{2\psi_{i,1} + \psi_{i-1,0} + \psi_{i+1,0} - 4\psi_{i,0}}{2d} = \frac{\partial\psi}{\partial n} + o(d^2)$$

Equations (7.11) and (7.12) are combined with this relationship to derive an equation for the spline coefficient $f_{i,-1}$:

$$\frac{1}{2d}(2\psi_{i,1} + \psi_{i-1,0} + \psi_{i+1,0} - 4\psi_{i,0}) = \frac{\partial\psi}{\partial n} = \frac{\partial f}{\partial y}(i d, 0) = \frac{1}{16d} \left[f_{i+1,1} + f_{i-1,1} - f_{i+1,-1} - f_{i-1,-1} + 6(f_{i,1} - f_{i,-1}) \right]$$

or

$$8(2\psi_{i,1} + \psi_{i-1,0} + \psi_{i+1,0} - 4\psi_{i,0}) = -f_{i-1,-1} - 6f_{i,-1} - f_{i+1,-1} + f_{i-1,1} + 6f_{i,1} + f_{i+1,1} \quad (7.13a)$$

A similar equation will hold at the vertical boundary, $x = 0$, namely:

$$8(2\psi_{1,j} + \psi_{0,j-1} + \psi_{0,j+1} - 4\psi_{0,j}) = -f_{-1,j-1} + f_{1,j-1} - 6f_{-1,j} + 6f_{1,j} - f_{-1,j+1} + f_{1,j+1} \quad (7.13b)$$

Equations (7.13a,b) yield 4L linear equations for the coefficients $f_{-1,j}$, $f_{j,-1}$, $f_{L+2,j}$, $f_{j,L+2}$, $j = 1, 2, \dots, L$.

It remains to find twelve more equations for the twelve coefficients, adjacent to the corners, e.g. $f_{-1,-1}$, $f_{0,-1}$, $f_{-1,0}$, etc. These equations will be derived by prescribing values for the 1st order,

and mixed 2^{nd} order derivatives of the spline at the corners.

Since f is an approximation to ψ_2 , it is possible to prescribe the velocity $\underline{u} = \underline{u}_1 + \underline{u}_2$ to be zero at the four corners by specifying that

$$\underline{u}_2 = (\partial_y f, -\partial_x f) = -\underline{u}_1 = (-u_1, -v_1). \quad (7.14)$$

Equation (7.14) yields eight more linear equations for the spline coefficients; in particular, at the corner $(x,y) = 0$ obtain,

$$\frac{\partial f}{\partial x}(0,0) = v_1(0,0) \text{ and } \frac{\partial f}{\partial y}(0,0) = -u_1(0,0)$$

or

$$\begin{aligned} 16dv_1(0,0) &= -f_{-1,-1} + f_{1,-1} - 6f_{-1,0} + 6f_{1,0} - f_{-1,1} + f_{1,1} \\ -16du_1(0,0) &= -f_{-1,-1} - 6f_{0,-1} - f_{1,-1} + f_{-1,1} + 6f_{0,1} + f_{1,1} \end{aligned}$$

It is now necessary to obtain four more equations to close the system. These are obtained by equating $\frac{\partial^2 f}{\partial x \partial y}$ at each of the four corners to the backwards, 2^{nd} order correct divided difference of the computed stream function ψ_2 . For example, at the corner $(x,y) = (0,0)$ set,

$$\begin{aligned} 9\psi_{0,0} - 12(\psi_{0,1} + \psi_{1,0}) + 3(\psi_{0,2} + \psi_{2,0}) + 16\psi_{1,1} - \\ 4(\psi_{1,2} + \psi_{2,1}) + \psi_{2,2} = 4d^2 \frac{\partial^2 \psi_2}{\partial x \partial y} + o(d^4). \end{aligned} \quad (7.15)$$

Using (7.11) and (7.12),

$$\frac{\partial^2 f}{\partial x \partial y} = \frac{1}{4d^2}(f_{-1,-1} - f_{-1,1} - f_{1,-1} + f_{1,1}).$$

Labeling the l.h.s. of (7.15) C1, the required equation is:

$$C1 = f_{-1,-1} - f_{-1,1} - f_{1,-1} + f_{1,1}.$$

Similar equations hold at the remaining corners. Thus, the number

In other words, multiply every $(L + 4)$ th equation by $-\mu$, beginning with the first. μ is a solution of the equation

$$\frac{1}{\mu} + \mu = 6.$$

Assuming, momentarily that S_{μ}^{-1} exists and is easy to obtain, premultiply every "block" row by S_{μ}^{-1} , obtaining

$$Q \underline{f} = \underline{z} \quad (7.18)$$

where,

$$Q = \begin{bmatrix} -I & 0 & I & & & & \\ & I & 6I & I & & & \\ & & I & 6I & I & & \\ & & & I & \cdot \cdot & & \\ & & & & \cdot \cdot & I & \\ & & & & & I & 6I & I \\ & & & & & & -I & 0 & I \end{bmatrix},$$

I is the identity matrix,

$$\underline{z} = (z_{-1}, z_0, z_1, \dots, z_{L+2})^T$$

and for $j = -1, 0, 1, \dots, L + 2$,

$$z_j = S_{\mu}^{-1}(y_{\mu})_j.$$

Transform (7.18) into

$$Q_{\mu} \underline{f} = \underline{z}_{\mu}, \quad (7.19)$$

where

$$Q_{\mu} \equiv \begin{bmatrix} \mu I & 0 & -\mu I & & & \\ & I & 6I & I & & \\ & & I & 6I & I & \\ & & & I & \dots & \\ & & & & \dots & \\ & & & & & I & 6I & I \\ & & & & & & -I & 0 & I \end{bmatrix},$$

and $\underline{z}_{\mu} \equiv (-\mu z_{-1}, z_0, z_1, \dots, z_{L+2})^T$. System (7.19) is now "uncoupled" and can be solved for \underline{f} by solving $L + 4$ tridiagonal systems of order $L + 4$ of the type,

$$S_{\mu} g_e = b_e, \quad e = -1, 0, 1, \dots, L + 2,$$

where,

$$b_e = (-\mu z_{e,-1}, z_{e,0}, z_{e,1}, \dots, z_{e,L+2})^T.$$

Consider now the evaluation of S_{μ}^{-1} . The system to solve is

$$S_{\mu} \underline{x} = \underline{b} \quad (7.20)$$

where S_{μ} is defined above; $\mu = 3 + \sqrt{8}$ is a solution of $\frac{1}{\mu} + \mu = 6$.

Note that by defining the vectors

$$\underline{e}_1 \equiv (1, 0, 0, \dots, 0)^T, \quad \underline{w} \equiv (0, -1, -\mu, 0, 0, \dots)^T$$

and the matrix

$$B \equiv \begin{bmatrix} \mu & 1 & & & & \\ & 1 & 6 & 1 & & \\ & & 1 & 6 & 1 & \\ & & & \dots & & \\ & & & & \dots & \\ & & & & & 1 & 6 & 1 \\ & & & & & & -1 & 0 & 1 \end{bmatrix}$$

then $S_\mu = B + \underline{e}_1 \underline{w}^T$. Hence, using the Sherman - Woodbury formula,

$$S_\mu^{-1} = B^{-1} - B^{-1} \underline{e}_1 (1 + \underline{w}^T B^{-1} \underline{e}_1)^{-1} \underline{w}^T B^{-1}.$$

The solution of (7.20) can thus be directly written as

$$\underline{x} = B^{-1} \underline{b} - B^{-1} \underline{e}_1 (1 + \underline{w}^T B^{-1} \underline{e}_1)^{-1} \underline{w}^T B^{-1} \underline{b}. \quad (7.21)$$

Define the scalar, $\beta \equiv \underline{w}^T B^{-1} \underline{b}$, then recalling the definition of \underline{w}^T , it follows that

$$\beta = - \left[(B^{-1} \underline{b})_2 + \mu (B^{-1} \underline{b})_3 \right], \quad (7.22)$$

where $(B^{-1} \underline{b})_e$ is the e^{th} component of the vector $B^{-1} \underline{b}$. The matrix B has a ready LU decomposition, namely,

$$B = LU = \begin{bmatrix} 1 & & & & & & & & & \\ & 1/\mu & & & & & & & & \\ & & 1 & & & & & & & \\ & & & 1/\mu & & & & & & \\ & & & & \cdot & \cdot & & & & \\ & & & & \cdot & \cdot & & & & \\ & & & & & & 1/\mu & & & \\ & & & & & & -1/\mu & 1/\mu^2 & & 1 \end{bmatrix} \begin{bmatrix} \mu & & & & & & & & & \\ & \mu & & & & & & & & \\ & & \mu & & & & & & & \\ & & & \mu & & & & & & \\ & & & & \cdot & \cdot & & & & \\ & & & & \cdot & \cdot & & & & \\ & & & & & & \mu & & & \\ & & & & & & & 1 & & \\ & & & & & & & & 1-1/\mu^2 & \end{bmatrix}$$

and it is now obvious why μ was chosen to satisfy $\frac{1}{\mu} + \mu = 6$. For stability choose

$$\mu = 3 + \sqrt{8}.$$

The evaluation of $B^{-1} \underline{b}$ for the first term of (7.21) is now trivial since B has been decomposed into a product of triangular matrices.

The evaluation of $B^{-1} \underline{e}_1$ can be done analytically by induction.

To solve the system

$$B\underline{y} = \underline{e}_1$$

where dimension (\underline{y}) = dimension (\underline{e}_1) = n , first solve

$$L\underline{y} = \underline{e}_1, \text{ getting}$$

$$\underline{y} = (y_1, y_2, \dots, y_n)^T, \text{ where}$$

$$y_j = \begin{cases} (-1/\mu)^{j-1}, & \text{if } j = 1, 2, 3, \dots, n-1 \\ (-1/\mu)^{n-2}(1 + 1/\mu^2), & \text{if } j = n \end{cases}$$

Having solved for \underline{y} , solve

$$U\underline{v} = \underline{y}, \text{ to find}$$

$$v_n = (-1/\mu)^{n+2} \left(\frac{1+\mu^2}{1-\mu^2} \right),$$

$$v_{n-1} = \left(-\frac{1}{\mu} \right)^{n-3} \frac{?}{1-\mu^2},$$

$$v_j = \left(-\frac{1}{\mu} \right)^{2n-j-4} \left(\frac{1 + \mu^2(2n-j-1)}{1 - \mu^2} \right), \text{ for } j = 1, 2, \dots, n-2.$$

It is then easy to verify that

$$\underline{w}^T B^{-1} \underline{e}_1 = -(1/\mu^2)^{n-3}, \text{ hence}$$

$$(1 + \underline{w}^T B^{-1} \underline{e}_1)^{-1} = \left[1 - (1/\mu^2)^{n-3} \right]^{-1}.$$

Since $\mu^2 \approx 33.92$ make the simplification that,

$$(1 - (1/\mu^2)^{n-3})^{-1} = 1;$$

which is valid since n will be a large integer (> 20). Therefore,

$$B^{-1} \underline{e}_1 (1 + \underline{w}^T B^{-1} \underline{e}_1)^{-1} \underline{w}^T B^{-1} \underline{b} = \beta B^{-1} \underline{e}_1,$$

where β has been evaluated previously by (7.22).

The computation of $\beta B^{-1} \underline{e}_1$ proceeds inductively as follows:

define

$$c \equiv \beta/(1 - \mu^2)$$

$$a_1 \equiv c(-1/\mu)^{2n-5}, \text{ and } b_1 \equiv \mu^{2(n-2)}$$

and if $\underline{z} = (z_1, z_2, \dots, z_n)^T \equiv \beta B^{-1} \underline{e}_1$, it follows that

$$z_j = a_j(1 + b_j), \text{ while for}$$

$$j = 2, 3, \dots, n,$$

if $a_j \equiv -\mu a_{j-1}$, $b_j \equiv b_{j-1}/\mu^2$, then

$$z_j = a_j(1 + b_j).$$

Thus, having evaluated \underline{z} , and recalling (7.21), the required solution is

$$\underline{x} = S_\mu^{-1} \underline{b} = B^{-1} \underline{b} - \underline{z}.$$

Q.E.D.

3. Timing tests for the linear equations

The calculation of the stream functions ψ_1, ψ_2 , requires solving 3 large linear systems. However, with the aid of modern techniques these large sparse matrices are quickly inverted.

The first large system to solve is $E\underline{\psi} = \underline{y}$, where $\underline{\psi}$ contains the grid values of ψ_2 , the stream function due to ξ_2 . To be able to use a cyclic reduction algorithm [6] these were always $2^e - 1$ interior grid points per side. That is, the domain D was covered with a square mesh of size $d = 1/64$, or $1/32$. A test was done to time a cyclic reduction matrix solver which I wrote to solve the system. When the r.h.s. of the system was initialized so that all components were exactly one, the routine solved the equation in .15 seconds

with a maximum error of 4×10^{-11} . The test was done with a mesh size of $d = 1/64$.

The second large matrix to invert is F of (7.9). The unknowns in (7.9) are the grid values of the function ψ_1 in the domain D_2 . The system (7.9) is of smaller dimension than the previous one since domain $D_2 \subseteq D$. However, it is hoped that D_2 will expand as much as possible to be able to cover as large a part of D as possible. Therefore, I was not at liberty to choose the number of points in D_2 at will. A typical case using a interior mesh size $d = 1/64$, would be to have 55^2 unknowns to compute in (7.9). Using 51 points per side and initializing \underline{y} in (7.9) to have all components equal to one, the routine took .45 seconds using 1 cyclic reduction, and .25 seconds using 2 cyclic reductions. The two tests had a maximum error of 4×10^{-11} and 5×10^{-11} respectively.

Both tests above were done on the LLL CDC 7600 machine using the CHAT compiler and the timing was done with the subroutine OPTIM.

The third system to solve is (7.16), for the spline coefficients $f_{i,j}$. Using a total of 67^2 coefficients, a test program was run to find the coefficients needed to interpolate the function $e^x \cos y$ on the unit square. The routine took 6.1×10^{-2} seconds and achieved an accuracy of 10^{-14} when the spline was evaluated at the grid points. This test was run on the LBL CDC 7600 machine and was timed with the routine SECOND.

4. Tridiagonal matrix inversion

The heart of the fast poisson solvers, and the time-splitting

difference scheme involves inverting tridiagonal matrices. For the sake of completeness, the following algorithm is presented (Isaacson and Keller [16]).

Let the matrix A have the tridiagonal form

$$A = \begin{bmatrix} a_1 & c_1 & & & \\ b_2 & a_2 & c_2 & & \\ & b_3 & \cdot & \cdot & \\ & & \cdot & \cdot & c_{n-1} \\ & & & b_n & a_n \end{bmatrix}.$$

An LU decomposition for A can be inductively computed,

$$A = LU = \begin{bmatrix} \alpha_1 & & & & \\ b_2 & \alpha_2 & & & \\ & b_3 & \alpha_3 & & \\ & & \cdot & \cdot & \\ & & & \cdot & \cdot \\ & & & & b_n & \alpha_n \end{bmatrix} \begin{bmatrix} 1 & \gamma_1 & & & \\ & 1 & \gamma_2 & & \\ & & 1 & \gamma_3 & \\ & & & \cdot & \cdot \\ & & & & \cdot & \gamma_{n-1} \\ & & & & & 1 \end{bmatrix}.$$

$$\alpha_1 = a_1, \quad \gamma_1 = c_1/a_1$$

$$\alpha_i = a_i - b_i \gamma_{i-1}, \quad i = 2, 3, \dots, n$$

$$\gamma_i = c_i/\alpha_i, \quad i = 2, 3, \dots, n-1.$$

The above algorithm was used throughout the computer program with obvious modifications, depending on the particular need. For example, the cyclic reduction algorithm needs to invert a matrix with ones along the super and sub-diagonals and constant elements along the diagonal; hence the algorithm is modified to use only $3n$

multiplications.

One advantage of the above method is that if the same matrix needs to be inverted several times to solve the system with different right hand sides, then only one LU decomposition need be performed.

8. Numerical experiments and results

1. Interpolation tests

Before the program was put together and run on the cavity problem, the many subroutines were individually tested. One such test was performed on the routine which interpolates the vortex blobs onto the grid. As described in §6.1, the linear interpolation chosen is equivalent to Peskin's discrete " δ " function [20]. Peskin's thesis, however, describes a different version using trigonometric functions. In that work, Peskin found that better results were achieved by using the function.

$$\delta_{ij}(x,y) = \alpha g(x - x_j)g(y - y_i) ,$$
$$g(x) = \begin{cases} 1 + \cos\frac{\pi x}{d} & , |x| < d \\ 0 & , \text{otherwise} \end{cases} ,$$
$$\alpha = \text{constant.}$$

The constant α insures that $\int \delta(x,y) = 1$.

This particular interpolating scheme was tested against the scheme described in §6.1. The test was done by interpolating vortex blobs at several places on the domain and checking which version gave the better velocity when compared to the field induced by the vortex blob. The results showed that the linear interpolating scheme was superior.

Another test compared the linear scheme with the function

$$\delta_{ij}(x,y) = \beta h(x - x_j)h(y - y_i) ,$$

$$h(x) = \begin{cases} 3 - 4|x| - \cos\pi x, & |x| < d \\ 0, & \text{otherwise} \end{cases},$$

β = normalization constant.

Once again, the linear interpolating scheme produced the better velocity, and it was therefore chosen as the candidate for the computer program.

It is, however, not surprising that the linear scheme worked best. As mentioned in §6, it is important that the remainder of the fluid not sense that an interpolation has occurred; hence, the vorticity carried by each blob should be spread to as few mesh points as possible. Thus, the support of the interpolating " δ " function is four mesh squares, and this function is simply a product of two one-dimensional " δ " functions. Since this one-dimensional " δ " function has a support of two mesh widths and is non-zero on at most two mesh points, linear interpolation will be the best choice. The situation will differ if a larger support is specified in which quadratic, cubic, trigonometric, or higher order interpolation may be better.

2. Circulating flow in a cavity

The aforementioned numerical method was tested on the square cavity flow problem. The fluid was initially at rest and at $t = 0$ the bottom edge was made to slide to the right with a unit velocity. This velocity of the edge remained constant throughout the calculations and I strived to attain the steady-state solution by solving the time dependent equations.

The program was run with the following sets of parameters:

$$R \equiv \text{Reynolds number} = 10^3,$$

$$d = \text{interior mesh size} = 1/64, 1/32$$

$$h = \text{boundary discretization length} = \begin{cases} 1/30, & \text{if } d = 1/64 \\ 1/20, & \text{if } d = 1/32, \end{cases}$$

$$k = \text{time step} = \begin{cases} 0.1, & \text{if } d = 1/64 \\ 0.2, 0.1, 0.08, 0.05, & \text{if } d = 1/32, \end{cases}$$

$$\delta = \text{dist}(\delta D, \delta D_2) = \begin{cases} 4d, & \text{if } d = 1/64 \\ 3d, & \text{if } d = 1/32. \end{cases}$$

The pictorial output will be of the form of Figure 6. The boundaries of the square coincide with the outside arrows. Each arrow represents the magnitude and direction of the velocity at the tail of the arrow. Due to the relatively large range of $|U|$ throughout the cavity, the lengths of the arrows vary as $|U|^{1/2}$, i.e. one arrow twice as long as another, signifies a velocity field four times larger in magnitude. Furthermore, the arrows are scaled so that the largest plotted velocity draws an arrow as long as the plot mesh size ($= 1/32$). Whether the interior mesh, d , equals $1/64$ or $1/32$, the plot mesh size remains the same. In both cases, ($d = 1/64$ or $1/32$), the boundary of D_2 coincides with the plotting mesh.

Generally speaking, the results are the same using an interior mesh size $d = 1/32$, or $d = 1/64$. Figures 6-11 are from runs with $d = 1/64$, while figures 12-15 are with $d = 1/32$.

In both cases, as the run begins, strong vortices of positive sign are shed from the sliding edge to induce a tangential velocity equal to 1. After several steps these blobs are interpolated onto

the edge of D_2 where they begin to form a large vortex. Figure 6 shows the beginnings of the large vortex which will eventually settle near the center of the cavity and dominate the flow. The path of this center vortex is traced in Figure 16 as it spirals in toward its central position.

As this large vortex spirals in, it generates a tangential velocity component, $\underline{u} \cdot \underline{s}$, which is positive along the four walls. To compensate for this, blobs of negative sign are generated at the stationary walls. These negative blobs are then swept up and pushed into the corners. Figure 7 shows one such negative blob being carried upwards into the stationary corner $(x,y) = (1,1)$. This sweeping is especially evident in Figure 14 for which $d = 1/32$. In that figure it is evident how negative vorticity is generated at the stationary walls and convected by the flow into the corners.

The blobs of negative sign which are thrust into the sliding edge in the lower left-hand corner of Figure 14 will cause a very strong positive blob to be generated at the next time step so that the condition $\underline{u} \cdot \underline{s} = 1$ may be satisfied. This strong blob of positive sign is responsible for the peculiar loop that the stream-lines make near the downstream corner of the sliding edge (Figures 9-11). Streamlines running parallel to the boundary $x = 0$ carry negative vorticity which has been shed from that edge. These streamlines then run into strong blobs of positive sign coming off the edge $y = 0$, generating a counter-clockwise motion, and it is this counter-clockwise motion which causes the loop.

At the beginning of each run, there were no vortices present in

the fluid since the flow was assumed to be at rest. After the first time step, some blobs of positive sign were shed from the sliding edge. These blobs in turn caused other blobs to be generated along the four walls in order that the tangential condition be satisfied. As time progresses some blobs may be lost through ∂D and others may enter into D_2 . When the equations of motion are solved in D_2 by the time-splitting scheme, other blobs are regenerated in D_1 by the "expanding mesh". Eventually a balance is reached between the number of blobs being generated on the expanding mesh and along ∂D , and the blobs lost through ∂D or entering into D_2 . This stable figure has an almost linear dependence on the time step (Figure 17).

The results for $d = 1/32$ were obtained by first running the program with $d = 1/32$, $k = .2$, $h = 1/20$ for 80 time steps. Using the resulting vorticity field as an initial condition, the program was restarted for the 4 different time increments, 0.05, 0.08, 0.1, 0.2.

As expected, halving the time step is roughly equivalent to doubling the stable number of vortices in D_1 . It seems that every vortex has a finite life expectancy in D_1 after which it is either lost through the boundary or enters into D_2 . Halving the time step allows the vortex to exist for twice as many time steps in D_1 . Since, approximately the same number of vortices is still generated per time step, twice as many vortices will be allowed to live inside D_1 .

The foregoing can be altered by allowing blobs to "merge" if they lie one on the other, and the computer program does exactly that.

At the time that the blobs need to be moved, it is necessary to calculate their distance from each other to calculate the velocities. If that distance is less than $\sigma/10$, the two vortices are merged and are treated as one. Thus, there is an upper limit to the possible number of vortices in D_1 . This linear dependence of the number of vortices on the time step would hold in exterior flows as well, so long as the boundary is finite and the domain D_1 enclosing the boundary be also finite.

The runs were terminated when I felt that no new information could be obtained from them. For the case $k = 0.1$, $d = 1/64$, the program was run for 260 time steps. For the case $k = 0.2$, $d = 1/32$, 160 time steps were run. For the case $k = 0.1$, $d = 1/32$, 140 time steps were run after initializing the field with the results of the 80th time step at $k = 0.2$. The cases $k = 0.08$, 0.05 were run to determine if the stable number of vortices exhibited the expected growth.

Figures 8, 12, 14 are plots of the velocity fields at one time step for three different runs. These plots are not atypical, but there is a slight variation of the field from one time step to another. There is always a large vortex present in the center of the cavity which dominates the flow, but the location of the vortex center does not remain entirely stationary. I feel that allowing the program to run for more time steps would eventually fix the location of the large vortex to one general area. The counter-rotating vortices in the corners, on the other hand, are not always present. At times they disappear for several time steps only to reappear in subsequent

time steps as new blobs are convected to the corners.

The program will probably never reach a true steady state in which there is absolutely no change between time steps, since there is always a random component present in the vorticity field. This randomness not only affects the vorticity ξ_1 in D_1 but also affects the boundary conditions of the time-splitting scheme, hence the finite difference scheme itself. Thus, I expect, the center of the large vortex to always oscillate slightly about some equilibrium point.

The corner vortices, being viscous in nature (Batchelor [2]), and smaller in size are more influenced by the randomness of ξ_1 . The negative vorticity composing them may, at any one time step, escape through ∂D only to be replaced by new blobs to satisfy the boundary conditions. Thus, a dynamic equilibrium will be attained; that is, as vorticity is lost, more will be generated by the system as it is needed. Whenever $\underline{u} \cdot \underline{s} > 0$ at a stationary wall, a blob of negative sign will be generated to cancel that component. This blob will diffuse into the fluid and be convected by the velocity field.

All this is to be expected in problems with high R . That is, there is always a randomness to any flow; the higher the Reynolds number, the more random the flow. Hence, to get a very detailed description it may be necessary to average several time steps.

Figures 9, 10, 11, 13, 15 represent the velocity fields after averaging. As is evident, the randomness of previous plots is smoothed out, and the finer details of the flow become apparent. The resulting graphs of the velocity profiles (Figures 18, 19) and the calculation of the extent of the spread of the corner vortex

were done using the average plot of 60 time steps (Figure 11).

Figure 20 merits some discussion. It lists the induced velocities on the sides of the cavity by the field plotted in Figure 11.

The velocities should be read along each side as that side is followed in the positive direction, i.e. keeping the domain D on one's left.

The first striking impression is that the tangential boundary condition

is not very well satisfied. For example, the horizontal velocity

U is never exactly one along the bottom sliding edge, nor is it

exactly zero along the top stationary wall. Similarly, the component

v is not exactly zero along the side walls. A detailed explanation

regarding why the plotted component $\underline{U} \cdot \underline{s}$ in Figure 20 does not

satisfy the required boundary condition is postponed until §9.

However, it should now be mentioned that the plotted velocity \underline{U} ($= \underline{U}_1 + \underline{U}_2$) takes into account only the vorticity already present

in the fluid from the previous time step. The viscous boundary

layer which is responsible for satisfying the tangential boundary

condition is not used in plotting, and, in fact, Chorin's scheme

assumes it to play only a local role at the time of its creation.

Hence, the tangential boundary condition is always satisfied by

construction. The boundary layer is omni-present but has a global

effect only upon having diffused into the fluid.

The extent of the spread of the corner vortices can be determined

by a close examination of Figure 20. In particular, along side 3

($y = 1$), the tangential velocity is first positive signifying flow

toward the corner $(x,y) = (1,1)$. Progressing down the table, signifies

movement in the direction of decreasing x ; the component U changes

sign becoming negative. The sign of U remains negative until the corner $(x,y) = (0,1)$ is approached when U changes sign once more, becoming positive, denoting the presence of another corner vortex.

The normal boundary component in Figure 20, on the other hand, cannot be so rationalized. The results in Figure 20 are a measure of the accuracy of the method described in §5, and could be improved by use of a finer mesh size d at the cost of larger demands of storage.

Finer mesh sizes, surprisingly, do not necessarily mean slower running times. A considerable portion of time is spent computing vortex interactions. Therefore, it is the number of vortices present in the fluid which primarily governs the speed of the method, and as previously noted it is the length of the time step which effectively dictates how many vortices will exist in D_1 . However, recalling the stability results of §5, it is not possible to reduce the interior mesh size independently of k , since the "expanding" mesh assumes that vorticity cannot travel much more than one mesh width in a time duration of $k/2$.

Figure 17 also includes approximate running times for the program once the number of vortices in D_1 had stabilized. The program was written in LRLTRAN, the Livermore Laboratory version of FORTRAN, and was run on the CDC 7600 R and S machines. The program was compiled by the CHAT compiler and the timing was done with the aid of the subroutine OPTIM. It should be noted that the times given in Figure 17 give the total time needed to complete one entire time step. The times listed include the work necessary to do such extraneous

things as plotting the velocity field every time step.

3. Comparison of results

As mentioned in §1, the large scale features of the flow for large R are well established. They have been observed experimentally by Pan and Acrivos [1] for Reynolds numbers as high as 4000, and numerically by Burgraff [5] ($R = 400$), Bozeman and Dalton [3] ($R = 10^3$), Runchal et al. [21] ($R = 10^4$), and Greenspan ($R = 2 \times 10^3, 10^4, 1.5 \times 10^4, 10^5$) among others. All of the above experimenters obtained their results by solving the steady-state equations by an iterative procedure. Most stopped their calculations when the difference between successive iterates was less than some predetermined constant.

Bozeman and Dalton sought to minimize the residual, defined as $r \equiv Ax - b$, when solving the linear system $Ax = b$. As pointed out by Bozeman, the above conditions simply imply the attainment of solutions which do not change appreciably from one iteration to another. I make no claim that the results presented here are the exact ones, but watching the large vortex form and spiral in towards an equilibrium state does lead me to believe that a correct steady-state is quickly becoming attained. It should be further mentioned that at no time did I experience a lack of convergence of the method. That is, making a judicious choice of k, h, d always led to the familiar large vortex core with counter-rotating vortices in the stationary corners.

Figure 21 exhibits the size of the upstream corner vortex as compared with other results. The size of the vortex is defined

as the distance that the back flow extends along the vertical side. This result agrees surprisingly well with the experiments of Pan and Acrivos and the calculations of Bozeman and Dalton and contrast with the calculation of Greenspan at $R = 2000$.

Figures 18 and 19 give the velocity profiles of U and V along horizontal and vertical lines through the vortex center. For comparison purposes, Figure 22 includes the results of Bozeman for $R = 1000$, Burgraff's for $R = 400$, and Burgraff's theoretical result as $R \rightarrow \infty$. Note the linear dependence of the velocity profiles near the vortex center signifying the constant vorticity core predicted by Batchelor [2].

Figure 23 plots the location of the center of the primary vortex as compared to other results. As evidenced by Figure 16, the location of this center does not remain motionless, but does seem to be spiralling in towards the point (.52, .45). My faith in the correctness of this conclusion is further enhanced by the fact that averages of the runs using different parameters give similar results. Comparing the location of this center with the results of Bozeman implies that the hybrid method described here places the vortex center closer to the physical center of the cavity itself.

9. Discussion of Results

The preceding was a presentation of a numerical method for use in fluid flow problems with high Reynolds numbers. It is a hybrid method utilizing a well established finite difference scheme in conjunction with the newer vortex scheme proposed by Chorin [8]. The method has yielded accurate results on the square cavity flow problem, an example of a steady circulating flow.

As a hybrid method, it must simultaneously satisfy the requirements imposed on each individual scheme, as well as to any other restrictions imposed by the combination process. As described in §5, the time-splitting scheme is unconditionally stable in regards to the time step k , and the mesh size d , when the familiar Von-Neuman stability analysis is used. However the solution of the tridiagonal matrices formally implies stability-like restrictions on the time step ((5.18) and (5.21)) to guarantee that no growth of errors occur in the inversion process.

A brief remark should be made regarding the conditions (5.18) and (5.21). They stem from the stability criteria (5.16a,b) which translate into the requirement that the matrices (5.15) be diagonally dominant. The more strongly the inequalities (5.16a,b) are satisfied, the more diagonally dominant the matrices become, and the less the unknowns $\xi_{1,j}^{m+1/2}$ depend on each other. This weaker dependence means that the boundary conditions and the values ξ_2^m takes on mesh points bordering ∂D_2 exert a weaker effect on the solution of ξ_2^m in the interior of D_2 . Hence, far from the domain D_1 , the vorticity ξ_2

is less influenced by the random fluctuations of ξ_1 .

Being relatively new, the vortex scheme does not yet possess well-established stability conditions. In particular, no known relationship exists between k and the boundary discretization length, h . One confusing aspect of the scheme is the apparent lack of satisfaction of the tangential boundary condition by the velocity \underline{U} (Figure 20). As noted in §8.2, the velocity \underline{U} does not include the numerical boundary layer which always exists to satisfy the tangential boundary condition. In fact, \underline{U} represents the velocity outside the boundary layer and is thus useful to spot unexpected features such as back flow or separation.

Although the numerical boundary layer can be evaluated at $t = t_m$, it has a global effect only at $t = t_{m+1}$ when it has been subdivided into vortex blobs and allowed to diffuse according to the heat equation. Roughly half of the blobs are immediately lost as they diffuse through ∂D and are thus discarded from the computation. The other half travel too far with the current choice of parameters. The structure of the blobs inside the cut-off length, $\sigma (= h/2\pi)$, was designed to have them exert a constant tangential velocity on ∂D as they diffused normally into the fluid. However, the diffusive component is a random step in time, and in a time duration of k , the blobs experience a random push with a standard deviation of $\sqrt{2k/R}$. The parameters chosen imply that those blobs diffusing into the fluid travel beyond σ after they have been created. Thus, although at the time of its creation the numerical boundary layer contained the correct amount of vorticity to cancel $\underline{U} \cdot \underline{s}$

at ∂D , after diffusing in the form of blobs, it may not be strong enough, and the tangential boundary condition must be satisfied by the creation of another layer.

It may then be inquired if it is better to choose a smaller k , or larger σ , i.e. larger h , to obtain better looking results than those depicted in Figure 20. Unfortunately, several factors need to be simultaneously weighed before decreasing k indiscriminately. As described in §8, smaller time steps mean more vortices, hence longer running times. Furthermore, the normal boundary component is canceled by the construction of a flow which is irrotational near ∂D . This flow is currently constructed by finite differences since no sharp gradients are expected to occur in it in areas bordering on ∂D . By experimenting with smaller time steps, I have observed a "rougher" looking velocity field near the boundary, which I attribute to the fact that the proximity of the new vortices to ∂D caused by the smaller k 's give less smooth normal boundary conditions which need to be cancelled by the irrotational flow. This, in turn, causes worse satisfaction of the normal condition. Thus, in trying to better satisfy the tangential condition one may worsen the normal one.

A less obvious consideration was proposed by Chorin [8]. The convective component in the vortex scheme is $O(k)$, while the diffusive component depends on the standard deviation of the random walk and is thus $O(k^{1/2})$. Hence, as k is decreased, the diffusive component may dominate, or at least exert a greater influence, over the convective component making the flow look more random.

Furthermore, less convection implies lower R , hence, care must be exercised that one is not solving a problem of the wrong viscosity.

On the other hand, increasing σ , hence h , means the boundary layer is subdivided into fewer blobs and is, therefore, a worse approximation. The entire intention is defeated, the tangential condition may be better satisfied, but only at fewer points along the boundary.

It should be asked whether it is fair to demand fine detail from the vortex scheme near ∂D . Fine detail is unavailable to finite difference methods, anyway, unless one uses prohibitively small mesh sizes, or as proposed by Dorodnicyn [11] discretizes different approximations (boundary layer, ideal-fluid, etc.) in their corresponding areas. The method described here can be made more accurate by simultaneously decreasing k and d at the expense of greater demands of storage and computer time. The improvement in the satisfaction of the normal boundary condition can be easily judged by looking at tables such as Figure 20. Although that flow represents the flow outside the numerical boundary layer, it should be parallel to ∂D . However, improvements in the satisfaction of the tangential boundary condition may be harder to judge and remains an open problem.

The domain interaction imposes no profound considerations or restrictions. As presented here, no restriction arises from the flow of ξ_1 into D_2 and only two criteria should be adhered to in regards to flow of ξ_2 into D_1 . First, the method assumes that vorticity cannot be convected in a normal direction much beyond one mesh size in a time duration of $k/2$. Secondly, the standard deviation of the

random walk should not be so large as to allow too much vorticity to diffuse beyond one mesh size in a time $k/2$. Therefore, it is not permissible to decrease d without simultaneously decreasing k .

It is not my intention, however, to paint too gruesome a picture, as none of the above ambiguities or restrictions caused convergence difficulties. The numerical program gave convincing results for all choices of parameters discussed. Initially, I expected to make δ no smaller than $O(R^{-1/4})$, but the results presented here were done when the edge of the "expanding" mesh was a distance $O(R^{-1/2})$ away from ∂D . This turned out to be $\delta = 4d$ when $d = 1/64$, and $\delta = 3d$ when $d = 1/32$. The results obtained give further credence to the usefulness of the vortex scheme to deal with the creation of the correct vorticity in problems at large R .

The hybrid method presented here was tested on a specific problem with a convenient geometry. The computer program exploited the particular geometry by using fast numerical techniques ready-made for these domains. However, the basic idea should be applicable in a wide class of flows. Basically one should try to subdivide the domain of interest into two subdomains, one bounded near the boundaries, and one away from the boundaries, possibly unbounded, depending on the problem. In the domain near the boundary the equations of motion are solved by the vortex method. In the other domain a suitable finite-difference method, or another method of a non-random type, can be chosen, with some care taken if this domain is unbounded.

10. References

1. A. Acrivos and F. Pan, "Steady flows in rectangular cavities," J. Fluid Mech., 28, (1967), pp. 643-655.
2. G.K. Batchelor, "On steady laminar flows with closed streamlines at large Reynolds numbers," J. Fluid Mech., 1, (1956), pp. 177-190.
3. J.D. Bozeman and C. Dalton, "Numerical study of viscous flow in a cavity," J. Comp. Phys., 12, (1973), pp. 348-363.
4. O. Buneman, "Subgrid resolution of flow and force fields," J. Comp. Phys., 11, (1973), pp. 250-268.
5. O. Burgraff, "Analytical and numerical studies of the structure of steady separated flows," J. Fluid Mech., 24, (1966), pp. 113-151.
6. B.L. Bugbee, G.H. Golub, and C.W. Nielson, "On direct methods for solving Poisson's equations," SIAM J. Numer. Anal., 7, No.4, Dec. 1970, pp. 627-656.
7. A.J. Chorin, "Numerical solution of the Navier-Stokes equations," Math. of Comp., 22, N. 104, (1968), pp. 745-762.
8. A.J. Chorin, "Numerical study of slightly viscous flow," J. Fluid Mech., 57, (1973), pp. 785-796.
9. R. Courant, K.O. Friedrichs, and H. Lewy, Mathematische Annalen, 100, (1928).

10. B. Davari, "Numerical study of viscous flows past a circular cylinder: application of Chorin's method," LBL-2480, UC-32, Mathematics and Computers, TID-4500-R61.
11. A.A. Dorodnicyn, "Review of methods for solving Navier-Stokes equations," Proceedings of IIIrd International Conference on Numerical Methods in Fluid Mechanics, Lecture Notes in Physics, Springer-Verlag, 1973, pp. 1-11.
12. D. Fischer, G. Golub, O. Hald, C. Leiva, and O. Widlund, "On Fourier-Toeplitz methods for separable elliptic problems," STAN-CS-73-375, July 1973.
13. D.G. Fox, J.R. Herring, R.H. Kraichnan, and S.A. Orszag, "Decay of two-dimensional homogeneous turbulence," J. Fluid Mech., 66, part 3, (1974), pp. 417-444.
14. D. Greenspan, "Numerical studies of prototype cavity flow problems," Comput. J., 12, (1967), pp. 643-655.
15. R.W. Hockney, "The potential calculation and some applications," Methods in Comp. Physics, 9, Plasma Physics, (1970), Academic Press, New York, pp. 136-210.
16. E. Isaacson and H.B. Keller, Analysis of Numerical Methods, (1966), John Wiley and Sons, New York.
17. M. Kawaguti, "Numerical solution for the Navier-Stokes equations for the flow in a two-dimensional cavity," J. Phys. Soc. of Japan,

16. (1961), pp. 2307-2315.
18. J. Lamperti, Probability, (1966), Benjamin, New York.
19. G. Marshall, "Computational methods in viscous flow problems," Ph.D. thesis, Polytechnic Institute, Delft, Holland, (1973).
20. C. Peskin, "Flow patterns around heart valves: a numerical method," J. Comp. Phys., 10, (1972), pp. 252-271.
21. A.K. Runchal, D.B. Spalding and M. Wolfshtein, "Numerical solution for the elliptic equations for the transport of vorticity heat and matter in two-dimensional flow," Phys. Fluids, 12, (1969), pp. II.21-II.22.
22. A.I. Shestakov, "Vortex blobs in a square cavity, application of Chorin's method," LBL-3393, UC-32 Mathematics and Computers TID-4500-R62.

11. Figures

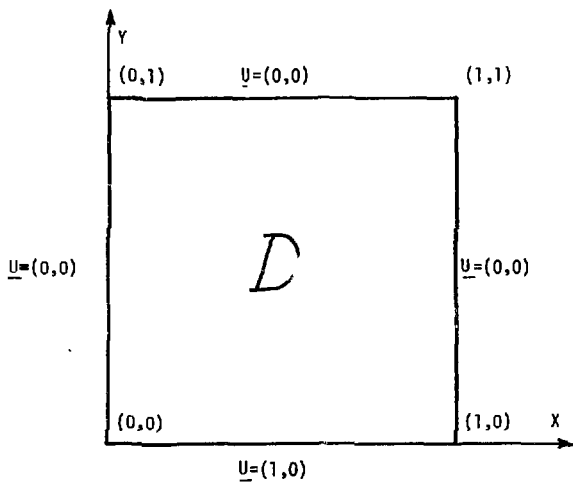


Fig. 1. Domain of interest and boundary conditions.

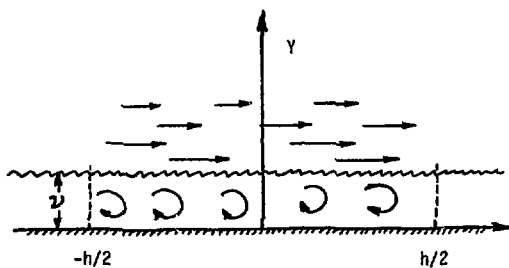


Fig. 2. Boundary layer along a wall.

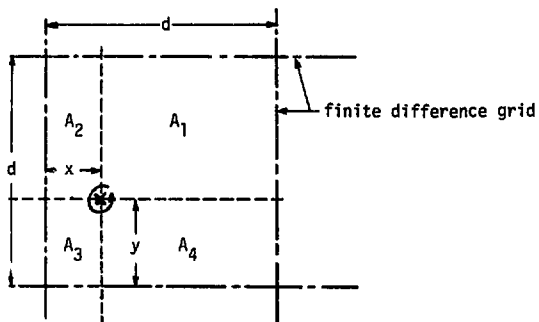


Fig. 3. Area weighting to interpolate vortex onto finite difference grid.

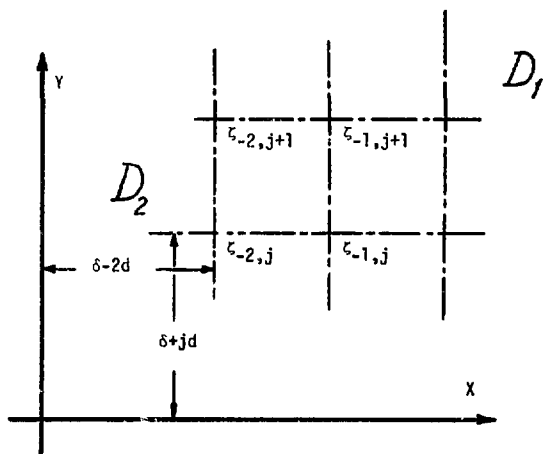


Fig. 4. Vorticity flow from D_2 into D_1 via "expanding mesh".

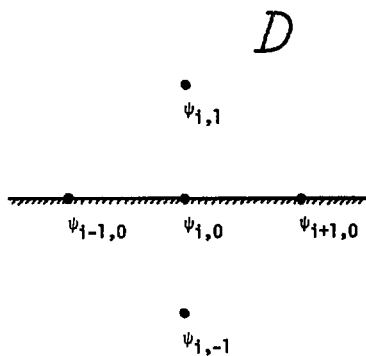


Fig. 5. Finite difference grid points on the boundary.

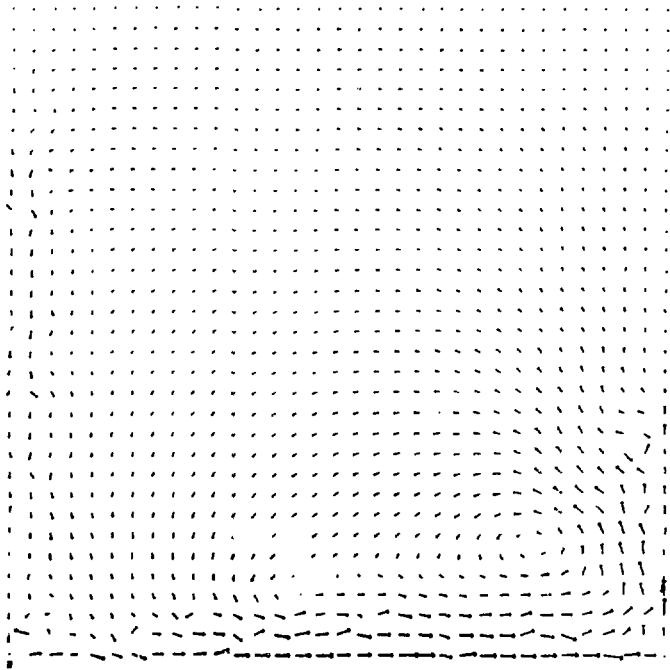
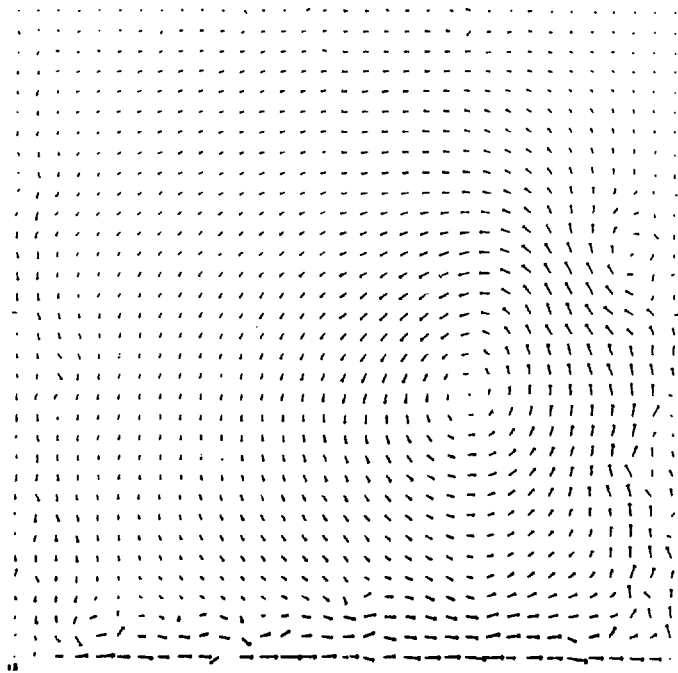


Fig. 6. Velocity plot, $k=0.1$, $d=1/6^c$ time step 22.

Creation of primary vortex.



**Fig. 7. Velocity plot, $k=0.1$, $d=1/64$, time step 53.
Example of creation of negative vorticity by
primary vortex.**

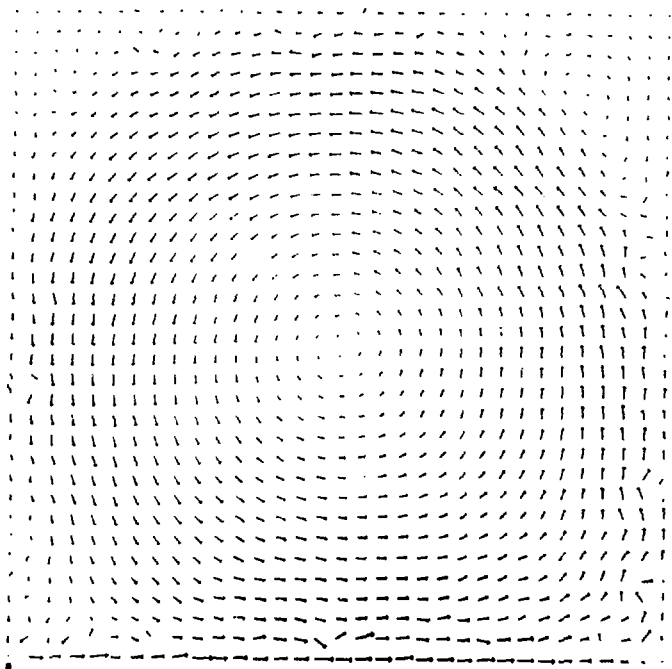


Fig. 8. Velocity plot, $k=0.1$, $d=1/64$, time step 238.

Typical velocity field at one time step; note counter-rotating vortex in the corner.

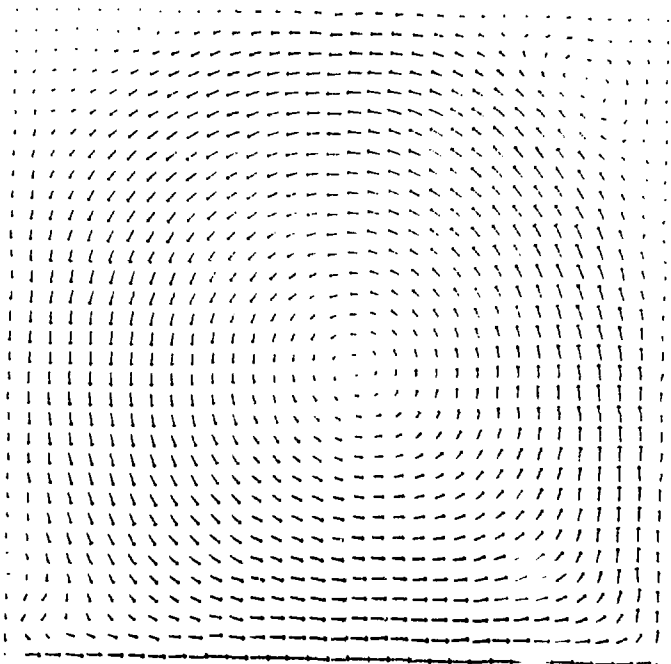


Fig. 9. Velocity plot, $k=0.1$, $d=1/64$. Velocity field after averaging over time steps 201-230.

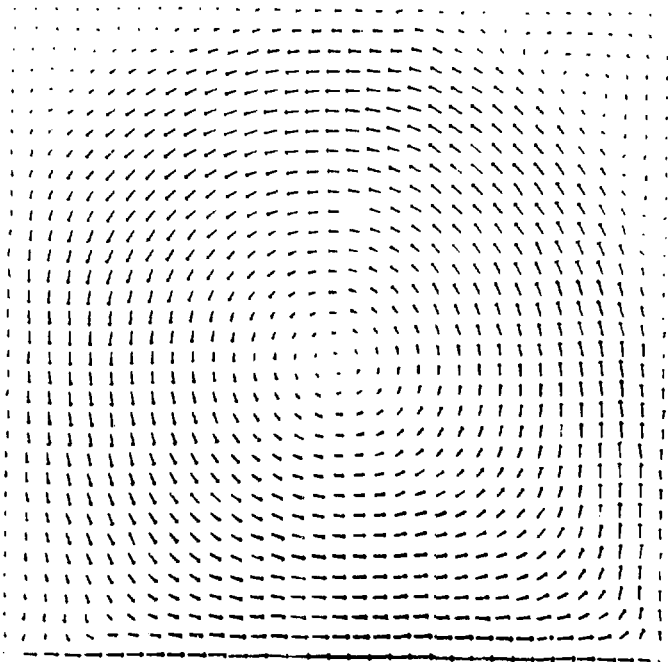


Fig. 10. Velocity plot, $k=0.1$, $d=1/64$. Velocity field after averaging over time steps 231-260.

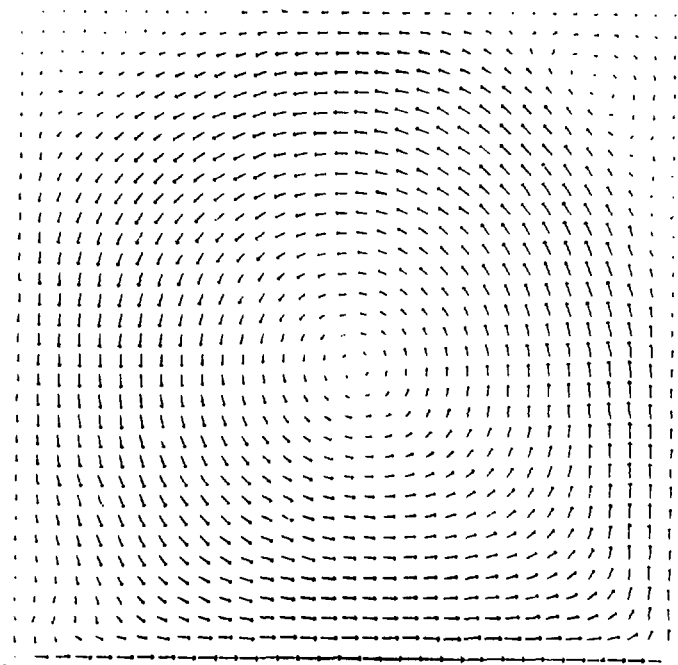


Fig. 11. Velocity plot, $k=0.1$, $d=1/64$. Velocity field after averaging over time steps 201-260 (i.e. Figs. 9 & 10).

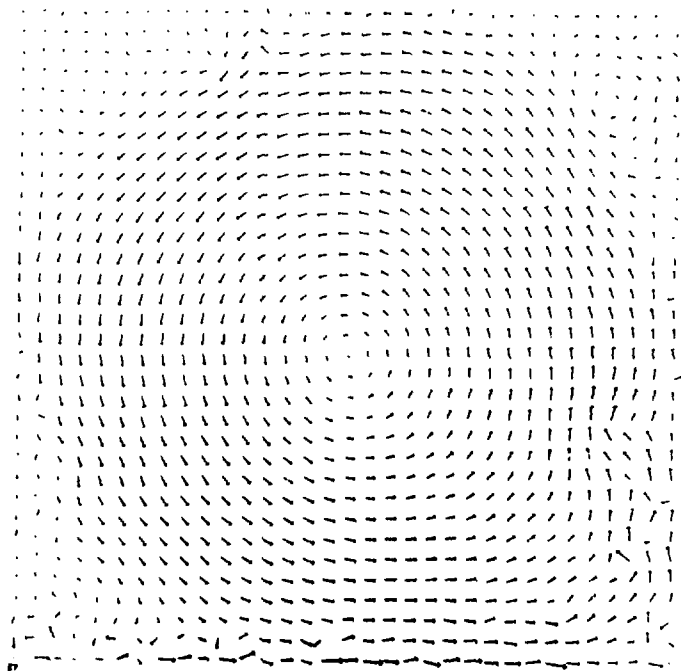


Fig. 12. Velocity plot, $k=0.1$, $d=1/32$, time step 217.

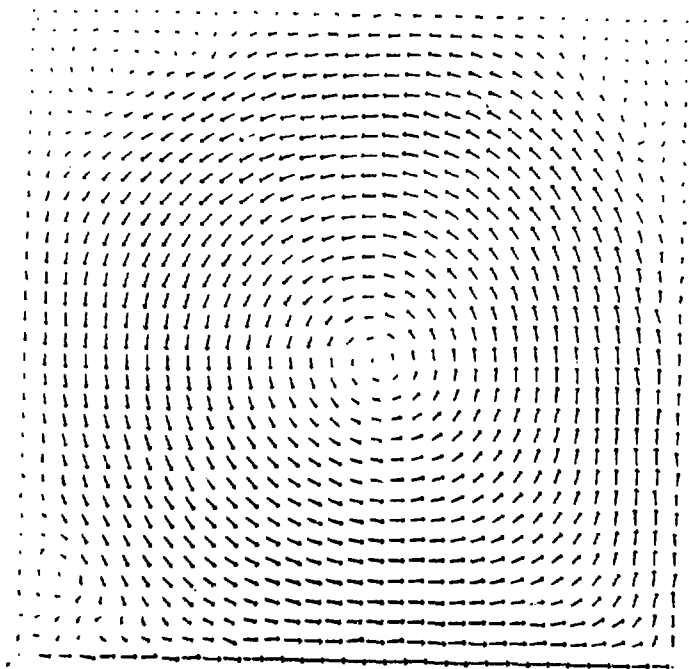
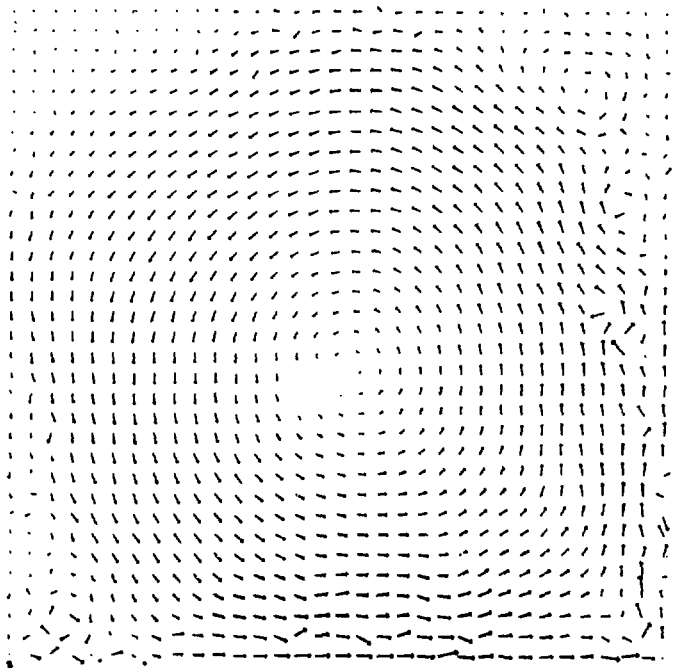


Fig. 13. Velocity plot, $k=0.1$, $d=1/32$. Velocity field after averaging over time steps 181-220.



**Fig. 14. Velocity plot, $k=0.2$, $d=1/32$, time step 105.
Example of convection of negative vorticity
into the corners.**

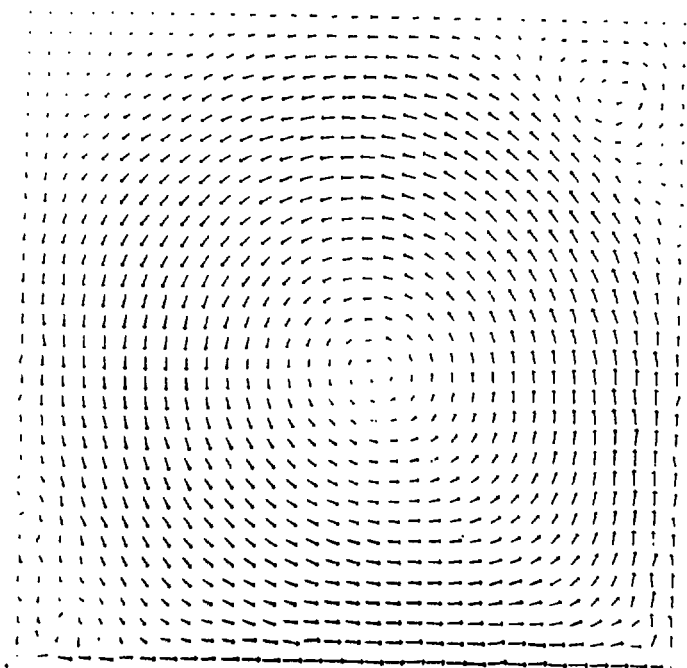
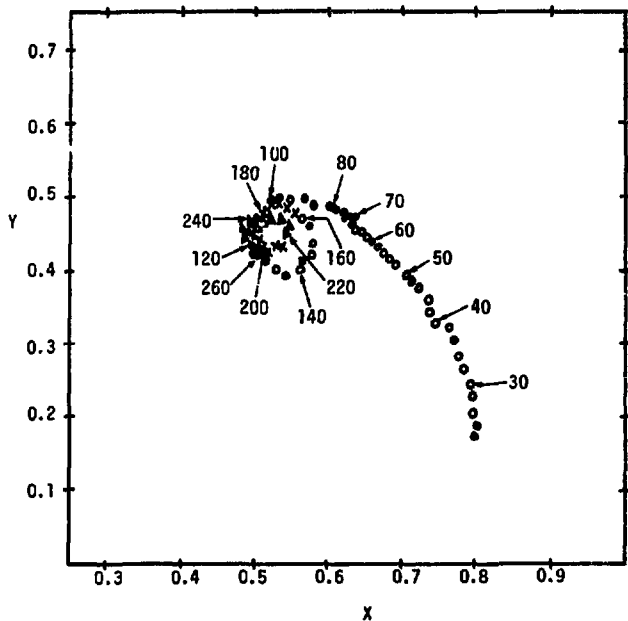


Fig. 15. Velocity plot, $k=0.2$, $d=1/32$. Velocity field after averaging over time steps 101-120.



● denotes time steps 22 - 160
 X denotes time steps 164 - 216
 ▲ denotes time steps 220 - 260

Fig. 16. Trace of path that center of primary vortex travels,
 $k=0.1$, $d=1/64$, $h=1/30$.

d	h	k	number of vortices	time/iteration (seconds)
1/32	1/20	0.05	1890*	29.5*
1/32	1/20	0.08	1620	24
1/32	1/20	0.1	1220	15
1/32	1/20	0.2	750	7
1/64	1/30	0.1	1550	25

d = interior mesh width

h = boundary discretization length

k = time interval

number of vortices = approximate stable number of vortices in D_1

time/iteration = running time necessary to complete 1 time step

Fig. 17. Comparison table for runs with different time steps and different mesh widths.

* Program was stopped for this run as only 2000 locations were allocated for the vortices. However, the number of vortices was stabilizing and should not have increased much beyond 2000.

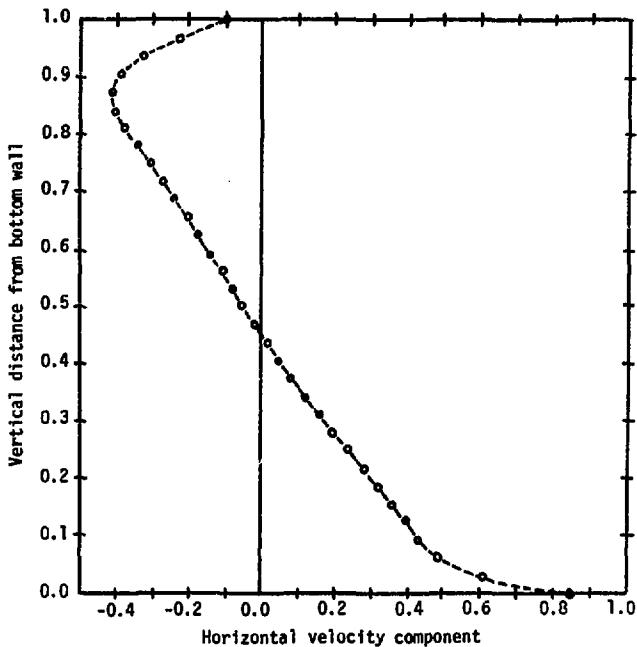


Fig. 18. Velocity profile of u-component along a vertical line through the vortex center.

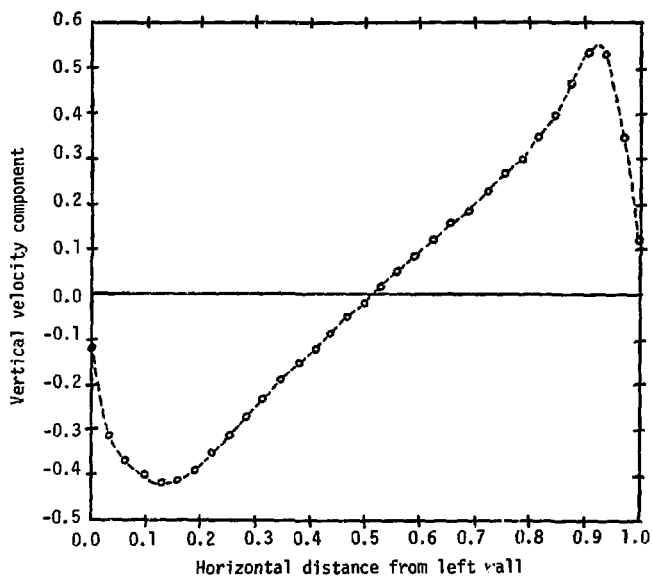


Fig. 19. Velocity profile of v-component along a horizontal line through the vortex center.

Fig. 20.

(on following page)

Velocities induced on the boundary by the field plotted in Fig. 11.

Read table by circumscribing the domain D in the positive \underline{s} direction.

Side 1 is the wall $y=0$, side 2 is $x=1$, side 3 is $y=1$, side 4 is $x=0$.

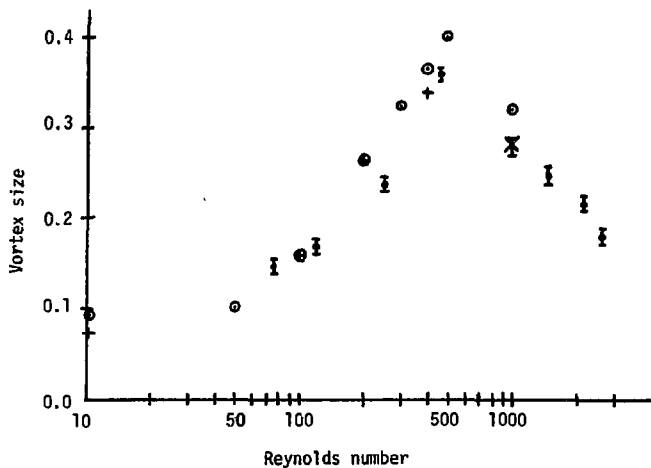
Index k in the left hand column is an increasing index along the boundary.

Along side 1, $k=1$ denotes the point $(x,y) = (0,0)$. Along side 2, $k=1$ denotes the point $(x,y) = (1,0)$.

K	SIDE 1		SIDE 2		SIDE 3		SIDE 4	
	U	V	U	V	U	V	U	V
1	0.	0.	0.	0.	0.	0.	0.	0.
2	0.410326	-0.023214	0.012996	0.255764	0.003473	0.000077	-0.000561	0.000147
3	0.452289	-0.001452	-0.001777	0.263143	0.013442	0.001047	0.000115	-0.001078
4	0.570850	-0.032727	-0.008771	0.248103	0.023087	-0.000949	0.000102	-0.009764
5	0.575703	-0.002114	-0.000624	0.256864	0.030004	0.000697	-0.001172	-0.016518
6	0.624472	-0.016317	0.005157	0.180125	0.030827	0.002781	-0.001228	-0.028831
7	0.61761	-0.006944	-0.017628	0.273952	0.019278	0.001560	-0.001807	-0.039193
8	0.677272	-0.003766	-0.000714	0.206638	0.008960	0.001302	-0.002453	-0.051448
9	0.670239	-0.002959	0.004334	0.290742	0.000995	0.002198	-0.000056	-0.071450
10	0.797408	-0.013922	-0.001893	0.162723	-0.011309	0.001702	-0.003271	-0.080880
11	0.770240	0.010163	0.012349	0.193444	-0.026873	-0.000621	-0.002921	-0.092462
12	0.776676	0.001975	0.007162	0.213963	-0.058198	0.003838	-0.007456	-0.107574
13	0.833026	-0.004533	0.008039	0.124285	-0.064574	0.001836	-0.008183	-0.092839
14	0.800366	0.007673	0.004482	0.174508	-0.088022	-0.004726	0.002548	-0.123989
15	0.879527	0.006544	0.005375	0.138339	-0.085062	0.003717	0.002048	-0.148417
16	0.847721	0.013315	0.005572	0.098977	-0.094271	-0.000957	-0.007489	-0.138374
17	0.825109	0.005127	0.002142	0.068150	-0.100764	-0.006584	-0.010473	-0.144213
18	0.875455	-0.006141	-0.003537	0.027883	-0.059561	0.002931	-0.003480	-0.131963
19	0.866425	0.008450	-0.000111	0.041181	-0.066352	0.002803	0.007211	-0.115917
20	0.879536	0.010100	0.003099	0.025099	-0.063949	0.006486	-0.009557	-0.084148
21	0.906121	-0.005907	-0.000640	-0.002596	-0.055177	-0.003247	0.003282	-0.112159
22	0.875243	-0.003487	-0.000649	-0.006989	-0.048203	0.001669	0.001349	-0.113547
23	0.826133	-0.002170	-0.012568	-0.021071	-0.015680	-0.002204	0.000918	-0.091038
24	0.841655	0.003915	-0.000207	-0.03045	-0.005666	-0.000080	-0.001626	-0.082341
25	0.844571	-0.002369	-0.004116	-0.030561	0.004554	0.001037	0.001660	-0.062350
26	0.789553	-0.002810	-0.001634	-0.025524	0.011529	0.001698	0.002494	-0.036821
27	0.783479	0.016098	0.000401	-0.017786	0.006246	0.001297	-0.000433	-0.041018
28	0.782164	0.010558	-0.002049	-0.017712	0.004847	0.000821	0.001508	-0.021527
29	0.701674	0.000918	-0.000357	-0.013682	0.004850	-0.001130	0.002843	-0.014139
30	0.771225	-0.012461	-0.000312	-0.020615	0.003820	-0.000032	0.000315	-0.043613
31	0.579872	-0.010789	0.000165	-0.013582	0.001184	-0.000217	0.005627	-0.039842
32	0.380096	-0.000812	-0.000133	-0.007157	0.000010	-0.000038	0.007510	-0.084925

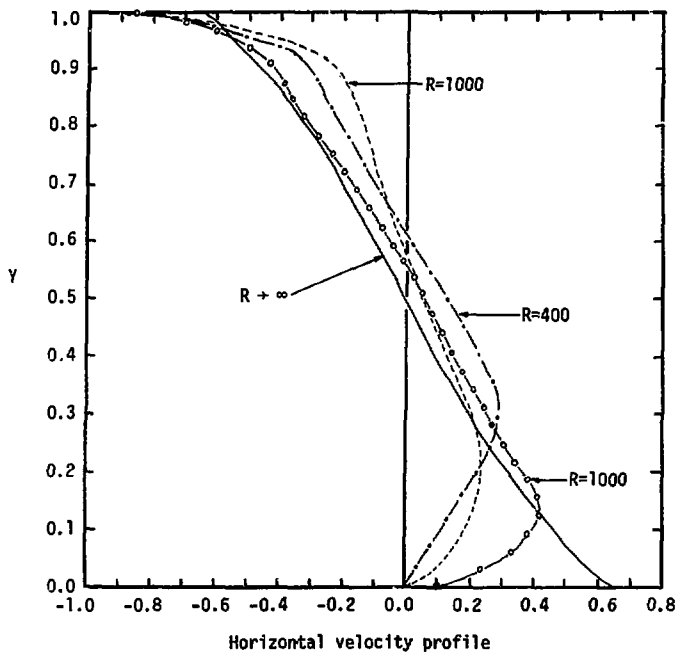
AVERAGE VELOCITIES

SIDE 1		SIDE 2		SIDE 3		SIDE 4	
UAV	VAV	UAV	VAV	UAV	VAV	UAV	VAV
0.712250	-0.002863	0.000455	0.096674	-0.022387	0.000585	-0.000695	-0.069456



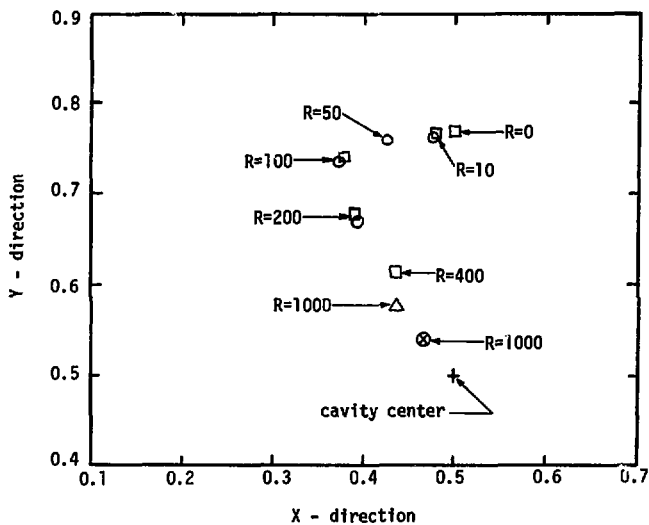
- ⌋ Experimental results (Pan and Acrivos [1])
- + Numerical results (Burgraff [5])
- o Numerical results (Bozeman and Dalton [3])
- x Numerical results (this study)

Fig. 21. Variation in size of upstream corner vortex vs. Reynolds number.



- R = 400 (Burgraff [5])
- - - R = 1000 (Bozeman and Dalton [3])
- R = 1000 (this study)
- R → ∞ (Burgraff [5])

Fig. 22. Comparison of velocity profiles of u-component along a vertical line through the vortex center.



- Burgraff [5]
- Bozeman and Dalton, CDD method [3]
- △ Bozeman and Dalton, UDD method [3]
- ⊗ this study

Fig. 23. Effect of Reynolds number on the location of the center of the primary vortex.