## Title
An Algorithm for Vortex Loop Generation

## Permalink
https://escholarship.org/uc/item/0mj7j52d

## Author
Summers, D.

## Publication Date
1991-09-01

LBL-31367
Preprint

Lawrence Berkeley Laboratory
UNIVERSITY OF CALIFORNIA

Physics Division

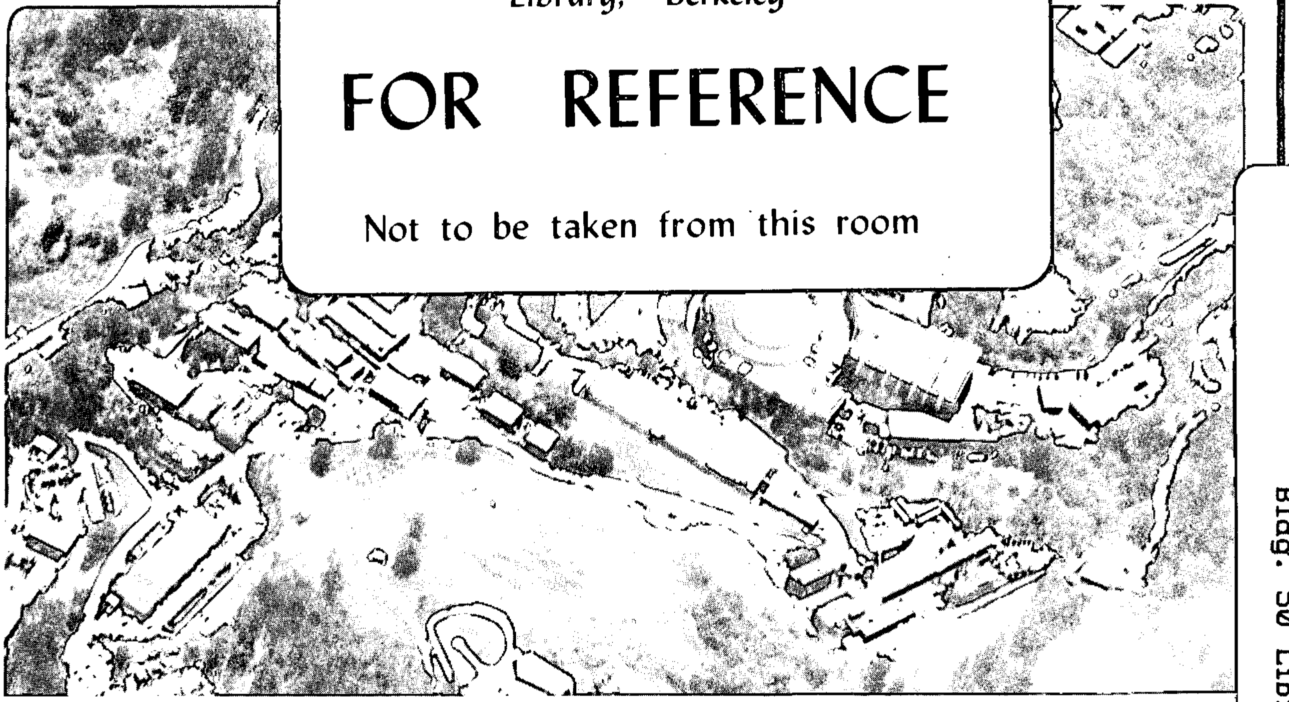
Mathematics Department

To be submitted for publication

# An Algorithm for Vortex Loop Generation

D. Summers

September 1991



Prepared for the U.S. Department of Energy under Contract Number DE-AC03-76SF00098

## DISCLAIMER

This document was prepared as an account of work sponsored by the United States Government. Neither the United States Government nor any agency thereof, nor The Regents of the University of California, nor any of 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. Reference herein to any specific commercial product, process, or service by its trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or The Regents of the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof or The Regents of the University of California and shall not be used for advertising or product endorsement purposes.

Lawrence Berkeley Laboratory is an equal opportunity employer.

## DISCLAIMER

This document was prepared as an account of work sponsored by the United States Government. While this document is believed to contain correct information, neither the United States Government nor any agency thereof, nor the Regents of the University of California, nor any of their employees, makes any warranty, express or implied, or assumes any legal 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. Reference herein to any specific commercial product, process, or service by its trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or the Regents of the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof or the Regents of the University of California.

LBL-31367

# AN ALGORITHM FOR VORTEX LOOP GENERATION[1]

David Summers[2]

Lawrence Berkeley Laboratory
and
Department of Mathematics
University of California
Berkeley, CA 94720

September 1991

[1] This work was supported in part by by the Applied Mathematical Sciences Subprogram of the Office of Energy Research, Office of Basic Energy Sciences, U.S. Department of Energy under Contract DE-AC03-76SF00098.

[2] Visiting from the Department of Mathematics, Napier Polytechnic of Edinburgh, 219 Colinton Road, Edinburgh EH14 1DJ, Scotland, UK.

# An Algorithm for Vortex Loop Generation

## Abstract

A vortex method in three dimensions to model unbounded flow using closed vortex filaments (loops) has been introduced by Chorin [9]. A vortex loop generation algorithm is developed here which relates this strategy to bounded flow. This algorithm is based on partitioning the boundary into a set of polygonal vortex loops which, in aggregate, effect the no-slip condition along a set of vorticity field-lines lying in the boundary surface. The algorithm is illustrated in application to flow over a flat plate.

## 1. Introduction

The random vortex method introduced by Chorin [4,5,6,7] and its subsequent elaboration and analysis has been extensively described in the literature (see, for example, the recent review by Puckett [14]). Briefly, the equations governing viscous incompressible fluid flow are solved numerically by discretizing the vorticity field into a finite collection of Lagrangian elements. These vortex elements interact with each other, this is to say each element is advected in the velocity field induced by the remaining elements in the collection. Viscous diffusion is represented by a stochastic model; to each element is imparted a gaussian random walk displacement.

In *two dimensions* at least, the notion of representing $\xi$, the continuum vorticity field (i.e. $\xi = \operatorname{curl} \mathbf{u}$, where $\mathbf{u}$ is the velocity field) by a distribution of interacting "particles" – specifically by a distribution of point vortices or desingularized versions such as blobs – has obvious appeal. For inviscid flow, the system of such particles is Hamiltonian. The inviscid interaction of vortices is well-understood, this being a Biot-Savart relationship. The velocity field at a point induced by a collection of vortices is the linear superposition of the fields induced by each member of the collection. The "particles" are passively transported in the resulting induced velocity field. The accuracy of this particle representation of the fluid dynamics can be estimated and its convergence to a continuum solution to Euler's equation (as the paramaterization is refined) can be analytically demonstrated at least for unbounded flow. Similarly the convergence for the case of viscous unbounded flow has been demonstrated [13]. A technology is developing for the choice of smoothing strategies; a number of core structures for blobs has now been studied. The method has found wide application.

The random vortex method provides a discretization which is naturally adaptive: the increasing structure associated with vortical flow is represented by concentrations of particles, and the method thus concentrates computational effort where the flow is most structured. If the model is suitably refined, the smaller scales of this structure should be modelled. The choice of an (idealized) point vortex (or vortex blob) as a Lagranian computational element does not imply that these elements have independent physical existence. They are in aggregate a particle discretization of a continuum field. However the vortex interaction between point vortex elements is identical to the classical understanding of the interaction between a point vortex model of a physical vortex. The point is, that the computational element is modelled after the physical species known as a line vortex.

The random vortex treatment of boundaries in two dimensions has been the focus of some effort. Such boundaries are of obvious importance to the study of viscous flow since an implication of the no-slip condition ($\mathbf{u} = 0$) at such a surface is the development of sheared flow and hence the creation of vorticity. The geometry of flow near such a surface suggests particular simplifications to the flow equations can be made in the neighbourhood of the surface; this is the boundary layer approximation. This approximation has been embedded in the formulation of the random vortex method, the blob computational element is expressed as a partitioned vortex sheet [5]. A rational strategy for transforming a sheet element into a corresponding blob element as the element moves from a neighbourhood of the solid surface into the flow interior, has been developed. There are a number of studies of the vortex sheet method in the literature (see [14]).

The random vortex method has been generalized to *three dimensions* in a variety of ways. A number of smoothing kernels (i.e. blob structures) have been proposed. One such generalization is the spherically symmetric blob developed by Beale and Majda [3]. Chorin [5] has introduced a filament consisting of a contiguous collection of vortex "segments", cylinders with uniform core radius. In some subsequent applications, the condition of connectivity of these segments has been relaxed, so increasing their resemblence to the blob representation. The convergence properties of the Beale-Majda blob [3] and the filament method (see Greengard [10]), have been explored for unbounded flow.

Whereas inviscid vortex dynamics in two dimensions has mathematically attractive properties – in particular a simple Hamiltonian property – in three dimensions the phenomenon of vortex stretching complicates the picture. The elements are subject to stretching consistent with the non-linear terms in the three-dimensional Euler equation. The complication this presents is considerable, because the stretching is typically accompanied by folding on smaller scales (to preserve overall vorticity) and, therefore, can induce in adaptive numerical schemes an increasing refinement in the discretization.

If the computational elements are to be "modelled after" macroscopic, physical, vortices, then one might expect such vortex elements to be consistent with the condition $\nabla \cdot \xi = 0$ which has the implication that the vorticity field is solenoidal, the vortex lines must form closed loops in an interior flow. Such a condition is satisfied (by construction) by using closed filament loops as elements; on the other hand blobs and disjointed segments are not constrained to satisfy the divergence-free condition, although they may well provide a field which is in some measurable sense approximately solenoidal.

The question of what criteria lead to a satisfactory particle discretization of a three-dimensional continuum vorticity field can be approached from an alternative direction. One can study the stationary statistical properties of an ensemble of interacting vortex elements, and try to determine which discretization re-creates the expected statistics of turbulent fluids [8]. Numerical experiment seems to suggest that the representation which is most successful in this respect is that of a system of closed vortex filaments with uniform core diameters.

## 2. Creating Divergence-free Vorticity

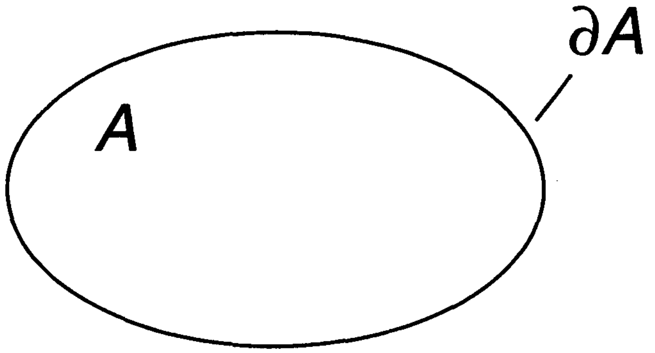
The original notion proposed by Chorin [4] for imposing the no-slip condition at a solid surface proceeds as follows: one conceived (in two dimensions) of an obstacle over which there is some flow. We assume at $t < 0$ there to be a free-slip condition at the surface. Impulsively at $t = 0$, at a discrete set of points $\{x_i\}$ on the obstacle surface, a vortex blob would be created such that the tangential velocity field it creates would exactly cancel this free-slip field. This blob would then diffuse into the flow interior in the subsequent time step, establishing again a non-zero velocity field at the set of points. The procedure can then be repeated for the next time-step. The sequential process of introducing a vortex element at each time step in this way is called the vorticity creation algorithm. Some authors have continued to use this strategy in recent work (see [1]).

This original idea was elaborated [5] to construct at a solid surface an explicit model of two-dimensional boundary layer dynamics. The no-slip condition was established by the creation of a "vortex sheet" which was then subdivided into sheet segments, and these participated in the flow according to a Lagrangian formulation of the Prandtl boundary layer equations.

The extension of the vortex sheet method to three dimensions is described in Chorin [6]. In this case a two-dimensional vortex sheet is subdivided into rectangular tiles, each carrying vorticity (a vector lying in the plane of the sheet) which exactly establishes no-slip at a point at the centre of the tile.

The preceding strategy in three dimensions is in itself an adequate algorithm for the pointwise establishing of the no-slip condition at a collection of points $\{\mathbf{r}_i\}$ on a solid surface. The expression "pointwise" is emphasized to indicate that the sheets or tiles that are created to establish no-slip are created independently of each other in the sense that no attempt is made to ensure the vorticity created satisfies $\nabla \cdot \xi = 0$, i.e. that the vortices form closed contiguous loops lying in the plane of the solid surface. If one wishes to use in the flow interior closed loops as computational elements, a strategy should be found to establish the no-slip condition in terms of precisely such closed loops.

To do this one may return to Chorin's original [4] strategy and seek to generalize this over a three-dimensional surface. We consider a general flow over a three-dimensional obstacle, $A$, bounded by closed surface $\partial A$



We consider the obstacle to be impermeable, and that there is a "free-slip" flow which is wholly tangential to the surface at time $t$. At a point on the surface ($\mathbf{r} \in \partial A$) there is a tangential velocity (which is not necessarily irrotational), $\mathbf{u}_s$. We wish to introduce a vortex (in fact a vortex sheet element) at this point whose vorticity will be directed in $\partial A$ perpendicular to $\mathbf{u}_s$, and which will induce a velocity which, when added to $\mathbf{u}$ will lead to a vanishing resultant. It is a simple matter to determine this vector $\xi = (\xi_{\mathrm{x}}, \xi_{\mathrm{y}})$, where $(x, y)$ are local coordinates lying in $\partial A$.

If $\mathbf{u}_s = (u_x, u_y)$, we can infer $\xi = (-u_y, u_x)\, h$ (where $h$ is a linear dimension of the sheet perpendicular to $\xi$). This $\xi$ will have the desired properties of being orthogonal to $\mathbf{u}_s$ and of the appropriate magnitude to induce a cancellation field.

Having determined (at, say, a regular mesh of points on the surface) the vector components of vorticity, we can exploit the divergence-free property of $\xi$ and the fact that $\xi$ is strictly tangent to $\partial A$ to construct a vector potential function $\Gamma\mathbf{n}$ such that

$$\xi = \text{curl } \Gamma\mathbf{n}$$

$\mathbf{n}$ is the unit outward normal to $\partial A$. The function $\Gamma\,\mathbf{n}$ is essentially a Hertz potential (see Section 1.10 of [12] for the development of this idea in an electromagnetic context) whose magnitude can be constructed from the line integration of the form:

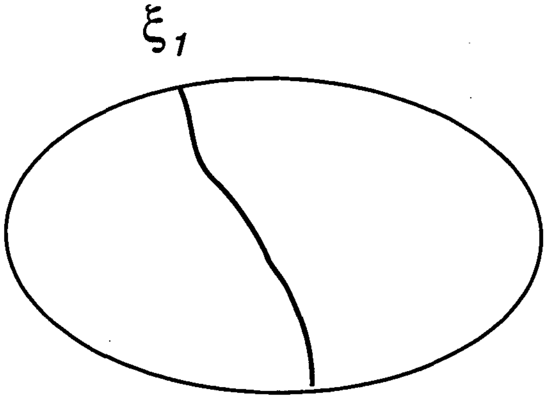
$$\begin{aligned}\Gamma(x,y) &= \int_o^y \xi_x dy - \int_o^x \xi_y dx \\ &= -h \int_{\mathbf{r}\in\partial A} \mathbf{u}\cdot d\mathbf{r} \qquad (1)\end{aligned}$$

The level curves of $\Gamma(x,y)$ on $\partial A$ constitute the lines of equivorticity on the surface. The fact that the constructed field $\xi$ satisfies $\nabla\cdot\xi = 0$ on $\partial A$ implies the level curves of $\Gamma$ form closed loops on $\partial A$. One can interpret a level curve of $\Gamma$ as being the location of a vortex filament (of uniform stength) required to exactly establish the no-slip condition.

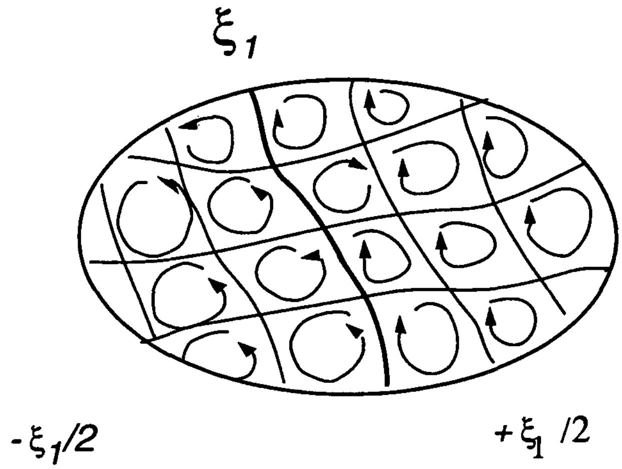
One can imagine how the creation of such a closed vortex loop along the surface of an obstacle can be embedded into a problem of evolving flow: the loop could be made to advect and diffuse in subsequent time steps. Conceivably an algorithm can be based on such a model; however such large computational structures raise various problems, such as how to ensure that diffusion near the surface (in particular at the first time step) does not effect any stretching of the element. It is not clear how a complete covering of the surface can be effected (i.e. how a set of level curves can lead to an effective partitioning of the surface) or how the notion of a vortex sheet can be incorporated into such a strategy.

## 3. Constructing a partition which conforms to vorticity level curves

Consider a single level curve, a closed loop of equivorticity with vortex strength, $\xi_1$ lying in $\partial A$. (For purposes of introducing the basic concept, we consider the line of equivorticy to be simply connected).

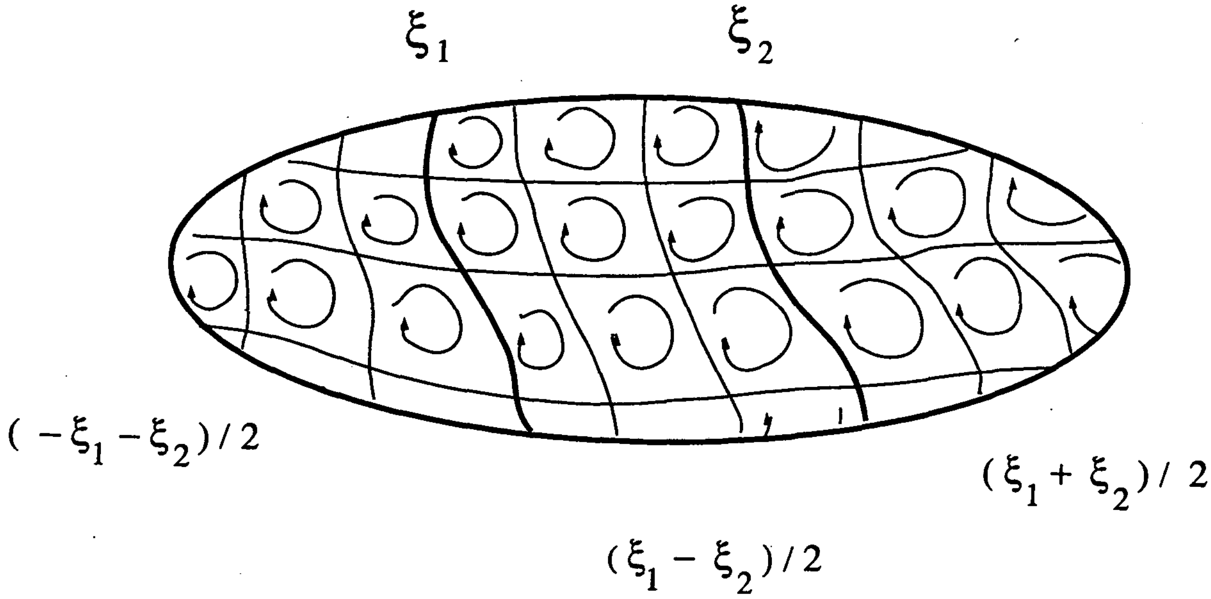


We can generate from this a partition of the entire surface of $\partial A$ We partition the surface to the left of the level curve into, say, quadrilaterals (later to be generalized to polygons) and consider each quadrilateral to coincide with a vortex loop of strength $-\xi_1/2$; similarly we partition the surface to the right of the level curve, and consider each quadrilateral to coincide with a vortex loop of strength $+\xi_1/2$. The situation is depicted as



(the sign of the vortex loop is indicated here by the direction of the arrow). With the exception of the level curve itself, along all the partition lines the neighbouring loops mutually cancel (vortex lines of equal strength and opposite sense coincide). However along the level curve the adjacent loops sum effectively to $\xi_1$.

Such a partitioning can be repeated for any number of level curves, and the results can be linearly superposed. For example if we consider two level curves associated with strengths $\xi_1$ and $\xi_2$, we could effect a partition of the form:



Again the effect of adjacent loops is to cancel except along the level curves themselves, where the adjacent loops sum to the values $\xi_1$ and $\xi_2$ respectively.

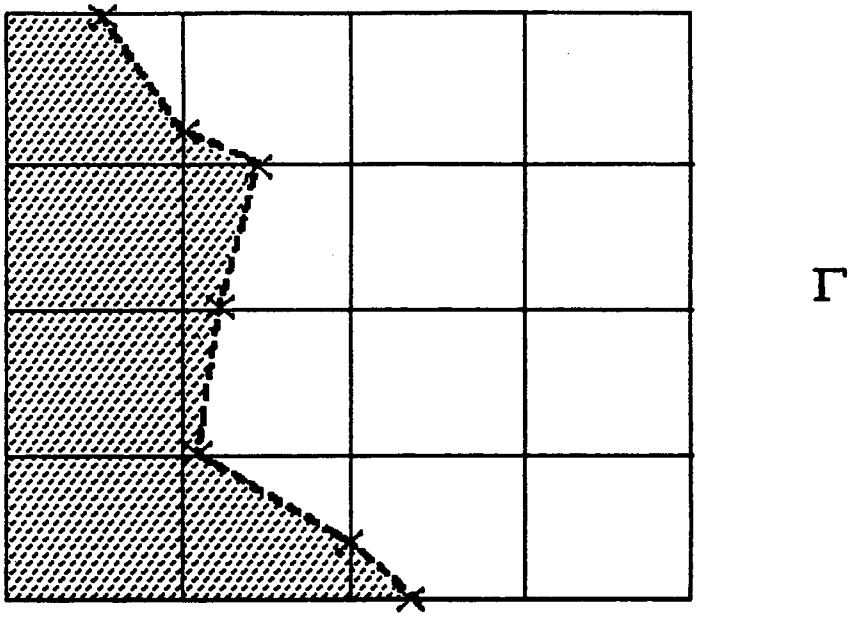
At this stage we have articulated an algorithm for vortex loop generation to establish a no-slip surface at each time-step of a flow evolution calculation. Two obvious refinement strategies suggest themselves: (1) the quadrilateral partition can be refined; (2) the number of level curves can be increased (and the increment between them reduced). The latter refinement would result, for $N$ specified level curves $\{\xi_i\}$, quadrilaterals (or more generally polygons) in $N+1$ subdomains of the surface, the vortex strength of the loops in the $m$th such subdomain being given by

$$-\frac{1}{2}\sum_{i=1}^{N}\xi_i + \sum_{i=1}^{m-1}\xi_i \tag{2}$$

It might be noted here how this distribution of vortex strengths differs from that resulting from the usual blob/sheet method. In the present method the greater strength of the created elements (and hence, ultimately, the greater density of computational vortex elements) occurs in the neighbourhood of stagnation points. In the usual blob/sheet method the greater strength of created elements occurs where the interior velocity is greater — typically away from stagnation.

Having established a system of loops lying in $\partial A$, these can be made to interact and otherwise participate in the flow. However, given the proximity of the adjacent loops in $\partial A$, one might expect instability to arise from the singularity in the Biot-Savart relationship. Even invoking a smoothing kernel (specifically a vortex filament of finite core size) the density of the elements in the neighbourhood of $\partial A$ suggests that Chorin's vortex sheet model of boundary layer flow could be introduced to model this region. Before discussing how the vortex sheet algorithm could be embedded in the present algorithm, we should discuss the question of how to partition a surface in a manner which conforms to, say, a system of highly structured, geometrically complicated level curves. The location and strength of the vorticity level curves derives from the surface distribution of "free-slip" field upon completion of the diffusion and advection of vortex elements in the preceding time-step. The point here is, this field is strongly determined by the inadequacy of the discretization to establish no-slip over a given time-step, and this may be conceivably highly dependent on the numerics of the random vortex method algorithm. Ideally one would like a robust strategy for partitioning a surface in anticipation of the complex surface field which can arise in a given time step.

One way to approach this problem is to consider an underlying fixed regular mesh. We assume that the "slip-field" on $\partial A$ has been calculated, and from this the scalar function $\Gamma$ has been determined at the nodes of this underlying grid. For example, consider $\partial A$ to be a flat surface, and the underlying grid to be rectangular:



We could select a particular level curve of Γ and determine using, for example, bilinear interpolation, the points of intersection of this curve with the underlying mesh. (For the illustrated example, the level set indicated by 'x'). Joining these points determines a set of polygons each of which we can consider to coincide with a closed-loop vortex filament. We can assign strengths to the loops either side of the level curve in the way described previously. (The shaded region in the diagram would correspond to one resulting subdomain; the unshaded region to another.) Furthermore we can proceed to determine a set of level curves, and superpose the resulting strengths in a manner very similar to that indicated in the previous discussion.

The discretization can be refined by refining the underlying mesh. Before discussing futher questions of refinement and convergence we discuss how a sheet representation might be introduced.

## 4. Sheet Representation of Surface Loops

We have proposed a rational criteria for creating a system of adjacent polygonal vortex loops on the surface $\partial A$ which will effect "no-slip" along pre-determined equivorticity lines. In the neighbourhood of the surface we would like to adopt the boundary layer approximation in order to represent correctly the symmetry condition at the wall which vorticity exhibits. This is to say the Prandtl equations are to be satisfied. Consider coordinates $(x, y)$ to be in $\partial A$.

$$\frac{\partial \xi_x}{\partial t} + (\mathbf{u} \cdot \nabla)\xi_x = \nu \frac{\partial^2 \xi_x}{\partial z^2} + \text{stretching terms}$$

$$\frac{\partial \xi_y}{\partial t} + (\mathbf{u} \cdot \nabla)\xi_y = \nu \frac{\partial^2 \xi_y}{\partial z^2} + \text{stretching terms}$$

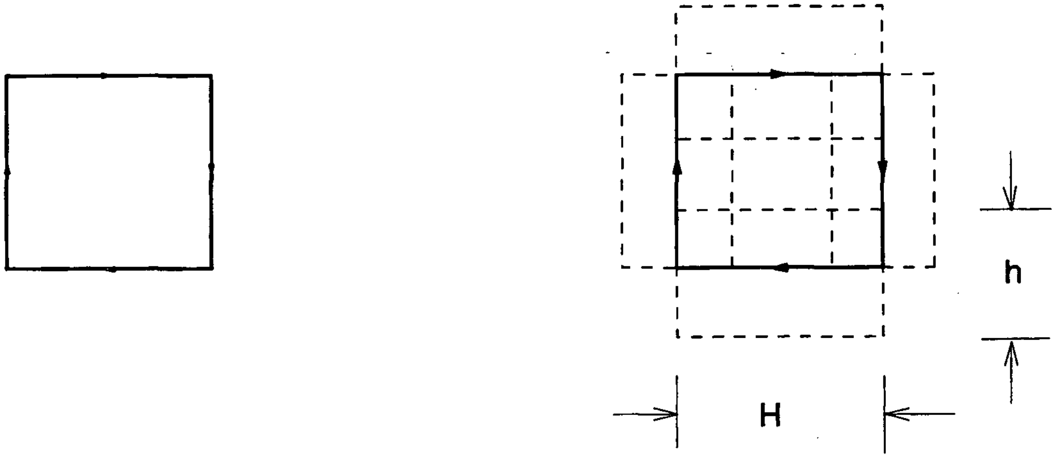
$$\xi_x = \frac{\partial u_y}{\partial z}$$

$$\xi_y = -\frac{\partial u_x}{\partial z}$$

$$\nabla \cdot \mathbf{u} = 0$$

Futhermore, we would like to represent the flow dynamics in the interior (i.e. away from the wall) by a collection of vortex loops of uniform strength.

To see how we might approach this problem, consider the case of a rectangular loop with vortex strength $\xi$. We could understand this as a 'cluster' of four tiles:



Each tile carries a two-dimensional vortex $(\xi_x, \xi_y)$. The four tiles in the cluster will have strengths per unit length, starting from the right hand side and going clockwise around the loop: $(0, -\xi/h)$, $(-\xi/h, 0)$, $(0, \xi/h)$, and $(\xi/h, 0)$. $h$ is the width of the tile in a direction perpendicular to the sense of $\xi$; $H$ is the length of the tile in the direction of the vorticity vector. As $h \rightarrow 0$ we retrieve a vortex line filament representation of the loop. We understand $\xi$ to be the velocity discontinuity associated with the vortex sheet, and $\xi/h$ is the vorticity per unit length of the sheet element, or vortex sheet strength. (From previous discussion we can see that $\xi/h = (u_x, u_y)$.) We note that in general $h \neq H$. In contrast with the usual blob/sheet random vortex method, the tiles overlap at the point of inception. We consider the clusters to move with a common random walk displacement (to avoid stretching induced solely by diffusion). Otherwise the tiles in the cluster interact independently in the sheet layer. The interaction associated with such tile elements has been described by Chorin [6]. If we consider the case $h = H$, a collection of tiles whose centres are located at $(x_j, y_j, z_j)$ will induce a velocity $\mathbf{u} = (u_x, u_y, u_z)$ at a point $(x_i, y_i, z_i)$ in the sheet layer given by

$$u_x = U_x(x_i, y_i) + \int_{z_i}^{\infty} \xi_y \, dz$$

$$u_y = U_y(x_i, y_i) - \int_{z_i}^{\infty} \xi_x \, dz$$

$$u_z = -\frac{\partial}{\partial x}\int_{z_i}^{\infty} u_x \, dz - \frac{\partial}{\partial y}\int_{z_i}^{\infty} u_y \, dz$$

where the velocity $\mathbf{U} = (U_x, U_y)$ represents the flow field associated with the interior. A discrete approximation for the horizontal components $(u_x, u_y)$ at $(x_i, y_i, z_i)$ is

$$\begin{aligned} u_x^{(i)} &= U_x^{(i)} + \sum_j \xi_y^{(j)} \, d^{(j)} \\ u_y^{(i)} &= U_y^{(i)} - \sum_j \xi_x^{(j)} \, d^{(j)} \end{aligned} \qquad (3)$$

with only those sheets included in the summation — located at $(x_j, y_j, z_j)$ — such that $0 \leq d^{(j)} \leq 1$, $z_j \geq z_i$, and the smoothing (or cutoff) function given by

$$d^{(j)} = \{(1 - \frac{1}{h}|x_i - x_j|)(1 - \frac{1}{h}|y_i - y_j|)\}$$

The vertical component, $u_z$ at $(x_i, y_i, z_i)$ can be determined by applying centre-difference approximation for derivatives:

$$u_z = -(I_+^{(x)} - I_-^{(x)})\frac{1}{h} - (I_+^{(y)} - I_-^{(y)})\frac{1}{h}$$

with

$$I_\pm^{(x)} = U_x^{(i)} - \textstyle\sum_\pm \xi_x^{(j)}\, d_\pm^{(j)}\, z_j^*$$

$$I_\pm^{(y)} = U_y^{(i)} - \textstyle\sum_\pm \xi_y^{(j)}\, d_\pm^{(j)}\, z_j^*$$

$$z_j^* = \min(z_i, z_j)$$

$$d_\pm^{(j)} = \{(1 - |x_i \pm \frac{h}{2} - x_j|\frac{1}{h})(1 - |y_i \pm \frac{h}{2} - y_j|\frac{1}{h})\}$$

The tiles located at $(x_j, y_j)$ are included in summation $\sum_-$ if $d_-^{(j)} \leq 1$; similarly $\sum_+$ sums over those elements at $(x_j, y_j)$ such that $d_+^{(j)} \leq 1$. Essentially sheets which do not overlap do not interact with each other. The smoothing functions $d_\pm^{(i)}$ or $d_\pm^{(j)}$ are the product of two linear tent functions: an implication of this representation is that the magnitude of the interaction between two sheets is related to their proportionate area of overlap. In fact, the preceding equations require to be modified slightly if one is computing the velocity of one tile induced by the remaining collection. We require, for example,

$$u_x^{(i)} = U_x^{(i)} + \frac{1}{2}\xi_y^{(i)} + \sum_j \xi_y^{(j)}\, d^{(j)}$$

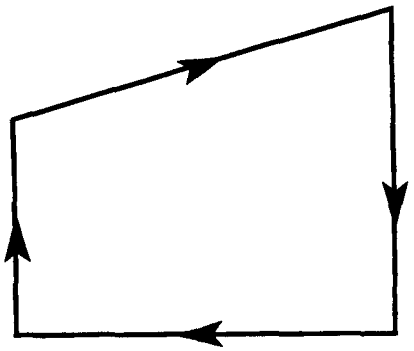
Since we require generally $h \neq H$, we indicate how the preceding discrete approximation must be modified to incorporate tiles interacting which have unequal dimensions. If we consider the dimensions of the $j$th tile to be $(h_x^{(j)}, h_y^{(j)})$, then the smoothing function above must become

$$d^{(j)} = \{(1 - \frac{1}{h_x^{(j)}}|x_i - x_j|)(1 - \frac{1}{h_y^{(j)}}|y_i - y_j|)\}$$

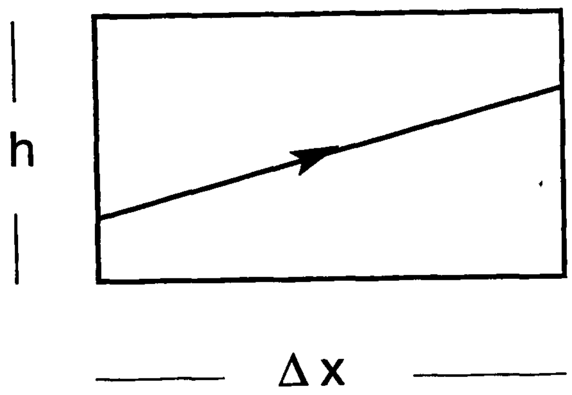
Also the smoothing function associated with the vertical component becomes

$$d_\pm^{(j)} = (1 - \frac{1}{h_x^{(j)}}|x_i - x_j \pm \frac{1}{4}(h_x^{(i)} + h_x^{(j)})|)(1 - \frac{1}{h_y^{(j)}}|y_i - y_j \pm \frac{1}{4}(h_y^{(i)} + h_y^{(j)})|)$$

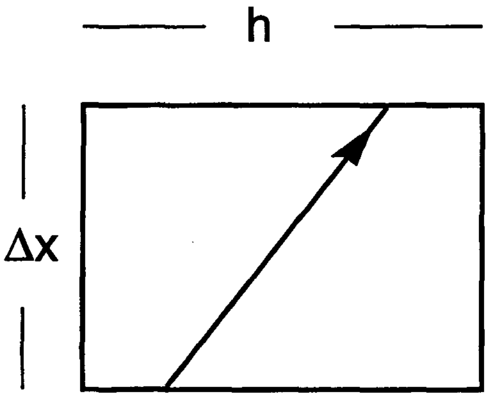
The more general case of a polygonal loop can be construed as a cluster of vortex tiles by adopting a similar rationale. For example, consider the quadrilateral



For the three rectilinear sides the tiles can be constructed as before. The case of the non-rectilinear side suggests that a convention be adopted, namely if the angle the vorticity vector $\xi$ makes with the horizontal axis is less than or equal to 45 degrees, the "width" of the tile is taken as in the vertical direction, and is the parameter $h$ which is numerically chosen and is common to all loops. The "length" of the tile is taken as the horizontal distance been base and tip of the vector, $\Delta x$. In the present illustration the angle is less than 45 degrees, so a tile is constructed:

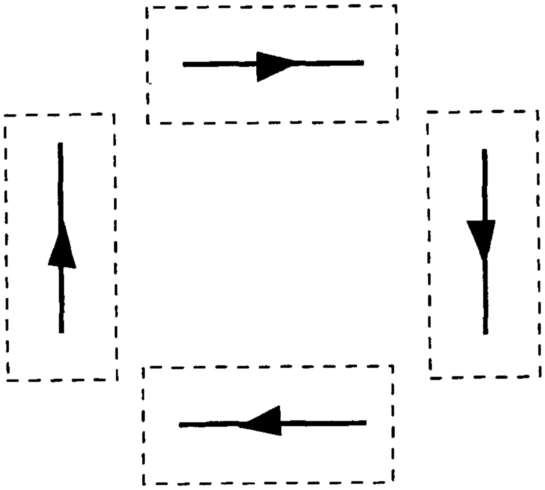


If the angle of the vorticity vector is greater than 45 degrees, the roles of $h$ and $\Delta x$ are reversed:

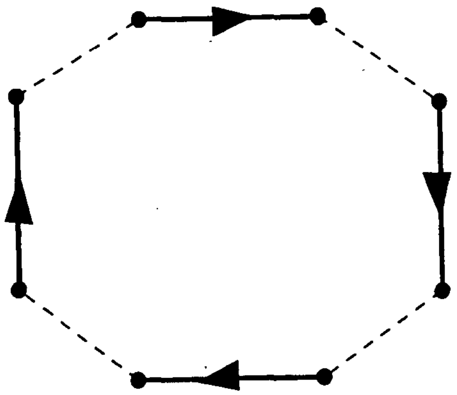


It is important to adopt some such convention: newly created tiles centred on non-rectilinear vortex vectors, but associated with two adjacent loops of differing vortex strength, must exactly overlap with each other. This implies a width factor $h$ which is independent of individual loop geometry.

When the tile clusters eventually leave the "sheet layer" they are made to regroup into a closed filament. For example if such a cluster upon leaving the layer (a distance typically several standard deviations thick from the surface) they might have the form (for example, after the individual tiles have advected in a non-uniform velocity field)



These would be re-connected into a loop of the form:



with strength $\xi$. The loop is characterised by this strength, and by its sign, and by the coordinates of the base and tip of each segment in the loop, and by a link number which points to the preceding segment in the loop. Each loop is assigned an integer label as well.

The model described differs in certain respects from that described by Chorin [6]. In the previous work, the notion of a partitioned vortex sheet was being invoked; furthermore no attempt was made to model vortex stretching. In the present model, more complex sheet structures (namely vortex clusters) are being introduced. Each cluster has, at point of inception, overlapping sheets, so the clusters can be self-interacting. There is the possibility in such a model to represent stretching of vorticity in a manner somewhat analogous to that exploited by Chorin [6] to model vortex filament stretching in the flow interior.

## 5. Matching Boundary Flow to That of the Interior

It is envisaged that the tiles created at $\partial A$ will enter the flow interior, rejoin themselves into polygonal loops, and subsequently interact with each other as closed loops of uniform core, all having uniform strength. Each will induce a field at a point r according to the Biot-Savart law:

$$\mathbf{u}(\mathbf{r}) = -\frac{1}{4\pi}\int \frac{\mathbf{a}\times\xi}{a^3}\,d\mathbf{r}'$$

with $\mathbf{a} = \mathbf{r} - \mathbf{r}'$, $a = |\mathbf{a}|$, where the integration is taken over the loop. If the vorticity field is a sum of $N$ such closed vortex filaments all of uniform strength $\xi$, then

$$\mathbf{u}(\mathbf{r}) = -\frac{\xi}{4\pi}\sum_{i\neq 1}^{N}\int \frac{\mathbf{a}\times\mathbf{s}}{a^3}\,ds$$

where $\mathbf{s} = \mathbf{s}(\mathbf{r}')$ is the unit tangent vector to the $i$th line, $s$ is the arc length along the $i$th line.

As the tiles leave the sheet-layer they become rejoined into polygonal loops, i.e. into loops which inevitably have acute angled corners. The flow evolution of discontinuities in vortex filaments leads to the development of "hairpins", elongating, tightly wound curling structures. These have significant effect on the local flow at smaller scales, but as one moves to larger scales these structures become less significant individually. This suggests heuristically that a strategy can be adopted to remove such hairpins, as they develop, from the numerical Lagrangian elements we have created. In fact physical observation indicates a limit to the fractal dimension of evolving vortex filaments which may be exceeded by computational vortex elements. To achieve this limitation numerically, hairpin removal strategies can be invoked. A more formal development of this idea is linked to renormalization theory (see Chorin [9]).

This will allow us to avoid the situation where we are investing computational effort into numerical artifices which do not significantly influence the flow calculation.

We wish all the loops in the interior to have the same strength, this to permit us to embed Chorin's algorithms which invoke renormalization strategies to simplify the description of the flow (specifically to allow hairpin removal, loop re-connection and de-connection, etc.). These algorithms require uniform vortex strength of each loop; the strength of the continuum vorticity field is thus to be represented by the density of such particles.

We wish to consider the loop filaments to have compact support; if we use some smoothing kernel to effect this, we shall require a rationale for determining the core diameter.

We commence the specification of these numerical parameters by nominating an interior loop vortex strength of $\xi$ (which may be positive or negative depending upon the 'sense' of the loop as viewed from, say, the origin). This is to say, we wish to represent the flow by a collection of vortex loops of uniform strength, $\xi$. In the algorithm described previously we have determined a covering of polygonal surface loops on $\partial A$ of strength $\kappa$, and these are then replaced by an integer $M$ loops each of strength $\xi$ such that $M\xi = \kappa$. Each such component loop of strength $\xi$ is then converted to a cluster of tiles, as previously described, these have vortex strength per unit length of $\xi/h$. When these tiles enter the interior, they are reformed into a loop, with strength $\xi$. There are further parameters to be chosen: the underlying uniform mesh dimensions $\Delta x$ and $\Delta y$; and the parameter $h$, the tile 'width' which should satisfy the condition $h \leq \min(\Delta x, \Delta y)$. The number of level curves and the incremental difference between the values of $\Gamma$ associated with each curve, must be chosen. Once $h$ is determined the core size, $\sigma$, of the interior loops is determined, since $\sigma = h/\pi$. (This choice ensures an interior loop segment located at the sheet-layer/interior interface will exert the same influence on the wall directly below it, as a sheet at this location would do.)

The flow dynamics in the interior has been described elsewhere [9]. The loops are made to advance their position during one time step using a fourth-order Runge-Kutta integration. The loops are subjected to a removal of sharp corners, to a re-connection if the loop structure is sufficiently "pinched" and to de-connection if segments (with opposite sense) belonging to two loops sufficiently coincide. Either using vortex images or using numerical integration, the impermeability condition at the wall must be established in the field induced by interior loops.

The interaction of interior loops, if performed directly, is an $O(N^2)$ operation; each segment must interact with the $N-1$ remaining segments. Fast solver strategies can reduce this computational effort to $O(N)$, for example the Greengard Rokhlin method [11] or the method of Anderson [2].

## 6. On the Numerical Determination of $\Gamma$

The partition of the surface which has been described in Section 3 depends upon an evaluation of the scalar function $\Gamma(\mathbf{r} \in \partial A)$, which in turn is determined (to within arbitrary additive irrotational functions) by the numerical integration of equation (1). The line integrations can be performed to a level of refinement greater than that of the underlying mesh: we can evaluate $\mathbf{u}_s$ at increments less than $h$, say.

This field, $\mathbf{u}_s$, is the "departure from no-slip" at $\partial A$ arising from the particle kinematics of the previous time-step. This field may be expected to reflect the numerics of the method: to be explicit, this field is a numerical residual, and need not be particularly simple geometrically. The act of numerical line-integration introduces, of course, some smoothing of this field.

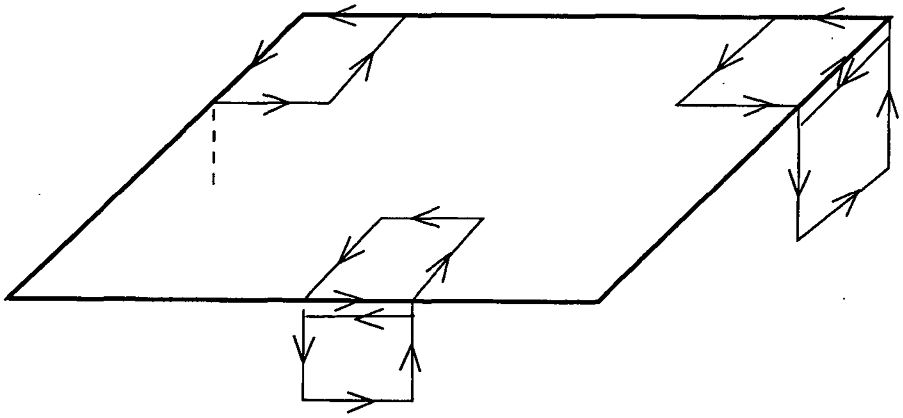
The physical significance of the level curves is reflected in the fact that the spatial gradients of $\Gamma$ are proportional to the local velocity field. Thus relatively large slip fields sould be associated with a greater density of level curves. This suggests the partitioning of $\partial A$ in this way has a naturally adaptive character, where the field $\mathbf{u}_s$ is highly structured, the mesh locally becomes more refined.

Obviously, once one has established level curves (or a partitioning consistent with these curves) one should expect the "departure from no-slip" velocity field to point normally to these curves. Given the character of the discretization of the line integration one would expect this condition of normal flow to be satisfied at best in approximation. Once the $N$ level curves, $\Gamma(\mathbf{r} \in \partial A) = c_i, i = 1, \cdots, N$, have been determined, one needs to evaluate the velocity $\mathbf{u}_i$ (including, importantly, the sign of $\mathbf{u}$ with respect the level curve) at least at one point on each of these curves (the velocity should be of equal magnitude along a given curve – this providing, incidentally, another test for validation of the numerics). This evaluation can be performed by using equations (3).

## 7. A Test-bed Example: A Finite Flat Plate

To illustrate numerically the algorithm we have described, we consider the case of flow (initially uniform with unit magnitude in the $x$ direction) over a square flat plate lying in the $x-y$ plane, and of dimensions $1 \times 1$ units. The $z$-direction points vertically upward from the plate into the interior flow which occupies the half-space $z > 0$.

The algorithm assumes full meaning on a closed surface $\partial A$ in three-dimensional space, thus the present illustration (chosen because of its obvious simplicity of metric) does introduce edge boundaries which need to be rationally modelled. If we conceive of the plate as lying in the $z = 0$ plane in $R^3$, we can consider the underlying mesh (and the subsequent repartitioning) to "wrap aournd" each edge, continuing on the underside of the plate. We can consider the flow above and below the plate to be symmetric, and that the plate itself shields the upper sheet dynamics from that below. We thus consider the upper surface alone: the surface loops along the edge of the plate are considered to be adjacent to identical loops (conceptually on the underside) which result in a cancellation of vorticity (at inception) of tiles centered along the plate edges.



The edge tiles regain their vorticity as they rejoin into loops upon leaving the sheet layer (the notion being that they, at this stage, rip themselves away from their underside counterparts). Once the tiles leave the sheet layer, and become loops, the impermeability condition at the plate is effected by introducing image loops.

The following steps are performed during a single time-step of the algorithm (subroutines refer to program lop6cray.f):

**Step 1** (subroutine mesher)

An underlying 8 × 8 mesh is constructed on the plate. This in turn is subdivided into a 31 × 31 mesh, at each node of which the tangential velocity field is calculated. This is integrated using trapezoidal rule to determine

$\Gamma(x, y)$ at each node of this fine inner mesh. (The integration is taken first along the bottom edge of the plate in the $x$ direction; using these as initial values for a set of 31 integrations along the $y$-directions.) The values of $\Gamma(x, y)$ on the $8 \times 8$ mesh points are stored. The maximum and minimum of this latter field is determined. The incremental values of the (*nlevs*) level curves are calculated from $(\Gamma_{max} - \Gamma_{min})/nlevs$

**Step 2** (subroutine repart)

¿From the $\Gamma(x_i, y_j)$ and previously determined values of the level curves, a repartitioning of the underlying mesh is constructed. Each polygon loop is identified by the coordinates of its vertices (xvert,yvert), the number of vertices (iahd), and the strength of the loop (this indicated by an integer, the *ik*th level). Also the strength (and sign) of the vorticity associated with each level is determined by taking a point on each of the level curves of $\Gamma$ and evaluating there $\mathbf{u}_s$. This is effectively the $\xi/h$ described in Section 2.

**Step 3** (subroutine create)

Having repartitioned the surface, the polygon loops are then superposed according to the equation (2); the resulting distribution of surface loops is then decomposed into a summation over loops of uniform, pre-assigned, strength. Each of these loops in turn is converted into a cluster of tiles. Each cluster is assigned a label. Each side of the polygon forms an individual tile characterized by its center, by its (common pre-assigned) width $h$, by its (individually determined) length; by its strength (a two-component vector), by the sign (or sense) of the originating polygon loop, and the number of tiles in a given cluster. Edge boundary values of this strength are assigned as discussed earlier. These tiles and clusters are attached to the existing list of tiles and clusters.

**Step 4** (subroutine getdt)

The incremental time-step is determined so that $h/u_{max} < dt$, essentially a CFL condition, where $h$ is the tile width, $u_{max}$ is the maximum velocity of flow at the sheet layer/ interior interface.

**Step 5** (subroutine stept)

The tile locations are advanced according to the algorithm described previously (and in Chorin [6]). Euler stepping is used for advection. Gaussian random walk in the $z$-direction models diffusion. The external flow consists of the initial free stream, plus the field induced by existant vortex loops in the interior.

**Step 6** (subroutine stepf)

The positions of loop elements in the interior are advanced according to the algorithm described in Chorin [6]. Fourth order Runge-Kutta is used for advection. The interaction formulae have been modified to incorporate image loops to effect impermeability.

**Step 7** (subroutine convert)

Tile clusters which have crossed the sheet-layer/ interior interface are rejoined into loops: each loop is characterized by an integer label, by coordinate positions of the two ends of each segment in the loop; by a link list which identifies the predecessor of each segment in the loop (at point of inception, "looking back" in an anticlockwise sense), the loop strength (preassigned) and the sense (or sign) of the vorticity of the loop at point of inception.

**Step 8** (subroutines delete, divide, deconn, reconn)

The loops are subjected to Chorin's [9] renormalization strategy for modelling loop-loop interaction dynamics. Specifically the possibility for loop reconnection and deconnection is incorporated into the kinematics. Sharp corners (or potential hairpins) in the loop structure are removed. These program contain various adjustible parameters.

**Step 9** (subroutine cleanup)

The tile clusters which have left the sheet layer to become loops (or alternatively deep into the wall to be annihilated) are erased from the list of such clusters; the labelling is re-ordered accordingly.

**Return to Step 1**

To illustrate the algorithm in the early stages of time evolution, a flat plate has been subjected to incident flow of unit magnitude. An underlying $6 \times 6$ mesh is constructed. The number of level curves to be considered has been chosen to be 5. (At the first step these curves coincide with various grid lines of the underlying mesh.) 78 clusters are then created (consisting of 778 sheets) as described previously, and are then made to diffuse during the first time step.

At the beginning of time step 2 the level curves are again computed. The resulting partition is illustrated in Fig. 1a. The polygons are quadrilaterals in this case. 714 sheets are created during this step. Also sheets emerge from the sheet layer to form 400 filament segments (which corresponds to 50 loops). After invoking the renormalization strategy there remain 342 filament elements. A plan view of these loops is shown in Fig. 1b.

The partition associated with time step 3 is illustrated in Fig. 2a. At this step 154 additional sheet clusters are created, resulting in a total of 1590 sheets in the sheet layer. Fig. 2b illustrates the filament loops which have entered the flow interior prior to, and during, this step. After renormalization, there are 1709 such filament elements.

Fig. 3a illustrates the partition for the 4th time step. In this step 574 sheets were created, bringing the total in the sheet layer to 1953. Fig. 3b illustrates the resulting (after renormalization) 3687 filament segments from the first four time steps. Fig. 3c shows an "elevation" view of these elements.

## 8. Further Work

An algorithm has been proposed for the creation of vortex loop elements at a no-slip surface. This has been implemented in a preliminary way for the case of a flat plate. Further investigation of this algorithm should determine improvements to the basic strategy. In particular the appropriate values of the various parameters associated with the renormalization algorithms still require to be numerically explored. Also some study of the accuracy with which level curves are numerically determined should be made.

## 9. Acknowledgements

The author thanks Alexandre Chorin for his crucial suggestions in the course of this work. Also Terry Ligocki is thanked for providing his own graphics software associated with Fig. 3c.

## 10. References

1. Anderson, C. R., Greengard, C., Greengard, L., and Rokhlin, V., "On the accurate calculation of vortex shedding", *Phys. Fluids*, A **2**, 883-885, 1990.

2. Anderson, C. R., "The Fast Multipole Method without multipoles", UCLA Computational and Applied Mathematics Report 90-14, 1-28, 1990.

3. Beale J. T. and Majda, A., "High order accurate vortex methods with explicit velocity kernels", *J. Comp. Phys.*, **58**, 188-208, 1985.

4. Chorin, A.J., "Numerical study of slightly viscous flow", *J. Fluid Mech.*, **57**, 785-796, 1973.

5. Chorin, A. J., "Vortex sheet approximation of boundary layers", *J. Comp. Phys.*, **27**, 428-442, 1978.

6. Chorin, A.J., "Vortex models and boundary layer instability", *SIAM J. Sci. Stat. Comput.*, **1**, 1-21.

7. Chorin, A.J., *Computational Fluid Mechanics — Selected Papers*, Academic Press, N.Y., 1989.

8. Chorin, A.J., "Equilibrium statistics of a vortex filament with applications", *Comm. Math. Phys.*, 1991 (in press).

9. Chorin, A.J., "Hairpin removal in vortex interactions II", 1991 (in press).

10. Greengard, C., "Convergence of the vortex filament method", *Math. Comp.*, **47**, 387-398, 1986.

11. Greengard, L., and Rokhlin, V., "A Fast algorithm for particle simulations", *J. Comp. Phys.*, **73**, 325-348, 1987.

12. Jones, D. S., *The Theory of Electromagnetism*, Pergamon Press, Oxford, 1964.

13. Long, D. G., "Convergence of the random vortex method in two dimensions", *J. A.M.S.*, 4, 779-804, 1988.

14. Puckett, E. G., "Vortex methods: An introduction and survey of selected research topics", to appear in *Incompressible Computational Fluid Dynamics — Trends and advances*, ed. R. A. Nicolaides and M. D. Gunzburger, Cambridge University Press, 1991.

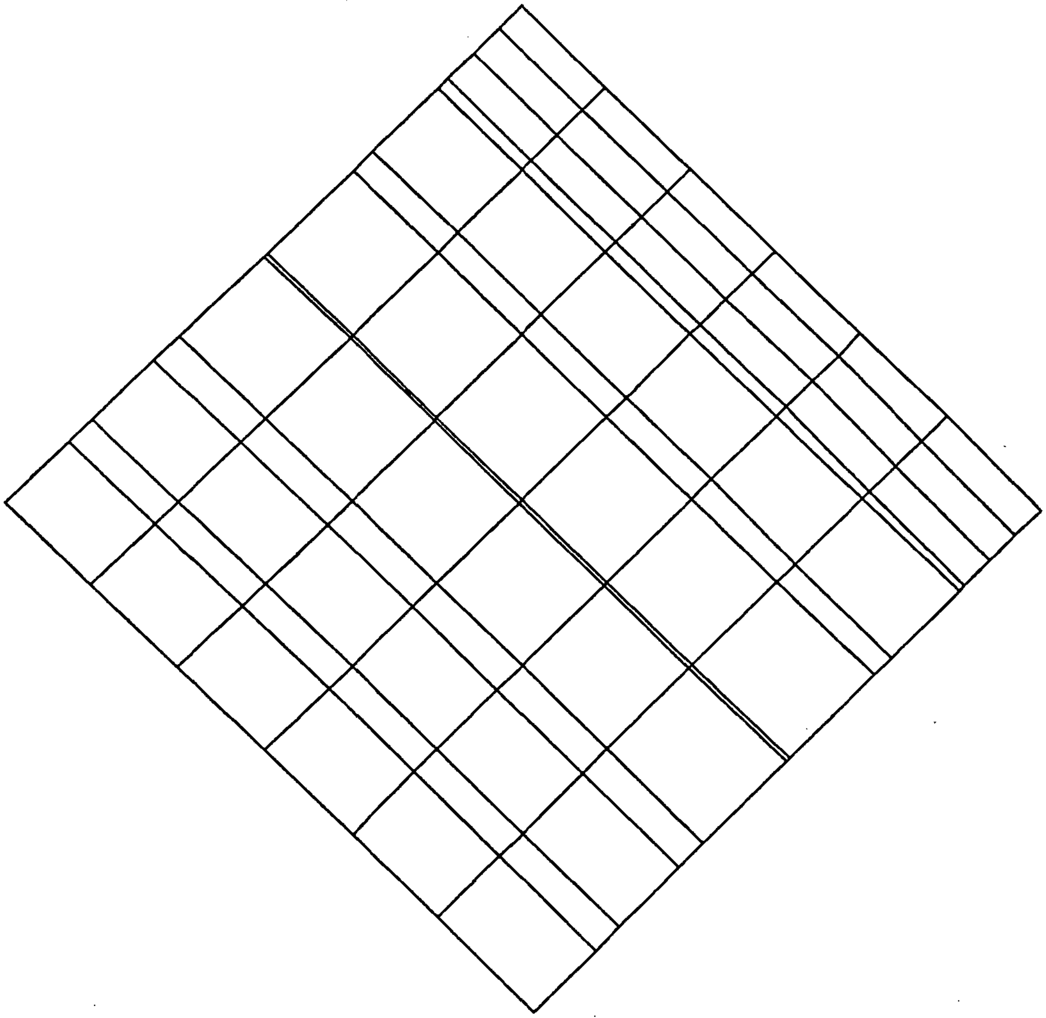


**Figure 1a** Partition of surface at beginning of time step 2.

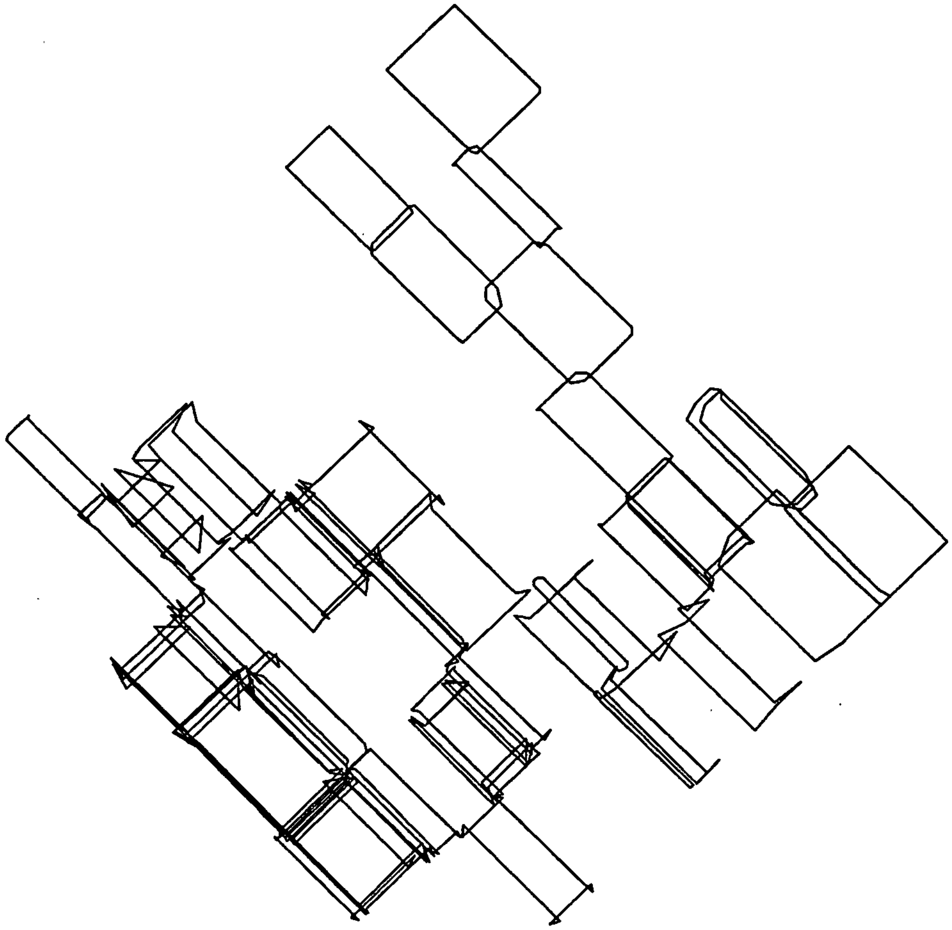


**Figure 1b** Loop filaments which have entered the flow interior from the sheet layer by completion of time step 2.

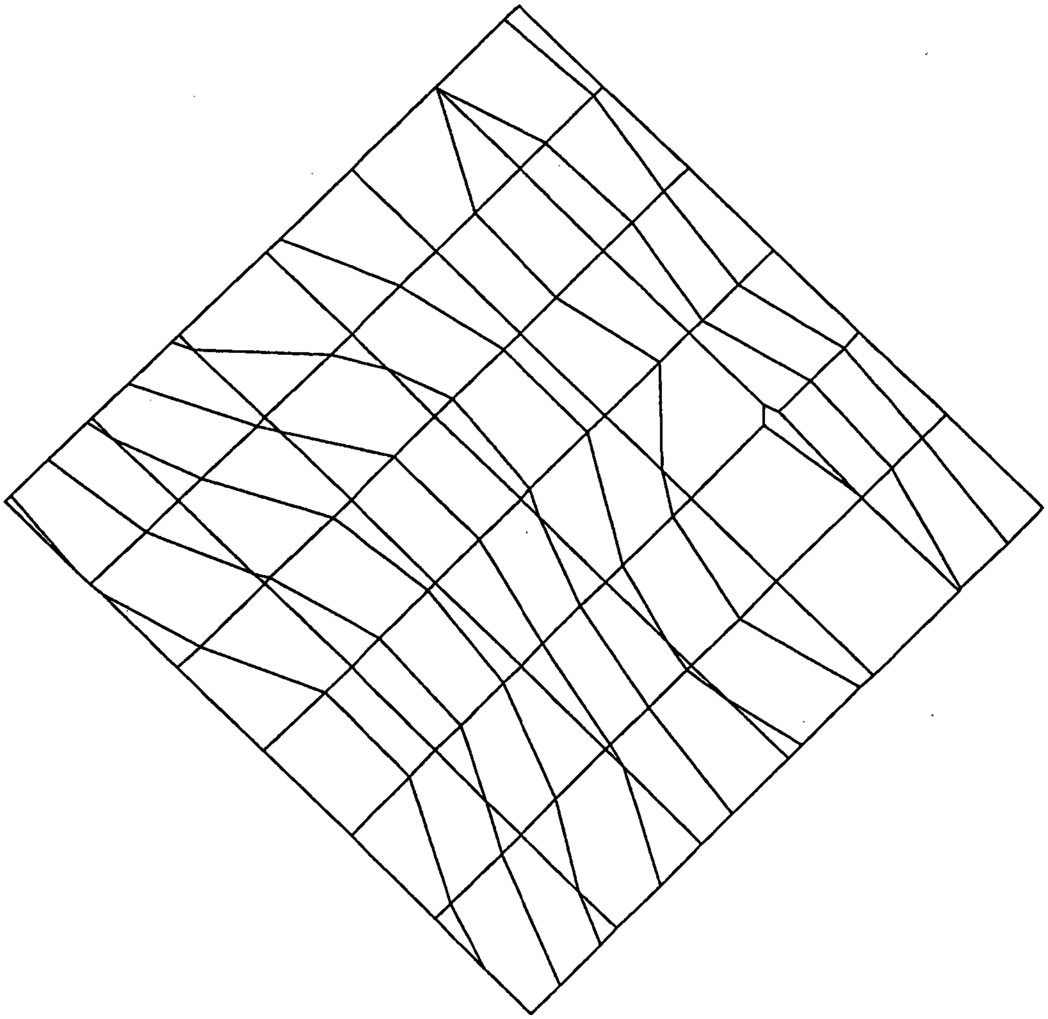


**Figure 2a** Partition of surface at beginning of time step 3.

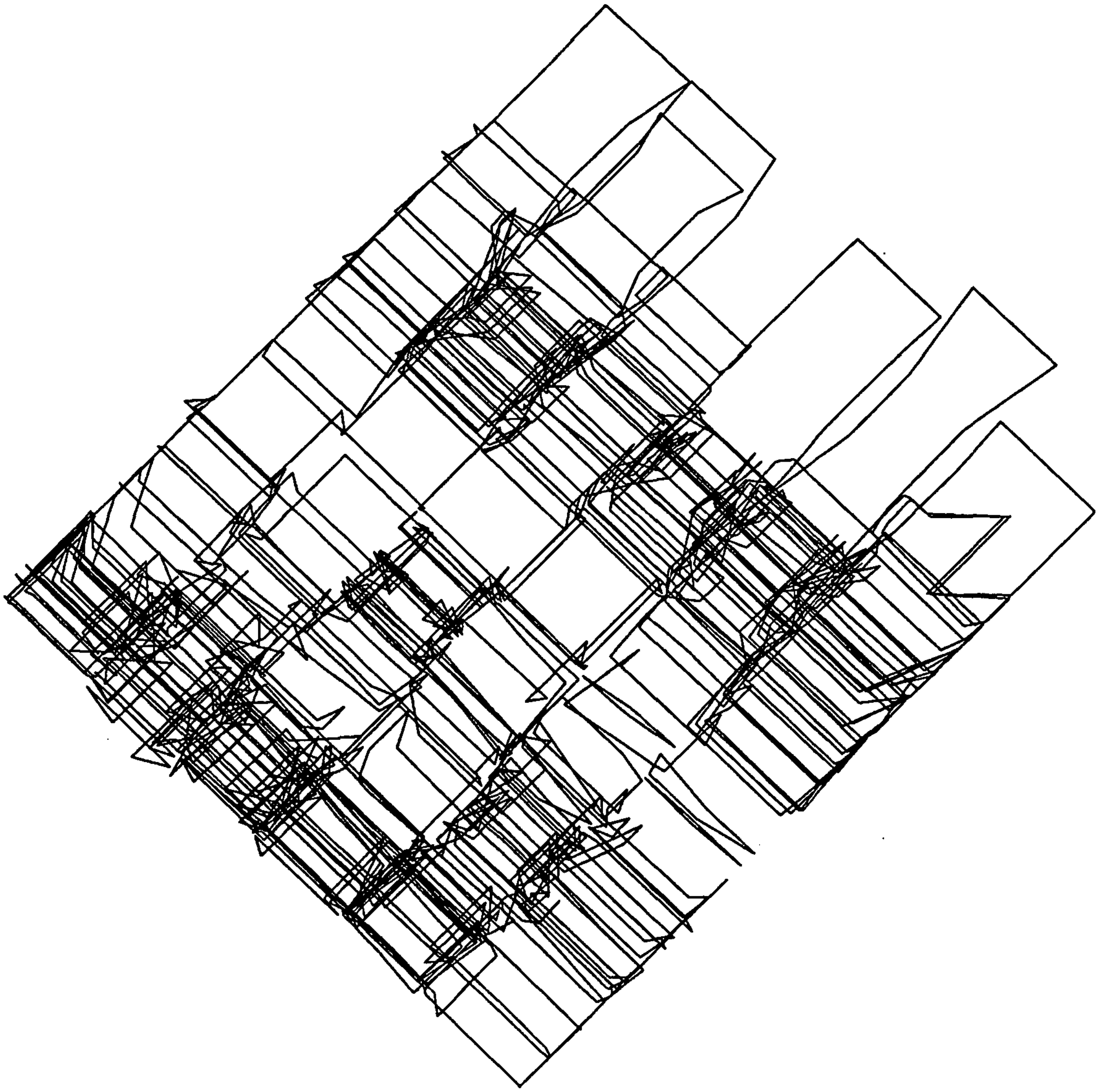


**Figure 2b** Loop filaments in flow interior at time step 3.

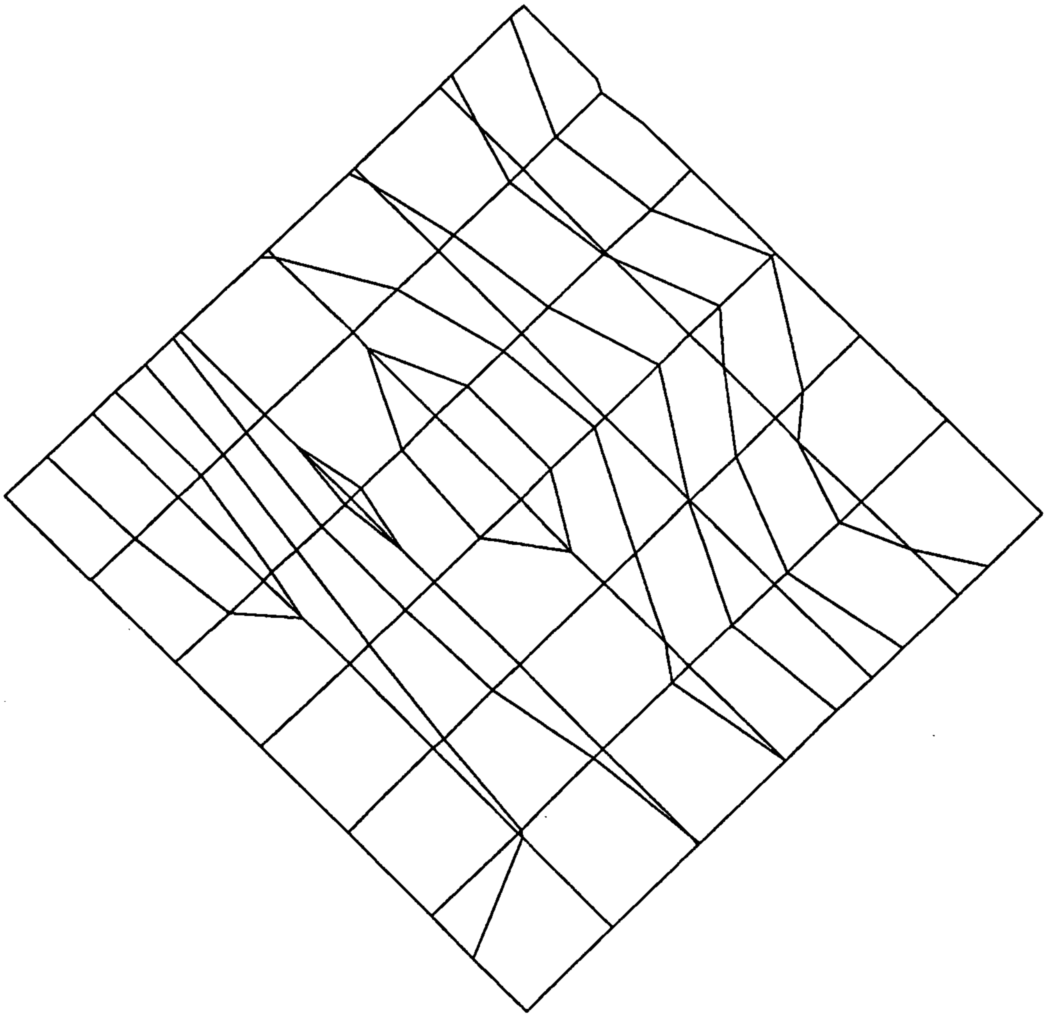


**Figure 3a** Partition of surface at beginning of time step 4.

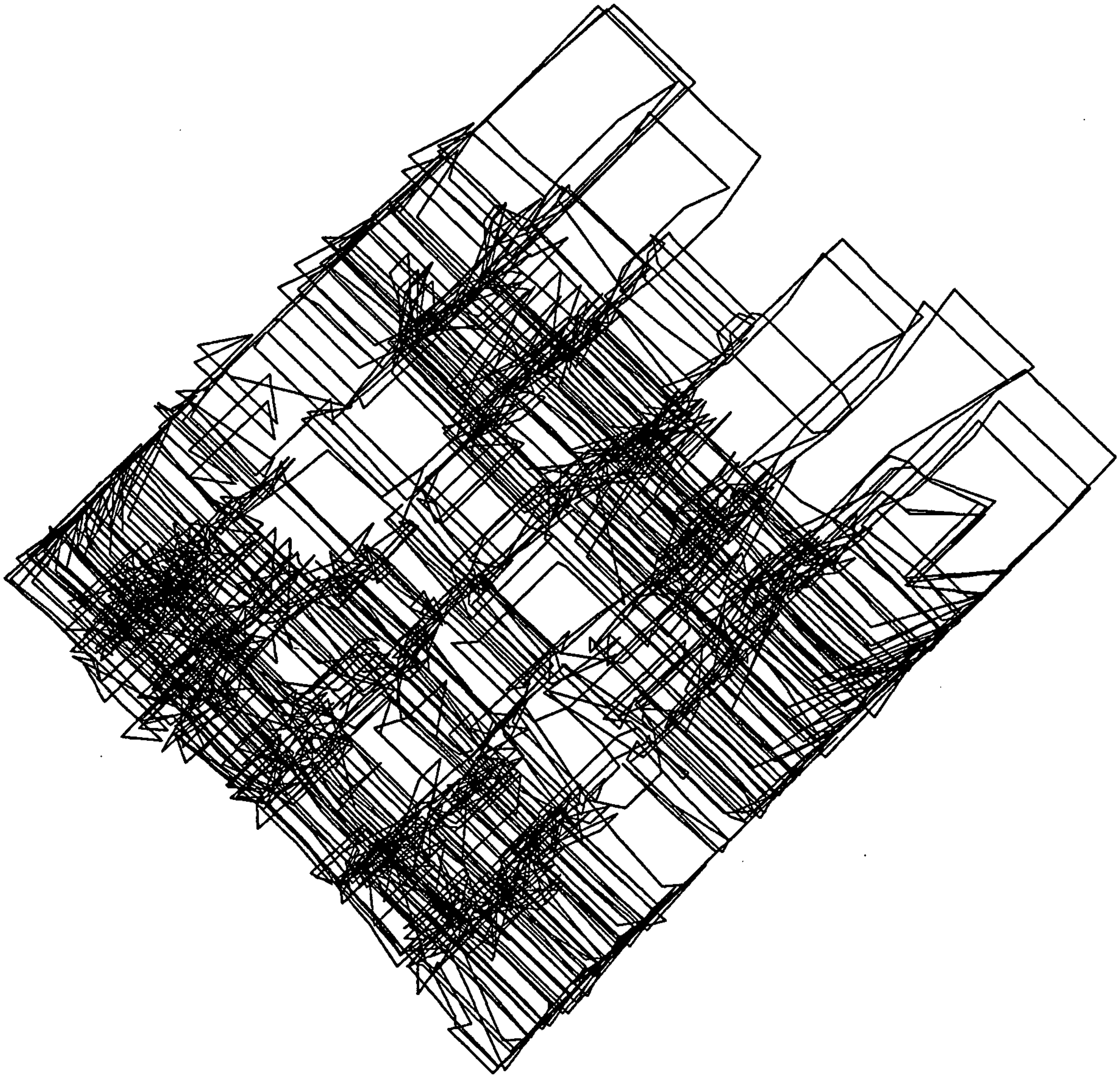


**Figure 3b** Loop filaments in flow interior at time step 4.

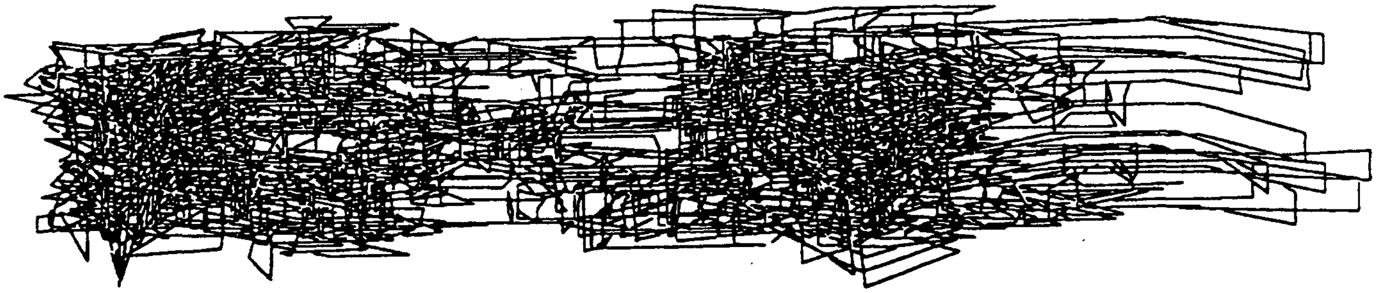


**Figure 3c** Loop filaments in flow interior at time step 4: an elevation view.

LAWRENCE BERKELEY LABORATORY
UNIVERSITY OF CALIFORNIA
INFORMATION RESOURCES DEPARTMENT
BERKELEY, CALIFORNIA 94720