LBL-35707
UC-405

Lawrence Berkeley Laboratory

UNIVERSITY OF CALIFORNIA

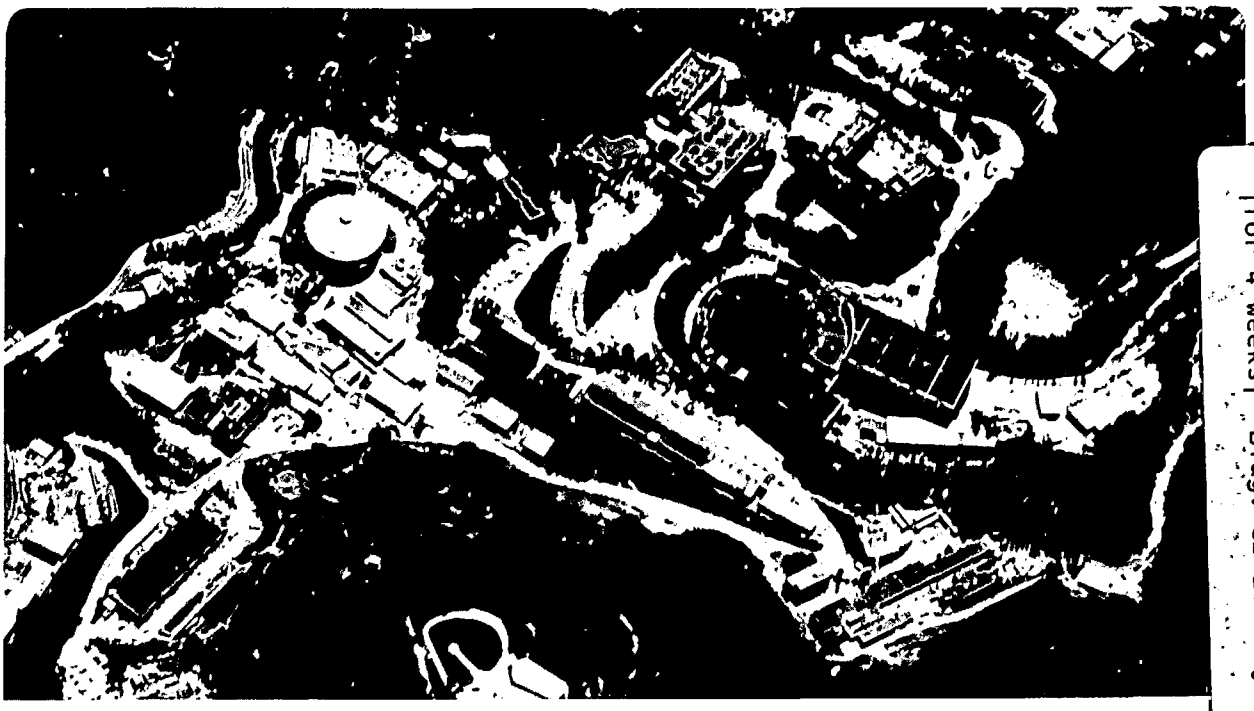
Physics Division

Mathematics Department

An Investigation of the Vortex Method

D.W. Pryor, Jr.
(Ph.D. Thesis)

May 1994



LOAN COPY
Circulates
for 4 weeks
Bldg. 50 Library.
LBL-35707
Copy 2

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. 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-35707

AN INVESTIGATION OF THE VORTEX METHOD[1]

Duaine Wright Pryor Jr.

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

Ph.D. Thesis

May 1994

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

An Investigation of the Vortex Method

Copyright © 1994

by

Duaine Wright Pryor Jr.

The U.S. Department of Energy has the right to use this thesis for any purpose whatsoever including the right to reproduce all or any part thereof

recycled paper

Contents

1 Introduction and Conclusions **2**
- 1.1 The Equations of Motion . . . 2
  - 1.1.1 The Navier-Stokes Equations . . . 2
  - 1.1.2 Vorticity and The Stream Function . . . 2
- 1.2 The Vortex Method . . . 4
  - 1.2.1 Two Dimensions . . . 4
  - 1.2.2 Smoothing and Core Functions . . . 5
  - 1.2.3 Three Dimensions . . . 8
- 1.3 Preview . . . 8
- 1.4 Overview Of Numerical Results . . . 9
- 1.5 Conclusions . . . 11

2 First Implementation and Experiments **14**
- 2.1 The Free Space Method . . . 14
  - 2.1.1 description of the method . . . 14
  - 2.1.2 Tests of the implementation . . . 15
- 2.2 Approximating Viscosity By The Random Walk . . . 15
- 2.3 The No Leak Boundary Condition . . . 16
  - 2.3.1 Tests of the Potential Solver . . . 17
- 2.4 The No Slip Condition . . . 18
- 2.5 First Results . . . 18
  - 2.5.1 Initial Computation . . . 18
  - 2.5.2 Conservation Form . . . 19
  - 2.5.3 Making Use Of Conservation Form . . . 19
- 2.6 Interim Conclusion . . . 21

3 Stretch Schemes **22**
- 3.1 Filament Schemes . . . 22
  - 3.1.1 Chorin's Filament Stretch and Helmholtz Stretch . . . 22
  - 3.1.2 Knio and Ghoniem's Scheme . . . 23
- 3.2 Unconnected Singular Schemes . . . 24
  - 3.2.1 Singular ω times ∇u Stretch . . . 24
- 3.3 Singular ω times ∇u^T . . . 25
  - 3.3.1 Exact Stretch . . . 25
- 3.4 Unconnected Smooth Schemes . . . 25
  - 3.4.1 Smooth Omega Times ∇u . . . 25
  - 3.4.2 Smooth Omega Times ∇u^T . . . 26

4 Ring Calculations I **27**
4.1 Connected versus Unconnected Schemes . . . 27
4.2 Blobs versus Sticks . . . 27
4.3 The Parameter Space . . . 27
4.4 Code Details . . . 28
4.4.1 Machines, parallelism and efficiency . . . 30
4.5 Code Verification . . . 30
4.6 Inviscid Single Filament Computations . . . 31
4.6.1 Observations . . . 75
4.6.2 Conjectures about causes . . . 76
4.7 Future Numerical Work . . . 78
4.7.1 Accurate Computation of Invariants . . . 78
4.7.2 Fourier Space Evolution and Filtering . . . 78
4.7.3 Modeling of Vorticity and Torsion Zeroes . . . 79
4.7.4 Front Motion Model Near a Vorticity Zero . . . 79

5 Theoretical Considerations **80**
5.1 Derivation of Conservation Form . . . 80
5.2 Divergence Of Vorticity Is Material . . . 81
5.3 Interpretation of Non-solenoidal Vorticity . . . 82
5.4 The Transpose Equation is an Euler Lagrange Equation . . . 82
5.5 Some Previously Known Results . . . 83

6 A Lax-Friedrichs Scheme for Vorticity Transport **85**
6.1 The Discretized VTE . . . 85
6.2 Boundary Conditions . . . 86
6.3 Evaluation of the Velocity Field . . . 86
6.4 Initial Conditions . . . 86
6.5 Checking the Code . . . 87
6.6 Test Problem Results . . . 87
6.7 What Was Learned . . . 90

7 On Projection Schemes To Enforce $\nabla \cdot \omega = 0$ **91**
7.1 Motivation . . . 91
7.2 Hodge Decomposition . . . 91
7.3 The Projection in a Lagrangian Context . . . 92
7.3.1 Derivation of An Algorithm . . . 92
7.3.2 Test Implementation . . . 93
7.3.3 What causes the large condition number . . . 93
7.4 Stability of the Problem . . . 93

8 Resampling and Numerical Non-uniqueness **95**
8.1 Methods of Updating the Weights . . . 95
8.2 Numerical Non-uniqueness of Vortex Weights . . . 96
8.3 Volume Estimation . . . 97
8.3.1 Using The Core Function . . . 98
8.3.2 Simple Density Estimates . . . 98
8.3.3 Numerical Results Summary . . . 99
8.4 Future Directions . . . 99
8.4.1 Spectral Or Galerkin Analysis Of Implied Basis . . . 99
8.4.2 How Good or Bad is Volumetric Resampling . . . 101
8.4.3 Can One Pick A Best Set Of Weights . . . 101

8.4.4 Analogy Between Integration and Resampling . . . 101

9 Vorticity Near A Zero and Reconnection **102**
9.1 Domains In Which To Study Reconnection . . . 102
9.2 Toy Models of Vorticity Near A Zero . . . 103
9.2.1 Generic Smooth Vorticity Near A Zero . . . 103
9.2.2 Toy Model I . . . 104
9.2.3 Toy Model II . . . 105
9.3 Smoothed and Singular Vorticity Near A Zero . . . 105
9.4 The Core Unmixing Phenomenon . . . 105
9.5 Some Possible Reconnection Schemes . . . 112
9.5.1 Reconnection Based on Non-uniqueness . . . 112
9.5.2 Reconnection Based on Matching the Linearization . . . 112
9.5.3 Reconnection Based on $\nabla \cdot \omega$ as an entropy . . . 112
9.6 Questions . . . 112

10 A Sketch of Vorticity Exchange Viscosity Algorithms **114**
10.1 A General Framework . . . 114
10.2 Fishelov Scheme . . . 115
10.3 Vortex Centered Finite Difference Scheme . . . 115
10.4 Cottets Scheme . . . 115
10.5 Mas-Gallic Scheme . . . 116
10.6 Adding Volume Estimation These Schemes . . . 116

11 Ring Calculations II **117**
11.1 The Parameter Space . . . 117
11.2 Discretization of the Core . . . 118
11.3 The Code . . . 118
11.3.1 Dirty Tricks . . . 118
11.4 Computations . . . 119
11.5 Observations . . . 139
11.5.1 Comparison to Single Filament Runs . . . 139
11.5.2 Failure to Conserve Estimated Volume . . . 139
11.5.3 Growth of $\nabla \cdot \omega$ and Z-impulse in filament methods . . . 140
11.5.4 "Discontinuities" in the Area of Cross Sections . . . 140
11.6 Future Work . . . 141
11.6.1 Invariants and Fourier Space Evolution . . . 141
11.6.2 Numerically demonstrate the non-uniqueness of particle values . . . 141

12 Boundary Conditions Schemes **142**
12.1 Chorin's Algorithm . . . 142
12.2 A Deterministic Boundary Condition Algorithm . . . 143
12.3 A Second Deterministic Algorithm . . . 144
12.4 Generating Connected Filaments . . . 146

13 Flow Past a Sphere **147**
13.1 The Code and Algorithms . . . 147
13.2 Verification . . . 147
13.3 The Parameter Space and the Algorithms . . . 148
13.4 Machines and Parallelism . . . 149
13.5 Dirty Tricks . . . 149
13.6 The Sphere . . . 150

13.7 Runs 150

Bibliography **153**

Chapter 1

Introduction and Conclusions

1.1 The Equations of Motion

1.1.1 The Navier-Stokes Equations

Newtonian fluid flow is described by the Navier-Stokes equations which express the conservation of several important physical quantities [2]. In the incompressible case, the dimensionless form of these equations reduces to

$$\partial_t u + u \cdot \nabla u = -\nabla P + \frac{1}{R}\Delta u \tag{1.1}$$

$$\nabla \cdot u = 0 \tag{1.2}$$

where u is the velocity of the fluid, P is the pressure, and R is the Reynolds number[2]. The first equation expresses conservation of momentum, while the second expresses conservation of mass[2].

1.1.2 Vorticity and The Stream Function

The curl of the velocity field, which is called *vorticity*, is important for a number of reasons of which two are used in this section. The support of the vorticity is typically much smaller than the support of the velocity (which is typically the whole domain). A set of equations governing the evolution of vorticity and determining the velocity and pressure can be derived. This new set of equations is equivalent to the Navier-Stokes equations for incompressible flow[18]. Thus, the vorticity representation is equivalent to the velocity-pressure description and has a natural adaptivity[18].

Vorticity is defined to be the curl of the velocity and usually designated by

$$\omega = \nabla \times u \tag{1.3}$$

Since $\nabla \cdot u = 0$,

$$u = \nabla \times A \tag{1.4}$$

for some vector field A, called the *stream function*. A is a scalar in the case of two dimensional flow and a vector in three dimensions[2].

The relation between A and ω is derived as follows. Choose A to be divergence-free. This can be accomplished by adding a gradient field to A, which leaves u unchanged. For such an A,

$$\omega = \nabla \times u = \nabla \times \nabla \times A = -\triangle A \tag{1.5}$$

$$\triangle A = -\omega \tag{1.6}$$

Assuming boundary conditions on A which allow the inversion of the Laplacian,

$$A = -\triangle^{-1}\omega \tag{1.7}$$

and

$$u = -\nabla \times \triangle^{-1}\omega \tag{1.8}$$

[2]. In free space, the fundamental solution to Laplaces equation, defined by $\triangle K(x) = \delta(0)$ and $K(\infty) = 0$ is used to express A as the convolution of the fundamental solution with the data[21].

$$A = K \star \omega \tag{1.9}$$

In two dimensions, $K(r) = \frac{1}{2\pi}\log(r)$, while in three dimensions, $K(r) = \frac{-1}{4\pi r}$[21].

A summary of the relations between u, w, and A follows.

- u determines ω exactly.
- u determines A to within a gradient of a scalar.
- ω, with appropriate boundary conditions on A, determines A.
- ω and boundary conditions on u determine u up to the gradient of a scalar.
- A determines u exactly
- A determines ω exactly

An evolution equation for vorticity may be derived from the evolution equation for velocity [2]. Since u determines ω exactly, the time evolution of u determines the time evolution of ω exactly. By taking the curl of the momentum equation, and using some vector identities along with the continuity equation, one arrives at the vorticity transport equation, which we will call the VTE,

$$D_t\omega = \partial_t\omega + u \cdot \nabla\omega = \omega \cdot \nabla u + \frac{1}{R}\triangle\omega \tag{1.10}$$

The VTE describes the evolution of vorticity with time[18]. $D_t = (\partial_t + u \cdot \nabla)$ is called the material derivative. While the pressure has been eliminated, the VTE involves both the vorticity and the velocity.

1.2 The Vortex Method

The vortex method is a numerical scheme for solving the vorticity transport equation. Chorin introduced modern vortex methods in [11]. The vortex method is a Lagrangian, grid free method which has less intrinsic diffusion than many grid schemes. It is adaptive in the sense that elements are needed only where the vorticity is non-zero. Our description of vortex methods begins with the point vortex method of Rosenhead [44] for two dimensional inviscid flow, and builds upon it to eventually cover the case of three dimensional slightly viscous flow with boundaries.

This section gives an introduction to the fundamentals of the vortex method. This is done in order to give a basic impression of the previous work and its line of development, as well as develop some notation and concepts which will be used later. The purpose here is not to give a full review of vortex methods or the contributions made by all the researchers in the field. Please refer to the excellent review papers in Sethian and Gustafson [26], chapters 1 Sethian[47], 2 Hald[28], 3 Sethian[48], 8 Chorin[16] provide a solid introduction to vortex methods, including convergence theory, application in two dimensions and connection to statistical mechanics and polymers. Much of the information in this review is taken from those chapters, Chorin and Marsden[18] and Batchelor [2] The chapters of [26] are also useful for their extensive bibliographies.

1.2.1 Two Dimensions

Free Space, Inviscid

In two dimensions, the stretch terms disappear from of the vorticity transport equation, and the vorticity itself has only one non-zero component[18]. Thus, the vorticity transport equation becomes an advection equation for a scalar vorticity $D_t\omega = 0$. This implies that vorticity is constant along particle trajectories.

This observation leads to the simplest vortex method which is the point vortex method of Rosenhead[44]. Divide the domain into some number of particles, or *vortex elements*, each of which has zero volume, but still carries some vorticity. Formally, the vorticity carried by a vortex element is a constant times a delta "function" centered at that element. The vorticity is thus $\omega(x) = \sum w_i \delta(x - x_i)$ where x_i is the position of the ith particle. Given the vorticity, a stream function may be constructed by convolving the vorticity with the fundamental solution to Poissons equation. In two dimensions, this kernel is $K(r) = \frac{1}{2\pi}\log(r)$, which leads to a formula for $A(x) = \frac{1}{2\pi}\sum w_i \log(|x - x_i|)$. u may be derived by taking the curl of A $u(x) = \nabla \times A = \frac{1}{2\pi}\sum w_i \nabla \times \log(|x - x_i|)$ The result of this process is that given the vorticity, one can construct the velocity. Given the vorticity and velocity, one can update the vorticity field by moving the vortex elements.

The full two dimensional free space point vortex algorithm is expressed by the following "pseudocode".

- Sample initial data onto vortex elements with positions, and strengths.

- While not done,
  - For each vortex element, j, compute $u(x_j)$
  - For each vortex element, j, compute a new position by integrating the velocity field for a time Δt

While the convergence of this method has been proved, it encounters numerical difficulties due to the singularity of the kernel, which allows arbitrarily large velocities to develop[28].

1.2.2 Smoothing and Core Functions

The basic idea of using smoothing functions is to approximate the delta function in the point vortex method by a kernel which is smooth everywhere, and converges to a delta distribution as the approximation is refined[11][3] [4][5][28][27]. There are three principal motivations for this approach. It places an upper bound on the velocity that one vortex element can induce on another. It also removes the need to evaluate the convolution of the Poisson kernel and the vorticity as a principal value integral. As long as the parameter which determines the smoothing radius is refined along with the other parameters, this scheme converges [3][4][5][28][27].

Given a function, $f(r) : \Re^+ \mapsto \Re$ which is smooth, decays at infinity, and has total integral equal to one, one can define a function $f_\delta(r) = \frac{1}{\delta^2} f(\frac{r}{\delta})$ in two dimensions or $f_\delta(r) = \frac{1}{\delta^3} f(\frac{r}{\delta})$ in three. f_δ approaches delta function in two (three) dimensions as δ goes to zero. f needs to have some set of moment properties for the convergence proofs to hold, but we will ignore those for now[28]. The vorticity may be written as $\omega_\delta(x) = \sum w_i f_\delta(x - x_i)$. Define $\kappa_\delta(r) = 2\pi \int_0^r f_\delta(b) b db$, $\kappa_\delta(r)$ is the integral of f_δ over a disk of radius r. The induced stream function and velocity field can be written as $A_\delta(x) = \sum \omega_i \kappa_\delta(x - x_i) K(x - x_i)$ and $u_\delta(x) = \nabla \times A_\delta(x)$

In three dimensions, there is an added complication. The vorticity just defined is not divergence-free[28]. One may define the vorticity in three dimensions to be $\omega_\delta(x) = \nabla \times u_\delta(x)$[28]. This vorticity induces the same velocity as the previously defined one, but is divergence-free[28]. It thus represents the divergence-free part of the previously defined vorticity[28]. It may be represented as $\omega_\delta(x) = \sum \omega_i Z_\delta(x - x_i)$ [28]. Z has two terms, one of which looks like $\kappa_\delta(r)$ and dominates in the near field, and another which looks like $\frac{1}{r^3}$ and dominates the far field[28].

The full two dimensional free space smoothed vortex method is written in "pseudocode" below.

- Sample initial data onto vortex elements with positions, and strengths.
- While not done
  - For each vortex element, j, compute $u_\delta(x_j)$
  - For each vortex element, j, compute a new position by integrating the velocity field for a time Δt

Viscous Flows

Two different classes of algorithms for dealing with the viscous term of the vorticity transport equation will be discussed. One is the *random walk* method of Chorin[11]. The other is the class of *deterministic diffusion* schemes [22][19][20]. There is a recent diffusion algorithm [32] which is interesting and does not fit either of the categories mentioned. This algorithm was invented after the work described in this document was nearly complete.

The Random Walk Method The random walk method relies on the fact that a large set of particles undergoing Gaussian random motion approximates the process of viscous diffusion. This representation of diffusion of vorticity is noisy, but this noise is greatly reduced when the vorticity is integrated to get the velocity[18]. Thus, this method represents the diffusion of velocity fairly well. The two dimensional vortex method with random walk may be written in pseudocode as follows.

- Sample initial data onto vortex elements with positions, and strengths.
- While not done
  - For each vortex element, j, compute $u_\delta(x_j)$
  - For each vortex element, j, compute a new position by integrating the velocity field for a time Δt
  - For each vortex element, j
    - choose a random vector from a Gaussian distribution of mean zero and variance $4\pi\Delta t/R$
    - add this vector to the position of element j

Deterministic Schemes Deterministic diffusion schemes developed as an alternative to the random walk method for several reasons. The convergence of the random walk method is slow[11][18]. In addition, some object to the random walk method on the basis that it is not a deterministic algorithm and thus can give different answers depending on the random numbers that are chosen. This lead to the development of deterministic diffusion schemes.[22][19][20] These rely on changing the vortex element strengths by an amount which depends on an approximation to $\Delta\omega_\delta$ Therefore these algorithms are also called *vorticity exchange algorithms.* The strengths, ω_i may be thought of as a point value times a volume, $\omega_i = dV_i * \omega(x_i)$ Diffusion of vorticity is governed by $\partial_t\omega = \frac{1}{R}\Delta\omega$ If into the RHS one substitutes the representation of ω_δ, and equates ω_i to $\omega(x_i)dV$, the result is

$$\partial_t\omega(x_i) = \frac{1}{R}\Delta\sum_j \omega(x_j)dV_j f_\delta(x_i - x_j) \tag{1.11}$$

Multiply through by dV_i and get

$$\partial_t dV_i\omega(x_i) = \frac{dV_i}{R}\Delta\sum_j \omega(x_j)dV_j f_\delta(x_i - x_j) \tag{1.12}$$

This is equivalent to

$$\partial_t \omega_i = \frac{dV_i}{R} \Delta \sum_j \omega_j f_\delta(x_i - x_j) \tag{1.13}$$

Let $\mathcal{L}_h$ be an approximation to the Laplacian which depends on a parameter h

$$\partial_t \omega_i \approx \frac{dV_i}{R} \mathcal{L}_h \sum_j \omega_j f_\delta(x_i - x_j) \tag{1.14}$$

In the scheme of Fishelov, $\mathcal{L}_h \sum_j \omega_j f_\delta(x_i - x_j) = \sum_j \omega_j \Delta(f_\delta(x_i - x_j))$ In the scheme of Cottet, $\mathcal{L}_h \sum_j \omega_j f_\delta(x_i - x_j) = \sum_j \omega_j \Lambda_h(x_i - x_j)$ where Λ_h is chosen so that $\omega \star \Lambda_h = \Delta\omega + \mathcal{O}(h^p)$

Boundary Conditions

Inviscid Flow (no leak) For inviscid flow around a body the boundary condition is that $u \cdot n = 0$, where n is the body normal vector[18]. Recall that ω determines u only up to the gradient of a scalar. It is then natural to ask if one can find a scalar ϕ such that

$$u_\delta + \nabla\phi = u_{total} = u_{vor} + u_{pot} \tag{1.15}$$

satisfies the boundary condition. [11] Since the resulting flow must be divergence-free, ϕ must satisfy $\Delta\phi = 0$[11]. The boundary condition on u translates into $n \cdot u_\delta + n \cdot \nabla\phi = 0$ ϕ is determined as the result of

$$\Delta\phi = 0 \tag{1.16}$$

$$n \cdot \nabla\phi = -u_\delta \cdot n \tag{1.17}$$

$$\tag{1.18}$$

provided that $\int u_\delta \cdot n = 0$, which is a consequence of incompressibility and the divergence theorem [11][18].

Viscous Flow (no leak and no slip) For viscous flows, the boundary condition is that $u = 0$ on the body. Using the method above, one can construct a flow which satisfies $u \cdot n = 0$. By abuse of notation, define $u \cdot \tau = u - u \cdot n$ This flow can be modified so that $u \cdot \tau = 0$ as well. A vortex sheet of strength β induces a jump in velocity across it of strength β When added to u the flow induced by a layer of vorticity on the surface of the body of strength $-u \cdot \tau$ will be zero. [11] One way to implement this is to represent the required layer using vortex elements. In the case that the body is piecewise linear, a somewhat more sophisticated representation uses small vortex sheets which in a layer near the body evolve according to a boundary layer approximation, the Prandtl equation[12]. As the sheets exit this layer they are converted to vortex elements in a manner which preserves total circulation[12]. The thickness of this numerical boundary layer is reduced as the other parameters of the algorithm are refined[12].

1.2.3 Three Dimensions

In three dimensions, the vorticity is a vector quantity and is not simply advected. It is amplified by stretching. $D_t = \omega \cdot \nabla u$ may be thought of as a set of ODE along characteristics of the flow. Thus the vorticity must be advected and stretched. Since the vorticity is a vector and $\omega = \nabla \times u$, the vorticity is constrained to be divergence-free. While these three factors do not have analogues in two dimensional flow, the notions of smoothing functions, random walk and deterministic diffusion schemes carry over immediately.

Stretch

A number of methods for modeling the amplification of vorticity are reviewed and classified in chapter 3. Only the stretch schemes used in chapter 2 are reviewed here.

One stretch scheme is based on the fact that $\omega \cdot \nabla u$ is a directional derivative[13]. Let $\hat{\omega}_i$ be a unit vector in the direction of ω_i Given a vortex element at x_i, the directional derivative may be approximated by

$$\omega_i \cdot \nabla u \approx \frac{u(x_i + h * \hat{\omega}_i) - u(x_i - h * \hat{\omega}_i)}{2h} \tag{1.19}$$

A second stretch scheme is based on formal differentiation of the sum which represents u[1]. Since $u_\delta(x) = \nabla \times A_\delta(x)$ and, $A_\delta(x) = \sum \omega_i \kappa_\delta(x - x_i) K(x - x_i)$, $\nabla u_\delta = \nabla\nabla \times A_\delta(x) = \nabla\nabla \times \sum \omega_i \kappa_\delta(x - x_i) K(x - x_i)$ A bit of vector gymnastics allows ∇u_δ to be evaluated as a convolution sum over the vortex elements[1].

Blobs, Sticks and Core Functions

The use of a core function to smooth out the singularity Poissons kernel carries over directly into three dimensions. In three dimensions, however, there are several alternate representation of vorticity which have been used. Instead of supporting the vorticity on blobs (spherically symmetric objects), one can support it on sticks (objects which cylindrical symmetry). In addition, these points or sticks can be thought of as connected in filaments or as independent points.

1.3 Preview

In chapter 2, which is included for historical reasons, the results of an early implementation of the three dimensional vortex method for flow past an arbitrary body are reported. The method used for imposing the boundary condition is also described. This implementation encounters great difficulty and must be considered to be a failure.

In chapter 3, different ways of treating the three dimensional stretch term are reviewed and classified.

In chapter 4, numerical results are presented from the computation of the evolution of a single filament ring using a wide variation of inviscid algorithms.

In chapter 5, the conservation form of the VTE is introduced. It is shown that the divergence of the vorticity is material. An interpretation of non-solenoidal vorticity is given and a group of published results which are related to the results of chapter 5 are reviewed.

Chapter 6, reports more numerical results. In this case, a finite difference scheme is used to verify some of the results of chapter 5.

In chapter 7, the fact that the vortex elements themselves do not define a divergence free field is examined. It is conjectured that a projection scheme could be used instead of the Z function of Hald [28] to enforce the condition $\nabla \cdot \omega = 0$. One such scheme is analyzed.

In chapter 8 the problems discovered in 7 are explored in greater detail. It is shown that ill-posedness of resampling a smooth function onto the vortex elements extends into the realm of stretch schemes and diffusion schemes. The vortex particle weights are shown to be numerically non-unique.

Chapter 9 is a collection of speculative results on the behavior of vorticity near a zero and near a zero of its torsion.

In chapter 10, deterministic diffusion schemes are reviewed. Volume estimation developed in chapter 8 is used to enhance these schemes.

In chapter 11, various viscous and inviscid vortex methods are applied to the evolution of a multifilament ring.

Chapter 12, returns to the original problem, flow past a body. Two new algorithms for the enforcement of the boundary conditions in the vortex method are proposed.

In chapter 13 a more recent attempt to compute flow past a sphere is described. This computation uses the new boundary algorithms along with a deterministic diffusion scheme in the interior.

1.4 Overview Of Numerical Results

This thesis contains a large number of numerical results. This section gives an overview of those results and some of the methods used in interpreting them.

The computations of Chapter 2 are the results of an attempt to build a program capable of computing arbitrary exterior flows. Upon testing this program, in many different ways, it became clear that something was causing an explosive growth of vortex elements. This growth was not consistent with what is known about the type of flow we wished to compute. The growth was widespread, not isolated, and it began at far too early a time.

In Chapters 4 and 11 the evolution of single filament and multifilament vortex rings is studied using the vortex method. Due to the failure of the computation of Chapter 2, the existence of a conservation form for the VTE, and the fact that impulsive startup of flow around a body represents singular initial data, it was decided that a study of free space vortex method algorithms

should be made. The intent of this study would be to attempt to determine in a very simple context, what about the vortex method is causing problems.

From initial single filament runs, it became clear that qualitative factors dominated both the individual computations and the differences between computations. Factors such as the preservation or altering of symmetry, creation or destruction of zeroes, presence or absence of helical waves dominated. For example, one method begins with six-fold symmetric initial data and produces a four-fold symmetric result, while another begins with the same initial data and produces a result with no apparent symmetry in a time where the equation should should still be governed by linear theory. In looking at all of the results, it is clear that the results of large core overlap are also important.

For the multifilament runs, important qualitative factors also emerged. Again symmetry and zeros played a major role, but in addition, discontinuities in the area of the ring itself also seemed important.

Some attempts to gather quantitative data were made. An important quantitative feature is the total energy of the ring. In order to make a fair comparison and respect the observation that effects of core overlap are important, one requires a method which will work with any core function and account for the effect of core overlap on the energy. The methods of which we are aware for energy evaluation violate one or both of these requirements. Two naive methods which asymptotically satisfy these criterion were tried. The energy integral was computed on a grid, and the energy integral was computed as an integral over the blob locations using the volume estimations of Chapter 8. However, as soon as significant mixing of cores occurred, the grid could no longer resolve the wave numbers at which energy appears. The integral over the blobs also does not appear to give reasonable values for highly irregular configurations. It is suspected that this is due to the singularity of the integral and the methods failure to account for it. Since the qualitative data was so interesting, work on improving these methods was not pursued. At the end of chapter 4, future attempts at accurately gathering this data are discussed.

In addition to the energy integral, the helicity, impulse, total vorticity, angular momentum and volume were computed both on a grid and as integrals over the blobs. As evaluated, the helicity of the initial data is zero in all cases. This value was preserved throughout all computations. The impulse direction and value as computed were preserved, past the point in time when results became qualitatively different. The angular momentum and volume as computed are also preserved past the point where qualitative difference appeared.

The minimum, maximum and average divergence of the field $\Omega(x) = \sum \omega_i f_\delta(x - x_i)$ was computed. These values were small until the onset of significant deformation of the original grid. The difference between Ω and its divergence free projection behaved similarly. The angle between the vortex elements and the fields Ω or ω also showed similar behavior. In fact, it rapidly goes from near zero to nearly π as soon as the configuration of points becomes more complex than in the initial data. The growth in $\nabla \cdot \Omega$ turn out to be important in the case of filament methods.

While in many cases, quantitative results seem to be inconclusive, or impossible to obtain, the qualitative factors seem to point to real differences between the algorithms and to flow structures which cause difficulty for the methods. Therefore, these qualitative results are emphasized. It is clear that these observations should be a point of departure for more precise and quantitative comparisons and analyses. It may also be possible to use these observations as a basis for further theoretical work.

1.5 Conclusions

- For the unconstrained VTE, $\nabla \cdot \omega$ is a material quantity.

  If the constraint, $\nabla \cdot \omega = 0$ is removed from the VTE, $Q = \nabla \cdot \omega$ becomes a material quantity. In the viscous case, it obeys $D_t Q = \frac{1}{R}\Delta Q$. This implies that it obeys a maximum principle and that any solution of the unconstrained VTE which is a limit of viscous solutions to the unconstrained VTE also obeys a maximum principle. $\nabla \cdot \omega$ acts as an entropy function for the unconstrained VTE.

- $\nabla \cdot \omega$ computed by the filament method does appear to obey a maximum principle

  In the filament method, the vorticity is often taken to be $\Omega(x) = \sum f_\delta(x - x_i)\omega_i$, where f_δ is an approximate delta function and ω_i is the strength of the ith particle. Numerical evidence indicates that for this definition of vorticity, the computed Ω has divergence which grows over time, which is not consistent with the unconstrained VTE. This indicates that if the convergence of the filament method is to be proved, then the divergence free part of Ω, $\omega(x) = \mathcal{P}(\Omega)(x)$, must be used as the vorticity. While this is not an obstacle to theory, it calls into question the analogy between physical vortex numerical filaments and the filaments used in the vortex method.

- The behavior of vorticity near its zeroes and zeroes of its torsion is important.

  Particularly in the single filament ring computations, the response of the method to the creation of new zeroes of the vorticity and/or its torsion is highly variable and in large part determines the outcome of the calculation. In backwards vortex method computations problems arise when the vortex method is asked to destroy existing zeroes. On the basis of numerical evidence, we conjecture that these zeroes are the location of a source of entropy and are related in some way to an entropy function. Zeroes of vorticity which correspond to zeroes of torsion appear in many computations. The motion of vortex lines near such a point is analogous to the motion of fronts in two dimensions.

- The conservative nature of the VTE is important to numerical schemes.

  The VTE has several hyperbolic properties whose effects become visible in numerical computations. The apparent discontinuities in the cross-section of a vortex ring in one ring computation

of chapter 11, are similar to those seen in solutions of the shallow water equations. The fact that the divergence of vorticity is both material and an entropy function, should force non-solenoidal vorticity to become more and more solenoidal. This is not seen in vortex filament computations but is seen in finite difference computations. The behavior of vorticity near a zero shows hyperbolic properties in several different ways. Although it has not been shown that discontinuities in the vorticity must form in the absence of numerical errors, there is evidence that they will form even with the smallest of compressible velocity errors.

- The geometric nature of the VTE is important to numerical schemes.

  Stretch schemes based on the transpose equation seem to be more successful than naive stretch schemes. The transpose equation is an Euler-Lagrange equation for a Lagrangian equal to the helicity. In addition, the derivative of its flow map is the map on the tangent spaces, while this is not true of the usual VTE. The behavior of the vorticity near a zero is also dominated by geometric concerns and the analogy to the geometry of moving fronts.

- The nature of the mixing introduced by the core function affects results.

  This conclusion is supported by the large dependence of vortex method results on the particular core function chosen and the "core unmixing phenomenon". While the core function provides some smoothing, the exact nature of this smoothing is not fully understood. In particular, Galerkin type subspace analysis is difficult for the vortex method. The map between the vorticity and the vortex particle weights is numerically non-unique.

- Straighforward extension of the vortex method to 3d is ineffective.

  In particular, straightforward extensions like those used in chapter 2, will not reproduce the success of the two dimensional vortex method in underresolved computations.

- Certain 3d flow structures are not handled properly by the vortex method.

  In particular, in the single filament ring computations of chapter 4, the behavior of the vortex method near zeroes of vorticity and in the presence of small scale and helical structures is highly variable and depends on the exact form of the stretch term and the exact form of smoothing which is used. This variability is also seen in multifilament ring computations.

- The results obtained by adding a diffusive term to the vortex method are inconclusive.

  Several computations of the evolution of vortex rings with viscosity, were made. In order to draw conclusions, however, a more extensive parameter study is necessary. While it appears that adding some viscosity does smooth solutions, it is not known how much viscosity is necessary to stabilize the method. The computer requirements for a complete parameter study are daunting, especially if one must use the $\mathcal{O}(N^2)$ method, which appears to be the case due to the interaction of fast summation methods with the core function. In particular,

the core function determines the smallest box size which in turn determines the cross over point.

- Volume estimation will work in combination with deterministic diffusion schemes

  The volume estimation scheme which resulted from work concerning resampling seems to work well in conjunction with deterministic diffusion schemes. Volume estimation allows the assignment of volumes to particles which are accurate enough for the evaluation of diffusion and stretch, However, the assigned volumes are not accurate enough to be used for the evaluation of singular integrals such as those which define the energy. Volume estimation allows the use of deterministic diffusion schemes near the boundary, without imposing the requirement that particles be given a volume as they are created.

- The boundary algorithms of Chapter 12 are promising but, the results are still inconclusive.

  It has been show that the boundary algorithms of Chapter 12 move vorticity off of the boundary and into the flow. It has also been shown that the second of the new algorithms has the property that at periodic intervals both the no slip and no leak boundary conditions are satisfied. The sort of parameter study, or theoretical work which would truly verify these algorithms has not been done. Again, the computer requirements are quite large.

- More theoretical work is necessary to extend the vortex method to three dimensional flow.

  Given the computer requirements for brute force numerical experimentation with the vortex method, it may be more prudent to concentrate on developing variants of the vortex method which attempt to deal with the issues raised so far, before attempting further numerical work.

Chapter 2

First Implementation and Experiments

A three dimensional vortex method was implemented for flow around an arbitrary body. This code was intended to run efficiently on serial vector processors (such as the Cray C90) and RISC based workstations, such as the IBM RS/6000 and DEC ALPHA). In this chapter, the algorithm used in that code, and the results from it are discussed. Our method consists of several separate algorithms which are combined by the standard time splitting technique. They are:

- a free space inviscid vortex method
- a potential solver to enforce the no leak condition
- a vorticity creation step to enforce the no slip condition
- a random walk method for the heat equation

2.1 The Free Space Method

2.1.1 description of the method

The initial implementation of the free space method was based on algorithm A from Anderson and Greengard [1]. In this method, the particle positions, x_i are updated according to

$$\partial_t x_i = \sum_j K_\delta(x_i - x_j)\omega(x_j)h^3 \tag{2.1}$$

while the particle weights are updated according to

$$\partial_t \omega(x_i) = \omega(x_i) \cdot \sum_j \nabla_x K_\delta(x_i - x_j)\omega(x_j)h^3 \tag{2.2}$$

The particle weights ω_i, are identified with $\omega(x_i) * h^3$. An evolution equation for the weights may be derived by multiplying the vorticity evolution equation for $\omega(x_i)$ by h^3 and then making the identification. [1] further recommends computing $\omega_i \cdot \nabla u$ by explicit differentiation of the sum u_{vor}.

$$u_{vor}(x,t) = \sum_j K_\delta(x - x_j)\omega_j \tag{2.3}$$

$$\omega_i \cdot \nabla u = \sum_j \omega_i \cdot \nabla(K_\delta(x - x_j)\omega_j) \tag{2.4}$$

The equations above are a set of ODEs for the evolution of the particle positions and weights.

$$\partial_t x_i = u_{vor}(x_i, t) \tag{2.5}$$

$$\partial_t \omega_i = \omega_i \cdot \nabla u_{vor}(x_i, t) \tag{2.6}$$

$$\tag{2.7}$$

The the second order smoothing function of Beale and Majda,

$$f(r) = \frac{3}{4\pi} e^{-r^3} \tag{2.8}$$

was used as the basis for the smoothing function. Heun's method (second order Runge-Kutta) was used to integrate the ODE.

2.1.2 Tests of the implementation

In order to be confident that this method had in fact, been implemented several tests and diagnostics were run. The gradient of the induced velocity field, u_{vor} computed by numerical differentiation and compared to the calculated exact value. The curl of the induced velocity field was taken numerically and compared to an exactly calculated value obtained by formally taking the curl of the expression for the velocity. The ode's were integrated with Heuns method and the evolution of four very long parallel sticks, placed in a circle was calculated. One would expect these to orbit each other just as two dimensional blobs do. The evolution of an unperturbed single filament vortex ring was calculated. One expects this ring to propagate and retain its shape almost exactly for a long time before numerical noise perturbs it enough that it becomes unstable.

2.2 Approximating Viscosity By The Random Walk

The random walk method [11] has been used in two dimensional vortex methods and extends in a straightforward manner to three dimensions[13]. Let ν denote the the viscosity of the fluid[2]. The diffusion alone is described by, $\partial_t \omega = \nu \Delta \omega$[18]. To find the solution in terms of the vortex elements at time, $T + \Delta T$, given the vortex elements at time, T, the following procedure is used[18].

- for each vortex element
  - choose a random vector from a three dimensional Gaussian distribution with mean zero and variance $4 * \nu \Delta T$
  - add the chosen vector to the position the element.
- end

When there is a body present in the flow, the random walk method must be slightly amended because a random step might take an element inside the body. If this occurs, or would occur, the element is reflected by the body, as if it were a perfectly elastic ball which collided with the body. This is consistent with the vortex sheet method[12].

2.3 The No Leak Boundary Condition

In order to enforce the no leak condition, a flow must be constructed which, when added to the vortex flow, causes $u_{total} \cdot \hat{n} = 0$ where $\hat{n}$ is the unit normal to the surface [18]. Recall that the gradient of a scalar may be added to u without affecting the vorticity since $\nabla \times (u + \nabla\phi) = \nabla \times u$ In order that the flow remain divergence free, It is also required that $\nabla \cdot (u_{vor} + \nabla\phi) = 0$ in the fluid. The boundary condition is equivalent to $(u_{vor} + \nabla\phi) \cdot \hat{n} = 0$ at each point on the boundary[11]. These stipulations taken together with the requirement that the velocity vanish at infinity are Laplace's equation in the interior with boundary conditions on the surface and at infinity[11].

$$\Delta\phi = 0 \tag{2.9}$$

$$\nabla\phi \cdot \hat{n} = -u_{vor} \cdot \hat{n} \tag{2.10}$$

$$\nabla\phi(\infty) = 0 \tag{2.11}$$

$$\tag{2.12}$$

In order to solve this equation for ϕ, a *panel method*, is used. It would be very difficult to give a better description of panel methods that the one given by Hess and Smith in[30]. The review of Hess and Smith has an extensive bibliography. Only the most basic notions will be described here.

Laplaces equation with boundary conditions is equivalent to a Friedholm integral equation of the second kind for a source distribution σ on the boundary which give rise to the potential ϕ, by a $\frac{1}{r}$ potential law. $\phi(x) = \int_{boundary} \frac{\sigma(y)}{|x-y|} dy$ The boundary condition then becomes an integral equation on the boundary for σ, namely,

$$\int_{boundary} \frac{\sigma(y)}{|x-y|} dy = -u_{vor} \cdot \hat{n} \tag{2.13}$$

The surface of the body is divided into quadrilateral surface elements called *panels* It is assumed that σ is constant on each panel. The value of σ on panel i is called σ_i. The value of the integral

above is then computed at one point,p_i, on each panel, called the *control point* of panel i. This gives rise to a matrix equation for the σ_i Define

$$A_{ij} = \int_{panel_j} \frac{1}{p_i - x} dx \tag{2.14}$$

This can be calculated exactly. Hess and Smith give a closed form solution for

$$\int_{panel_i} \frac{1}{x - y} dy \tag{2.15}$$

over any quadrilateral, or triangular panel. An approximation to the integral equation resulting from the boundary condition is given by

$$A_{ij}\sigma_j = -u_{vor}(p_i) \cdot \hat{n}(p_i) \tag{2.16}$$

$$A\sigma = -u \cdot \hat{n} \tag{2.17}$$

This matrix equation can be solved for σ_i, which can then be used to evaluate ϕ and $\nabla\phi$

Note that the matrix A is fully determined by the body itself, while the right had side contains all the information about the external flow needed to calculate the boundary condition. If A is small enough, and the problem is to be solved many times, with different right hand sides, it is efficient to first build an LU decomposition of A (which is fully determined by the body alone) and then simply do the back substitution for each right hand side. In this algorithm, a potential flow will be required at each time step in order to enforce the no leak condition. Given the body, the matrix A is constructed. The LU decomposition is computed and stored using the routine SGECO from LINPACK. Each time a potential is required in order to satisfy the no leak condition, the right hand side of the matrix equation is computed and then the routine SGESL from LINPACK is called to generate the strengths σ_i. This gives rise to a velocity field,

$$u_{pot}(x) = -\nabla \sum_i \sigma_i \int_{panel_i} \frac{1}{x - y} dy \tag{2.18}$$

such that

$$u_{tot} = u_{vor} + u_{pot} \tag{2.19}$$

satisfies the no leak condition.

2.3.1 Tests of the Potential Solver

Several efforts were made to verify that the code actually implements the panel method. The solution was evaluated at each point on the boundary and compared to the boundary condition. The Laplacian of the solution was evaluated on a grid in the region of the flow and compared to zero. The solution was evaluated on a grid near the body and interpolated values at the boundary were compared to the boundary condition. These tests were run for a body containing no corners, such as a sphere, and a body with corners, such as a cube. They were also run for bodies made up of quadrilateral panels only, triangular panels only, and a combination of quadrilateral and triangular panels

2.4 The No Slip Condition

For viscous flows the condition which must be satisfied by a body in the fluid is that the velocity of the fluid on the body is zero relative to the body[2].

The no leak condition may be expressed as

$$u \cdot \hat{n} = 0 \tag{2.20}$$

For viscous flows, it is required in addition that for any $\hat{\tau}$ tangent to the body,

$$u \cdot \hat{\tau} = 0 \tag{2.21}$$

To arrive at an algorithm for enforcing this condition, one considers what would happen to the vorticity if the velocity on the boundary were forced to be zero. Let v be a given velocity field such that $v = u_{tot}$ in the interior, and $v = 0$ on the boundary. One then computes the vorticity distribution which corresponds to v. $\nabla \times v$ is equal to the curl of u_{tot} except on the boundary, where the curl is singular.

$$\nabla \times v = \nabla \times u_{tot} + \delta(d)\hat{n} \times u_{tot} \tag{2.22}$$

where d is the distance to the boundary. A singular layer of vorticity, a three dimensional vortex sheet at the boundary, is added to the vorticity distribution in order to satisfy the boundary condition.[11][12] This argument motivates the following algorithm due to Chorin [11].

- for each panel
  - Evaluate the velocity, $u = u_{vor} + u_{pot}$ at the control point, p_i
  - Compute $\lambda = \hat{n} \times u$
  - Compute $\omega = A \cdot \lambda$, where A is the panel area.
  - Create a vortex element or several vortex elements with total strength ω
  - for each element just created
    - Choose a random number b from a Gaussian distribution.
    - place that element at $p_i + |b|\,\hat{n}$
  - end (for each element)
- end (for each panel)

2.5 First Results

2.5.1 Initial Computation

Using this code, an attempt was made to compute flow around a unit cube, with an incoming flow of speed 1. The results of this calculation were very disappointing. The calculation

quickly becomes intractable due to the growth of the vorticity and thus the number of vortex elements necessary to represent it. The machine being used could compute a time step for approximately 5000 elements in several hours. Even with very small time steps, the calculation becomes intractable by T = 0.2. One expects the stretching of vorticity to give areas of exponential growth of vorticity and the number of elements. However, one expects these regions to be localized at least in space, if not in time as well. One also expects these regions to be in the wake. Several aspects of the observed growth were inconsistent with these expectations.

- The growth appeared to be exponential almost everywhere.
- The growth started well before any significant wake had developed.
- The growth started well before time 1.0
  - This means that it had to be due only to boundary layer effects, whereas much of the boundary layer flow should be initially smooth.
  - Since the growth appears at this time, it is associated with the impulsive startup of the system. This startup is a singular perturbation (the creation of vortex sheets on the boundary of a potential flow) to a smooth flow which is expected to have (initially) smooth results.
- Reduction of time step size did not seem to have any effect the growth.

These observations suggested that the growth could be due to an instability of the algorithm, not instability of the flow.

2.5.2 Conservation Form

Due to the fact that this exponential growth appeared almost everywhere and the fact that it was associated with a failure to properly handle a singular perturbation, it was hypothesized that there could be a conservation law or a hyperbolic subsystem of the vorticity transport equation causing these problems. It was found that the vorticity transport equation has a conservative form. (See 5 With a bit of vector gymnastics, the VTE can be rewritten as

$$\partial_t \omega_i + \nabla \cdot (u\omega_i - \omega u_i) = 0 \tag{2.23}$$

or

$$D_t \omega_i = \nabla \cdot (\omega u_i) \tag{2.24}$$

2.5.3 Making Use Of Conservation Form

Defining Conservation Form In A Lagrangian Setting

Since the VTE has a conservative integral form, an algorithm in conservation form may be easier to understand and stabilize than one that is not. The basic idea behind conservation form is

to use the divergence theorem on the integral equation and then approximate the fluxes across the boundary of the region of integration. This gives a numerical flux which is then consistent with the continuous flux, and thus consistent with the integral equation.[36]

An algorithm in "conservation form"

Given a vortex stick, it could be considered as a right circular cylinder of constant vorticity, parallel to the axis of the cylinder. By applying the divergence theorem to the integral form of the VTE, one obtains a semi-discrete equation

$$D_t \int_C \omega_i = \int_C \nabla \cdot (\omega u_i) \tag{2.25}$$

The divergence theorem implies

$$D_t \int_C \omega_i = \int_{\partial C} u_i \omega \cdot dA \tag{2.26}$$

The surface of the cylinder is considered in three sections, the cap away from which the vorticity points, called cap_p, the cap toward which the vorticity points, called cap_m, and the remaining cylindrical section which will be called S

$$D_t \int_C \omega_i = \int_{cap_p} + \int_{cap_m} + \int_S u_i \omega \cdot dA \tag{2.27}$$

The value of the integral over S is zero, since ω is parallel to the surface and hence normal to the surface normal.

$$D_t \int_C \omega_i = \int_{cap_p} + \int_{cap_m} u_i \omega \cdot dA \tag{2.28}$$

If it is further assumed that the velocity over each end cap is constant, that the area of each cap is A, and the length of the cylinder is l, one may conclude that

$$D_t \int_C \omega_i = A\omega(u_i(cap_p) - u_i(cap_m)) \tag{2.29}$$

$$D_t(Al\omega_i) = A\omega(u_i(cap_p) - u_i(cap_m)) \tag{2.30}$$

$$D_t(\omega_i) = \frac{\omega_i(u_i(cap_p) - u_i(cap_m))}{l} \tag{2.31}$$

A large number of assumptions have been made in order to derive this semi-discrete form. They are listed together below.

- The vorticity is confined to right cylinders of circular cross section.
- The vorticity in each cylinder is constant.
- The vorticity in each cylinder is parallel to the axis of the cylinder.
- The velocity is constant over each cylinder end cap.

$$D_t(\omega_i) = \frac{\omega_i(u_i(cap_p) - u_i(cap_m))}{l} \tag{2.32}$$

Numerical Tests

In the program which implements the vortex method which has been described, the exact pointwise evaluation of the stretch term was replaced by the discrete version just derived. Both of these discrete stretch terms are consistent with the VTE. However, after this modification, the number of elements which was previously reached by $T = 0.2$, was not reached until $T = 2.0$ However, the growth is still exponential and widespread. While both computations appear to be unstable, the second seems to be less so.

Properties of the "Conservative" Scheme

The scheme just derived is the same as the stick scheme of Chorin in the case where the initial configuration of sticks is connected [14][13]. This scheme has a number of desirable properties.

- Connected sticks remain connected which implies that $\nabla \cdot \omega$ remains bounded if the initial stick configuration is connected.
- Locally, the scheme is consistent to first order with the integral equation.

It also has some undesirable properties.

- Geometric Problems
  - Since each cylinder is right, the end caps of connected sticks do not coincide, leaving small gaps and making flux calculations suspect at best.
  - The cylinders do not tile space.
  - These problems are not bounded as the flow evolves.
- A volume whose boundary is initially a right circular cylinder does not remain so as the flow evolves. The best one could hope for is that the boundary remains a ruled surface with end caps.
- Numerical tests indicate that if $\nabla \cdot \omega$ is small at some time, but the sticks are unconnected, $\nabla \cdot \omega$ can and does become large at a later time.

2.6 Interim Conclusion

Given the difficulties with this initial implementation, it was decided that the most likely cause of problems was somewhere in the free space vortex method. More tests of the free space vortex method were necessary to understand what was happening to this more complex calculation. The existence of a conservation form also indicates that more theoretical work may help in understanding the instability of this initial calculation. Subsequent chapters will deal with these issues.

Chapter 3

Stretch Schemes

In this chapter, various discrete versions of the VTE stretch term are explored. Each has a different motivation and will be seen to have different properties. They will be used and compared in chapters 4 and 11 on free space calculations. Several schemes which have been used in previous calculations, will be presented along with two new schemes. These discrete stretch methods fall into two major classes, those which are based on connected filaments of vorticity, and those which are not.

3.1 Filament Schemes

A vortex filament is a curve, $\gamma(s)$, which is tangent to the vorticity at all points[18]. Vortex filaments have several useful properties. The circulation around a vortex filament is constant and if s is an arc length parameterization of γ, then $\omega(\gamma(s)) = \Gamma(\gamma)\gamma'(s)$, where Γ does not depend on s but may depend on the curve γ itself [18]. Infinitesimal vortex lines are transported materially by the flow[18]. That is, they are transported like tiny sticks which stretch with the fluid. This fact is called the Helmholtz theorem[18]. By integrating this result, one finds that vortex filaments are transported materially by the flow. These observations form the basis for several ways to discretize the stretch term. The schemes of Chorin [13][14] and of Knio and Ghoniem[33] will be briefly described

3.1.1 Chorin's Filament Stretch and Helmholtz Stretch

The stretch scheme used by Chorin in his vortex filament calculations makes use of the fact that $\omega \cdot \nabla u$ is a directional derivative and can be evaluated as such. It is also based on the Helmholtz theorem.

This method uses vorticity supported on blobs and transports these these blobs as material sticks in the flow. The total vorticity of a blob is given by its circulation, which does not vary over

time, times its length, which does vary over time. $\omega = \Gamma l$, where l is the vector extent of the stick[14][13].

More formally, if γ is an integral curve of ω and s is its arclength parameterization,

$$\omega \cdot \nabla u = \frac{d}{ds}(u(\gamma(s))) \tag{3.1}$$

If $\hat{\omega}_i$ is the unit vector in the direction of ω_i at x_i, then $\omega \cdot \nabla u$ can be approximated by

$$\omega \cdot \nabla u \approx |\omega| \frac{u(x_i + \frac{h\hat{\omega}}{2}) - u(x_i - \frac{h\hat{\omega}}{2})}{h} \tag{3.2}$$

Assume the vorticity is supported on sticks define x_i to be the center of stick i and h to be the length of the stick. The scheme then becomes

$$\frac{D\omega}{dt} = \omega \cdot \nabla u \approx |\Gamma l| \cdot \frac{u(x_i + \frac{l}{2}) - u(x_i - \frac{l}{2})}{|l|} \tag{3.3}$$

$$\frac{Dl}{dt} \approx u(x_i + \frac{l}{2}) - u(x_i - \frac{l}{2}) \tag{3.4}$$

There are several interesting facts to note about this scheme

- Sticks which are initially connected, will remain so for all time.
- The algorithm still makes sense for unconnected sticks.
- For stick elements, this is the scheme for which a conservation form was derived in chapter 2
- This scheme can be implemented by simply moving the endpoints of each stick and using the new locations to compute a new length, and hence a new vorticity for the stick. Thus implemented, it requires only one evaluation of the velocity per particle[13].
- This scheme has some smoothing due to the use of finite differences, but the smoothing effects only one direction.
- This scheme approximates a vortex filament by placing marker particles on that filament and then constructing a piecewise linear approximation to the filament from the particles. This approximation is used to evaluate the tangent to the filament.

3.1.2 Knio and Ghoniem's Scheme

The stretch scheme used by Knio and Ghoniem [33] is similar to that of Chorin in the fact that it uses connected filaments, however, instead being supported on sticks, the vorticity is supported on blobs along a filament. In addition a second order approximation to the tangent vector of the curve is used. The filament is advected as a material object[33]. Marker particles are placed along the filament and derivatives are approximated using the mean value theorem.

$$\gamma'(x_i) \approx \frac{\gamma(x_{i+1}) - \gamma(x_{x-1})}{x_{i+1} - x_{i-1}} \tag{3.5}$$

The vorticity is considered to be a circulation times a length as in Chorin's scheme $\omega_i = \Gamma l_i$ and a discrete evolution equation is derived for l[33].

$$\frac{Dl_i}{dt} = l_i \cdot \nabla u = \frac{u(x_{i+1}) - u(x_{i-1})}{2} \tag{3.6}$$

Note that this is equivalent to moving the curve by moving the marker particles and then reconstructing the vorticity through a second order approximation to the derivative, γ' along the filament. Of note about this scheme are

- Filaments must be used and remain connected for all time.
- The scheme can be implemented by moving marker particles on the filament, just as efficiently as Chorin's scheme[33].
- Any smoothing or aliasing which results from the finite differencing, only effects one direction.

3.2 Unconnected Singular Schemes

These are schemes in which the objects on which the vorticity is supported are not considered to be connected in filaments. This is more convenient for calculation of flow past bodies, since neighbors need not be determined for the vorticity which is generated on the boundary. However it is possible that problems could arise due to the vorticity not being divergence free.

If stretching takes one set of particle strengths directly to another, without computing the smooth vorticity field, this is called singular stretch, because one might imagine a vorticity distribution which is made up of delta functions at each particle with vorticity equal to the weights. If the smooth field is calculated on the way to updating the strengths, the scheme will be called smooth.

3.2.1 Singular ω times ∇u Stretch

This is the most naive possible implementation of the stretch term. A finite difference approximation $\nabla_h u$ to ∇u is computed [1]. Next, one computes

$$\frac{D\omega_i}{dt} = \omega_i \cdot \nabla_h u \tag{3.7}$$

To motivate this, one starts with the VTE at x_i,

$$\frac{D\omega(x_i)}{dt} = \omega(x_i) \cdot \nabla u(x_i) \tag{3.8}$$

Assuming that $\omega_i = V_i \omega(x_i)$, and using the fact that $\nabla_h u$ is close to ∇u, multiplying both sides of equation 3.8 by V_i gives equation 3.7 [1].

In the continuous case, this method is equivalent to the stick scheme or the filament schemes, but can be very different in practice. In particular, connected filaments need not remain so, and

any smoothing or aliasing which results from the use of finite differences is now present in all three dimensions, not just the direction of the vorticity.

3.3 Singular ω times ∇u^T

This is similar to the previous scheme, except the stretch operator is taken from the transpose equation instead of the VTE (see section 5.4)[43][25]. A finite difference approximation $\nabla_h u$ to ∇u, is computed and then the stretch for particle i is computed as

$$\frac{D\omega_i}{dt} = \omega_i \cdot (\nabla_h u)^T \tag{3.9}$$

Again this scheme is equivalent to the others in the continuous case, but is different in the discrete case. In particular, since the VTE and the transpose equation are connected through an identity which is true only when $\nabla \cdot \omega = 0$, they may behave very differently in the case that $\nabla \cdot \omega \neq 0$[43]. Also, note that both this method and the previous one are computationally very costly. Each requires six evaluations of the velocity field in order to compute $\nabla_h u$ if derivatives are approximated by central differences.

3.3.1 Exact Stretch

The "exact" evaluation of the stretch term which is suggested by Anderson and Greengard [1] was the early numerical experiments of Chapter 2. ∇u can be calculated by differentiating the sum which forms u

$$\omega_i \cdot \nabla u = \nabla \sum_j \omega_j K_\delta(x_i - x_j)\omega_i \cdot \nabla u = \sum_j \omega_i \cdot \nabla \left(\omega_j K_\delta(x_i - x_j)\right) \tag{3.10}$$

Note that the terms of the sum can be evaluated analytically.

3.4 Unconnected Smooth Schemes

These schemes use unconnected objects, but they operate on the vorticity, not directly on the particle strengths. The strengths are updated by a volumetric resampling procedure. (see chapter 8)

3.4.1 Smooth Omega Times ∇u

Recall that in three dimensions, there are two objects which may be considered as the vorticity. The non solenoidal $\Omega(x)$ and the divergence free $\omega(x)$ given by

$$\Omega(x) = \sum_i \omega_i f_\delta(x - x_i) \tag{3.11}$$

$$\omega(x) = \sum_i \omega_i Z_\delta(x - x_i) \tag{3.12}$$

where $\omega_i Z_\delta$ is the divergence free part of $\omega_i f_\delta$[28]. The motivation here is to combine the motivation for the singular stretch term with some smoothing and some projection. The smoothing will come from using the smooth vorticity, while the projection will come from stretching only the divergence free part. $\omega(x)$ is used as the vorticity. The smooth stretch is computed as

$$\frac{D\omega}{dt} = \omega(x) \cdot \nabla_h u \tag{3.13}$$

Volumetric resampling is used to update the particle strengths given the smooth stretch

$$\frac{D\omega_i}{dt} = V_i \omega(x) \cdot \nabla_h u \tag{3.14}$$

where V_i is the volume assigned to particle i.

3.4.2 Smooth Omega Times ∇u^T

This is simply the argument of the previous subsection applied to the transpose equation. The algorithm is then,

$$\frac{D\omega_i}{dt} = V_i \omega(x) \cdot (\nabla_h u)^T \tag{3.15}$$

Chapter 4

Ring Calculations I

There are several different types of vortex methods that have been used or suggested for free space problem. They may be classified along several different axes.

4.1 Connected versus Unconnected Schemes

Vortex methods are divided into two broad categories, *Filament Schemes* and *Unconnected Schemes*. Filament schemes keep the vortex elements as connected filaments. Thus, each element has a forward neighbor and a backward neighbor along the filament to which it belongs. Unconnected schemes on the other hand, do not keep such neighbor information and allow each vortex element to evolve without reference to its neighbors. A number filament and unconnected stretch schemes are described in chapter 3.

4.2 Blobs versus Sticks

Vortex methods are also divided into two categories in an other manner, *Stick schemes* and *Blobs schemes*. Stick schemes have vortex elements which are one dimensional. Each element is a one dimensional stick of vorticity which is then smoothed by convolution. Blob schemes have zero dimensional or point elements which are also convolved with a suitable core.

4.3 The Parameter Space

The following is a list of the parameters which will be studied through an inviscid computation of the evolution of a single filament. After each possible value for a parameter are abbreviations which will be used in tables later.

- Algorithm Type

  - Filament
    * Chorin filament (C)
    * KG filament (KG)
  - Unconnected
    * Helmholtz Theorem based stretch (stick) (Helm)
    * exact stretch (exact) ($\omega_i \cdot \nabla u$)
    * grad stretch (grad) ($\omega_i \cdot \nabla_h u$)
    * grad transpose (trans) ($\omega_i \cdot \nabla_h u^T$)
    * smoothed stretch (smooth) ($\omega \cdot \nabla_h u^T$)
- Element Type and Core Function
  - Blob
    * Second Order (2)
    * Fourth Order (4)
    * Gaussian (G)
  - Stick
    * Stick With Chorin's 2d Core (C)
    * Stick With 2d Core (2d)
    * Stick With 3d Core (3d)
- Cutoff
- Time Step
- Number of Points

The parameter space defined by this classification of algorithms is enormous. If the cutoff, time step, and number of points, are ignored and the core function is restricted to one of the three listed above, there are $8 * 2 * 3 = 48$ different algorithms. A number of representative algorithms will be explored. Table 4.1 describes each of the filament algorithms which will be used.

Table 4.2 is a table of unconnected algorithms which will be used. Algorithms will be named in two ways, by algorithm number in the tables above and, by description as a point in the parameter space. For example algorithm 2 is KG.blob.2 or (KG,blob,2)

4.4 Code Details

Two separate programs were implemented. One implements the filament algorithms and the other implements the unconnected algorithms. These implementations will be referred to as

Table 4.1: Table of Filament Algorithms

Filament Algorithms			
Run	Stretch.	Element Type	Core
1	KG	Blob	2
2	KG	Blob	4
3	KG	Blob	G
4	C	stick	C
5	C	blob	2

Table 4.2: Table of Unconnected Algorithms

Unconnected Algorithms			
Run	Stretch.	Element Type	Core
7	Helmholtz	blob	2
8	exact	blob	2
9	$\omega_i \cdot \nabla_h u$	blob	2
10	$\omega_i \cdot \nabla_h u^T$	blob	2
11	$\omega \cdot \nabla_h u^T$	blob	2

the connected and unconnected codes, respectively. Both implementations use the C programming language. Core functions and their derivatives and integrals are implemented as macros, which allowed them to be easily switched at compile time. This was also done is such a way that two methods differing only in core function execute exactly the same code with the exception of the core function macros. Different stretch algorithms may be chosen at run time. This is implemented by indirect function call. Again, this is done in such a way that two algorithms which differ only in their computation of the stretch term execute exactly the same code except for the subroutine which computes the stretch of an element. First and second order Runge-Kutta methods for integration in time were used in the filament code, while the unconnected code has first, second and fourth order Runge-Kutta solvers. It was found that fourth order integration was sometimes necessary for the ($\omega \cdot \nabla_h u^T$,blob,*) algorithms. The result of these details is

- Between algorithms implemented by the same code, there is a great deal of shared code. In fact, everything but the core function and the subroutine which computes the stretch is shared.
- Between algorithms implemented by different codes, there is no shared code.

In exploring so many different variations, one must have some way of being sure which variation one is exploring at any given time. Two different approaches were used, one for compile time selectable options and another for run time selectable ones. For every compile time switch, an ascii string, readable by the unix "what" command was inserted into the code describing the choice that was made, and the version number of the files used to build the program. Thus, at any time, one

might find out which executable version one is running, using the unix "what" command. Run time selectable values are listed as keyword value pairs in a parameter file. The first item in the diagnostic output file is a copy of the parameters with which the run was made.

4.4.1 Machines, parallelism and efficiency

These codes run efficiently on the following uniprocessors, sun sparc models ELC,1,2 and 10, IBM rs6000 models 220, 550, 560, dec alpha 3000-400, cray XMP, cray YMP, cray C90, cray 2s, sgi indigo (mips R4000), dec decstation (mips R3000). Of the uniprocessor workstations, the code performed best on the IBM rs6000/560 which was followed closely by the dec alpha 3000-400. The unconnected code has been parallelized for the Cray XMP,YMP,C90 and the Cray 2S, achieving a speedup of 15.8 on a 16 processor C90. The unconnected code has also been parallelized for a multiprocessor sun sparc 10, achieving a speedup of 3.95 on a 4 processor machine.

4.5 Code Verification

In order to verify that the program implements the vortex method, several test were performed.

1. The velocity and strain induced by a single vortex element on several points outside the core was computed by hand. The value of the subroutine which computes the influence of one element was then compared to the hand computed values.

2. The evolution of four very long parallel vortex sticks placed in a circle was compared to the evolution of four two dimensional vortex blobs, which it should approach as the length becomes infinite.

3. The evolution of an unperturbed single filament ring was computed with each method. One expects it to continue unperturbed for a very long time.

4. The velocity of propagation of a single filament ring was compared to that given by the formula of Saffman[33]. For a ring of radius R, with a Gaussian core of standard deviation σ this is $V = \frac{\Gamma}{4\pi R}\left(log(\frac{8R}{\sigma}) - .558\right)$ [33].

5. The pictorial results of [33] for a single filament ring were duplicated. These are shown as run reproduce.1

6. Two separate programs were implemented, one for filament algorithms, and another for unconnected algorithms. However, if an algorithm which uses the Helmholtz based stretch scheme has initially connected elements, they will remain connected. Therefore, in this case, such an algorithm reduces to the Chorin filament algorithm, with blob elements. The two programs

were in qualitative agreement when compared in this manner. They are not in exact agreement due to slight differences in the way the splitting of large elements are handled. Compare, for example, runs base.5 and base.7. They match if one is rotated 180 degrees about the z axis. The initial data also matches itself under this rotation.

The results of each of these tests are in good agreement with the corresponding comparison values, with one exception. Method 8, unconnected blobs with exact evaluation of the stretch term did not pass test number 3. It could not propagate a ring significantly longer than $T = 10$ without becoming unstable. However, the stretch which it does calculate is the same as a hand computed value.

4.6 Inviscid Single Filament Computations

Table 4.3 is a list runs used in this chapter. In each case the ring has radius 1.0, and circulation 2.0. Unless otherwise specified, second order integration in time was used. Note that for such a run, the velocity of propagation of the ring is about 0.5, so $\Delta t = 0.005, \Delta x = 2\pi/128 = .05$ gives a CFL number of 0.05 for the advection. Thus with the largest Δt and the smallest Δx used, the CFL number will be less than 0.5, even if the velocity should grow ten fold.

For each algorithm, a base run was made with a cutoff of 0.3 and $\Delta t = 0.005$, $\Delta x = 0.1$ This implies a CFL number of 0.025. Next, a run with Δt of half its value in the base run was made. Next, a run with Δx equal to half the value in the base run was made. These runs were chosen for display because they show the range of behavior which we saw in all of the runs. Runs were also made with perturbations at wave numbers of 1 through 12. The wave number 6 runs showed interesting behavior earlier than the others. Also, runs were computed with different core sizes, with results similar to the ones displayed. (Text continues on page 75)

Table 4.3: Table of Single Filament Runs

Run	Stretch	Type	Core	Cutoff	dt	N	Wave	perturb	Tmax
reproduce.1	KG	Blob	2	0.3	0.005	96	6	0.02	18
base.1	KG	Blob	2	0.3	0.005	64	6	0.05	75
base.2	KG	Blob	4	0.3	0.005	64	6	0.05	176
base.3	KG	Blob	G	0.3	0.005	64	6	0.05	55
base.4	C	stick	C	0.3	0.005	64	6	0.05	12
base.5	C	blob	2	0.3	0.005	64	6	0.05	63
base.7	Helmholtz	blob	2	0.3	0.005	64	6	0.05	55
base.8	exact	blob	2	0.3	0.005	64	6	0.05	9
base.9	$\omega_i \cdot \nabla_h u$	blob	2	0.3	0.005	64	6	0.05	17
base.10	$\omega_i \cdot \nabla_h u^T$	blob	2	0.3	0.005	64	6	0.05	146
base.11	$\omega \cdot \nabla_h u^T$	blob	2	0.3	0.005	64	6	0.05	307
RefineDt.1	KG	Blob	2	0.3	0.0025	64	6	0.05	79
RefineDt.2	KG	Blob	4	0.3	0.0025	64	6	0.05	125
RefineDt.3	KG	Blob	G	0.3	0.0025	64	6	0.05	55
RefineDt.4	C	stick	C	0.3	0.0025	64	6	0.05	15
RefineDt.5	C	blob	2	0.3	0.0025	64	6	0.05	62
RefineDt.7	Helmholtz	blob	2	0.3	0.0025	64	6	0.05	56
RefineDt.8	exact	blob	2	0.3	0.0025	64	6	0.05	9
RefineDt.9	$\omega_i \cdot \nabla_h u$	blob	2	0.3	0.0025	64	6	0.05	16
RefineDt.10	$\omega_i \cdot \nabla_h u^T$	blob	2	0.3	0.0025	64	6	0.05	146
RefineDt.11	$\omega \cdot \nabla_h u^T$	blob	2	0.3	0.0025	64	6	0.05	203
RefineDx.1	KG	Blob	2	0.3	0.005	128	6	0.05	56
RefineDx.2	KG	Blob	4	0.3	0.005	128	6	0.05	570
RefineDx.3	KG	Blob	G	0.3	0.005	128	6	0.05	63
RefineDx.4	C	stick	C	0.3	0.005	128	6	0.05	7
RefineDx.5	C	blob	2	0.3	0.005	128	6	0.05	52
RefineDx.7	Helmholtz	blob	2	0.3	0.005	128	6	0.05	56
RefineDx.8	exact	blob	2	0.3	0.005	128	6	0.05	11
RefineDx.9	$\omega_i \cdot \nabla_h u$	blob	2	0.3	0.005	128	6	0.05	16
RefineDx.10	$\omega_i \cdot \nabla_h u^T$	blob	2	0.3	0.005	128	6	0.05	155
RefineDx.11	$\omega \cdot \nabla_h u^T$	blob	2	0.3	0.005	128	6	0.05	203
RefineSigma.1	KG	Blob	2	0.2	0.005	64	6	0.05	160
RefineSigma.2	KG	Blob	4	0.2	0.005	64	6	0.05	233
RefineSigma.3	KG	Blob	G	0.2	0.005	64	6	0.05	13
RefineSigma.4	C	stick	C	0.2	0.005	64	6	0.05	4
RefineSigma.5	C	blob	2	0.2	0.005	64	6	0.05	72
RefineSigma.7	Helmholtz	blob	2	0.2	0.005	64	6	0.05	53
RefineSigma.9	$\omega_i \cdot \nabla_h u$	blob	2	0.2	0.005	64	6	0.05	42
RefineSigma.10	$\omega_i \cdot \nabla_h u^T$	blob	2	0.2	0.005	64	6	0.05	119
RefineSigma.11	$\omega \cdot \nabla_h u^T$	blob	2	0.2	0.005	64	6	0.05	48

Figure 4.1: Run reproduce.1

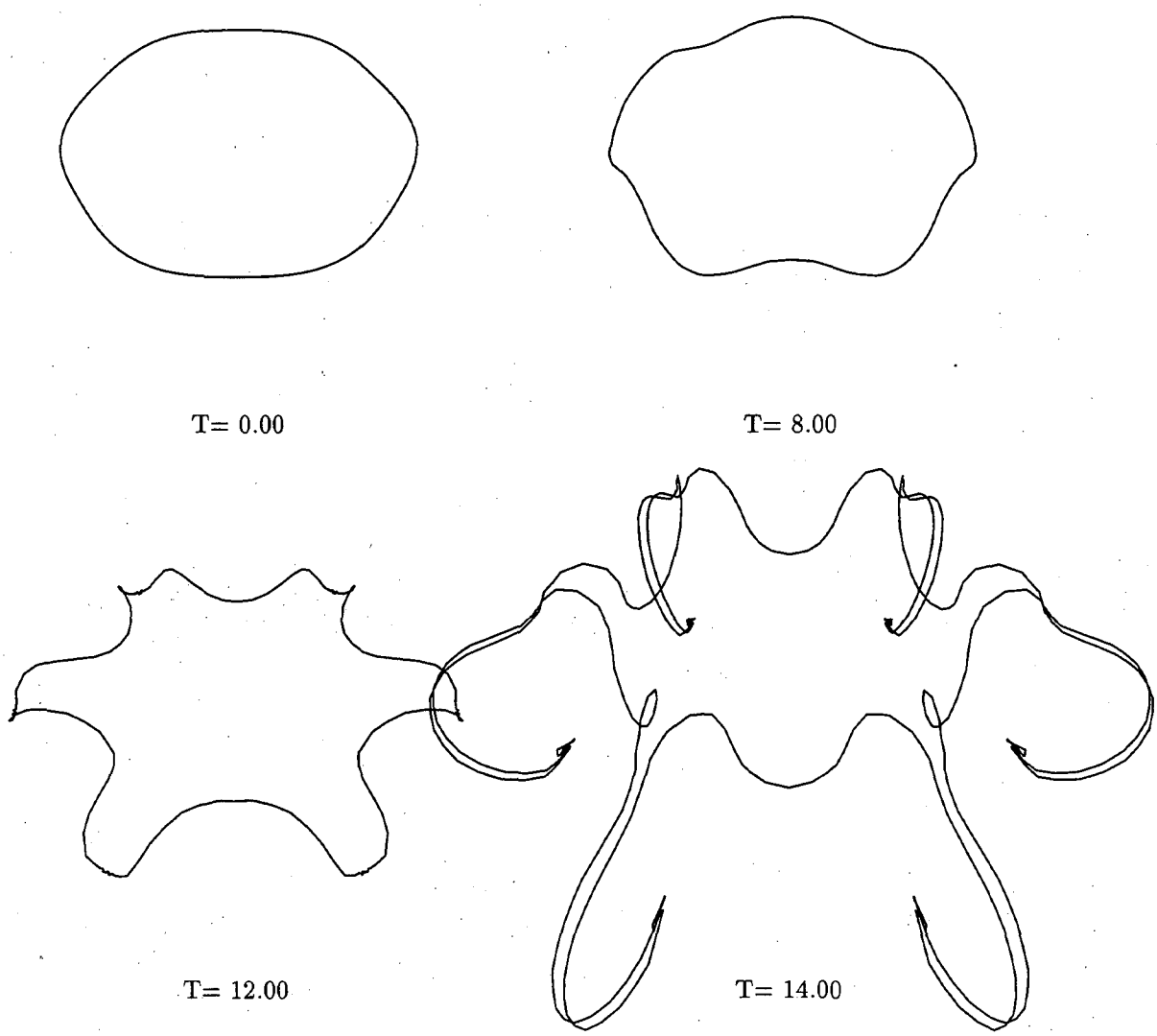


Figure 4.2: Run base.1

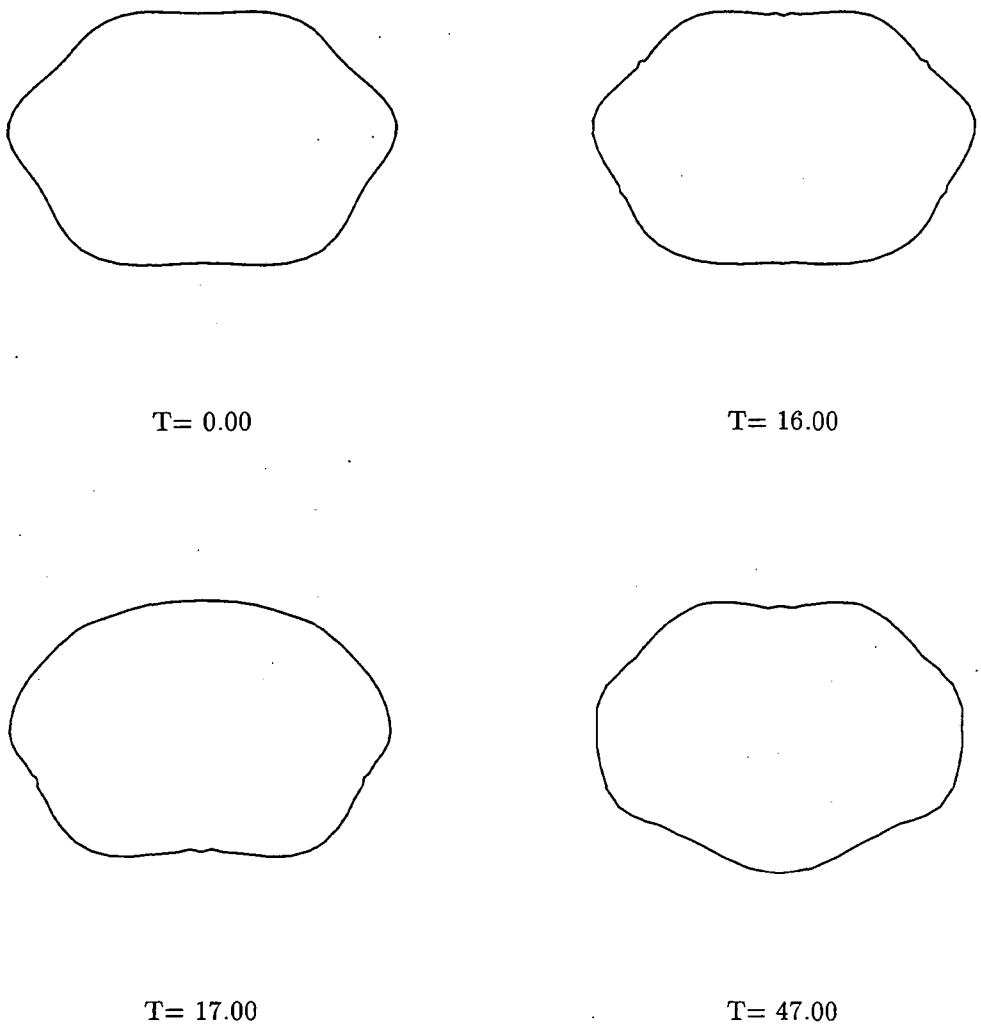


Figure 4.3: Run base.1

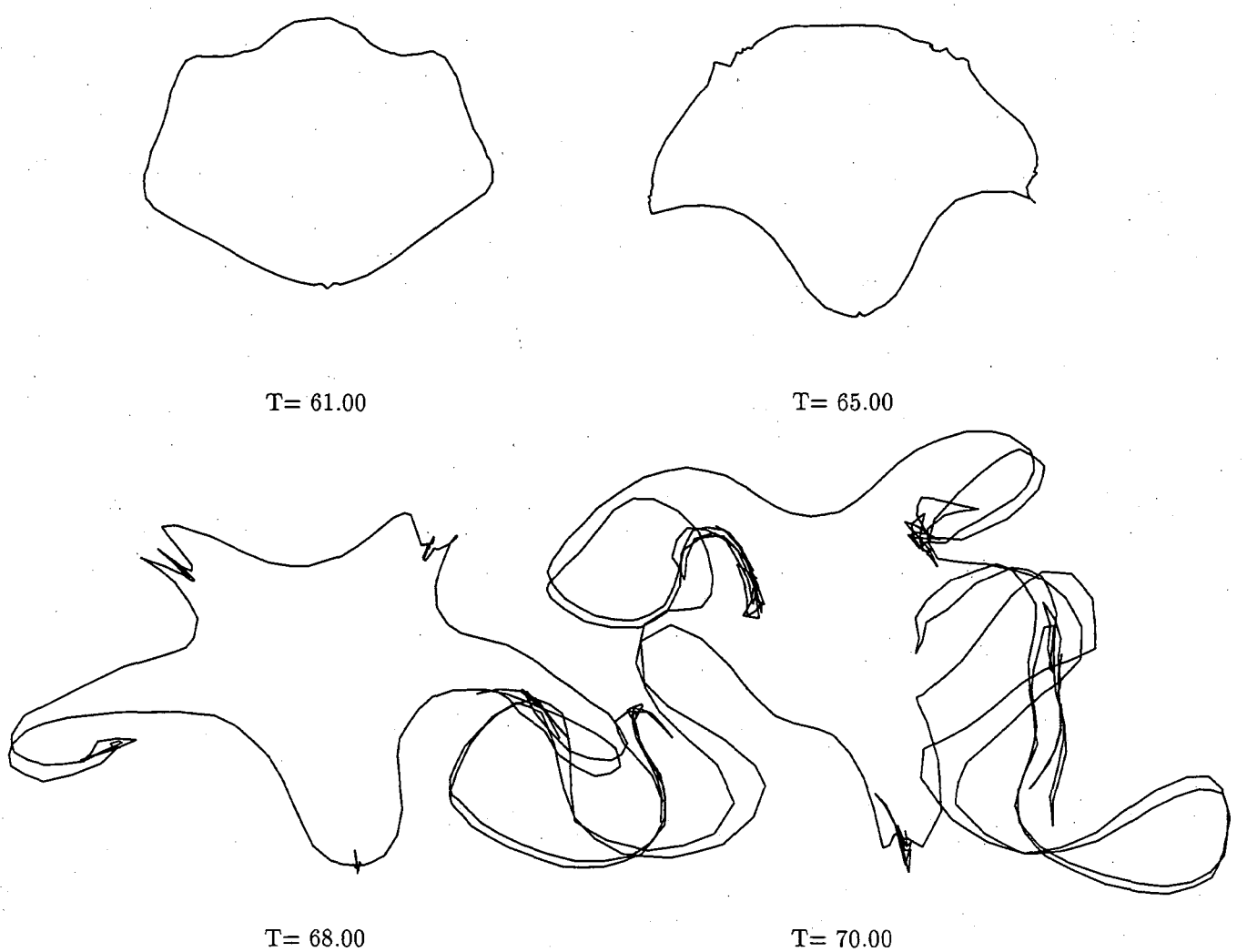
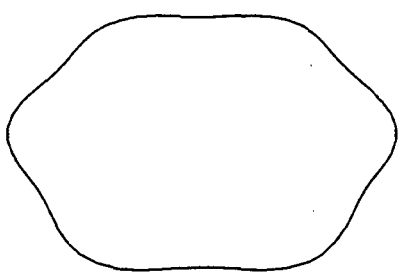
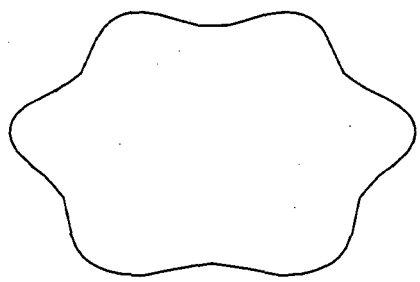


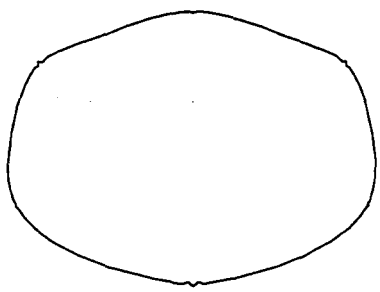
Figure 4.4: Run base.2



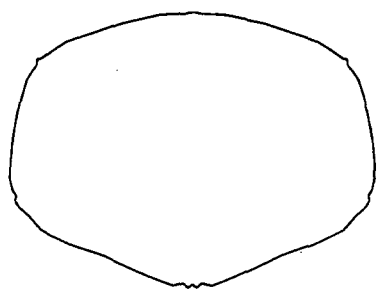
T= 0.00



T= 8.00

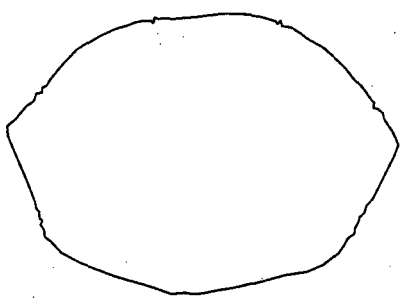


T= 17.00

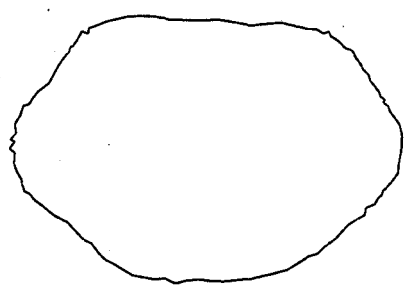


T= 79.00

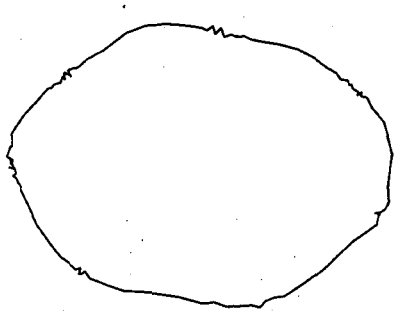
Figure 4.5: Run base.2



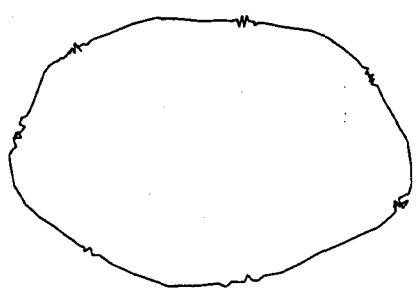
T=120.00



T=139.00



T=156.00



T=159.00

Figure 4.6: Run base.2

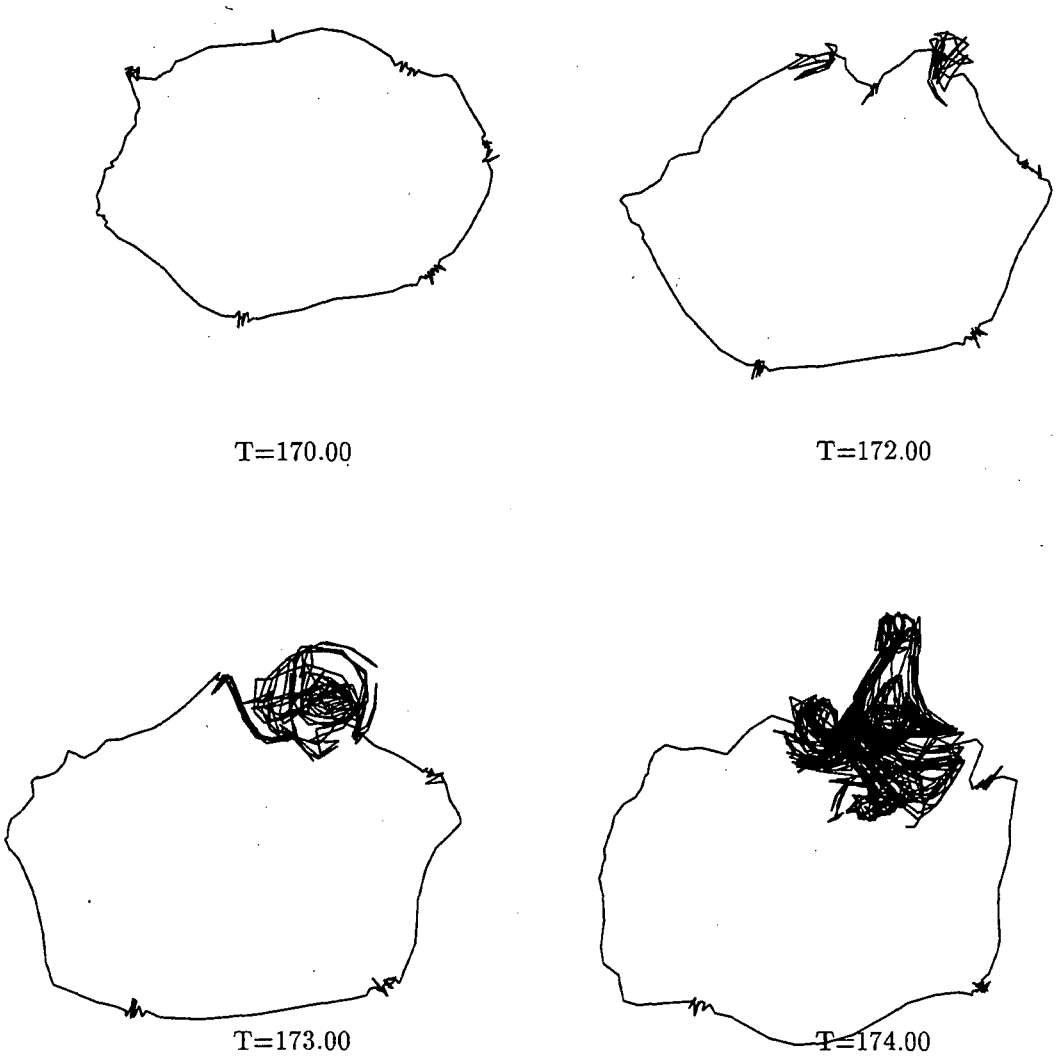


Figure 4.7: Run base.3

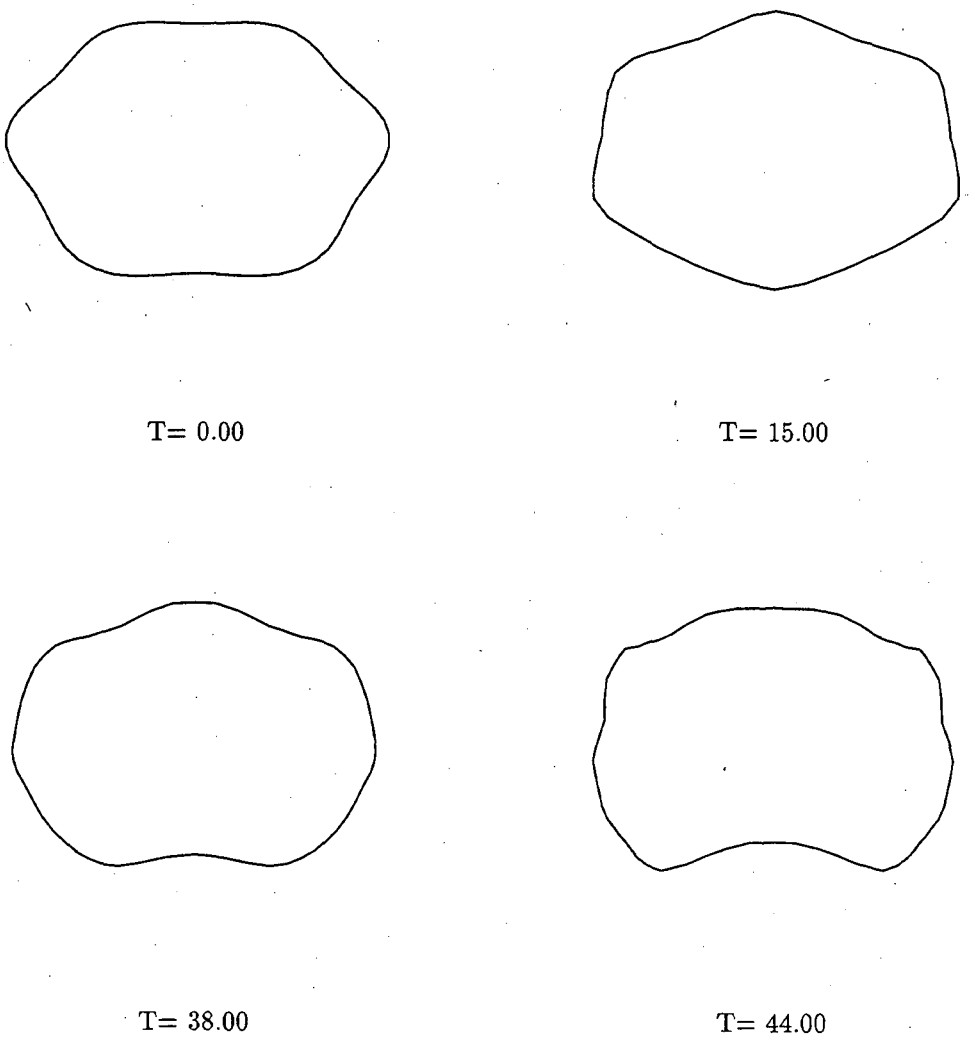


Figure 4.8: Run base.3

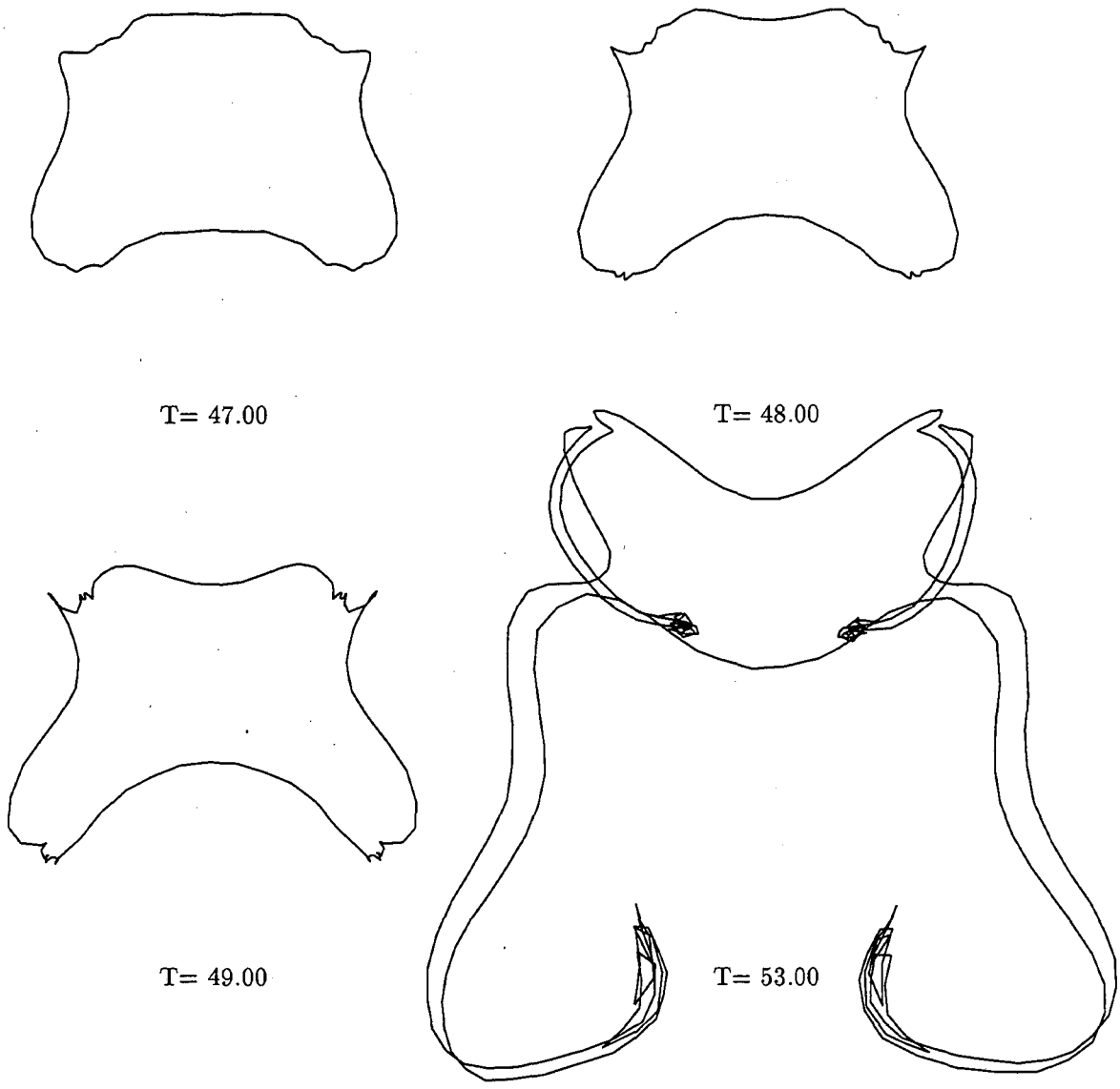


Figure 4.9: Run base.4

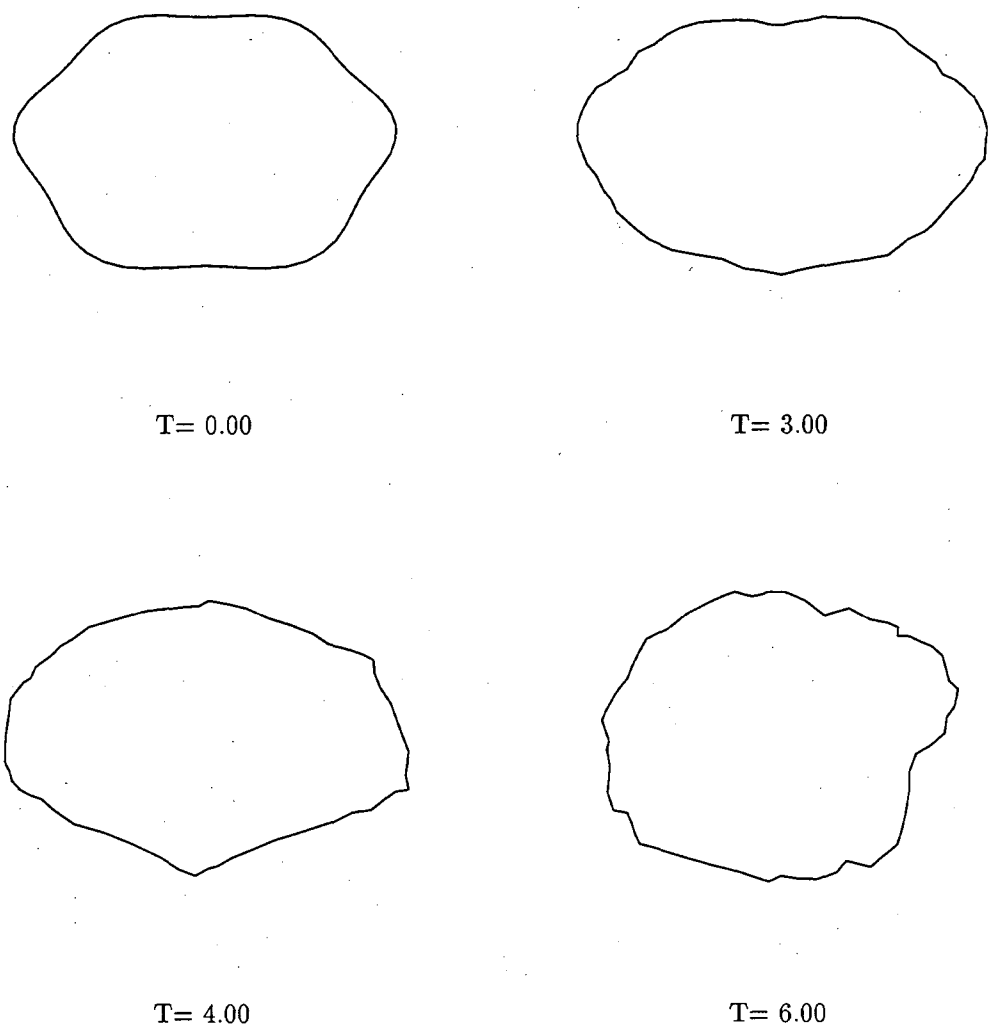
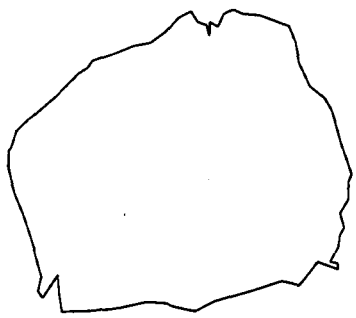
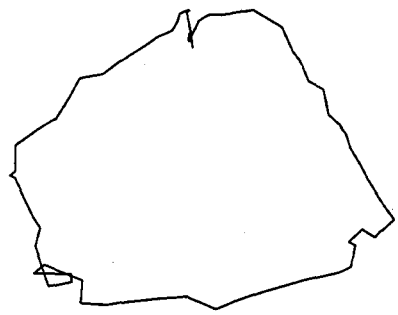


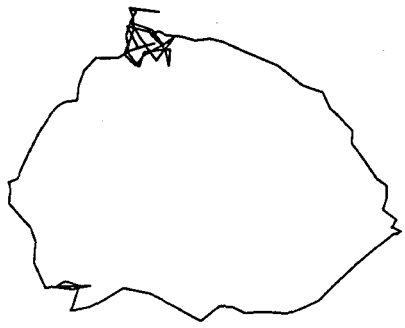
Figure 4.10: Run base.4



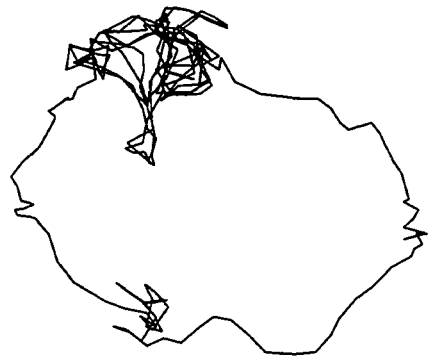
T= 8.00



T= 9.00



T= 10.00



T= 11.00

Figure 4.11: Run base.5

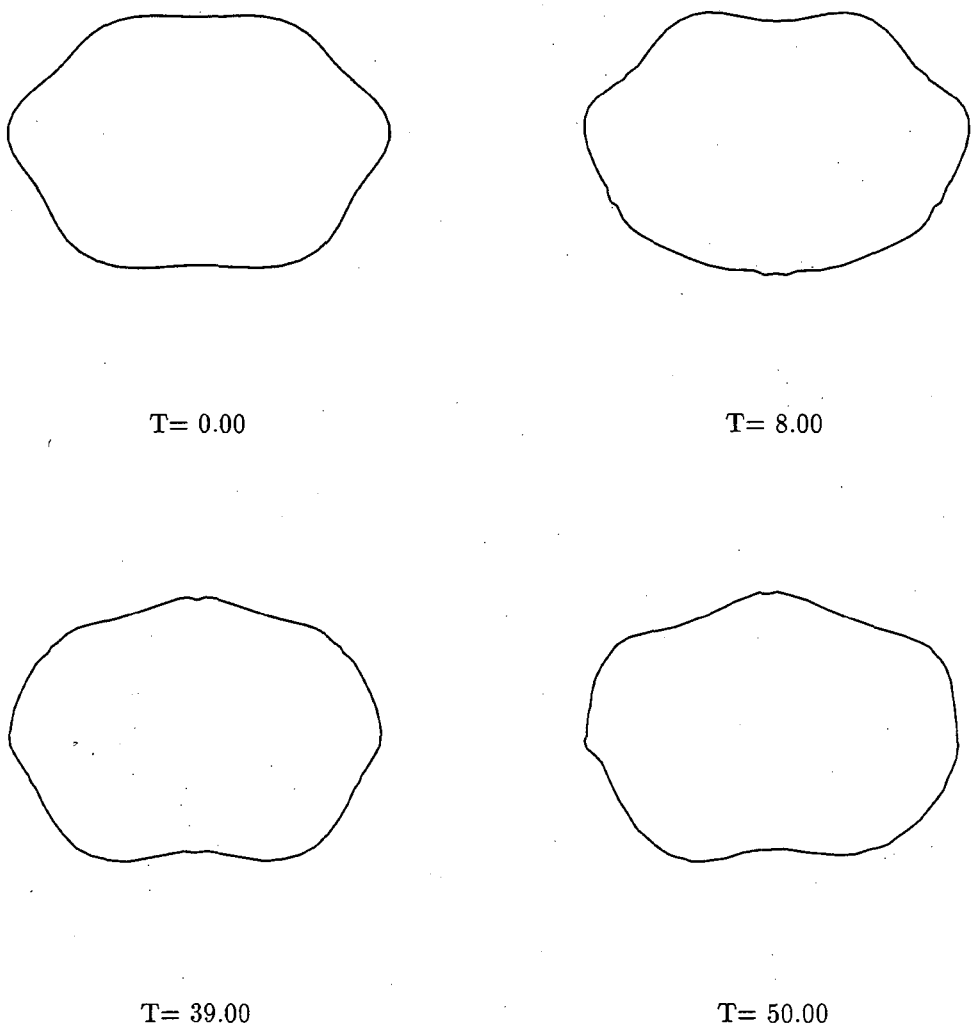


Figure 4.12: Run base.5

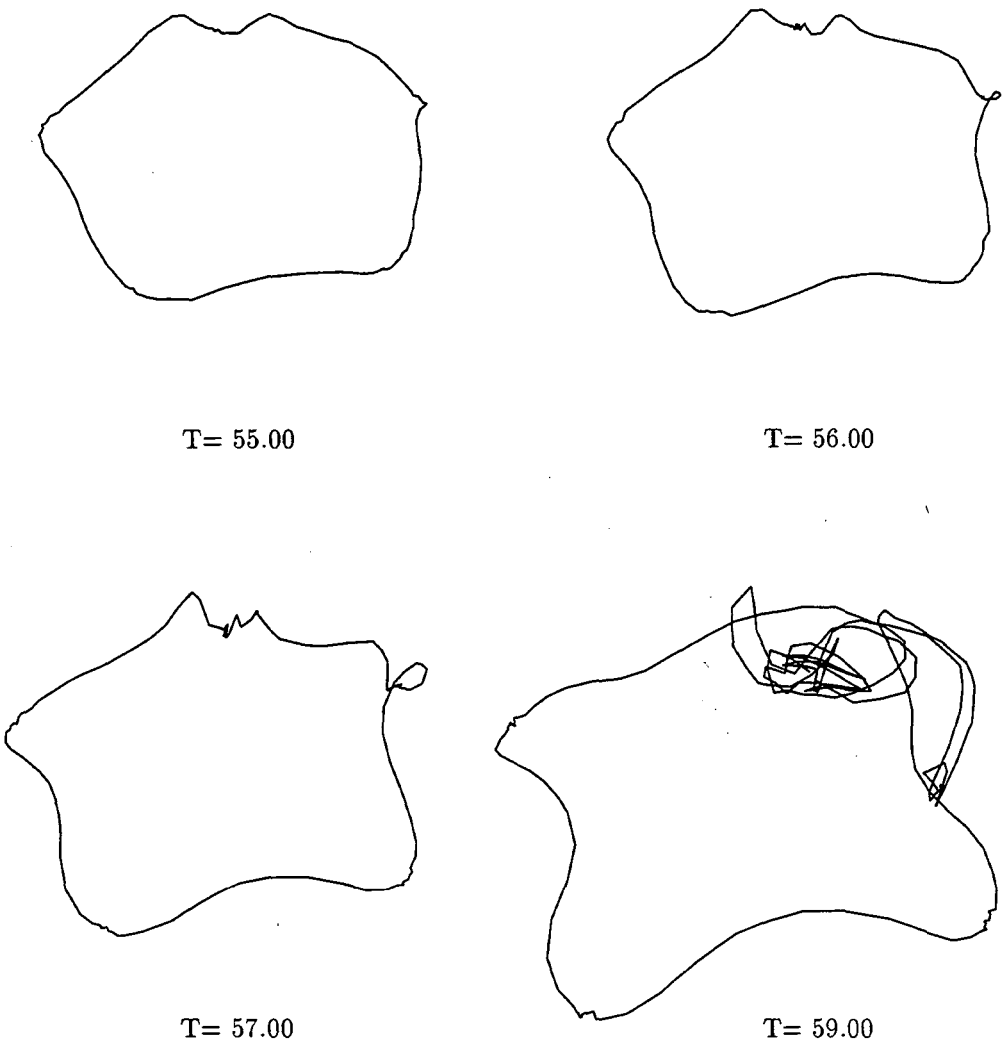


Figure 4.13: Run base.7

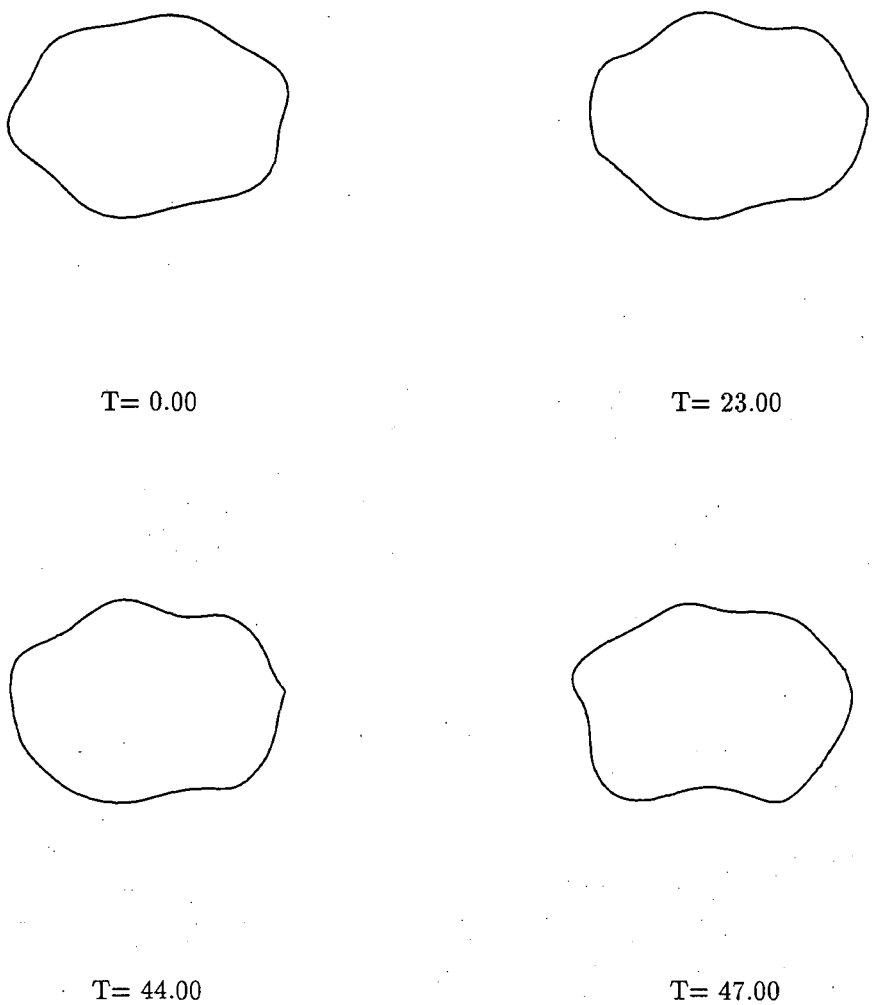


Figure 4.14: Run base.7

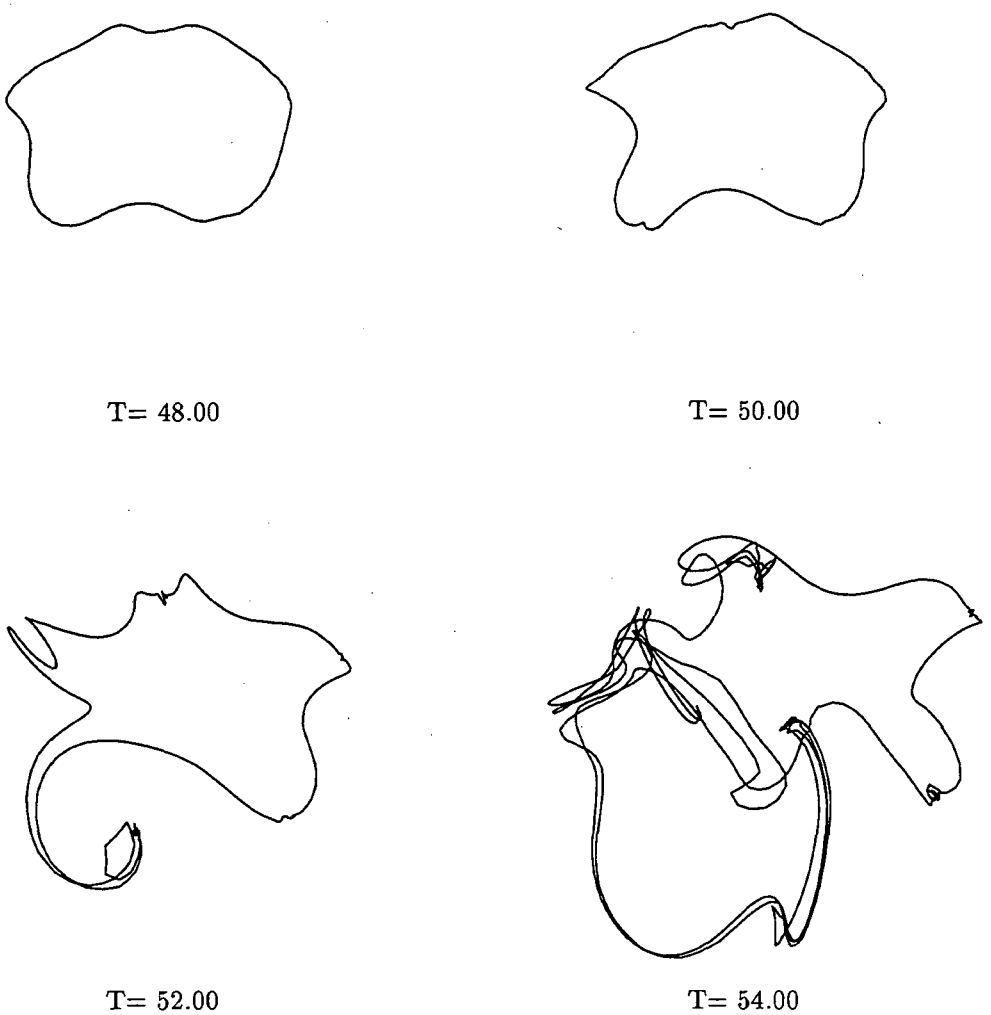


Figure 4.15: Run base.8

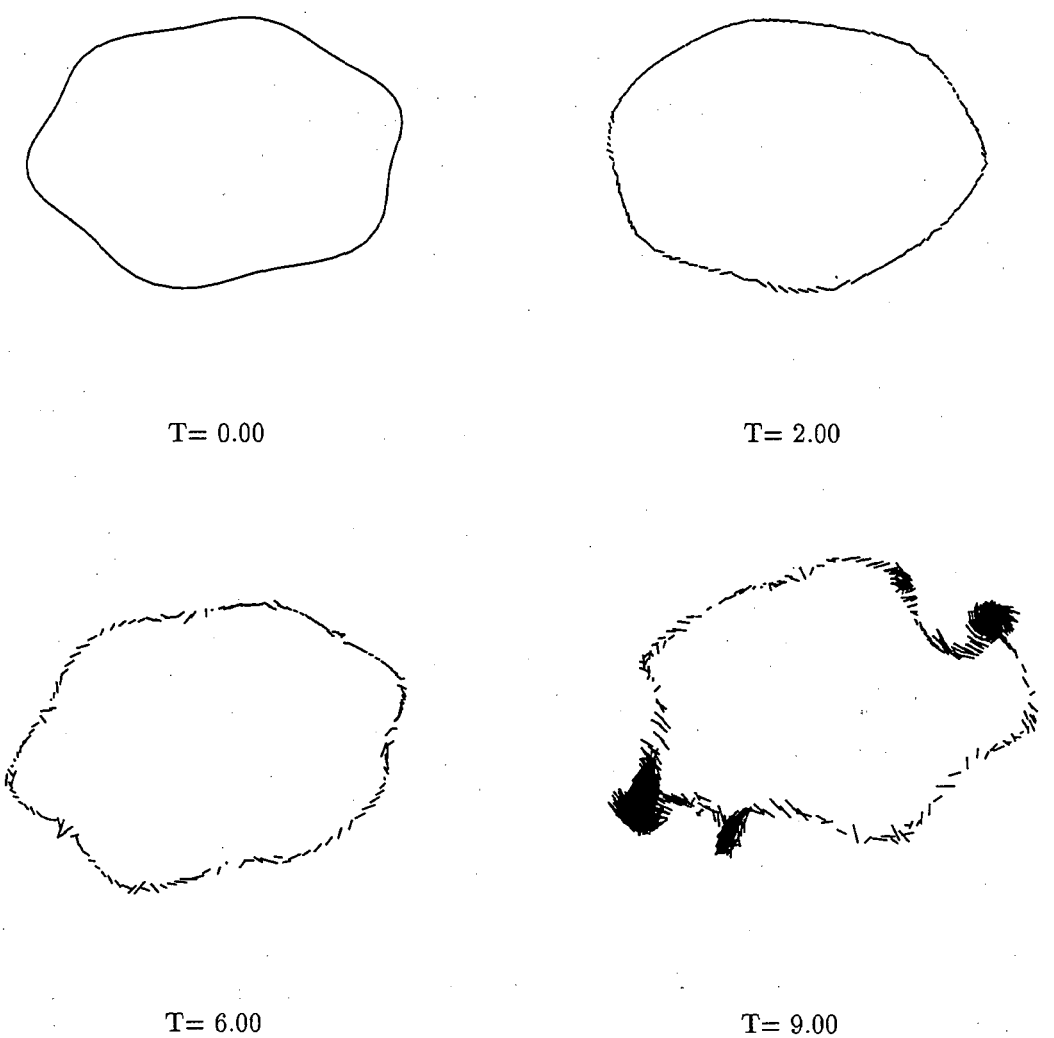


Figure 4.16: Run base.9

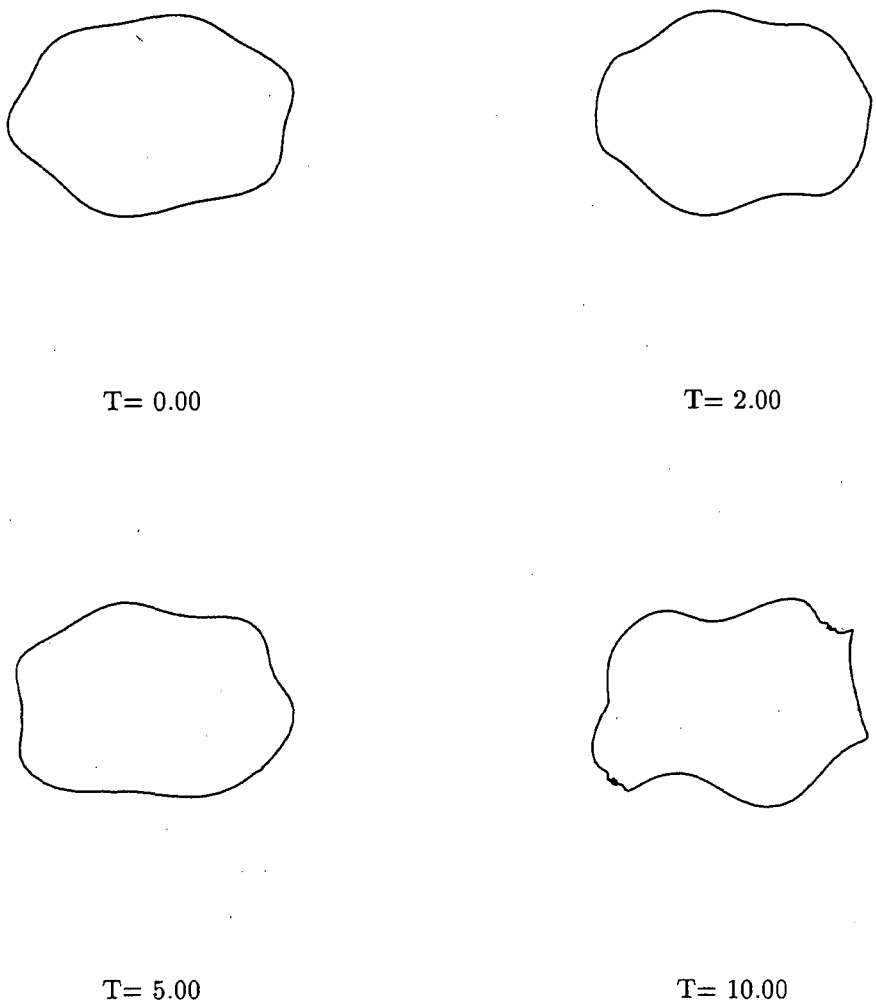
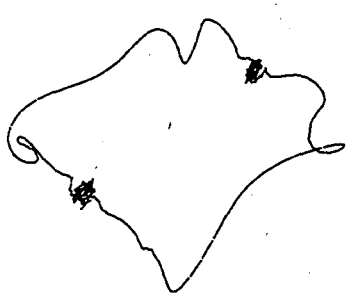
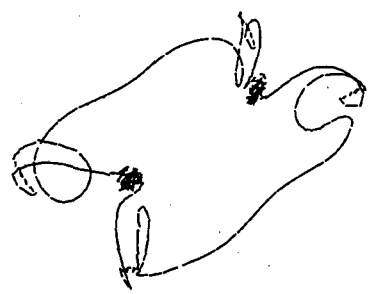


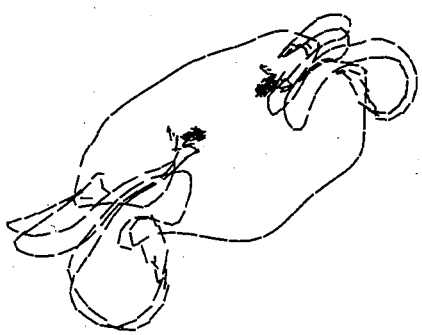
Figure 4.17: Run base.9



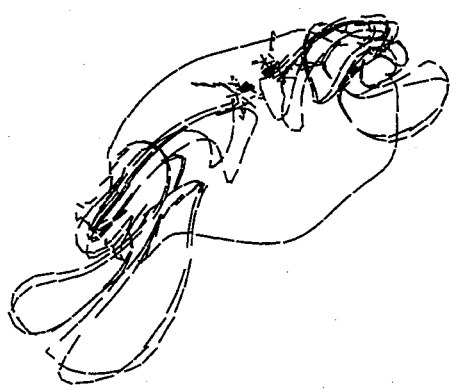
T= 13.00



T= 14.00

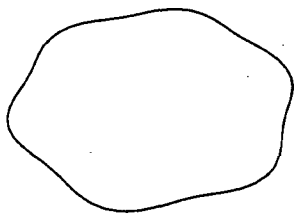


T= 15.00

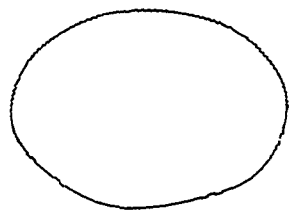


T= 16.00

Figure 4.18: Run base.10

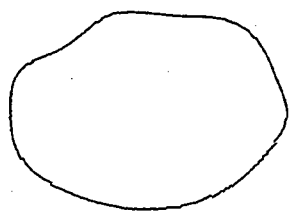


T= 0.00



T= 13.00

T= 41.00



T= 76.00

Figure 4.19: Run base.10

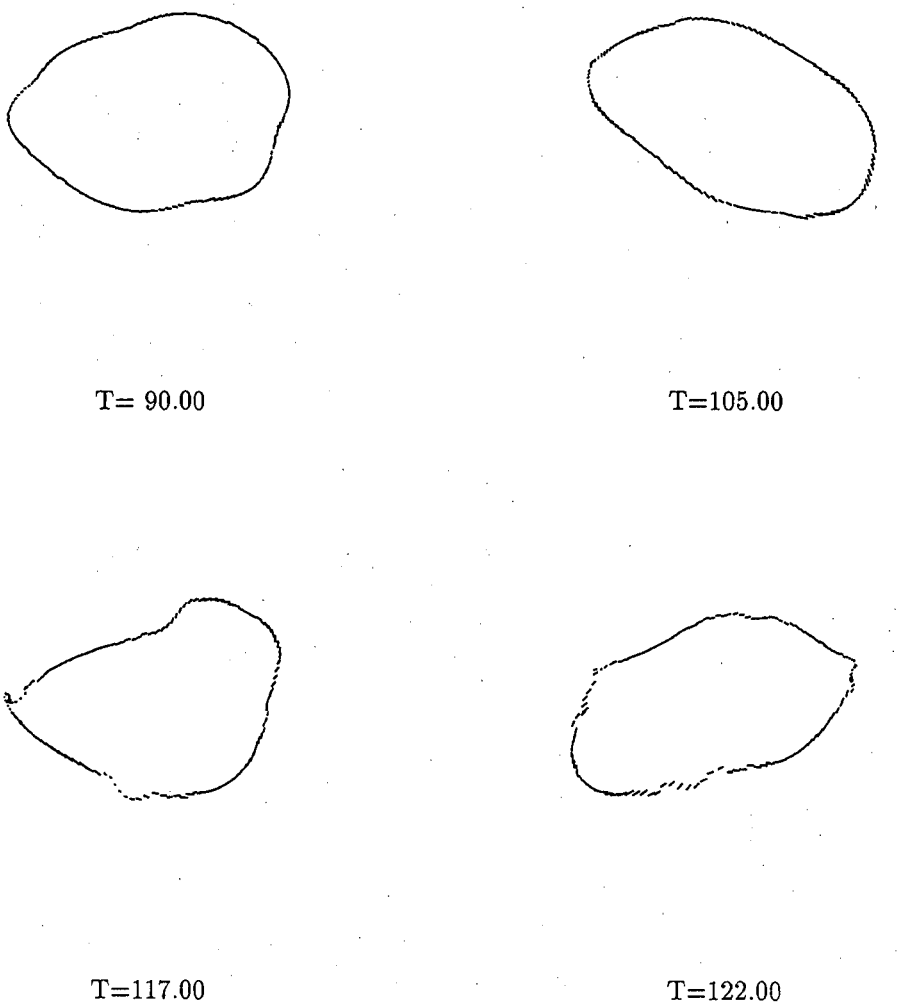


Figure 4.20: Run base.10

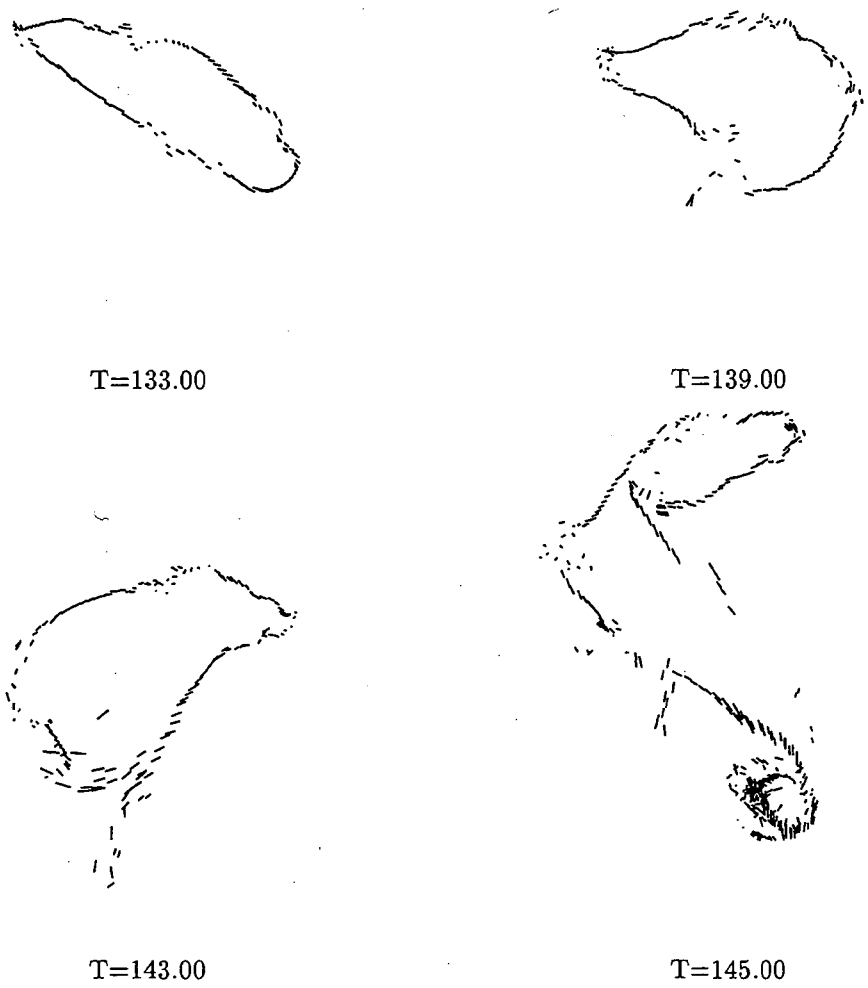


Figure 4.21: Run base.11

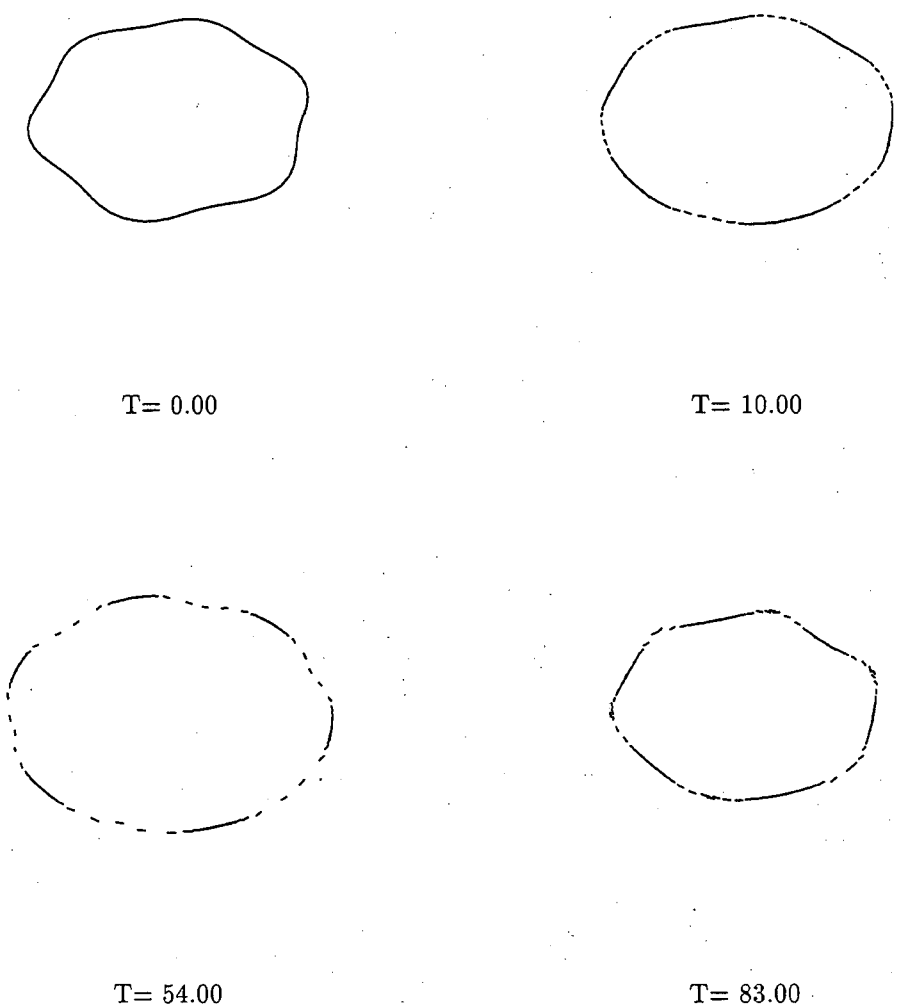


Figure 4.22: Run base.11

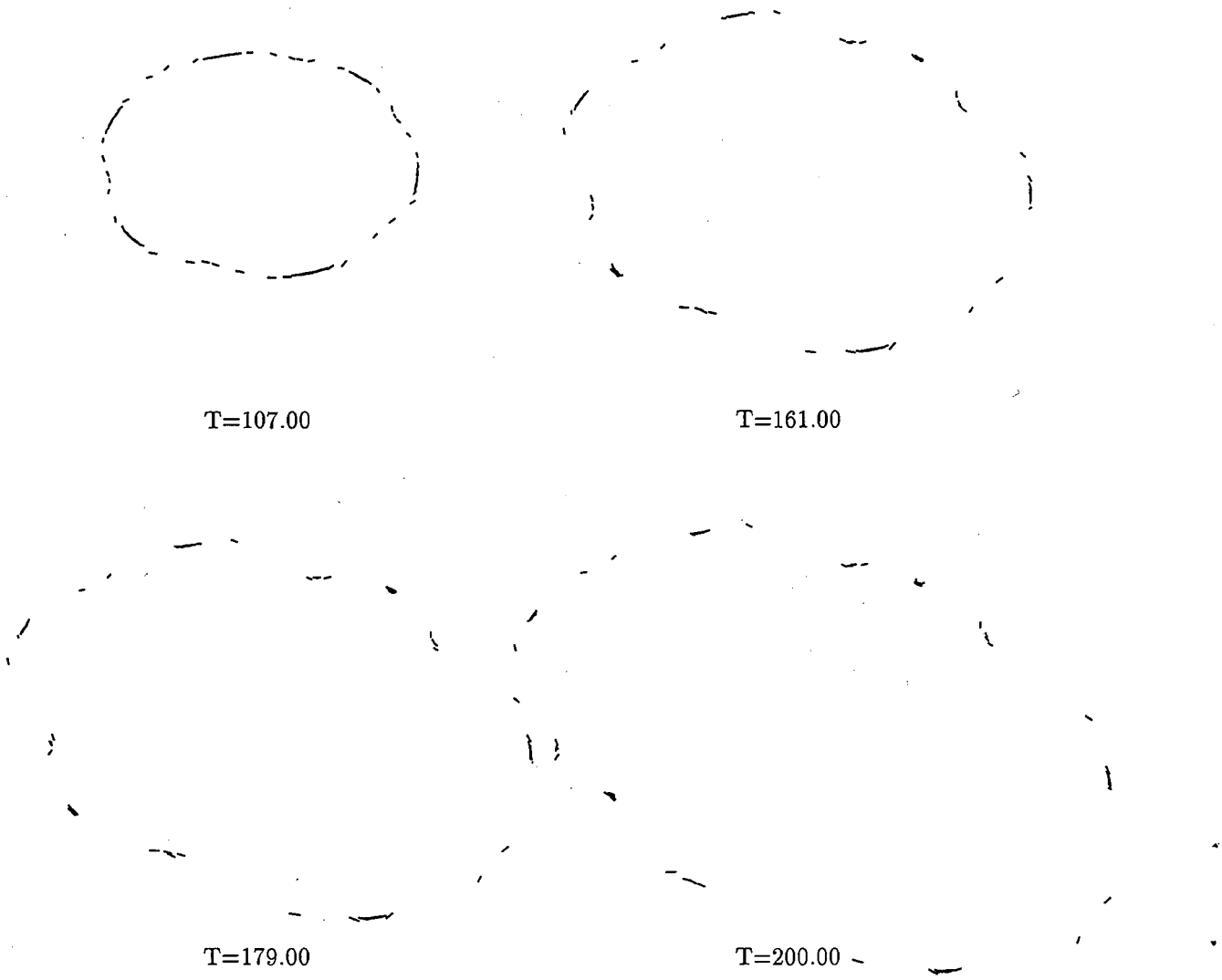


Figure 4.23: Run refinedt.8

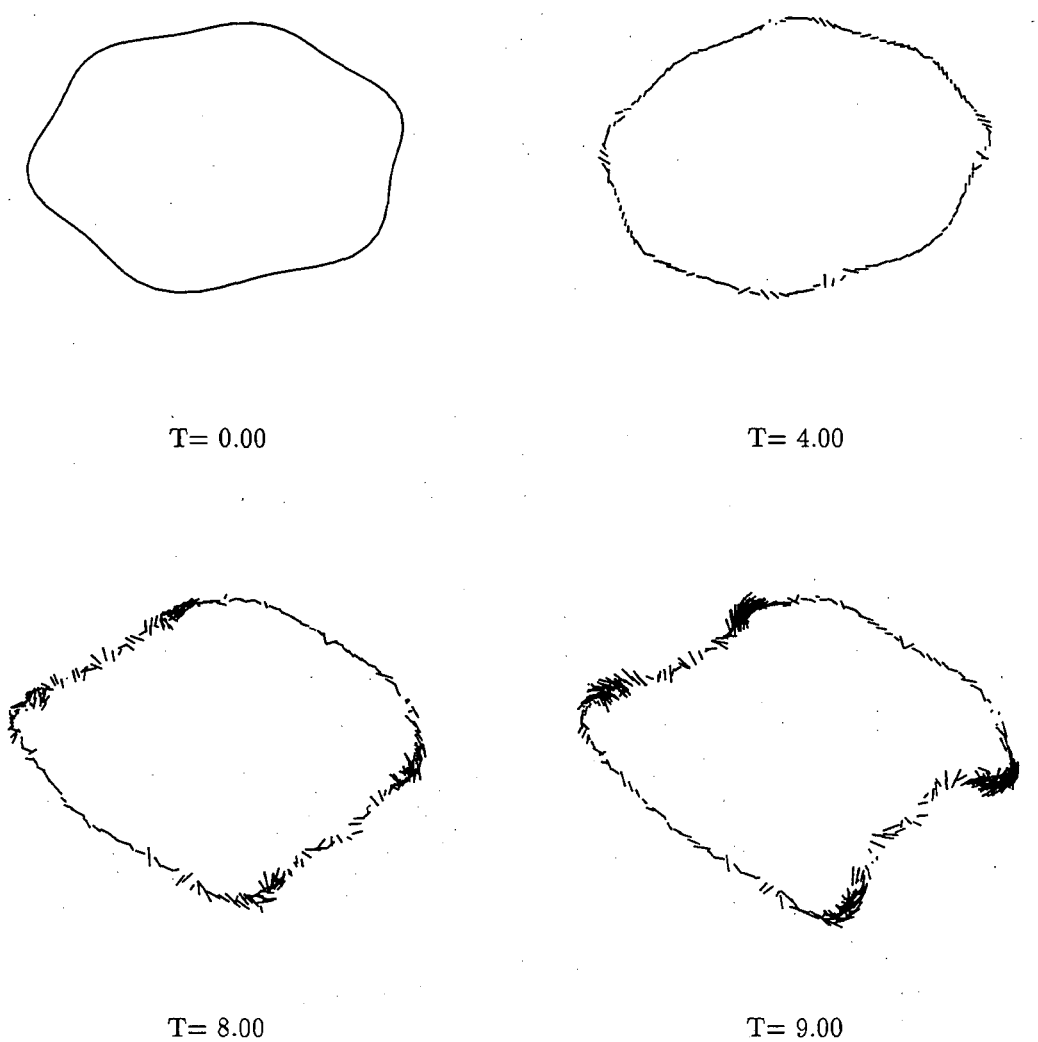
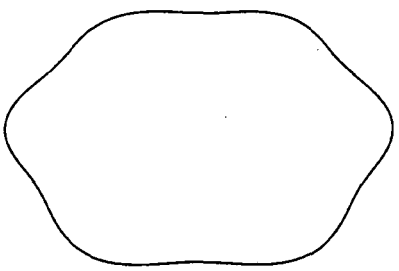
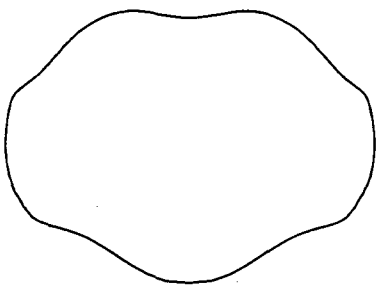


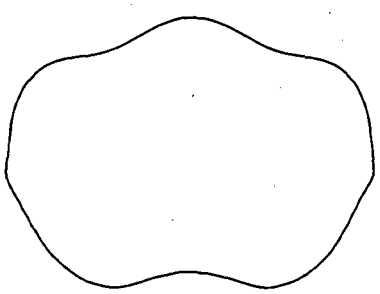
Figure 4.24: Run refinedx.1



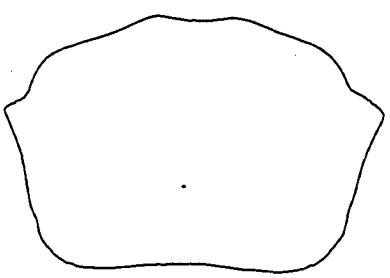
T= 0.00



T= 17.00

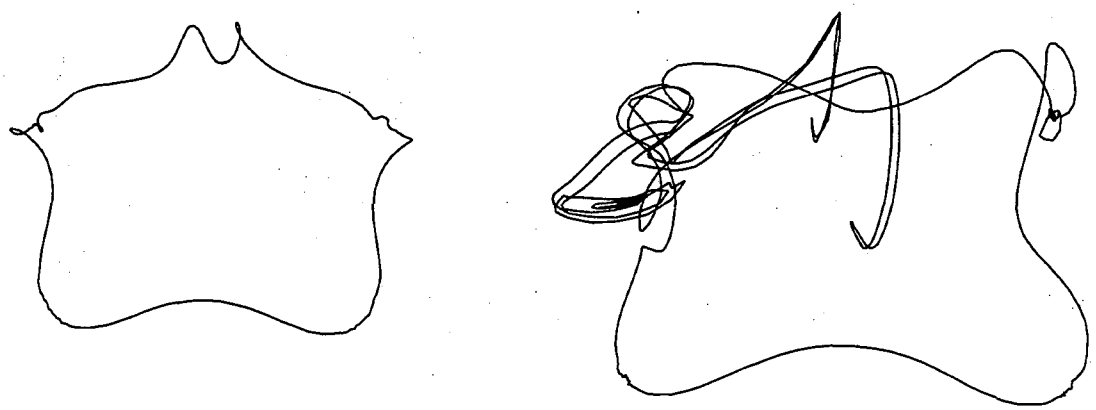


T= 43.00



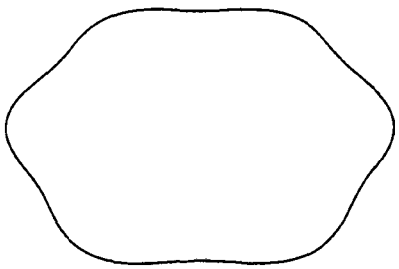
T= 48.00

Figure 4.25: Run refinedx.1

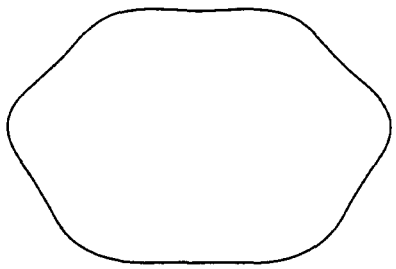


T= 50.00 T= 53.00

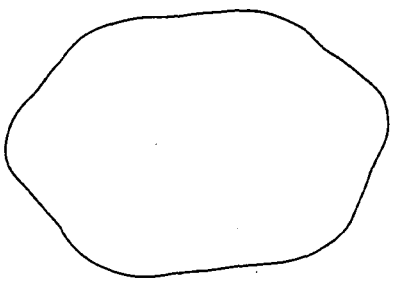
Figure 4.26: Run refinedx.2



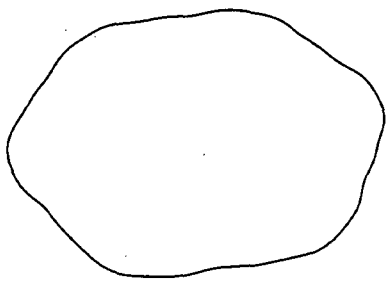
T= 0.00



T= 35.00

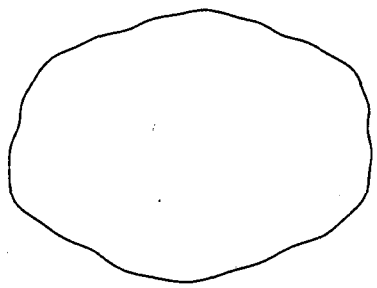


T=215.00

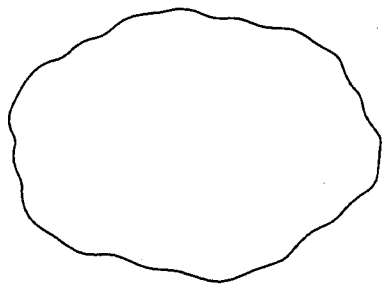


T=250.00

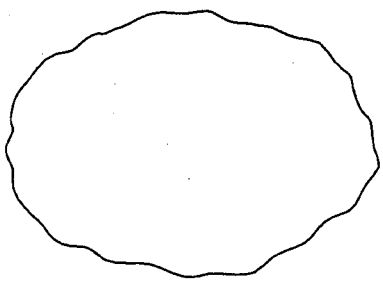
Figure 4.27: Run refinedx.2



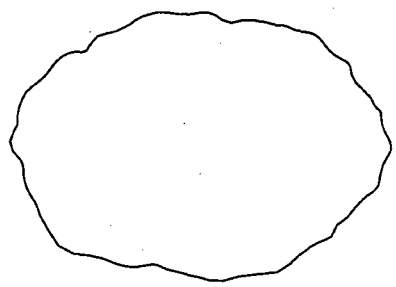
T=300.00



T=502.00

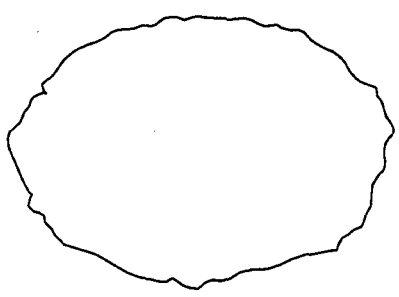


T=540.00

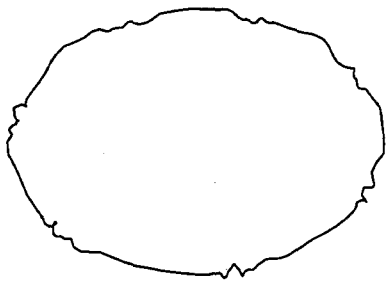


T=550.00

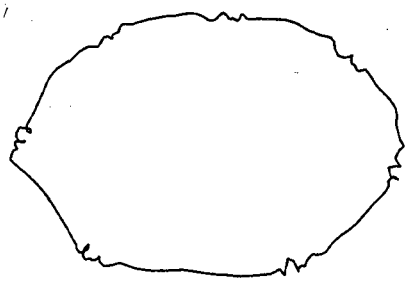
Figure 4.28: Run refinedx.2



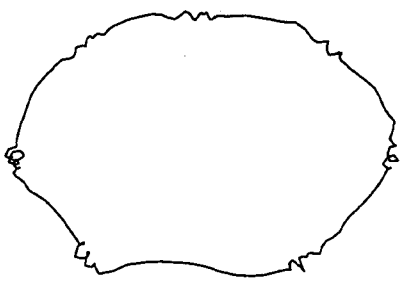
T=561.00



T=564.00

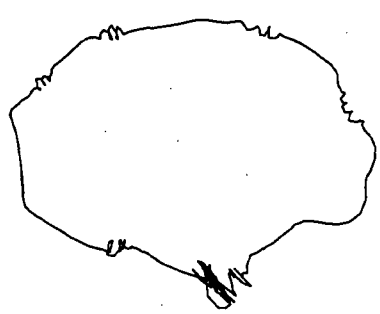


T=566.00



T=567.00

Figure 4.29: Run refinedx.2



T=570.00

Figure 4.30: Run refinedx.10

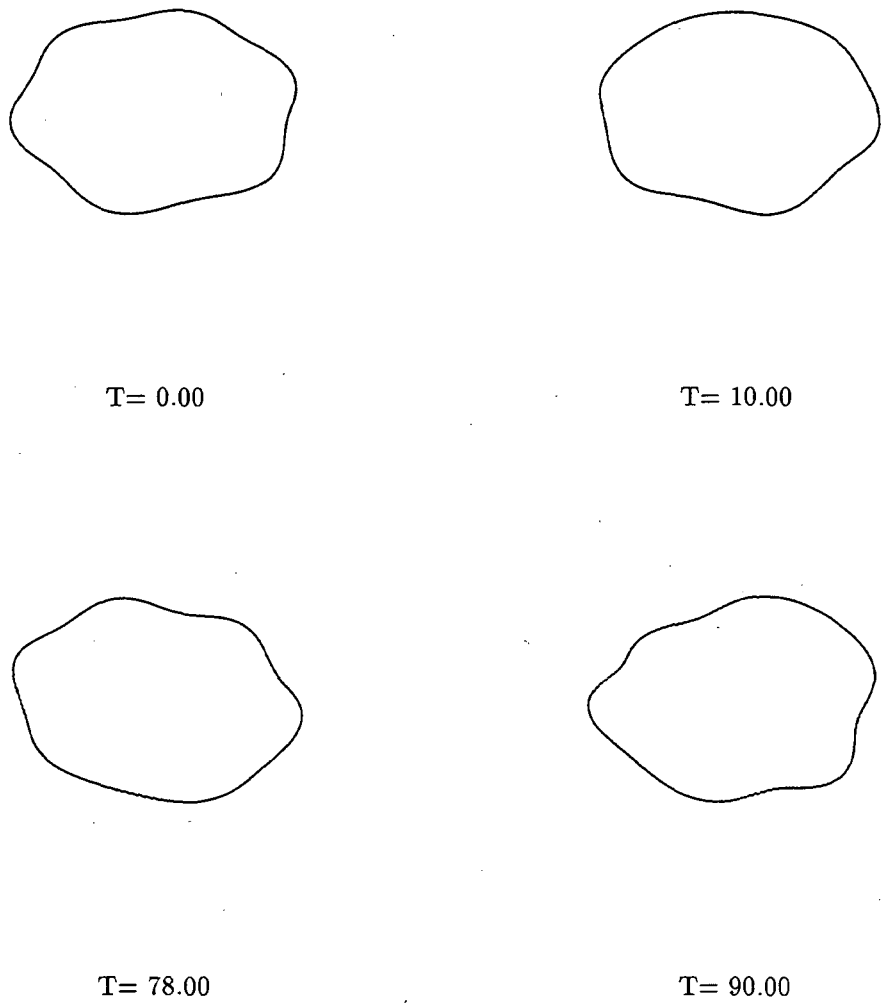


Figure 4.31: Run refinedx.10

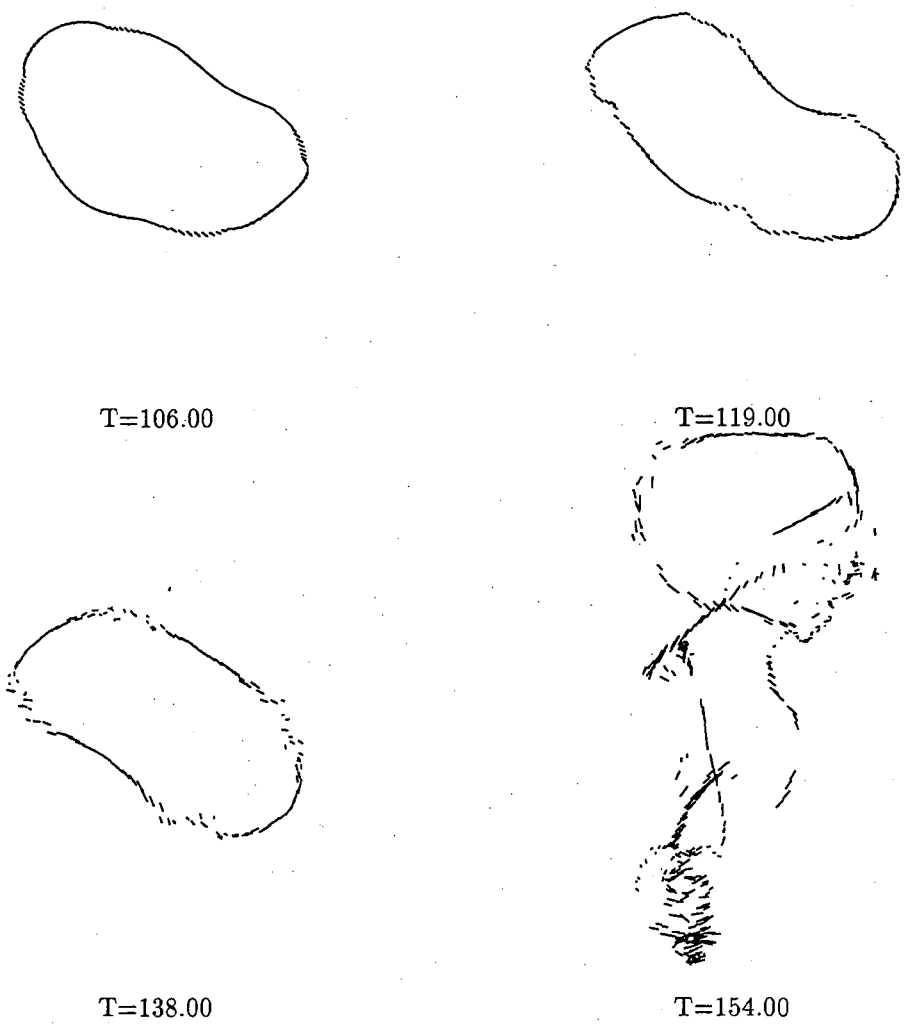
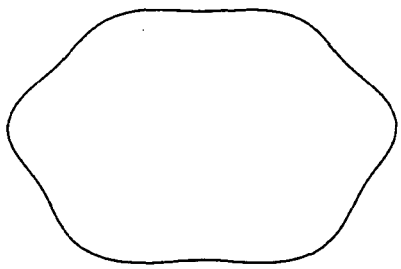
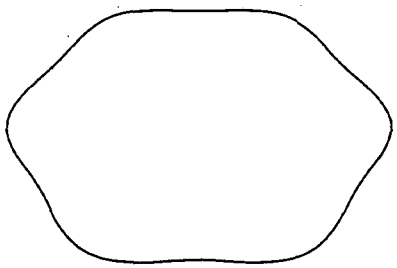


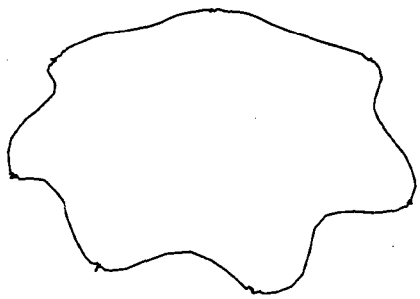
Figure 4.32: Run refinesigma.1



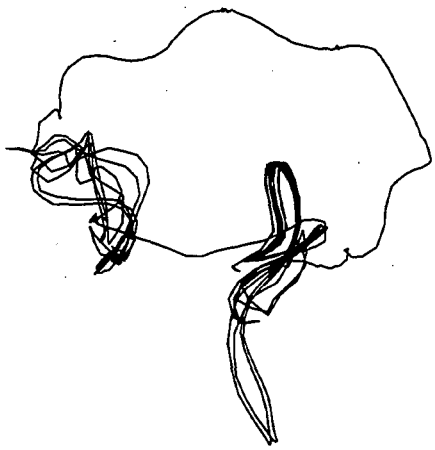
T= 0.00



T= 68.00



T=157.00



T=160.00

Figure 4.33: Run refinesigma.2

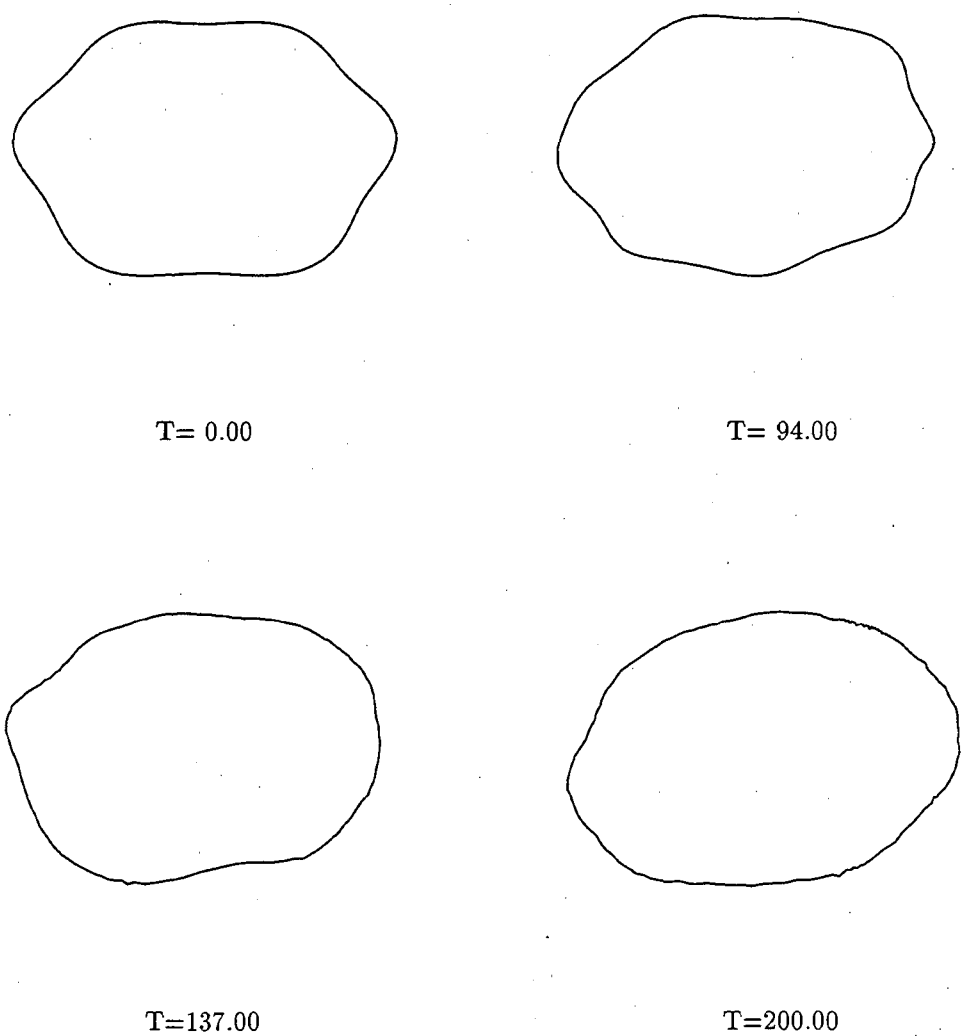
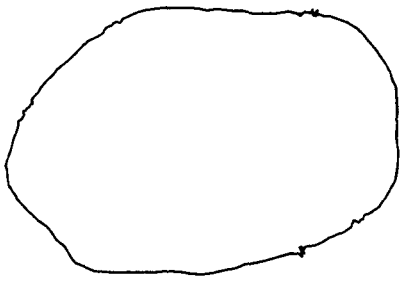
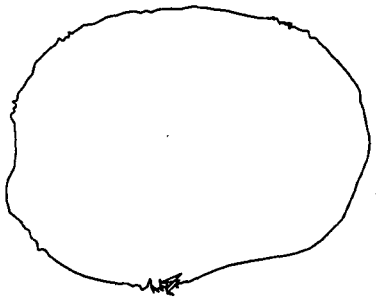


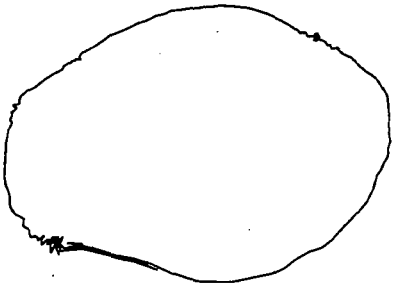
Figure 4.34: Run refinesigma.2



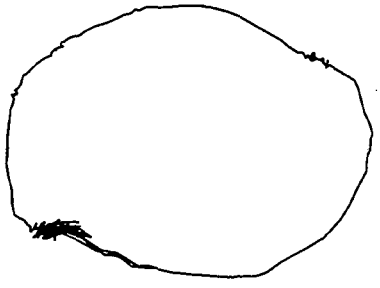
T=215.00



T=227.00



T=229.00



T=232.00

Figure 4.35: Run refinesigma.3

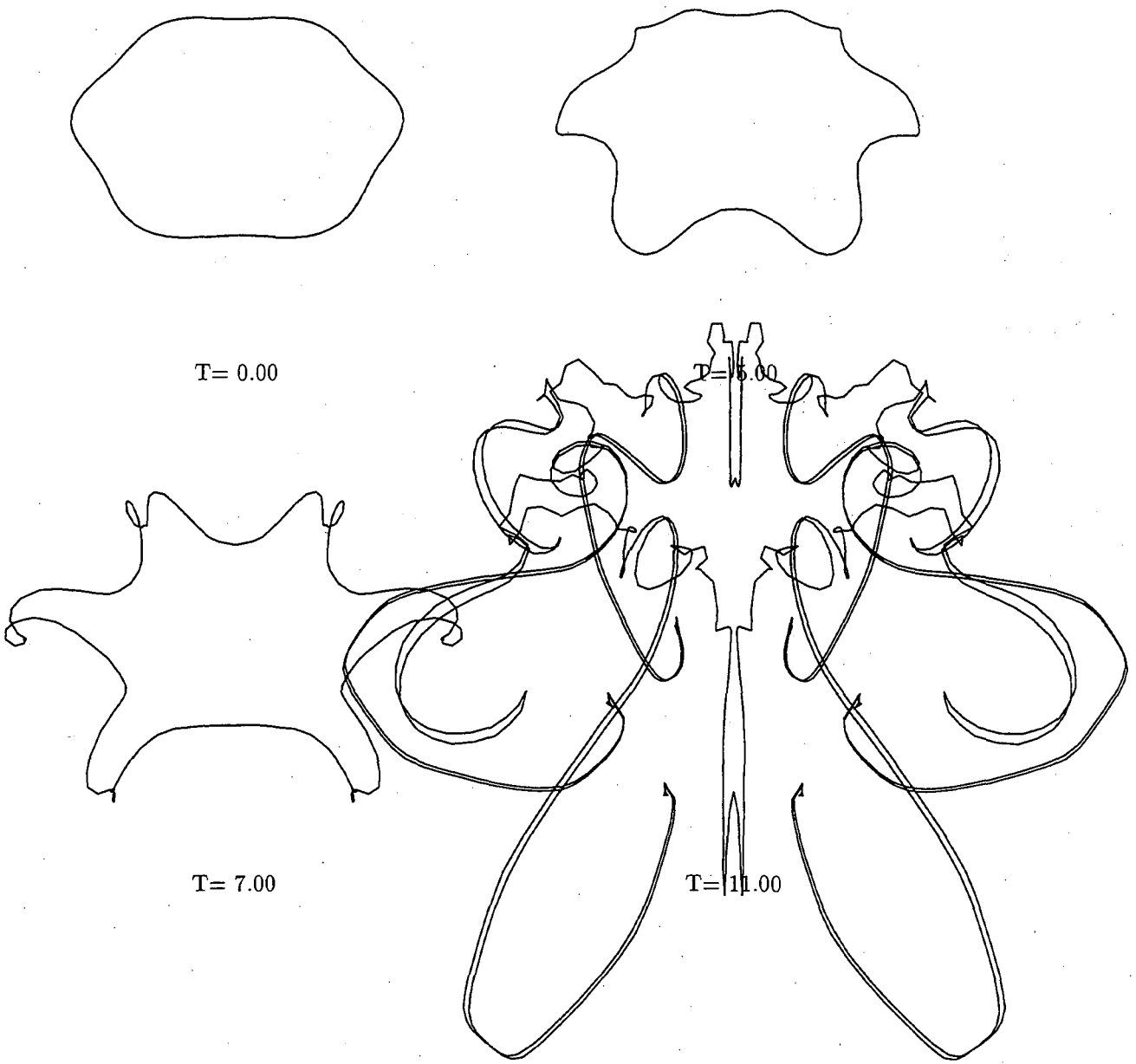


Figure 4.36: Run refinesigma.5

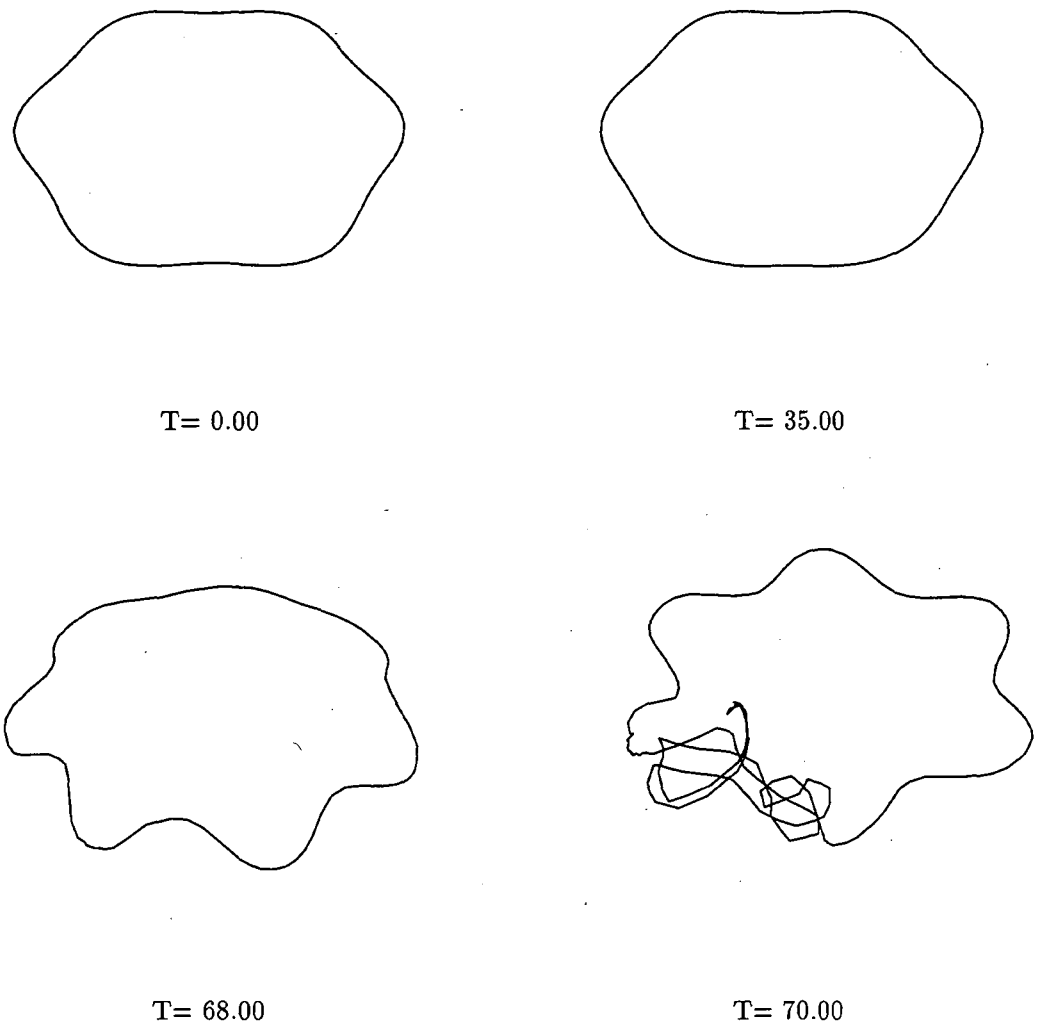


Figure 4.37: Run refinesigma.7

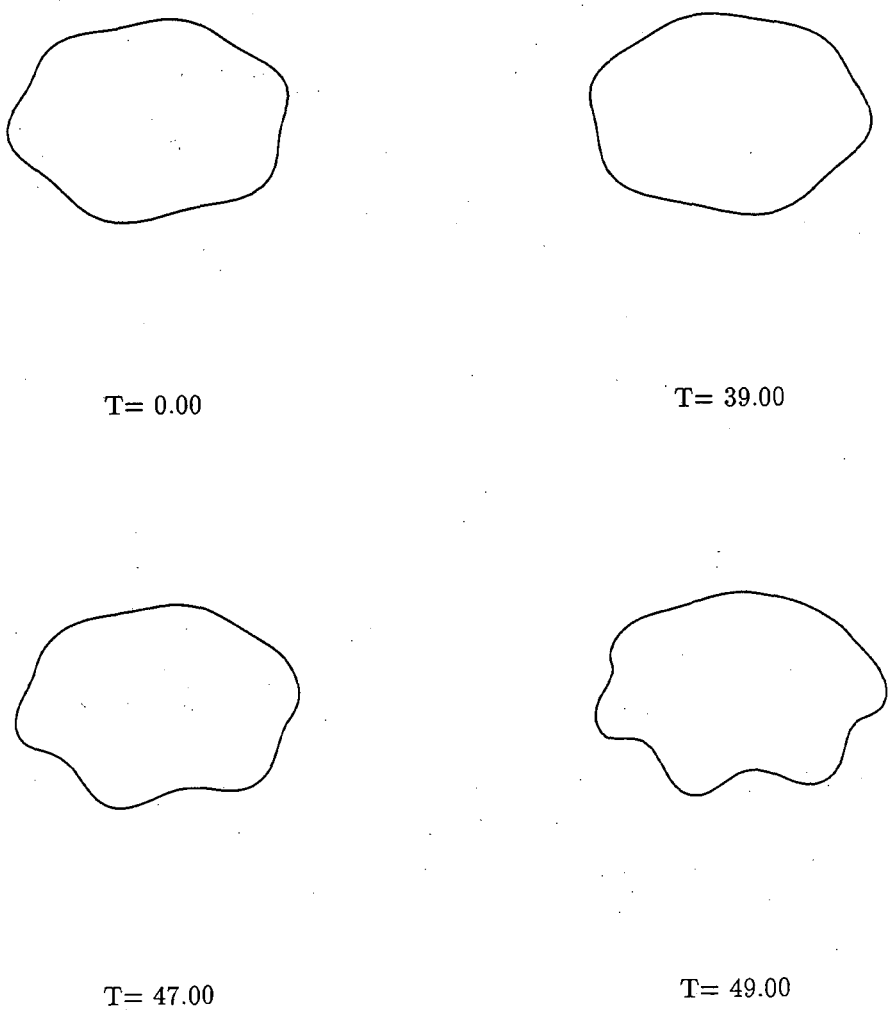
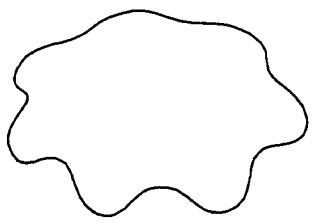
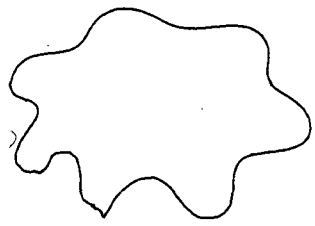


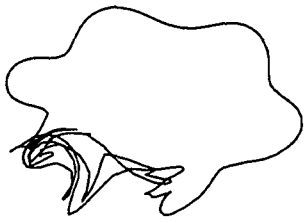
Figure 4.38: Run refinesigma.7



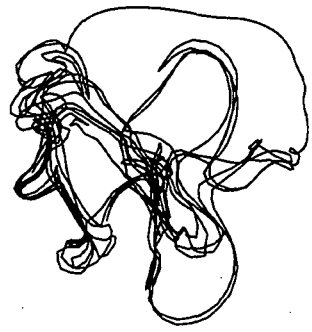
T= 50.00



T= 51.00



T= 52.00



T= 53.00

Figure 4.39: Run refinesigma.9

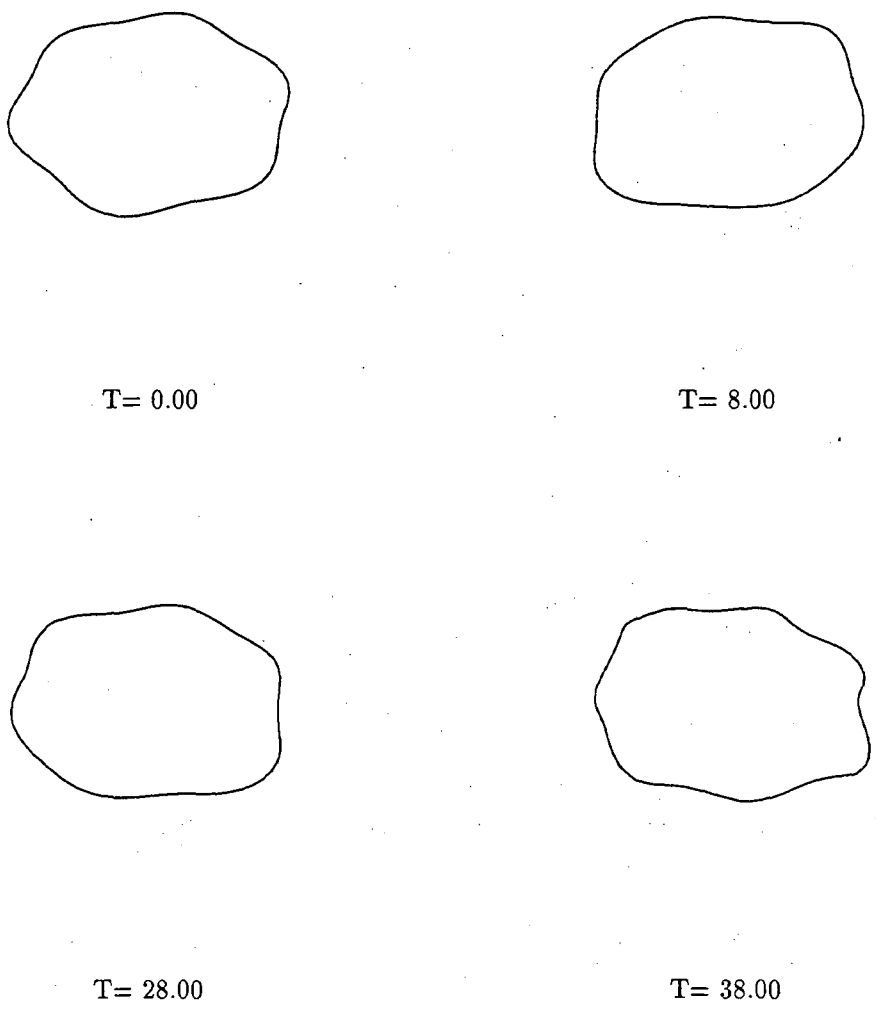
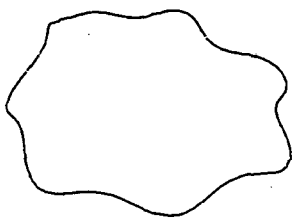
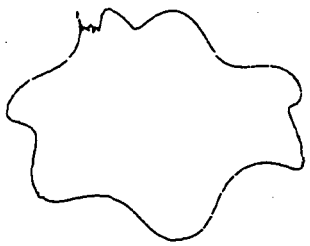


Figure 4.40: Run refinesigma.9



T= 39.00



T= 40.00

T= 41.00

Figure 4.41: Run refinesigma.10

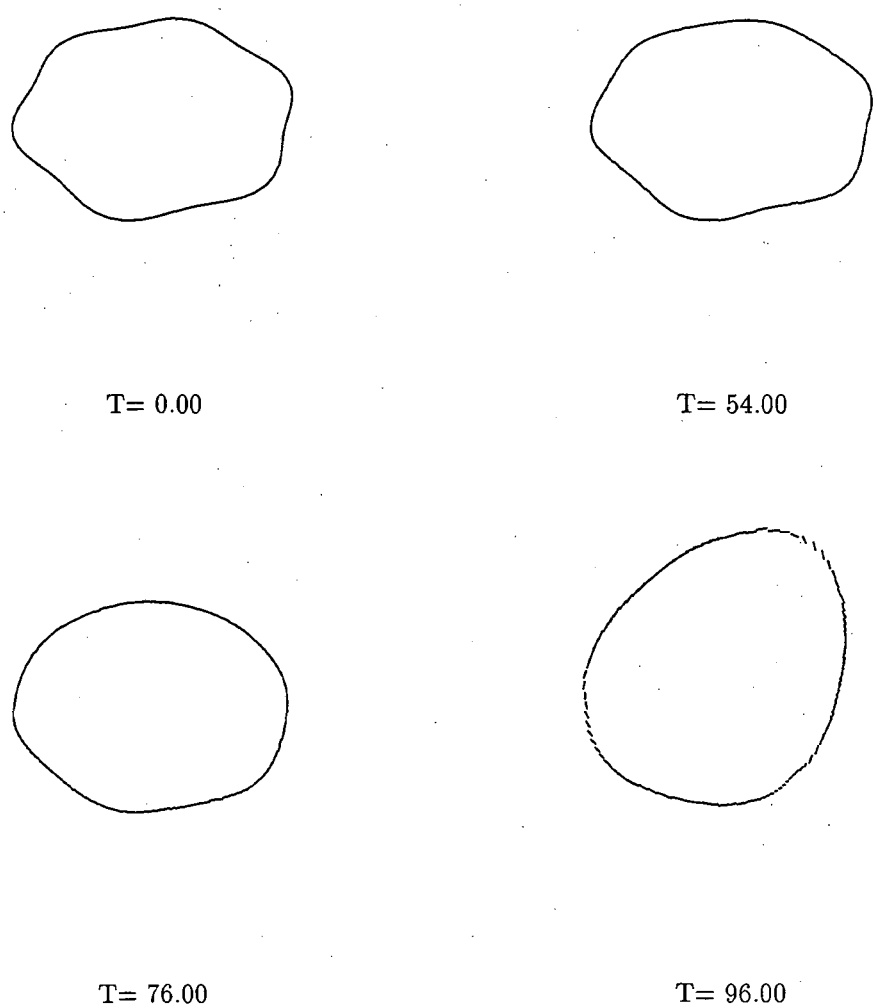
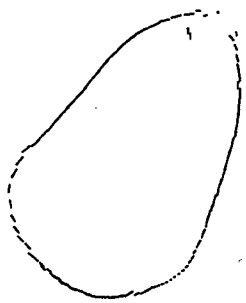


Figure 4.42: Run refinesigma.10



T=104.00

T=111.00

T=118.00

4.6.1 Observations

The original intent of the work presented in this chapter was to make a full comparison of known vortex methods applied to the problem of a single filament ring. This turned out not to be an appropriate objective. From initial runs, it became clear that qualitative factors dominated both the individual computations and the differences between computations.

While quantitative results seem to be inconclusive, or impossible to obtain, the qualitative factors seem to point to real differences between the algorithms and to flow structures which cause difficulty for the methods. Therefore, these qualitative results are emphasized. It is clear that these observations should be a point of departure for more more precise and quantitative comparisons and analyses. It may also be possible to use these observations as a basis for further theoretical work.

Robustness of Results

With the exception of algorithm 2, whatever behavior an algorithm exhibits in the base run seems to persist when Δx and Δt are refined. Algorithm 2, exhibits a large qualitative change when Δx is refined. In particular, when Δx is refined, helical waves appear These waves persist for a long time before the computation breaks down. In many cases the refinedt and refinedx runs, are nearly identical to the corresponding base runs. In other cases the runs are identical under some symmetry transform of the initial data. When this is the case, the refined runs are omitted from the figures.

Catastrophic Growth of The Number Of Elements

In all of the runs except those using algorithm 11, at some point there is a catastrophic growth in the number of vortex elements, which makes the computation intractable beyond that point. Algorithm 11 was specifically designed to smooth the stretch term to such a point that it does not suffer from this problem. It suffers from the problem of oversmoothing. Although it is not visible in the runs, in each case algorithm 8 (exact,blob,2) had the most catastrophic growth. In each case, within $t = 0.4$ of the last picture shown, the number of elements had grown to over 45,000. Algorithm 4, (Chorin Filament,stick,Stick cutoff) seems to suffer from this growth earlier in the computation than do the other algorithms.

Growth Near Low Values of Torsion and Near Zeroes of Vorticity

In all cases, except 8, the growth seemed to occur in several localized areas which then grew to dominate the entire computation. These regions began as nearby nearly antiparallel filaments, or kinked filaments which appear to have near zero torsion. In each of these regions, there is also a zero of the vorticity which is created by the smoothing of the core function. These zeroes are typically isolated. One can infer this from the visualization since the parts of the filament in question appear

to lie in a plane, or in a surface with very small curvatures. This in also seen in the run which reproduces the results of Knio and Ghoniem, as well as in other runs from [33]. The instability of vortex structures near low values of torsion was also noted by Qi in [42].

Loss of Symmetry

In each case, the initial data has a six fold symmetry. In each case, except algorithm 11, this symmetry is lost at some time in the evolution. After this point, smaller scale structures appear to become more important. This loss of symmetry occurs differently for each algorithm. In some cases, a four fold symmetry appears. In others, a five fold symmetry develops. In some cases a two fold symmetry develops. In still others, a complete loss of symmetry occurs. This is unexpected, especially since the linear instability of perturbations of wave number six should force this six fold perturbation to grow in amplitude and thus reenforce the six fold symmetry.

Movement of Particles Along the Filament

In all of the computations, but especially in the filament computations, oscillatory movement of the particles along the filament develops. Particles bunch up at some points, while spreading apart in others. The compressed regions then are rarefied, while the rarefied regions become compressed. The particles continue to oscillate in this fashion.

Development of small scale structure

In each case, very small scale structures eventually become part of the solution. These structures appear to have components of wave numbers which which are at or above the Niquist frequency of a uniform grid of the same spacing as the particles. These structures often start as "hairpin" structures or kinks and often develop into the sort of low torsion areas or isolated zeroes which were mentioned earlier.

Development of nearly helical structures

In many of the runs, structures develop which appear to be helical or nearly so. This is very clear in run RefineDX.2, where helical waves appear and move around the ring. These appear similar to the Hasimoto soliton solution of the local induction approximation[29]. In other cases, the development of these nearly helical structures follows shortly after the development of other small scale structure and only shortly precedes the end of the computation.

4.6.2 Conjectures about causes

The previous section described observations made directly from the visualization of the vortex particles. In this section, speculation on the causes of those phenomena is presented. These

will be expanded upon in later chapters. It is suspected that there are some configurations of vorticity which are not handled correctly by the vortex method and which result in an incorrect solution. The exact nature of these configurations and the exact nature of the miscalculation is not known. These ideas are introduced here and expanded in later chapters.

Instability Of Marker Particles

Sethian [46] showed that using marker particles to follow curves in two dimensions is unstable. In general, particle methods do not do well in cases where the hyperbolic nature of the system causes discontinuities or admits discontinuous solutions. The problem of maintaining arc length parameterization of a curve is difficult[46]. This is what a filament algorithm proposes to do. It is following a curve which is moving in three space dimensions, while trying to keep the filament well resolved by adding points where necessary. Recall that there was oscillation of the markers along the filament. Small low wave number perturbations introduced in the area of expansion will become high wave number perturbations when that area contracts. Also, the compression zones and rarefaction zones are remiscent of one dimensional gas dynamics.

The VTE has hyperbolic character

The VTE has an integral conservation form 5 and is derived as the limiting case of hyperbolic systems. In addition, there is an asymptotic theory of thin filaments with circular cores which links them to the shallow water equations. The numerical evidence presented in chapter 2 on the use of an algorithm in conservation form also points in this direction.

Importance of Zeros of Vorticity

The beginnings of catastrophic growth were seen near isolated zeroes of vorticity. It is possible that these points require special treatment. In particular, near such points the topology of the numerical filaments does not match that of its smoothed counterpart. This will be explored in greater detail in chapter 9.

Importance of Helical Structure

The development of singularities in the curvature of the filament in the solutions seems to occur shortly after the development of small scale helical structures. This suggests that such structures are important to understanding the behavior of the VTE and the vortex method.

Importance of Smoothing

Since the results are dependent upon the amount and type of smoothing used, one would suspect that this is an important parameter in the vortex method. It also suggests the possibility

that each smoothing scheme selects a different solution and that a method is needed to decide which solution is the correct one.

Connection to Lagrangian Mechanics and the Transpose Equation

In chapter 5, it is argued that the VTE is equivalent to the Euler-Lagrange equations for a system with Lagrangian density equal to the helicity density. This Lagrangian system cannot be immediately transformed into a Hamiltonian system because the helicity is not convex as a function of ω. Hamiltonian systems are know to pose special difficulties to numerical methods.

4.7 Future Numerical Work

To further pursue this subject, there are several tools which could be used on the problem. In this section, possible future work is discussed.

While this study has focused on qualitative behavior of the algorithms, along with the calculation of a few interesting quantities, a more quantitative study could also be interesting.

4.7.1 Accurate Computation of Invariants

The Euler equations for incompressible flow have a number of invariants, such as total vorticity, impulse, angular momentum, energy and helicity. The integrands for some of these quantities in vorticity variables are singular, making them difficult to evaluate. The volume estimates of section 8.3 have been used as a means for evaluating these integrands. Total vorticity, impulse angular momentum and helicity seem to be handled well by this scheme, but the (singular) self energy is greatly misestimated. In the work of Chorin, the self energy is evaluated exactly, while the interaction energy is evaluated numerically based on the assumption of the core size being zero. This neglects influence of the core function on the interaction energy of nearby filaments. It is precisely this near neighbor interaction which seems to be part of the problem. Recent work by Strain [52] is aimed at evaluating exactly these kind of integrals and could be used to monitor the invariants associated with the flow.

4.7.2 Fourier Space Evolution and Filtering

The evolution of the system in a transformed set of variables may make more clear what is happening. It would be interesting to take a Fourier transform at each time step and see if the evolution matches expectations. In addition, the system could be filtered in Fourier space as was done by Krasny in two dimensions[34][35]. This could be done either with a standard Fourier transform, or the non-equidistant Fourier transform of Rohklin, of which we learned in [51]. An attempt to evaluate the invariants could also be made using the Fourier space data.

4.7.3 Modeling of Vorticity and Torsion Zeroes

If suitable boundary conditions could be posed, the behavior of vorticity near an isolated zero and near a zero of vorticity could be investigated using a finite difference scheme, such as the projection method[9]. These results could then be compared to the results of the vortex method in these regions.

4.7.4 Front Motion Model Near a Vorticity Zero

Another approach would be to model the vortex lines near a zero as a series of moving fronts, using the level set algorithms of Sethian and Osher [50]. It is shown in Chapter 9, that near a zero of vorticity, the vortex lines are essentially two dimensional. This combined with the fact that vortex filaments are materially advected yields a set of moving front problems in two dimensions.

Chapter 5

Theoretical Considerations

In this section, the vorticity transport equation is cast in a conservative form. Other theoretical developments are also discussed, which show that the VTE has connections to several types of equations which are know to be difficult to handle numerically, and which require the use of some kind of diffusive term in order to get a physically relevant solution and/or a solution which is no more unstable than the physical solution. There are indications that the VTE is connected to Lagrangian mechanics, Hamiltonian systems and hyperbolic conservation laws.

5.1 Derivation of Conservation Form

The starting point for this derivation is the vorticity transport equation

$$\partial_t \omega_i + u \cdot \nabla \omega_i = \omega \cdot \nabla u_i \tag{5.1}$$

Since both $\nabla \cdot u$ and $\nabla \cdot \omega$ are zero, terms proportional to them may be added to both sides.

$$\partial_t \omega_i + u \cdot \nabla \omega_i + \omega_i \nabla \cdot u = \omega \cdot \nabla u_i + u_i \nabla \cdot \omega \tag{5.2}$$

Next, the vector identity, for F a vector and f a scalar, will be required

$$\nabla \cdot (fF) = f \nabla \cdot F + F \cdot \nabla f \tag{5.3}$$

$$\partial_t \omega_i + \nabla \cdot (u\omega_i) = \nabla \cdot (\omega u_i) \tag{5.4}$$

$$\partial_t \omega_i + \nabla \cdot (u\omega_i - \omega u_i) = 0 \tag{5.5}$$

The previous equation is an Eulerian conservative form for the vorticity transport equation. This can also be expressed in a Lagrangian form using the material derivative.

$$D_t \omega_i + \nabla \cdot (-\omega u_i) = 0 \tag{5.6}$$

This indicates that along the characteristics of the advection part of the equation, the evolution of the values of the vorticity takes a conservative form.

5.2 Divergence Of Vorticity Is Material

Some vortex methods use various devices to keep the vorticity field divergence free, while other calculations have been carried out which do not impose this constraint (see chapter 3). In order to examine the importance of this constraint, an evolution equation for $Q = \nabla \cdot \omega$ is developed in the absence of the constraint that $Q = 0$

Suppose that the VTE is given, but that the restriction, $\nabla \cdot \omega = 0$, is removed. In this case, an equation for the evolution of $Q = \nabla \cdot \omega$ can be derived. Start with the three equations

$$\partial_t \omega_i + u \cdot \nabla \omega_i = \omega \cdot \nabla u_i + \epsilon \Delta \omega_i \tag{5.7}$$

where ϵ is $\frac{1}{R}$, and R is the Reynolds number[2]. For each equation, take the derivative with respect to x_i and sum the resulting three equations and put everything in coordinates. The result is

$$\begin{aligned}
\sum_i \partial_i \partial_t \omega_i + \sum_i \partial_i \sum_j u_j \partial_j \omega_i &= \sum_i \partial_i \sum_j \omega_j \partial_j u_i + \sum_i \epsilon \Delta \omega_i \\
\partial_t \sum_i \partial_i \omega_i + \sum_i \sum_j \partial_i (u_j \partial_j \omega_i) &= \sum_i \sum_j \partial_i (\omega_j \partial_j u_i) + \epsilon \Delta (\nabla \cdot \omega) \\
\partial_t \sum_i \partial_i \omega_i + \sum_i \sum_j (\partial_i u_j \partial_j \omega_i + u_j \partial_{ij} \omega_i) &= \sum_i \sum_j (\partial_i \omega_j \partial_j u_i + \omega_j \partial_{ij} u_i) + \epsilon \Delta (\nabla \cdot \omega) \\
\partial_t \sum_i \partial_i \omega_i + \sum_i \sum_j u_j \partial_j \partial_i \omega_i &= \sum_i \sum_j \omega_j \partial_j \partial_i u_i + \epsilon \Delta (\nabla \cdot \omega) \\
\partial_t (\nabla \cdot \omega) + \sum_j u_j \partial_j (\nabla \cdot \omega) &= \sum_j \omega_j \partial_j (\nabla \cdot u) + \epsilon \Delta (\nabla \cdot \omega) \\
\partial_t Q + u \cdot \nabla Q &= \epsilon \Delta Q
\end{aligned}$$

The last equation indicates how $Q = \nabla \cdot \omega$ is transported, when it is left unconstrained and reveals several interesting facts about this case. First, $\nabla \cdot \omega$ is a material quantity. As such, $\nabla \cdot \omega$ obeys a maximum principle; if $\epsilon > 0$ and if the initial condition for the vorticity transport equation has $|\nabla \cdot \omega| < \alpha$ for some $\alpha > 0$, then solutions of the vorticity transport equation will have $|\nabla \cdot \omega| < \alpha$ for all time. Any solution for this equation with $\epsilon = 0$ which is a limit of solutions with $\epsilon > 0$ also obeys a maximum principle.

Thus, any method in which $\nabla \cdot \omega$ does not satisfy a maximum principle is computing a solution which is not a limit of viscous solutions.

One would also like to conclude that any method which does not require initial data with $Q = 0$ exactly, must contain some artificial viscosity in order to be stable, since otherwise, instabilities in the underlying transport equation for Q will lead to non-physical solutions with $\nabla \cdot \omega$ unbounded. However, since we have not been able to demonstrate truly discontinuous or non-unique solutions to the VTE itself, including $\nabla \cdot \omega = 0$, this conclusion cannot yet be drawn.

A hyperbolic subsystem of the VTE, other than simple advection, has been demonstrated.

5.3 Interpretation of Non-solenoidal Vorticity

When the quantity Ω, which is supposed to be the vorticity, is allowed to be non-solenoidal, Ω can be written $\Omega = \omega + \nabla\alpha$, where $\omega = \nabla \times u$. That is, Ω is allowed to drift off of the subspace to which it should be confined. When this freedom is allowed, some interesting facts surface. If the vorticity starts on this subspace, and there is any viscosity, then the vorticity will remain on this subspace. If the vorticity starts off of this subspace, and there is any viscosity, and the average of $\nabla \cdot \Omega$ is zero, the vorticity will eventually get as close as desired to this subspace.

The case of zero viscosity is now investigated. There are several ways to interpret what might happen. One way is to notice that a gauge freedom has been added to the system which is not part of the original equations. This causes solutions to be nonunique, with each solution corresponding to a different gauge. Note that this freedom does not effect the velocity, since the velocity induced by Ω is equal to the velocity induced by ω. Numerically, with no viscosity, one might expect to get large contributions to Ω from the most unstable (in time) of the possible solutions, Ω, since those are the ones which will be most amplified (in time) by the numerical scheme.

5.4 The Transpose Equation is an Euler Lagrange Equation

The vorticity transport equation, with $\nabla \cdot \omega = 0$ is equivalent to an equation called the *transpose equation* [43][25]. One may start with

$$D_t\omega = \omega \cdot \nabla u \tag{5.8}$$

and apply the vector identity $(a \cdot \nabla b - a \cdot (\nabla b)^T) = \nabla \times u \times \omega$ together with $\nabla \cdot \omega = 0$, see that

$$D_t\omega = \omega \cdot (\nabla u)^T \tag{5.9}$$

It will be shown that, in the case of discrete vortex elements, the transpose equation is actually the Euler-Lagrange equations for a Lagrangian density equal to the helicity of the system. The vorticity transport equation possesses at least two quadratic invariants, the energy $\int u \cdot u dx$ and the helicity $\int \omega \cdot u dx$. Suppose that a Lagrangian density equal to the helicity density, is postulated. The resulting Euler-Lagrange equations determine the motion of the postulated system. In the case of discrete vortex elements, $L(x_i, \dot{x}_i, t) = \sum_i w_i \cdot u(x_i, t)$. Viewing the x_i^j as the coordinates, usually denoted by q_i^j, the Euler-Lagrange equations for this system are derived as follows

$$\begin{aligned} D_t\dot{q}_i^j - \frac{dL}{dq_i^j} &= 0 \\ D_t\omega_i^j - \frac{dL}{dx_i^j} &= 0 \\ D_t\omega_i^j - \partial_{x_i^j} \sum_k \omega_i^k u^k(x_i) &= 0 \end{aligned}$$

$$
\begin{aligned}
D_t\omega_i^j - \sum_k \omega_i^k \partial_{x_i^j} u^k(x_i) &= 0 \\
D_t\omega_i^j - \sum_k \omega_i^k \frac{\partial u^k}{\partial x^j}(x_i) &= 0 \\
D_t\omega_i - \omega_i \cdot (\nabla u)^T &= 0 \\
D_t\omega_i &= \omega_i \cdot (\nabla u)^T
\end{aligned}
$$

Given a Lagrangian system with a convex Lagrangian, the Legendre transformation will give an equivalent Hamiltonian system. Unfortunately, the helicity is not convex as a function of $\dot{q} = u(x_i)$ this path cannot be followed in order to find a Hamiltonian for the vorticity system.

5.5 Some Previously Known Results

In this short section, references to some previously know results are given. These results also connect the incompressible flow system and the VTE to areas similar to the ones of this chapter.

Incompressible flow is abstractly Hamiltonian

Marsden and Weinstein proved that the incompressible flow system is Hamiltonian in an abstract sense connected with symplectic geometry [38].

In velocity-pressure variables, incompressible flow is the projection of a hyperbolic system

The incompressible flow system in velocity-pressure variables is a hyperbolic system which has been restricted to a subspace, namely the incompressible subspace of $\mathcal{L}^2$. This restriction eliminates certain types of discontinuities, and the hyperbolicity of the system. However, it still leaves some properties of hyperbolic systems intact[23][10].

In magnetization variables, incompressible flow is Hamiltonian

Buttke has discovered that the VTE is equivalent to a system in variables called *magnetization variables*. This system is Hamiltonian with a Hamiltonian function equal to the energy [6][7][41]. The derivation of this Hamiltonian from the energy is similar to the derivation of the Euler-Lagrange form of the transpose equation. The magnetization variables automatically satisfy the condition, that ω be incompressible. The price of this is that the size of the support of the magnetization may grow very rapidly.

Analogy to the Shallow Water Equations

The system governing core area varying waves of a vortex ring with a core of circular cross section is analogous to the shallow water equations. The argument here is reproduced from [53] who drew the result from [37] and [39]. First, one writes incompressible flow equations for a vortex filament in velocity-pressure variables, in axisymmetric cylindrical coordinates, with the z-direction in the direction of the vorticity and the origin on the center of the filament. Let r be the radial component, z, the axial component, u the radial velocity, w the axial velocity the system becomes

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial r} + w\frac{\partial u}{\partial z} - \frac{\nu^2}{r} = \frac{-1}{\rho}\frac{\partial p}{\partial r} \tag{5.10}$$

$$\frac{\partial w}{\partial t} + u\frac{\partial w}{\partial r} + w\frac{\partial w}{\partial z} = \frac{-1}{\rho}\frac{\partial p}{\partial z} \tag{5.11}$$

One then assumes that u and $\frac{Du}{dt}$ may be neglected. Under these assumptions, the axial pressure gradient can be derived.

$$p - p_\infty = -\frac{\rho\Gamma^2}{8\pi A} \tag{5.12}$$

where, A is the area of the circular core and Γ is the circulation. The equations then become

$$\frac{\partial A}{\partial t} + \frac{\partial Aw}{\partial z} = 0 \tag{5.13}$$

$$\frac{\partial w}{\partial t} + w\frac{\partial w}{\partial z} = -\frac{1}{A}\frac{\partial}{\partial z}\left(\frac{\Gamma^2 \ln A}{8\pi}\right) \tag{5.14}$$

These equations are the shallow water equations with ρ replaced by A and p replaced by $\frac{\Gamma^2 \ln A}{8\pi}$. This connects core area varying waves on a vortex ring with a core of circular cross section to the movement of waves in shallow water which is known to be hyperbolic and known to develop discontinuous solutions[36].

Chapter 6

A Lax-Friedrichs Scheme for Vorticity Transport

We have seen that the vorticity transport equation can be cast in a conservative form and that that form has at least two terms which can be considered as strictly hyperbolic. However, we do not know how to solve a Riemann problem for this system. Even posing a Riemann problem for this system is difficult. There is reason to conjecture that this equation has some hyperbolic character, since the equivalent system in velocity pressure variables is hyperbolic and in the equivalent system in magnetization variables is a Hamiltonian system. In addition, we have seen that the vorticity transport equation is an Euler-Lagrange equation.

Given this conjecture combined with the lack of Riemann problem, the upwind schemes are out of reach. A numerical option which respects the hyperbolic nature of the equation is the Lax-Friedrichs algorithm. In order to explore the conservation form of the equations numerically, this algorithm was implemented, and several test problems were calculated. Until June 1, 1995, the code for this scheme along with the initial data and example output files are available from the author (pryor@math.berkeley.edu) upon request by electronic mail (pryor@math.berkeley.edu).

Recall that the conservative form of the vorticity transport equation for vorticity component n is given by

$$\partial_t \omega_n + \nabla \cdot (u\omega_n - \omega u_n) = 0 \tag{6.1}$$

6.1 The Discretized VTE

If we define the following quantities,

$$\begin{aligned}\omega_n^{ave}(i,j,k) = \frac{1}{6}(&\omega_n(i+1,j,k) + \omega_n(i-1,j,k) \\ &+\omega_n(i,j+1,k) + \omega_n(i,j-1,k)\end{aligned}$$

$$+\omega_n(i,j,k+1)+\omega_n(i,j,k-1))$$

$$H_n(i,j,k) = u(i,j,k)\omega_n(i,j,k) - \omega(i,j,k)u_n(i,j,k)$$

$$\begin{aligned}\mathrm{Div}_0[H_n](i,j,k) = &\frac{H_n(i+1,j,k)-H_n(i-1,j,k)}{\Delta x}\\ &+\frac{H_n(i,j+1,k)-H_n(i,j-1,k)}{\Delta y}\\ &+\frac{H_n(i,j,k+1)-H_n(i,j,k-1)}{\Delta z}\end{aligned}$$

then the Lax-Friedrichs scheme for the vorticity transport equation is

$$\omega_n(i,j,k,t+dt) = \omega_n^{ave}(i,j,k) + \Delta t \cdot \mathrm{Div}_0[H_n](i,j,k)$$

Given the vorticity and velocity at any given time t, this allows us to find the vorticity at time $t+\Delta t$. The only details remaining are evaluation the velocity from the vorticity, boundary conditions and initial data.

6.2 Boundary Conditions

Since the interest here is in exploring the nature of the equations, not in solving real problems, the boundary conditions are taken to be periodic in all three dimensions.

6.3 Evaluation of the Velocity Field

To calculate the velocity given the vorticity, a package is used to solve Poisson's equation for the stream function. The stream function is then numerically differentiated to obtain the velocity.

This was tested with two different packages. CFFT3D was used on the Cray XMP,YMP and C90 to find the Fourier transform of the vorticity. The Poisson equation was then solved by division in the Fourier domain and the result was inverse transformed to give the numerical stream function. On other machines, as well as on the Cray, HW3CRT from FISHPAK was used to solve the Poissons equation.

6.4 Initial Conditions

The method of initializing the vorticity which gave the best results was to initialize a grid with a velocity field and then numerically take its curl to produce the initial vorticity field. One other method was tried. In this case, the grid was initialized by point sampling a vorticity field or a low pass filtered vorticity field and then projecting the result onto the space of (numerically)

divergence free fields. This typically produced vorticity fields which were either much weaker than the original field due to the low pass filtering, or whose support was too small to be resolved on the grid due to the point sampling and the low order of the method.

6.5 Checking the Code

This code that was written makes some attempt to check the results. At each time step, several diagnostic quantities are evaluated.

- the residual resulting from the solution of the Poisson problem
- the amount by which the right hand side fails to meet the compatibility condition.
- the L^1, L^2 and max norms of the divergence of the velocity
- the L^1, L^2 and max norms of the divergence of the vorticity
- the total kinetic energy
- the difference between the curl of the induced velocity field and the vorticity field which gives rise to it.

6.6 Test Problem Results

The first test problem that was solved with this scheme is a two dimensional periodic shear layer. This problem has been solved many times, so one knows what to expect[35]. Several things are of interest here. Does the scheme preserve the two dimensional nature of the problem? Under what conditions is the scheme stable? Does the scheme keep the divergence of the vorticity small? Are energy, helicity, impulse and angular momentum non-increasing? What is the effective numerical viscosity?

The second problem which was explored is a shear layer with a three dimensional perturbation. In this problem, vortex stretching is important and the solution is not know beforehand.

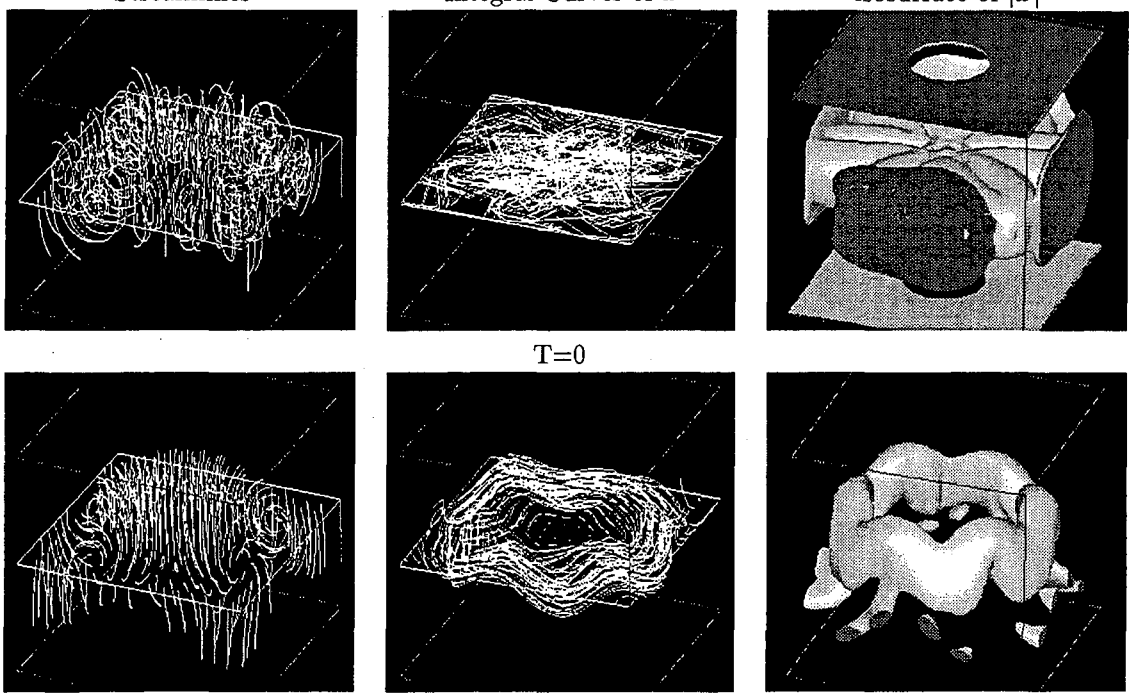
A periodic array of vortex rings was the last problem for which this code was used. In order to get any interesting results in this case, a variable time step was needed, since this reduced the effective viscosity as the strength of the ring decayed, allowing it to exist for longer times.

In all cases, the divergence of the vorticity field remained small and the conserved quantities were non-increasing. The scheme appears to be stable for CFL numbers less than $\frac{1}{2}$. For the two dimensional problem, the symmetry which yields the two dimensional nature of the problem was retained throughout the calculation without explicit enforcement of this symmetry.

Figures 6.6 and 6.6 show the computed evolution of a Gaussian ring with circulation 1, core cutoff of 0.1, centerline radius 0.25, perturbed at wave number 6 with a perturbation of size 0.3. The grid used was 64x64x64

Figure 6.1: Perturbed Ring On A 64x64x64 Grid

Streamlines | Integral Curves of ω | isosurface of $|\omega|$

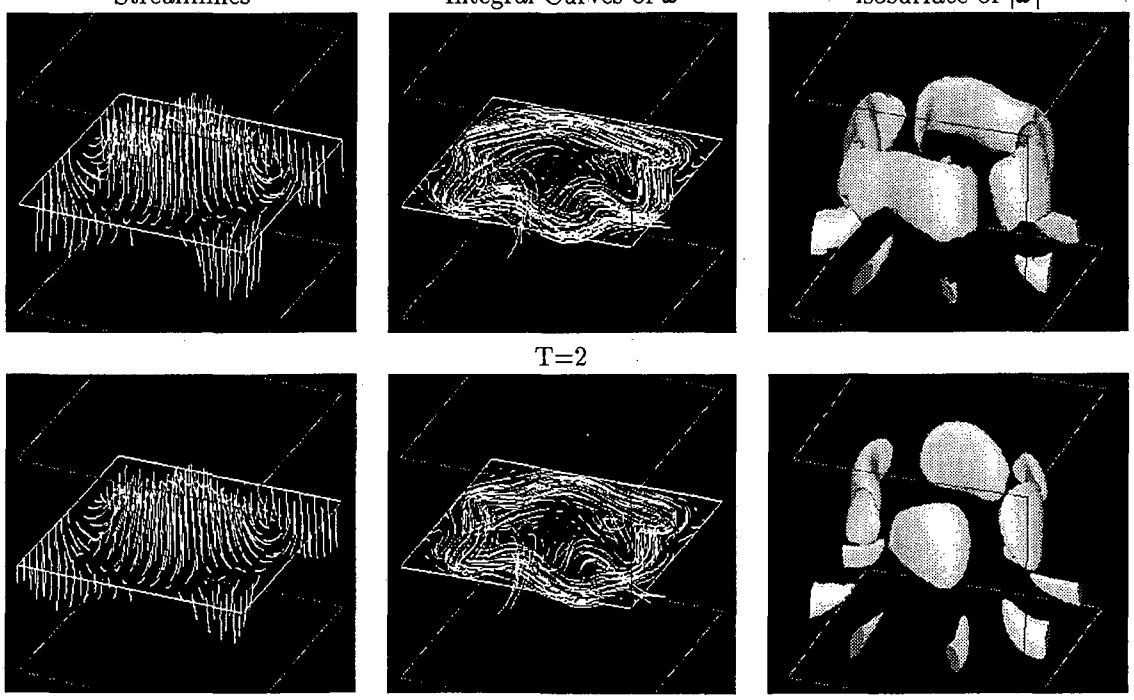


T=0

T=1

Figure 6.2: Perturbed Ring On A 64x64x64 Grid (contd.)

Streamlines | Integral Curves of ω | isosurface of $|\omega|$



T=2

T=3

6.7 What Was Learned

The conclusions which were drawn from this exercise are as follows

- Given enough viscosity, one can stabilize both $J = \nabla \cdot \omega$ and vortex stretching in an Eulerian setting.
- A Lax Friedrichs scheme for the unconstrained vorticity transport equation is stable for CFL number < 0.5
- The LF scheme gives the correct evolution of $J = \nabla \cdot \omega$
- A projection scheme for the vorticity, without added viscosity is not stable. J remains small, but the conserved quantities grow. So, projection is not enough to stabilize the system.
- The LF scheme causes the conserved quantities to be non-increasing.
- Central differencing on this problem is unstable.
- The LF scheme in three dimensions preserves the two dimensionality of two dimensional problems.
- The LF scheme adds too much numerical viscosity to be useful for reasonable grid sizes.

Chapter 7

On Projection Schemes To Enforce $\nabla \cdot \omega = 0$

7.1 Motivation

Given that the constraint that the divergence of the vorticity remain zero seems to be very important, but is not naturally enforced in inviscid flow, one should look for a way to enforce it.

The approach of defining the problem away, by using the Z function of Hald is appealing, [28]. However, this solution has one disturbing property, which is most easily seen by considering the point vortex method. The case with a core function is similar. In this case, Z is $\frac{2}{3}$ times a delta function, plus a term which decays like $\frac{1}{r^3}$. This implies that the velocity at the location of the particle, determines, in part the motion of vorticity very far away from the particle, and that this effect does not vanish as the size of the smoothing radius is reduced. This appears to be inconsistent with the material transport of vortex lines. However, it is important to note that convergence has been proved for this method when the vorticity starts and remains sufficiently smooth [28].

In this chapter, results of attempts to impose the constraint on the vorticity via a projection scheme are reported. In doing so, several interesting properties of the representation of vorticity used in vortex methods were discovered. These will be discussed more fully in chapter 8 on resampling, however they were first discovered in the context of projection.

7.2 Hodge Decomposition

One possible method to keep the vorticity divergence-free would be to use the Hodge decomposition which states that any smooth vector field, u can be written as the sum of a divergence-free field and the gradient of a scalar.

$$u = v + \nabla\phi \tag{7.1}$$

Note that this implies a decomposition of L^2 into two perpendicular subspaces, one of which contains the gradients of scalars and the other of which contains all divergence-free fields. What is desired here is a projection onto the space of divergence-free fields. Following Chorin[9] and taking the divergence of both sides, one obtain the differential equation

$$\nabla \cdot u = \Delta \phi \tag{7.2}$$

This together with suitable boundary conditions determines ϕ, and thus determines the projection. This procedure is the basis of the projection method of Chorin for incompressible flow [9][10]. In that case, he uses the projection at each time step to keep the velocity field solenoidal. In theory, one should be able to use such a projection to keep the vorticity field divergence-free.

7.3 The Projection in a Lagrangian Context

7.3.1 Derivation of An Algorithm

It is clear how to implement such a projection in the context of a grid based method. This was in fact done as part of the experiments with conservative difference schemes for the VTE. An attempt was made to implement it in the context of vortex methods. Recall that in a vortex method, each blob or stick carries a strength which is related to the total circulation about the filament to which it belongs. For particle i, call this weight w_i. What is sought then, is a prescription for altering these weights in such a way that the induced vorticity field is solenoidal, or nearly so in some sense. Begin with the smooth non-solenoidal vorticity.

$$\omega(x) = \sum_i w_i * f_\delta(x - x_i) \tag{7.3}$$

$$\nabla \cdot \omega \neq 0 \tag{7.4}$$

It is easy to demonstrate that the divergence-free part of ω is given by $\nabla \times u$ where u is the velocity field induced by ω through the Biot-Savart law[28]. So, one possible way to find a divergence-free smooth field would be to solve the following equation for new weights W_i

$$\nabla \times u(x) = \sum_i W_i * G(|x - x_i|/\delta) \tag{7.5}$$

The above equation is a linear equation for W_i. If divergence-free is chosen to mean divergence-free at each of the particle positions, what results is a set of N equations in N unknowns, of the form $Ax = b$.

$$\nabla \times u(x_j) = \sum_i W_i * G(|x - x_i|/\delta) \tag{7.6}$$

The solution of this equation then provides a prescription for altering the w_i to get W_i which leads to a divergence-free vorticity field.

7.3.2 Test Implementation

Code was written which creates the matrix A and solves the equation $Ax = b$ by first taking an LU decomposition of A. and then back solving for x. The LU decomposition was used in order to evaluate the condition number of A in order to see if the problem is well posed and well conditioned. It was found that the closer the vortex elements got to each other and the more their core functions overlapped, the worse the condition of A became. In fact, the condition number became huge ($> 10^9$) for reasonable configurations of elements. This is a mystery, since the projection itself should be well conditioned.

7.3.3 What causes the large condition number

The matrix equation to be solved was then examined further. For a core function G which is a Gaussian, A is the forward evolution operator for the heat equation. Thus, solving $Ax = b$ is equivalent to solving the backwards heat equation. The easiest way to see this is to note that the spectrum of a Gaussian has exponential decay at infinity, hence the inverse of convolution with a Gaussian must cause wave numbers to be multiplied by a function which grows exponentially with wave number. This makes any algorithm for solving the inverse heat equation unstable since the errors can be made to grow arbitrarily quickly by choosing it to be in a high enough wave number.

Any core function which is both infinitely differentiable and decays exponentially at infinity will have the property that its spectrum decays exponentially at infinity, which will lead to the instability of the inverse convolution. This fact has implications for any resampling of vorticity or rearrangement of the weights which will be more fully explored in chapter 8

7.4 Stability of the Problem

In this section the problem of projecting the vorticity onto the space of divergence-free fields by using the vorticity weights is examined.

Suppose once again that one is given particles with strengths. $\sum_i \Omega_i f_\delta = \sum_i \omega_i g_\delta$ where g_δ is the divergence-free field induced by a vorticity distribution f_δ. This system then has the form $Ax = By$, where both A and B are very ill-conditioned.

$x = A^{-1}By$

Suppose this problem is brought back into the continuous domain so that A_δ and B_δ are operators. A_δ is a smoothing convolution. Let P be the projection onto the divergence free part of L^2. P takes divergence-free fields to themselves with eigenvalue 1 and its null space is the gradients of scalars (eigenvalue 0). In this case, $B = PA_\delta$.

Thus, the problem looks like $A_\delta v = PA_\delta u$, and the solution is $v = A_\delta^{-1} PA_\delta u$, if we can compute it.

It may now be shown that even in the fortunate case that A_δ and P commute, the problem is still ill posed. In the case that A_δ is a Gaussian, A_δ is a forward time evolution operator for a heat equation. In this case, it is easy to analyze what is going on by using the connection to the heat equation.

$$\partial_t u = \Delta u \tag{7.7}$$

$$\partial_t \nabla \cdot u = \Delta \nabla \cdot u \tag{7.8}$$

$$u(x,t) = A_\delta u(x,0) \tag{7.9}$$

$$\nabla \cdot u(x,t) = A_\delta \nabla \cdot u(x,0) \tag{7.10}$$

$$\nabla \cdot u(x,t) = \nabla \cdot A_\delta u(x,0) \tag{7.11}$$

$$A_\delta \nabla \cdot u = \nabla \cdot A_\delta u \tag{7.12}$$

A_δ takes divergence-free fields to divergence-free fields, hence preserving that subspace in the forward direction. A_δ also takes fields of average divergence k and maximum divergence $0 < J_{max}(0) \neq 0$ to fields of average divergence k and smaller but nonzero maximum divergence. $0 < J_{max}(1) < J_{max}(0)$. A_δ takes non-divergence-free fields to non-divergence-free fields. Thus one may conclude that A_δ and P commute. $v = A_\delta^{-1} P A_\delta u = Pu$

However, in the case which is useful to us, u is a function which is a sum of delta functions, and it is demanded that v also be a sum of delta functions, supported at the same locations as those which define u. Pu makes sense for such u since we can solve $\Delta w = \delta$. However, in this case, the projection is no longer represented by a sum of point masses. It is a sum of point masses and functions which are $\nabla \frac{a}{|x-x_0|}$ for some point x_0 and real number a

The point of this exercise is to show two points

- The projection P is not well defined on the space which is used to represent the vorticity.
- The process of resampling onto the space on which the vorticity is represented is unstable.

The combination of these two observations makes it unlikely that any stable formulation of the projection problem will be found.

Chapter 8

Resampling and Numerical Non-uniqueness

In this section, the relation between the singular and smooth vorticity fields is explored. In addition, the mapping of operations from one domain to the other is investigated. This map is unimportant in two-dimensions, since the weights are not changed. However, in three dimensions, there is an evolution equation for the weights as well as the positions. This phenomenon was first discovered in the context of the projection problem discussed in chapter 7.

8.1 Methods of Updating the Weights

There are two ways of manipulating the weights of vortex particles which have been used in the context of the vortex method. The weights must be manipulated for stretching and for deterministic diffusion schemes. The first method, which will be called *deconvolution resampling* is to compute the evolution of the smooth field and then attempt to resample this back onto the vortex elements, by solving a linear equation which undoes the effects of the core function. This is the method which caused so much trouble in chapter 7 on projection. It will be seen later that this method is in general unstable. The second, which will be called *volumetric resampling*, is to assign a volume to each particle and then point sample the changes in the smooth field back onto the vortex elements. As an illustration, both schemes are detailed for the stretch term. First, deconvolution resampling:

$$D_t\omega = \omega \cdot \nabla u \tag{8.1}$$

$$\omega(x_i) = \sum_j \omega_j f_\delta(x_i - x_j) \tag{8.2}$$

$$D_t\omega(x_i) = \omega(x_i) \cdot \nabla u(x_i) \tag{8.3}$$

$$\sum D_t\omega_j f_\delta(x_i - x_j) = \omega(x_i) \cdot \nabla u(x_i) \tag{8.4}$$

$$A = (A_{ij}) \tag{8.5}$$

$$A_{ij} = f_\delta(x_i - x_j) \tag{8.6}$$

$$A_{ij} D_t\omega_j = \omega(x_i) \cdot \nabla u(x_i) \tag{8.7}$$

$$\tag{8.8}$$

Note that A is symmetric. The last in N equations for the N $D_t\omega_j$ Solving this would give high order accuracy, but it looks like the backwards heat equations.

Second, volumetric resampling:

$$D_t\omega = \omega \cdot \nabla u \tag{8.9}$$

$$\omega(x_i) = \sum_j \omega_j f_\delta(x_i - x_j) \tag{8.10}$$

$$\omega_j = v_j\omega(x_j) \tag{8.11}$$

$$\omega(x_i) = \sum_j v_j\omega(x_j) f_\delta(x_i - x_j) \tag{8.12}$$

$$D_t\omega(x_i) = \omega(x_i) \cdot \nabla u(x_i) \tag{8.13}$$

$$v_i D_t\omega(x_i) = v_i\omega(x_i) \cdot \nabla u(x_i) \tag{8.14}$$

$$D_t\omega_i = \omega_i \cdot \nabla u(x_i) \tag{8.15}$$

$$\tag{8.16}$$

The difficulty with this is in the assignment of volumes, since the flow map distorts shape of the volumes so greatly, but the integration remains spherically symmetric.

8.2 Numerical Non-uniqueness of Vortex Weights

The vorticity field that one would like to converge to a solution of the incompressible Navier-Stokes equation is the curl of the smooth velocity field. One may take the point of view that the vortex elements and weights themselves are a purely artificial construct, while the induced velocity and vorticity fields may be given some physical meaning.

Given this point of view, several interesting questions naturally arise, which are related to the uniquness of the vortex elements and their weithts.

- Does $\omega(x) = G * \Omega$ have solutions which are unique?
- Does $\omega(x) = G * \Omega$ have solutions which can be computed?
- There are natural notions of "close" in the smooth domain, How do they map back into the singular domain?

The answer to the first question is yes and the answer to the second is no, since the equation is the backwards heat equation. The third is more interesting. It turns out that many different distributions of vortex elements can represent flows which to the machine look the same.

Suppose that $|G * \Omega - G * \Omega'| < \epsilon$, for some function norm, then how may Ω and Ω' differ. This is easiest to answer in the context of a specific core function, and norm, although the argument may be made for any core function. For this example, a Gaussian core and the max norm will be used.

Suppose that $G = \frac{1}{\sqrt{\pi}\alpha^2} e^{-\frac{x}{\alpha}^2}$ then, $\hat{G} = \frac{1}{\sqrt{2\pi}} e^{\frac{\alpha^2 k^2}{4}}$. By how much may $\hat{\Omega}(k)$ differ from $\hat{\Omega}'(k)$ and still have convolutions with G which differ by at most ϵ. A bit of algebra shows that $k > \frac{2}{\alpha}\sqrt{|\log \sqrt{2\pi}\epsilon|}$ implies that $\hat{G}(k) < \epsilon$ For fixed ϵ the k for which $\hat{G} < \epsilon$ is proportional to $\frac{1}{\alpha}$, which should be expected with a Gaussian core.

For $\epsilon = 2^{-24}$ (single precision numbers), the above condition on k becomes $k > \frac{7.92}{\alpha}$ for example, $\alpha = .066$ implies that for $k > 128$, $\hat{G}(k) < \epsilon$ This implies a freedom to rearrange the high wave numbers of Ω and still describe induced velocity and vorticity fields, which are close.

This ill-posed inverse convolution lurking in any resampling of the vorticity raises several interesting questions and several possibilities.

- Why does this not manifest itself in the 2-d vortex method? Because in two dimensions, the weights are never manipulated.
- Does this cause the vortex filament or stick description of a smooth vorticity field to be non-unique? It is numerically non-unique, and unique only in the sense that solutions to the backwards heat equation are unique
- Is it possible to pick on of these solutions which is more tractable than another? Is it possible to use this nonuniqueness to our advantage?
- Is there any stable way of rearranging the weights?
- Is this part of the reason that the accuracy of vorticity exchange methods for the viscous part of flow in two dimensions is much lower than expected, even when definite volumes can be assigned to particles?

8.3 Volume Estimation

Volumetric resampling schemes used in the vortex method are based upon writing the weight as an integral of the vorticity over some volume, such as one of the following.

$$\omega_i \approx \omega(x_i) dV_i \tag{8.17}$$

$$\omega_i \approx \int_i \omega dV \tag{8.18}$$

In this section, two methods of assigning volumes to the points are explored.

8.3.1 Using The Core Function

The first method uses the core function and the estimate:

$$\int f_\delta(x - x_i)dV \approx \sum_j f_\delta(x_j - x_i)dV_j \tag{8.19}$$

for some assignment of volumes dV_j.

Assume that for x_i close to x_j, dV_i is close to dV_j. This is basically assuming a smooth variation in density. Then since only x_j close to x_i contribute to the integral, one may write,

$$\int f_\delta(x - x_i)dV \approx dV_i \sum_j f_\delta(x_j - x_i) \tag{8.20}$$

or

$$dV_i \approx \frac{\int f_\delta(x - x_i)dV}{\sum_j f_\delta(x_j - x_i)} \tag{8.21}$$

$$dV_i \approx \frac{1}{\sum_j f_\delta(x_j - x_i)} \tag{8.22}$$

Using volume estimate to do volumetric resampling is convenient in several ways. It allows one to avoid the problem of assigning volumes to particles created on the boundaries of the flow and it accounts for the fact that the flow map radically distorts volumes.

8.3.2 Simple Density Estimates

Very Naive Density Estimates

The second method uses a simple estimate of the density of particles in the region surrounding a particle. In the simplest version of this method, one counts the number of particles in a sphere or radius R about a particle. Call the number of particles $N(R)$ and the volume of the sphere $V(R)$. One then assigns that particle a volume $dV_i = V(R)/N(R)$, or $\frac{1}{\overline{\rho}(R)}$, where $\overline{\rho}(R)$ is the average density of particles in a sphere of radius R.

Slightly Less Naive Density Estimates

The difficulty with the naive scheme is that particles are either in or out of the volume. Unless there are a very large number of particles in the volume, this will make the volume estimates very noisy. To remedy a smoother counting method is required. One way to do this is by the connection to integration which was used in the core function based method. The naive density method is based upon integrating the function 1 over the sphere $S(R)$, and assuming that the local variations in density can be neglected. Suppose the same assumptions are made, but the function

integrated is $g(r)$ where $g(r) = 1$ for $r < R/2$ and $g(r) = 2r/R$ for $R/2 < r < R$ This assigns a volume

$$dV_i = \frac{\int_{S(R)} g(r)}{\sum g(x_j - x_i)} \tag{8.23}$$

where the sum is taken over all x_j such that x_j is in $S(R)$. This seems to smooth the volume estimates.

8.3.3 Numerical Results Summary

When combined with the vortex method, the different estimation schemes performed very differently. The very naive method was noisy and did not give values for energy helicity, impulse and momentum which were close to those obtained by integrating over a grid. However, the estimate of the total volume obtained by using this method was very good and did not vary greatly over time.

The core function based method behaved in an almost opposite manner. It gave initial estimates for the energy which were very close to those on a grid, but the estimates of the volume were not conserved over time.

The slightly less naive scheme gave estimates for energy, helicity, etc, which were better than the naive scheme, but not as good as the core function based scheme. It gave a good volume estimate which was conserved over time.

None of the schemes gave good estimates of the energy, etc integrals as the blob distribution evolved over time, especially once the initial grid like configuration of the particles was lost. This is to be expected, since those integrals have singular terms which are neglected by simple estimation using such volume estimates.

Based on these numerical experiments, the slightly less naive scheme for volume estimation was chosen for later use, since it conserves volume over time and is not noisy.

8.4 Future Directions

In this section, a number of incomplete ideas, related to the resampling problem are discussed. This section should be regarded as conjecture or motivation. Very little here is rigorous.

8.4.1 Spectral Or Galerkin Analysis Of Implied Basis

It may be useful to analyze the vortex method in terms of the subspaces which are used to represent the vorticity. The vortex method naturally defines a number of subspaces.
Given,

- a set of N points $\{x_i(t)\}$ which depend on time
- a cutoff parameter, δ,

- a cutoff function, $f_\delta(r) = \frac{1}{\delta^3} f(\frac{r}{\delta})$, where f is $\mathcal{C}^\infty$ and satisfys some moment conditions.

one can define the set of functions $y(x) : \mathcal{R}^3 \to \mathcal{R}^3$ which may be written as $y(x) = \sum_i y_i \cdot f_\delta(x - x_i)$, for some N elements of $\mathcal{R}^3$, y_i. Call this set $\mathcal{S}(x_i(t), \delta, f)$, or simply $\mathcal{S}(t)$. $\mathcal{S}(t)$, at a given fixed time, has a number of properties. For any fixed t, $\{x_i\}$, δ, f,

- $\mathcal{S}(x_i(t), \delta, f)$ is a vector space.
- $\mathcal{S}(x_i(t), \delta, f)$ has dimension equal to the number of points in $\{x_i\}$.
- $\mathcal{S}(x_i(t), \delta, f)$ is a complete space (definition plus completeness of reals).
- For $\delta > 0$, $\mathcal{S}(x_i(t), \delta, f)$ is a subspace of $\mathcal{L}^2$
- For $\delta > 0$, $\mathcal{S}(x_i(t), \delta, f)$ is a subspace of $\mathcal{C}^\infty$
- For $\delta = 0$, $\mathcal{S}(x_i(t), \delta, f)$ is a subspace of $\mathcal{L}^1$, but not of $\mathcal{L}^2$
- A basis for $\mathcal{S}(x_i(t), \delta, f)$ is formed by the functions $f_{\delta i}(x, t) = f_\delta(x - x_i(t))$

The vorticity weights which are used in the vortex method are the coordinates of ω in the basis $f_{\delta i}$. The fact that the basis $f_{\delta i}$ is ill conditioned is essentially the result of section 8.2 Since this basis is so ill conditioned, one may expect some trouble in an attempt to rearrange the weights.

From this viewpoint, one may ask, why does the 2d vortex method work? What will emerge here is the fact that since the y_i do not change in the two dimensional vortex method, the instability of the basis is irrelevant. The coefficients are never constructed from the solution, except for the representation of the initial data.

The two dimensional vortex method may be analyzed in this context. Start with some initial data $\omega(x, t = 0)$ and some N points $x_i(t = 0)$, which are the initial positions of the vortex particles. Let f be the function from which the core function is constructed. The first task is to find an element of $\mathcal{S}(x_i(t), \delta, f)$, which is close to the initial data. There are two ways in which one might attempt to do this. The first is to try to do the deconvolution using very high precision arithmetic. The second is to realize that as δ goes to zero, keeping all other parameters fixed, $f_{\delta i} = \delta(x - x_i)$ and the basis for $\mathcal{S}$ becomes ever better conditioned as we progress to this limit. One makes an error by constructing coefficients using a basis with δ smaller than the δ used to reconstruct the function from its coefficients. This error depends on δ and goes to zero as δ goes to zero. So, there are at least two ways to accomplish this task and each has error which goes to zero with δ. The first requires infinite precision arithmetic as δ goes to zero, but can be solved using very high, but still finite, precision arithmetic so long as $\delta > 0$. For δ, small enough, the cost of the high precision arithmetic will become prohibitive. In any case, initial coefficients $y_i(t = 0)$ such that $y(x, t = 0)$ is close to the initial data $\omega(x, t = 0)$ can be obtained. This argument extends to three dimensions.

Given a representation of $\omega(x, t) = \sum_i \omega_i f_\delta(x - x_i)$, the next task is to move forward in time to $\omega(x, t + dt)$. Suppose the locations of the points are advanced using the velocity field. This changes

the relevant function space from $S(x_i(t), \delta, f)$ to $S(x_i(t+dt), \delta, f)$. However, $\sum_i \omega_i f_\delta(x - x_i(t+dt))$ is in $S(x_i(t+dt), \delta, f)$ and close to $\omega(x, t+dt)$, so no calculations involving the basis of S must be done.

In three dimensions, one first moves the particles, then, one must find an element of $S(x_i(t+dt), \delta, f)$ which is close to the solution of $\frac{d\omega}{dt} = \omega \cdot \nabla u$ The normal prescription is given by $\omega_i := dt\omega_i \cdot \nabla u(x_i)$ which defines a function in the required space, which is close to $\omega(x, t+dt)$ for ω and ∇u sufficiently smooth, but which does not include any additional smoothing of the solution. If such smoothing is necessary, it is not clear which function in $S(t+dt)$ is an appropriate choice.

8.4.2 How Good or Bad is Volumetric Resampling

Volumetric resampling is used implicitly in the computation of the stretch term in the vortex method. It is used explicitly in deterministic diffusion schemes, and we will later use it explicitly in both a boundary layer algorithm and an alternate approach to the stretch term. It is not known how accurate this type of resampling is, or even if it converges in the case that the particles are allowed to move.

8.4.3 Can One Pick A Best Set Of Weights

The non-uniqueness of the vortex weights, may allow an opportunity to pick a "best" set of weights to represent the vorticity field. However, one must know what one means by best and one must be able to stably get that set of weights. I have not found a way to exploit this.

8.4.4 Analogy Between Integration and Resampling

There is a connection between volumetric resampling and integration formulae, in the fact that both assign volumes to points. It may be possible to build a higher order volume estimation scheme by requiring that the volumes which are assigned exactly integrate some set of orthogonal functions. This has not yet been fully investigated.

Chapter 9

Vorticity Near A Zero and Reconnection

The numerical work in chapters 4 and 11 indicates that part of what is causing trouble for the vortex method is the behavior of vorticity near a zero of the vorticity and/or a zero of its torsion. This chapter consists of a collection of preliminary observations about zeroes of vorticity and reconnection of vortex filaments.

9.1 Domains In Which To Study Reconnection

There are several simplified regimes in which one can study reconnection of vortex filaments and in which such study has been done. Four such areas are loosely defined. Each of the following assumptions defines such a regime.

- Vorticity is confined to a lattice
- Vorticity is confined to isolated filaments
- Vorticity is confined to smoothed isolated filaments
- Vorticity is smooth.

Chorin has extensively studied the renormalization based reconnection of vortex lines in the context of vortices on a lattice, and of isolated filaments, using techniques from statistical mechanics[17] [15]. The relevance of vortex reconnection to the vortex method is the hope that an understanding of renormalization and its relation to vortex reconnection will allow one to reconnect the filaments of vortex elements in a way which is consistent with the Euler equations, and which avoids the type of blow-up seen in chapter 4.

In Chorin's studies, the vorticity is assumed to be supported on filaments of zero volume. In addition it is assumed that these filaments are sparse. In the present chapter, it is assumed that the vorticity itself is a smooth field which is determined by smoothing a filament system. This filament system need not be sparse. This allows one to investigate mixing of cores and areas near a zero of vorticity which appear to be numerically important. It also allows one to study volume effects which should be important on very small scales. The present contribution to this effort is an attempt to understand what happens near a zero of smooth vorticity fields, especially in the presence of small compressible terms and small viscous terms.

Since zeros of vorticity and its torsion appear to be numerically important, the neighborhoods of these phenomena are studied.

9.2 Toy Models of Vorticity Near A Zero

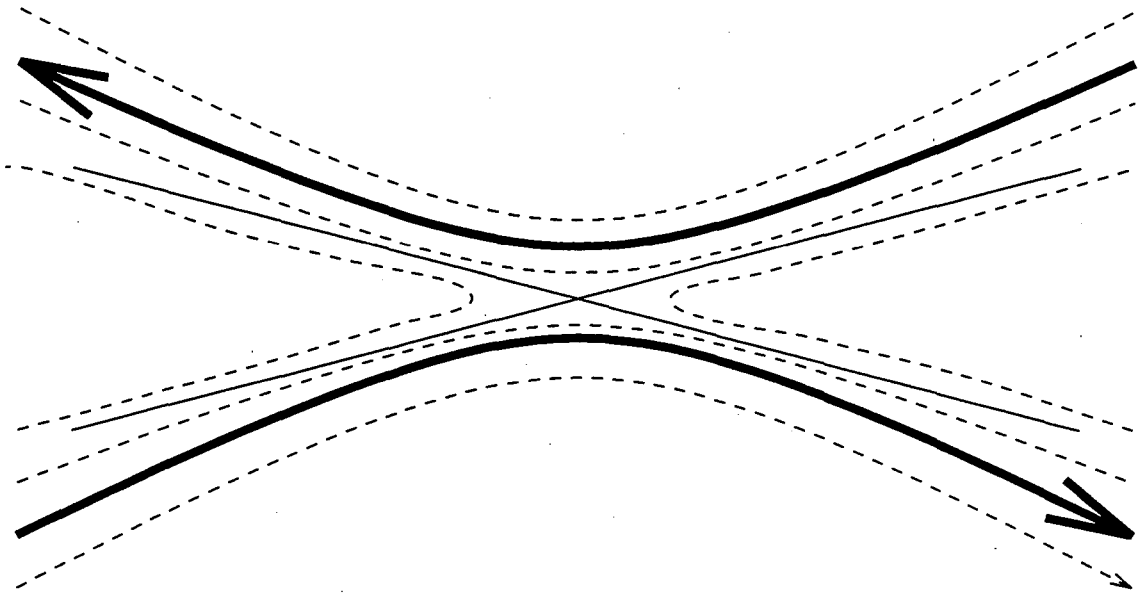
In this section, the vorticity near an isolated zero of the vorticity field is examined. The initial motivation to study this comes from the observations in chapter 4. Structures develop in the numerical method which consist of antiparallel filaments, which when convolved with a core function, cause a zero of the smooth vorticity. Such points seem to cause problems in the computations of that chapter.

9.2.1 Generic Smooth Vorticity Near A Zero

Recall that in chapter 4, it was noted that many of the points of very large growth occurred near zeroes of the smoothed vorticity field and near zeroes of torsion.

Assume a smooth vorticity field in three dimensions. Generically, close enough to an isolated zero of the field, the field and its flow are isotopic to that of the linearization of the field near the zero[31]. Since the field itself is divergence free, the linearization is trace free. Therefore, there are only two choices for the signs of the eigenvalues of the linearized field, (-,+,+) or (+,-,-). The dynamics of the field are all in the space spanned by the eigenvectors with positive eigenvalues, so in the first case, they are restricted to a plane and in the second to a line. Thus in the first case the torsion of the linearized field is zero and in the second, both the torsion and curvature are zero at the field zero. This result can also be obtained by computing the Frenet frame for a vortex filament. This implies that the asymptotic dynamics of the vorticity field are confined to a surface in the first case and a curve in the second.

In figure 9.2.1, we see a schematic representation of cross section through a (+,-) plane of the linearized vorticity near a zero, with signature (+,-,-). The thin solid lines are the eigendirections of the linearization, the dotted lines represent integral curves of the vorticity. If it is further assumed that this field is derived from the smoothing of singular vortex filaments, then the thick solid lines would represent those filaments. It is also assumed that the core is small relative to



the size of the region depicted.

9.2.2 Toy Model I

Suppose that the vorticity is smooth and three dimensional, with an isolated zero. Suppose that one sits at such a zero point and moves with it. Suppose also that the eigenvalues are (+,-,-) Note that since the velocity is divergence free, that ∇u is trace free. The relative velocity u_{rel} is zero at the zero of vorticity by construction. In this toy model both fields are replaced by their linearizations at the zero, and only the y component of the vorticity is considered. While a rigorous derivation of this toy model is not given, it appears similar to the VTE near a zero. Formally, it is conjectured that this system has solutions which match to first order those of the VTE near a zero in a plane containing the positive eigenvalue of the linearized vorticity. The coordinate system used has the y direction in the direction of the largest positive eigenvalue of ∇u_{rel}. The x direction is perpendicular to that and in the attractive plane. The z direction is then given by a cross product. Let $\alpha(x)$ represents the line length through a given point on the x axis or, the value of the y component vorticity at such a point.

Consider the following differential equation, where a and b are positive real numbers.

$$\partial_t \alpha - a * x \partial_x \alpha = b * \alpha \tag{9.1}$$

This is a one dimensional quasi-linear hyperbolic system with a source term along characteristics. The characteristics are the integral curves of $\partial_t x = -a * x$, which are $x(t) = x_0 e^{-at}$ If the initial data is given by $\alpha(x, 0) = f(x)$, it is easy to check that the solution is given by $\alpha(x, t) = (f(xe^{at}))^{bt}$ This model suggests several conclusions.

- Given any finite resolution, if one waits long enough, one will no longer be able to resolve the flow.
- At any finite resolution, structures will eventually appear which cannot be differentiated from discontinuities at that resolution. That is, the first N Fourier components will look like those of a discontinuity.
- If any velocity field with $\nabla \cdot u \neq 0$ is added to the field ax, discontinuities will eventually occur in the model. If this compressibility allows the destruction of a length of scale of $L < 1.0$, then shocks can occur by time $-\ln L$.

9.2.3 Toy Model II

Figure 9.2.1 also suggests the following model. Consider the case in which the linearized vorticity with eigenvalues (+,-,-) near a zero. Consider the vorticity which is in a plane containing the positive eigenvector. Suppose furthermore that the maximum eigenvalue and one other eigenvalue of ∇u lie in that same plane, and that the eigenvalues of ∇u are (+,-,-). Furthermore, assume that the zero of vorticity is also a zero of torsion. (For the linearized field, this last assumption follows from the first.) This sort of configuration is seen in the numerical examples of chapter 4. The integral curves of the vorticity may then be drawn. Each of these lines is then materially transported by the flow, and is moving toward a plane or line. The problem of a front or fronts evolving in a space with codimension 1 is known to be a hyperbolic problem and to cause problems for marker particle methods[46]. Further study of both of these analogies is planned.

9.3 Smoothed and Singular Vorticity Near A Zero

In the vortex method, one may think of the computational filaments as a singular vorticity field, which gives rise to the smooth field by convolution with a smooth kernel. These fields are related, but they may be quite different. For example, numerical results from computations like those in chapters 4 and 11 indicate that these two field may have point values which point in opposite directions.

9.4 The Core Unmixing Phenomenon

The state of run base.7 at time 53 was taken as a starting point for the computation of this section. The particle strengths, Ω_i, were replaced by their opposites, $-\Omega_i$. This configuration was then posed as initial data to the vortex method. Since the time reversal of the inviscid VTE is once again the inviscid VTE, this is a reasonable thing to do. There are several interesting things

which appear in this run. They appear to be related both to the properties of the vortex method near a zero and the arguments of section 8.2.

Note that in the initial data, $T = 53$, several isolated zeroes of vorticity are located in the "tendril" which hangs down from the ring. Also, there are sections of the filament which are indistinguishable at the resolution of the graphics. At $T = 51.0$ and $T = 50.5$ symmetric helical waves form on antiparallel filaments. (They are visible at $T = 51.5$ but not at the resolution of the plot) Although the fused part of the tendril has not separated, much of the tendril has. It is this fused portion and the waves which are visible by $T = 51$ which become unstable so that by $T = 47.5$, it is clear that the ring will not stably evolve back to its original shape.

It is, of course, unreasonable to ask that the vortex method do anything reasonable with the initial data of this example. It is surprising that it gets as far as it does. What is interesting is not the instability itself, but the fact that the problems seem to be isolated to areas which are related to zeroes of vorticity. This creation of waves upon the separation of antiparallel filaments will be called the *core unmixing phenomenon*, since it is also associated with the unmixing of the core functions. This relates it to section 8.2 which is concerned about the relationship between the computed vorticity and the strengths of the elements.

It is also interesting to note that up to $T = 44$, the computation base.7 may be run stably backwards to $T = 0$. The result is not the same as the initial data, but it is qualitatively similar in the sense that it has the same symmetry as the initial data and about the same total arc length.

Results from base.9 and base.10 were run them backwards from late times. The phenomena in base.9 are similar to those in base.7, and they are displayed. Base.10 also becomes unstable when run backwards, but the mechanism is not clear. This run is not displayed

Figure 9.1: Run Backwards.7

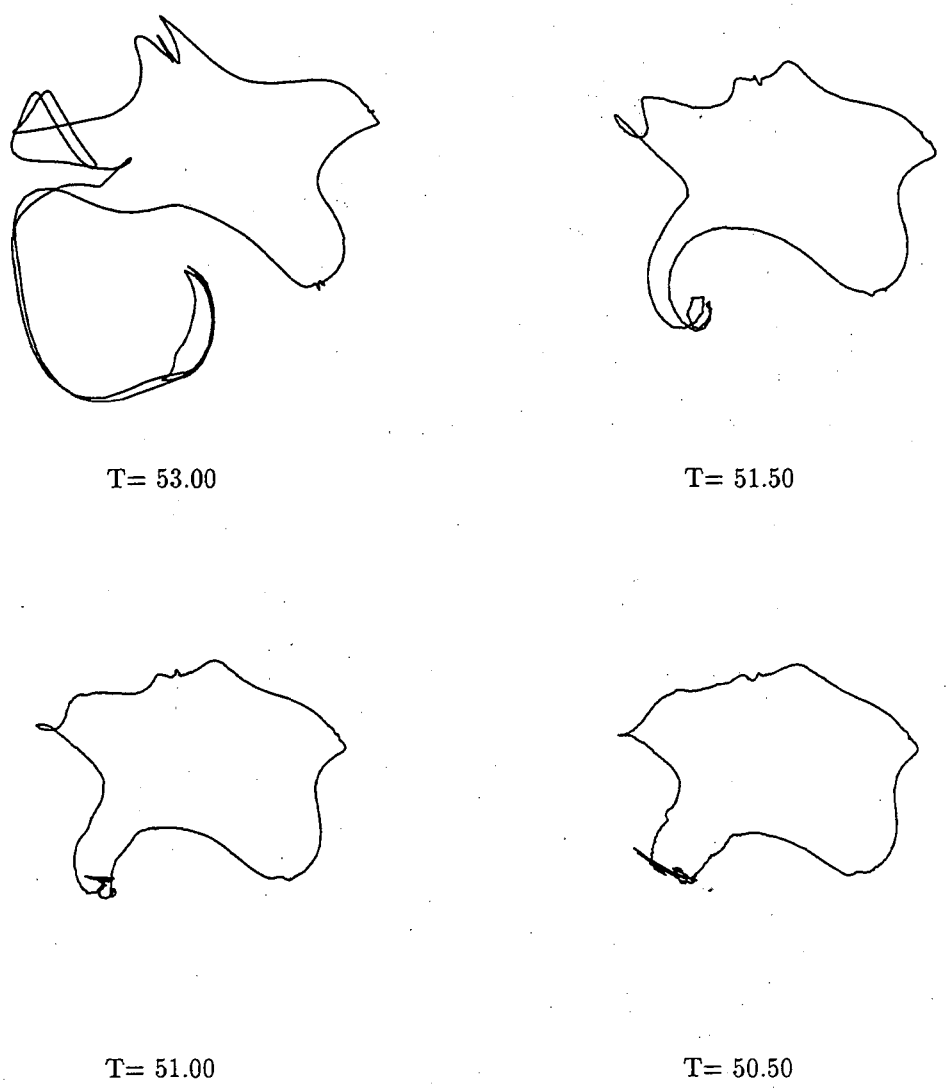
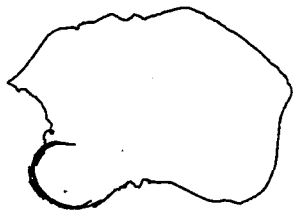
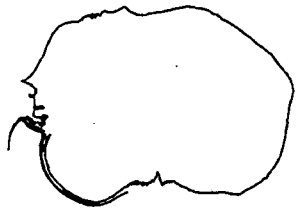


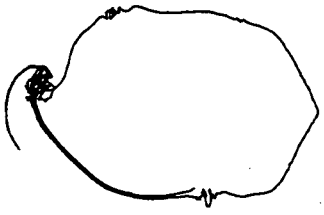
Figure 9.2: Run Backwards.7



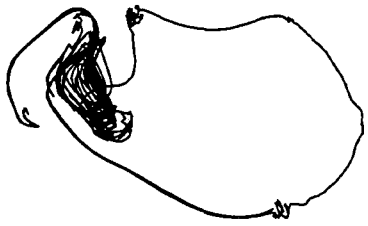
T= 49.50



T= 48.50



T= 46.50



T= 44.00

Figure 9.3: Run Backwards.9

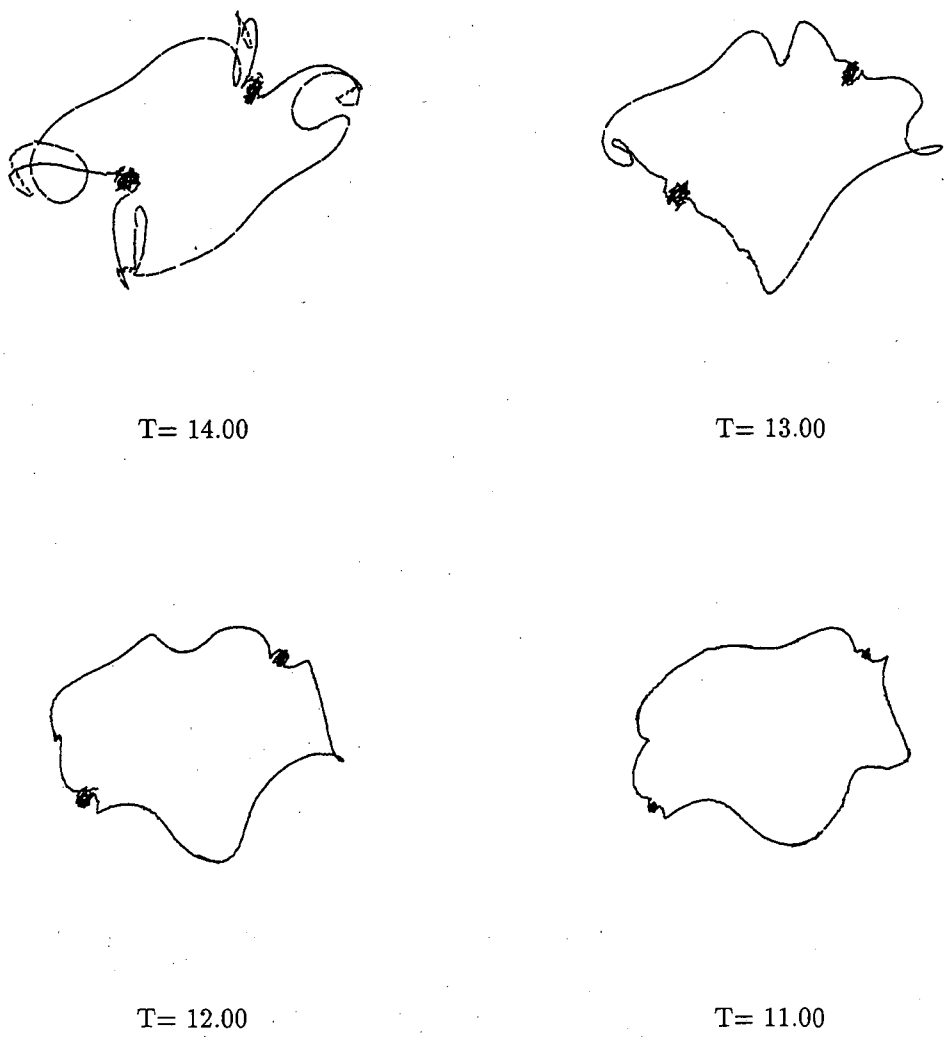
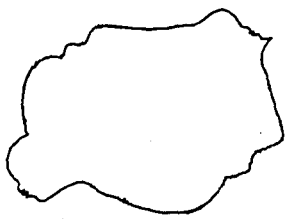
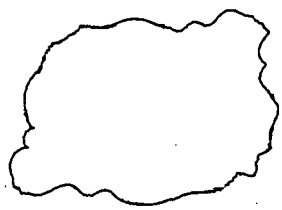


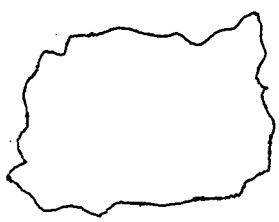
Figure 9.4: Run Backwards.9



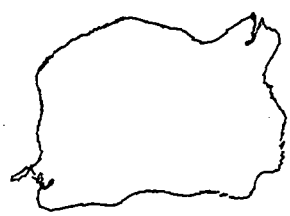
T= 10.00



T= 9.00



T= 8.00



T= 7.00

Figure 9.5: Run Backwards.9

T= 6.00 T= 5.00

9.5 Some Possible Reconnection Schemes

In this section, ideas for reconnection schemes are proposed which are loosely based on the observations of this chapter and those of chapter 8. Although the renormalization based hairpin removal and reconnection algorithm of Chorin presented in [17] was implemented, it was found to be very sensitive to the choice of the parameters $cos\theta_{min}$ and δ_{min}. However, the parameters used were very much smaller than those used in [17].

9.5.1 Reconnection Based on Non-uniqueness

In chapter 8, it was noted that the singular vorticity field is numerically nonunique in the sense that two such distributions which differ only in sufficiently high wave numbers could not distinguished. Call the initial singular vorticity distribution Ω and the reconnected distribution Ω^R Allow the replacement of Ω by Ω^R, in Chorins reconnection algorithm only if

- Ω^R and Ω differ only in sufficiently high wave numbers that they are indistinguishable.
- Ω^R is smaller in high wave numbers than Ω.

9.5.2 Reconnection Based on Matching the Linearization

The idea here is to force the numerical filaments to topologically match the linearized smooth vorticity at some resolution. Loosely, one computes the linearization and its eigen-directions at a zero. These eigenvectors and eigenvalues determine the topology. It is then asserted that the topology of the linearized field must match that of the smooth field at a distance ϵ from the zero, where ϵ determines the scale at which we wish to force the match. Then, if the numerical filament crosses the eigen-directions, it is reconnected so that it no longer crosses.

9.5.3 Reconnection Based on $\nabla \cdot \omega$ as an entropy

Recall that in chapter 5, it was shown that the divergence of the vorticity acts as a type of entropy. We therefore suggest allowing reconnection in the cases where it reduces the divergence of the smooth vorticity field.

None of these algorithms has yet been explored numerically.

9.6 Questions

We end this exploratory chapter with a list of questions inspired by numerical observation for which we would like to someday see answers.

- Is the core unmixing phenomenon physically relevant or simply a numerical artifact?

- Under what circumstances does reconnection and renormalization introduce an error which is dissipative?
- Under what circumstances could reconnection cause core unmixing?
- Under what conditions, if any, does the error due to mixing of the core functions fail to be dissipative?
- Is the divergence of the velocity error large enough to cause the collision of vortex lines?
- If so, is the correct entropy condition enforced?
- Do any of the proposed reconnection/resampling algorithms have the property of introducing a purely dissipative error.
- How should the parameters in Chorins renormalization, hairpin removal and reconnection algorithm be chosen?

Chapter 10

A Sketch of Vorticity Exchange Viscosity Algorithms

In this chapter, a very brief overview of vorticity exchange schemes for viscosity is presented These schemes will be used in later computations.

The motivation for an investigation of diffusion algorithms is the hope that without fully understanding the mechanism producing problems for the vortex method, it can be fixed through the addition of the right amount of viscosity.

The convergence of the random walk method for diffusion[11] is slow, $\mathcal{O}(\sqrt{R})$, and In addition, some object to it on the basis that it is not a deterministic algorithm and thus can give different answers depending on the random numbers that are chosen. Our concerned is that it introduces noise into a system whose long time evolution is unstable (see section 12.1 on Chorin's 1973 random walk based boundary condition algorithm). This led to the development of deterministic diffusion schemes.

10.1 A General Framework

In this section, a general framework in which to view deterministic diffusion schemes is built. Vorticity exchange schemes rely on changing the vortex element strengths by an amount which depends on an approximation to $\Delta\omega_\delta$ The strengths, ω_i are thought of as a point value times a volume, $\omega_i = dV_i * \omega(x_i)$ Viscous diffusion is governed by $\partial_t\omega = \frac{1}{R}\Delta\omega$ Substitute the representation of ω_δ, into the RHS and represent ω_i as $\omega(x_i)dV$. The result is

$$\partial_t\omega(x_i) = \frac{1}{R}\Delta\sum_j \omega(x_j)dV_j * f_\delta(x_i - x_j) \tag{10.1}$$

Multiply through by dV_i.

$$\partial_t dV_i \omega(x_i) = \frac{dV_i}{R} \Delta \sum_j \omega(x_j) dV_j * f_\delta(x_i - x_j) \tag{10.2}$$

This is equivalent to

$$\partial_t \omega_i = \frac{dV_i}{R} \Delta \sum_j \omega_j * f_\delta(x_i - x_j) \tag{10.3}$$

Now, let $\mathcal{L}_h$ be an approximation to the Laplacian which depends on a parameter h

$$\partial_t \omega_i \approx \frac{dV_i}{R} \mathcal{L}_h \sum_j \omega_j f_\delta(x_i - x_j) \tag{10.4}$$

This scheme may be symmetrized by adding subtracting a constant from the RHS and using the fact that the Laplacian of a constant is zero [20][19]. The symmetrized form is

$$\partial_t \omega_i \approx \frac{dV_i}{R} \mathcal{L}_h \sum_j (\omega_j - \omega_i) * f_\delta(x_i - x_j) \tag{10.5}$$

The advantage of the symmetrized form is that it conserves the total vorticity.

10.2 Fishelov Scheme

In the scheme of Fishelov[22], $\mathcal{L}_h \sum_j \omega_j * f_\delta(x_i - x_j) = \sum_j \omega_j \Delta(f_\delta(x_i - x_j))$, and the unsymmetrized form is used. The parameter h is δ used for the core function.

10.3 Vortex Centered Finite Difference Scheme

One obtains a vortex centered finite difference scheme by setting $\mathcal{L}_h(x_j) = \Delta_h(x_j)$, where $\Delta_h(x_j)$ is a finite difference approximation to the Laplacian, with the stencil centered at x_j.

10.4 Cottets Scheme

In the scheme of Cottet [19],

$$dV_i \mathcal{L}_h \sum_j (\omega_j - \omega_i) f_\delta(x_i - x_j) = \sum_j (dV_i \omega_j - dV_j \omega_i) \Lambda_h(x_i - x_j) \tag{10.6}$$

where Λ_h is chosen so that $\omega \star \Lambda_h = \Delta\omega + \mathcal{O}(h^p)$ In addition, in a later version of this scheme, Λ_h is normalized differently at each point, based on the local density of particles. This yields better stability properties. In this case, h is a cutoff parameter for the core Λ

10.5 Mas-Gallic Scheme

The scheme of Mas-Gallic [20] only fits this framework after time has been discretized, due to its use of the kernel of the heat equation. It is equivalent to exactly solving the heat equation and then resampling, using volumetric resampling. Let Δt be the time step of the calculation and $G(x,t)$ be the fundamental solution of the heat equation,

$$\Delta t \mathcal{L}_h \sum_j (\omega_j - \omega_i) f_\delta(x_i - x_j) = \sum_j (\omega_j - \omega_i) G(x_i - x_j, \frac{\Delta t}{R}) \tag{10.7}$$

10.6 Adding Volume Estimation These Schemes

Note that in all of these schemes, the volume of each particle, dV_i , is used in the calculation. In each of these schemes, it is assumed that volumes are assigned to the initial data and then kept up to date as the calculation progresses. As an alternative, one can also use the volume estimation schemes of section 8.3 to assign particle volumes at each time step.

Chapter 11

Ring Calculations II

In this chapter, results of computations using multifilament discretizations of perturbed vortex rings are reported.

11.1 The Parameter Space

For the multifilament ring computations, several different stretch algorithms, core functions and diffusion algorithms were used. Only blob based elements were employed. The following outline describes the different algorithm parameters. The stretch algorithms are discussed in chapter 3 and the diffusion algorithms are described in chapter 10

- Stretch Algorithm
  - Helmholtz Theorem based stretch
  - Stretch based on $\omega_i \cdot \nabla_h u$
  - Stretch based on $\omega_i \cdot \nabla_h u^T$
  - Stretch based on $\omega(x) \cdot \nabla_h u^T$ with resampling
- Diffusion Algorithm
  - Fishelov's algorithm
  - Cottet's algorithm
  - Mas-Gallic's algorithm.
- Core Function
  - Second order core function based on $f(r) = \frac{3}{4\pi} e^{-r^3}$
  - Fourth order core function based on $f(r) = \frac{3}{4\pi}(\frac{5}{2} - \frac{3}{2}r^3)e^{-r^3}$

Due to finite computer time, many interesting algorithms were not studied or not studied in sufficient detail.

11.2 Discretization of the Core

An additional issue that must be addressed with a study of multifilament rings is the discretization of the cross section of the ring. The ring cross section is divided into annuli over a central filament. The central filament will be called annulus 0. Annulus j for $j \neq 0$ contains 6*j filaments. So, a 3 layer ring, consists of $12 + 6 + 1 = 19$ filaments. Each filament is given the same strength which is equal to the total circulation desired divided by the number of filaments.

11.3 The Code

The unconnected vortex method code was used. This program, along with the verification procedures used, was described in chapter 4.

While this code is efficient on the machines used, in the sense of megaflops achieved versus the theoretical maximum speed of the machine, one might be inclined to use an $\mathcal{O}(N \log N)$ algorithm for the velocity computations instead of the naive $\mathcal{O}(N^2)$ algorithm which was used for the smaller problems. There are however complications with using such a scheme. It was observed in chapter 4 that the computations are highly sensitive to the core function. The fast solver would have to be implemented and used in such a way that it did not effect the core functions contribution or it would invalidate the intended study. Suppose that a fast solver based on "divide, approximate and conquer" such as the fast multipole method, the algorithm of Anderson, or a panel method based fast solver, were used. Assuming a Gaussian core of standard deviation σ, points whose distance is $2 * 5\sigma$ or more could be considered as well separated. For typical core sizes and a ring of radius one, a ball of radius 5σ is bigger than the entire support of the vorticity. Thus, if a fast solver were used, it would either be ineffective, or result in unknown effects. The results are perplexing enough without added complications.

Although the tradeoffs have not yet been rigorously analyzed, it is noted that the method of local corrections provides a compromise solution to the conflict between core size and fast solvers. It represents the core function to some finite resolution, while still allowing reasonably sized smallest boxes. Research in this area is in progress.

11.3.1 Dirty Tricks

Both the second order and fourth order core functions which were used have Laplacians with undesirable properties. Their Laplacians are zero in the center, rise rapidly to a maximum and then decay. This is not what one intuitively expects, and it seems to cause problems for Fishelov's

algorithm. Therefore, in Fishelov's algorithm, the Laplacian of the core function is replaced by the Laplacian of a Gaussian (for second order), or a super-Gaussian core (for fourth order). This seems to give more stable results.

11.4 Computations

The solution of two different problems was attempted. Both problems have vortex rings as initial data. In both cases, the ring has radius 1 and core radius 0.2. The ring is perturbed at wave number 6, with a perturbation of size 0.05. In the first case, the fluid is assumed to be inviscid. In the second, the fluid has dimensionless viscosity of 0.001 This gives a Reynolds number between 500 and 1000 depending on how it is defined. Table 11.1 describes the computations.

Due to computer time limitations a number of compromises were made. An initial time step size of 0.01 was chosen for the base runs, which gives an initial CFL number of 0.1 for the advection term. Computations were aborted when the number of elements reached approximately 10000. The number of computations attempted was small and more computations are necessary, especially to investigate the effects of the different diffusion algorithms. (Text continues on page 139.)

Table 11.1: Table of Multi Filament Runs

Run	Stretch	ODE	Type	Core	Cutoff	Delta T	N	Layers	Wave	perturb	Diffuse	Nblobs	Time	Machine
Base Runs														
1	Helmholtz	2	blob	2	0.3	0.01	64	3	6	0.05	0	22584	8	Done
2	$\omega_i \cdot \nabla_h u$	2	blob	2	0.3	0.01	64	3	6	0.05	0	11160	8	Done
3	$\omega_i \cdot \nabla_h u^T$	2	blob	2	0.3	0.01	64	3	6	0.05	0	12456	31.5	Done
4	$\omega \cdot \nabla_h u^T$	2	blob	2	0.3	0.005	64	3	6	0.05	0	2148	3.5	Done
5	$\omega \cdot \nabla_h u^T$	4	blob	2	0.3	0.01	64	3	6	0.05	0	2168	17	done
Refine DT														
1	Helmholtz	2	blob	2	0.3	0.005	64	3	6	0.05	0	10004	7.2	Done
2	$\omega_i \cdot \nabla_h u$	2	blob	2	0.3	0.005	64	3	6	0.05	0	6840	7.5	pi11
3	$\omega_i \cdot \nabla_h u^T$	2	blob	2	0.3	0.005	64	3	6	0.05	0	2432	15.5	mb
5	$\omega \cdot \nabla_h u^T$	4	blob	2	0.3	0.005	64	3	6	0.05	0		.25	c
Fourth Order Core														
1	Helmholtz	2	blob	4	0.3	0.01	64	3	6	0.05	0	19620	8	Done
2	$\omega_i \cdot \nabla_h u$	2	blob	4	0.3	0.01	64	3	6	0.05	0	10580	12	Done
3	$\omega_i \cdot \nabla_h u^T$	2	blob	4	0.3	0.01	64	3	6	0.05	0	14330	15.75	Done
5	$\omega \cdot \nabla_h u^T$	4	blob	4	0.3	0.01	64	3	6	0.05	0		0	
Cottets Diffusion														
1	Helmholtz	2	blob	2	0.3	0.01	64	3	6	0.05	c	9365	7.5	pi
2	$\omega_i \cdot \nabla_h u$	2	blob	2	0.3	0.01	64	3	6	0.05	c	8816	8.5	pi10
3	$\omega_i \cdot \nabla_h u^T$	2	blob	2	0.3	0.01	64	3	6	0.05	c	2412	21.0	
Fishelovs Diffusion														
1	Helmholtz	2	blob	2	0.3	0.01	64	3	6	0.05	f	7808	7.5	pi4
2	$\omega_i \cdot \nabla_h u$	2	blob	2	0.3	0.01	64	3	6	0.05	f	4716	7.2	
3	$\omega_i \cdot \nabla_h u^T$	2	blob	2	0.3	0.01	64	3	6	0.05	f	2416	9.7	
Refine Sigma														
	Helmholtz	2	blob	2	0.2	0.01	64	3	6	0.05	0	9818	8	pi9

Figure 11.1: Run base.1

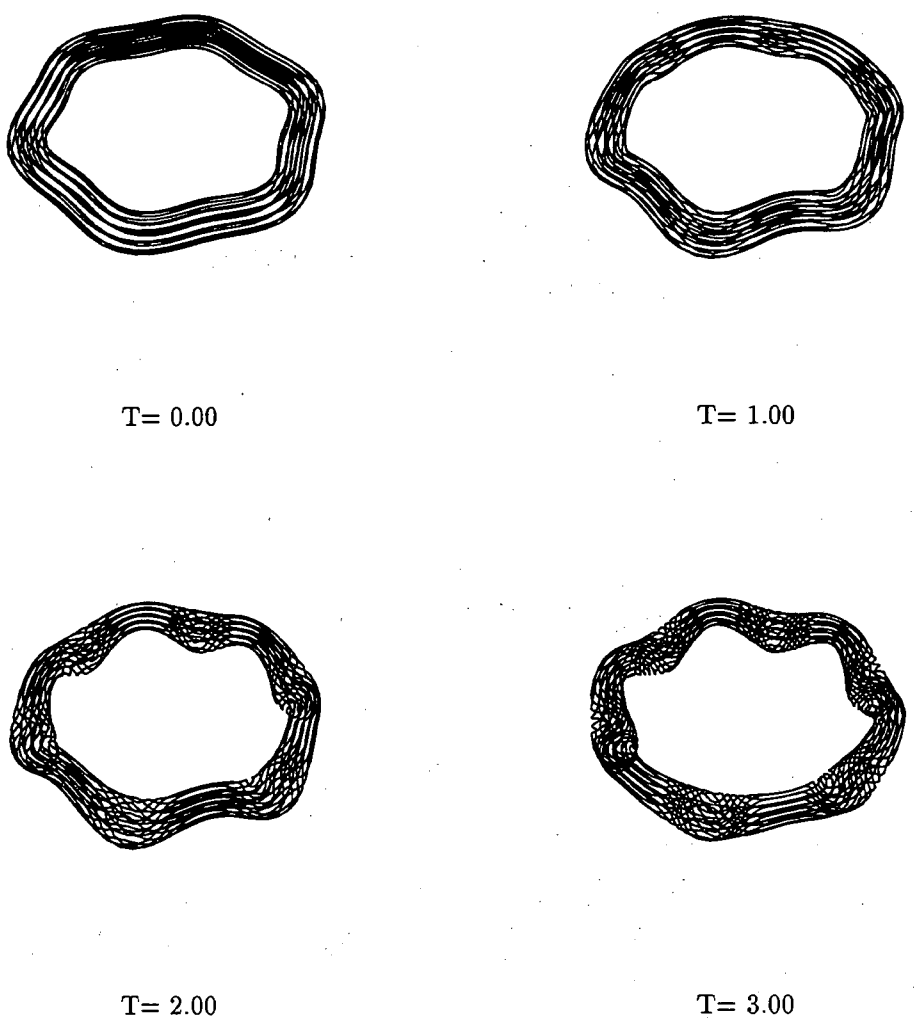
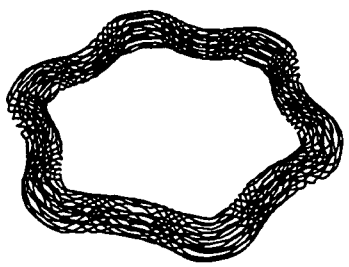
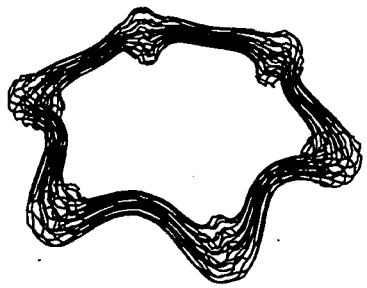


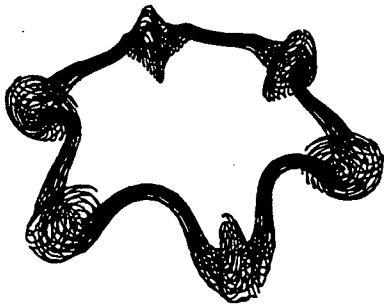
Figure 11.2: Run base.1



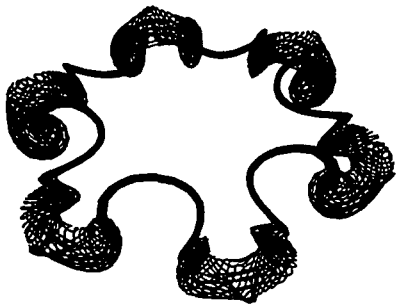
T= 4.00



T= 5.00

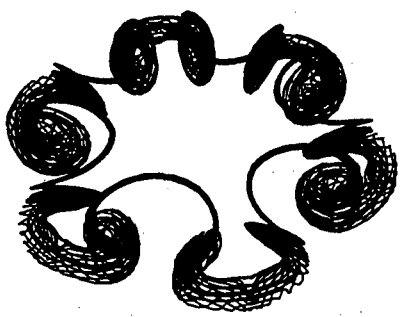


T= 6.00

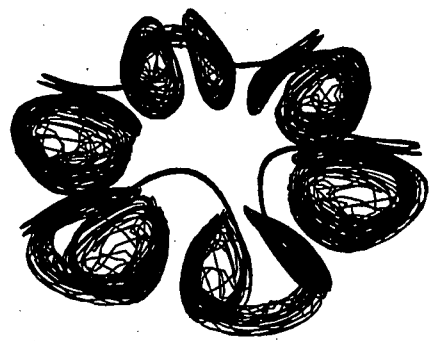


T= 7.00

Figure 11.3: Run base.1



T= 7.25



T= 7.50

T= 7.75

Figure 11.4: Run base.2

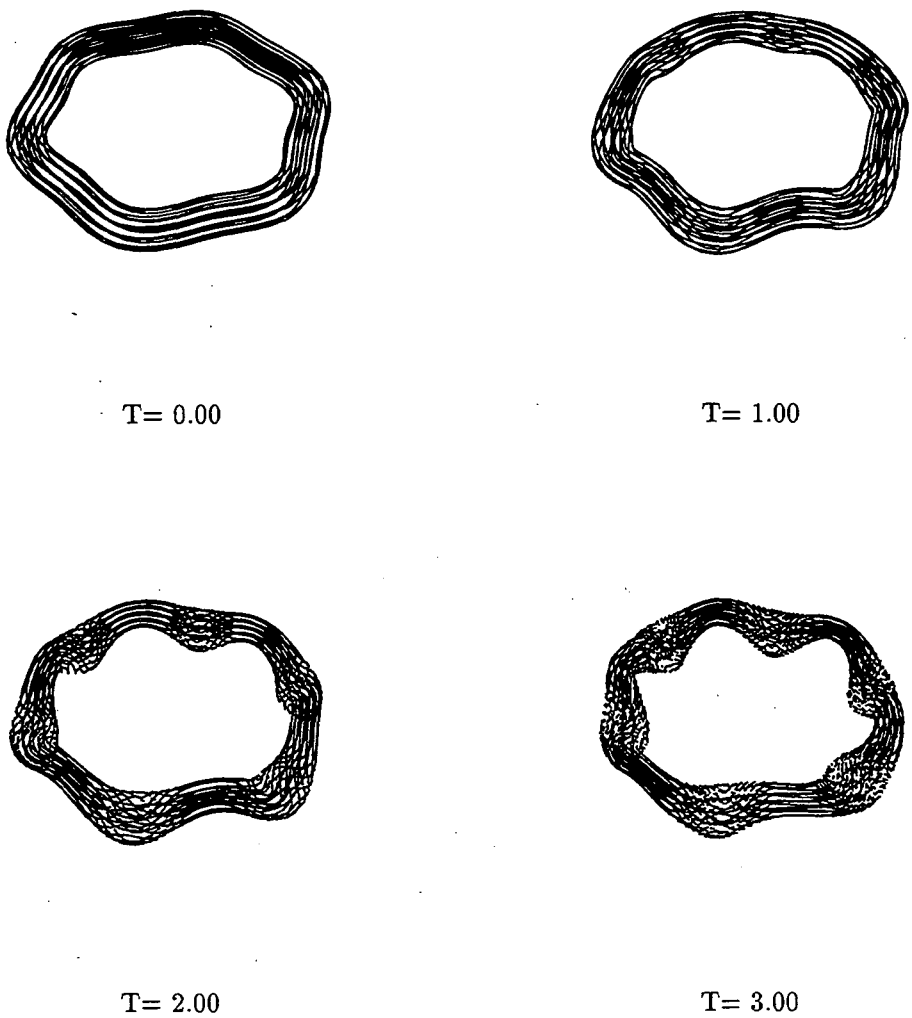
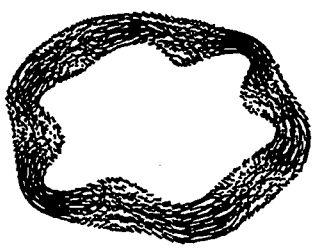
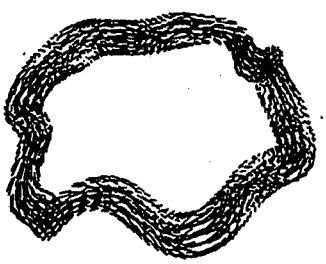


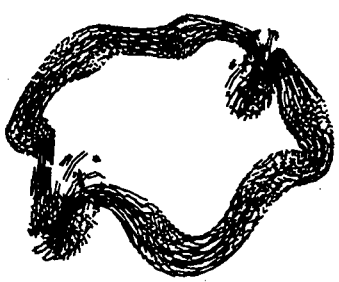
Figure 11.5: Run base.2



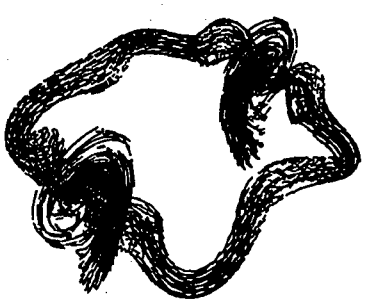
T= 4.00



T= 5.00



T= 6.00

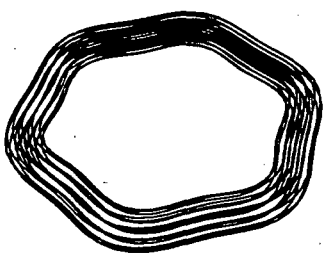


T= 7.00

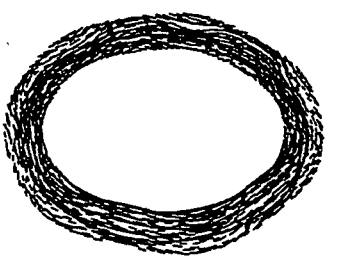
Figure 11.6: Run base.2

T= 8.00 T= 8.50

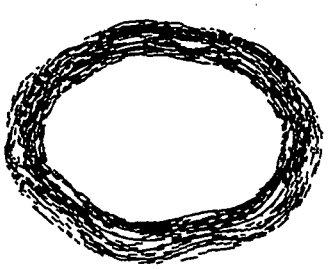
Figure 11.7: Run base.3



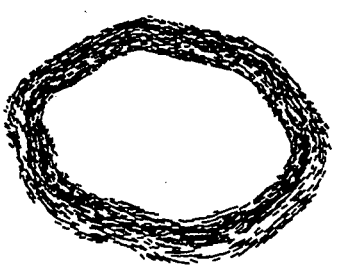
T= 0.00



T= 8.00



T= 12.00



T= 19.00

Figure 11.8: Run base.3

T= 24.00

T= 26.00

T= 27.00

T= 29.00

Figure 11.9: Run base.3

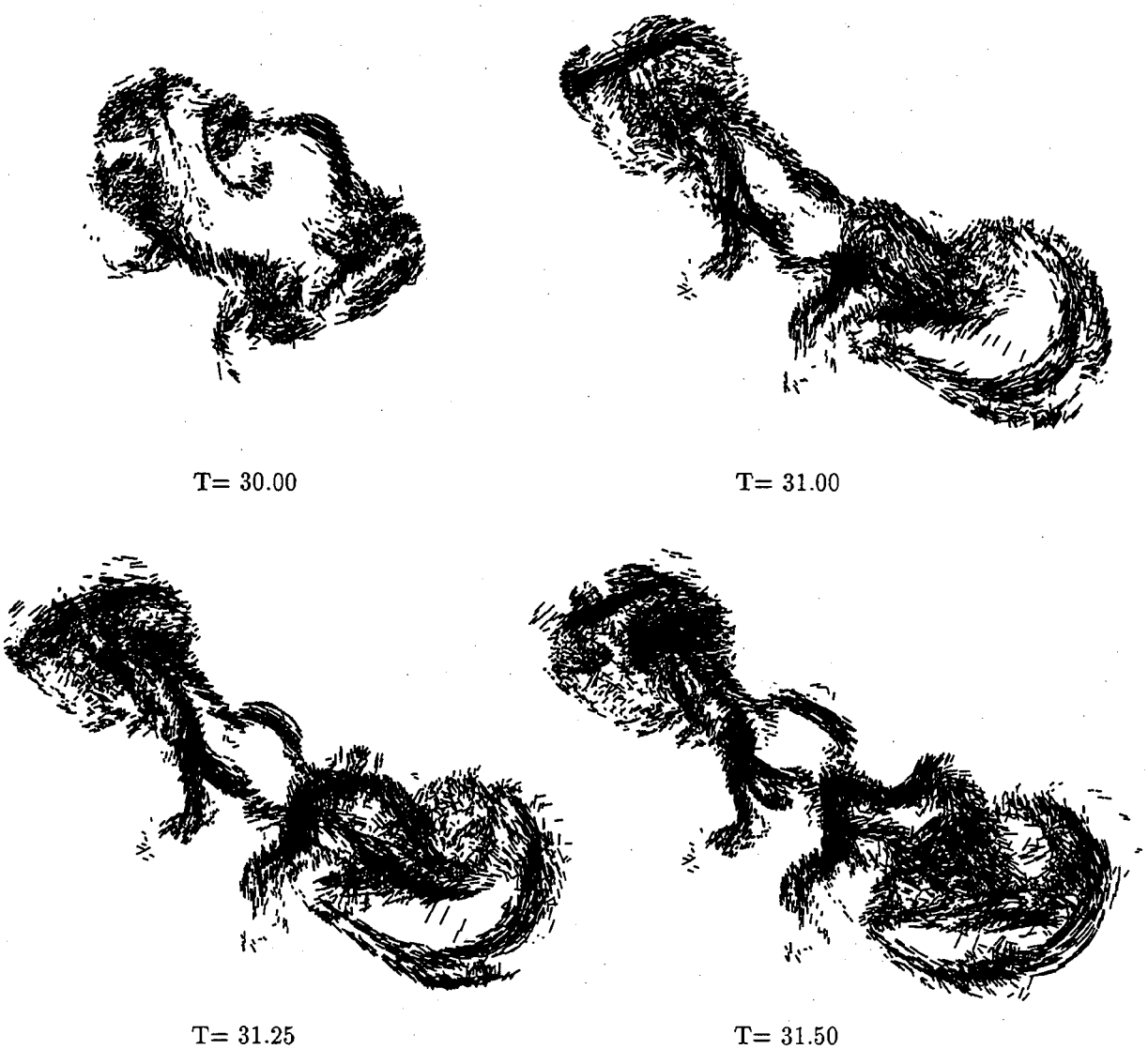


Figure 11.10: Run base.5

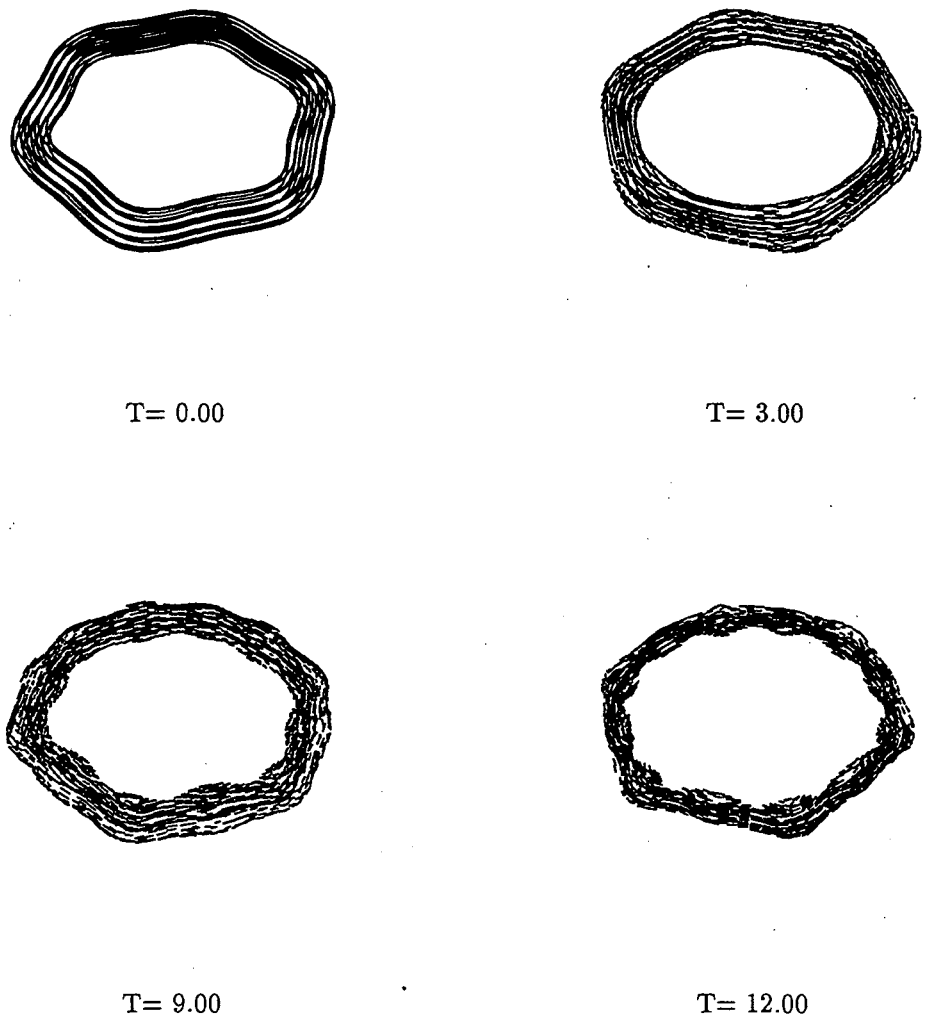
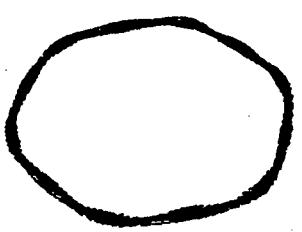
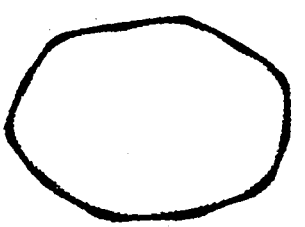


Figure 11.11: Run base.5



T= 15.00



T= 16.00

Figure 11.12: Run fourth.1

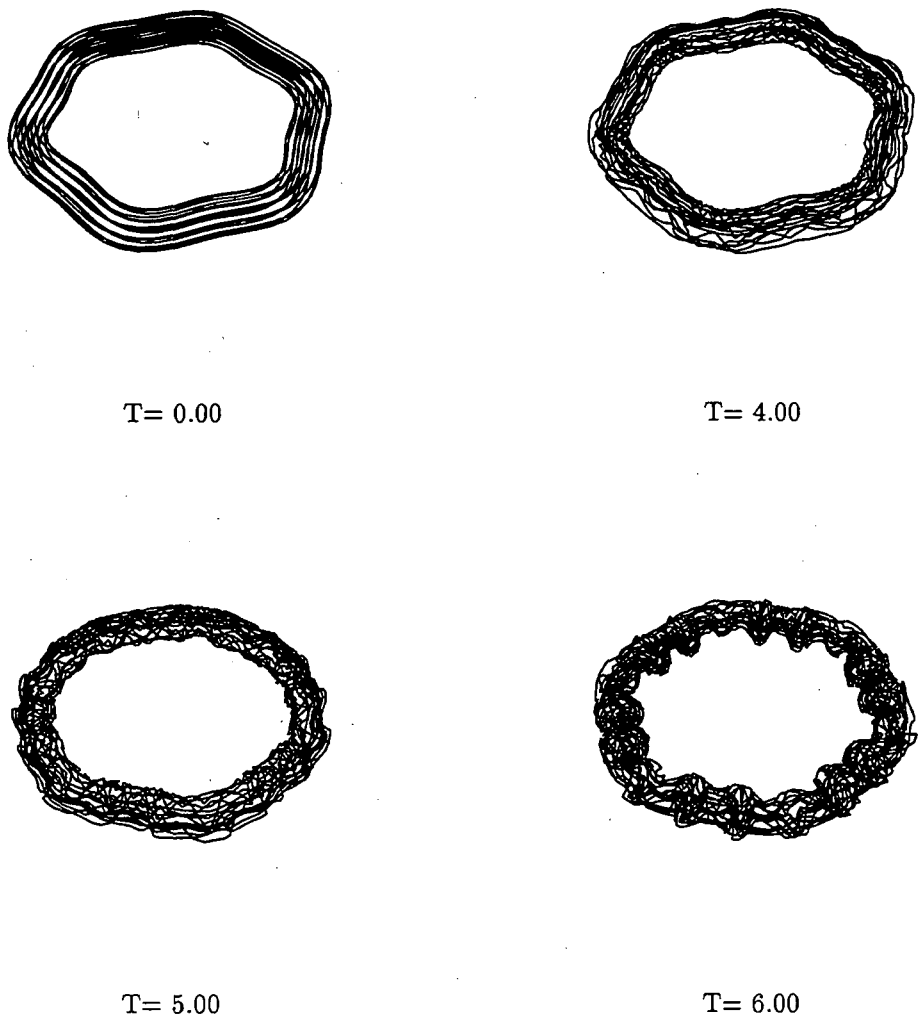
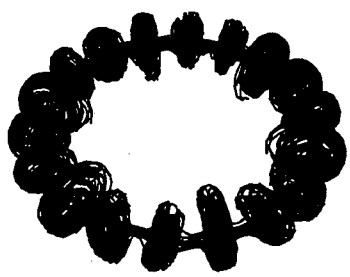


Figure 11.13: Run fourth.1

T= 7.00

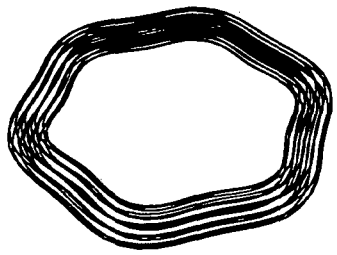
T= 7.50



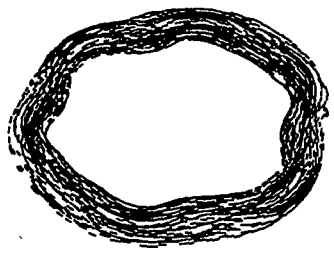
T= 8.00

T= 8.25

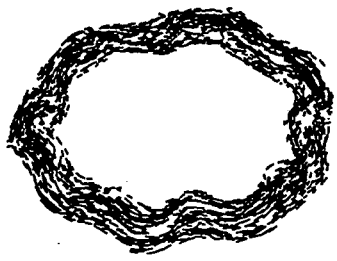
Figure 11.14: Run fourth.2



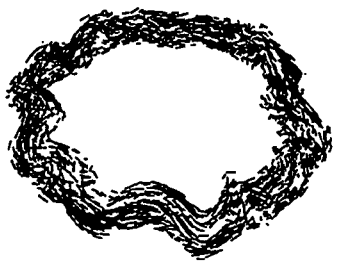
T= 0.00



T= 4.00

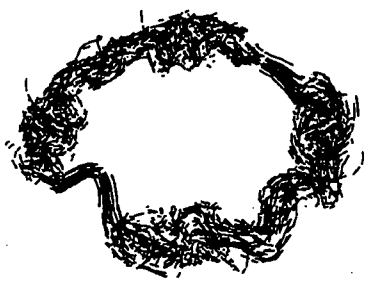


T= 8.00

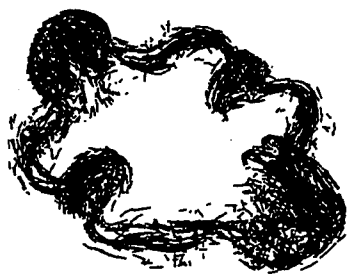


T= 9.00

Figure 11.15: Run fourth.2



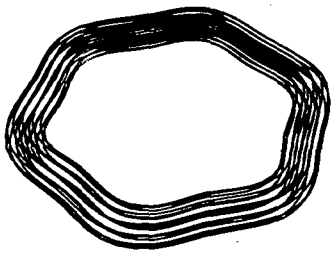
T= 10.00



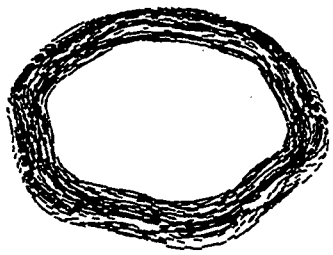
T= 11.00

T= 12.00

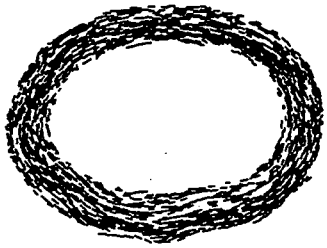
Figure 11.16: Run fourth.3



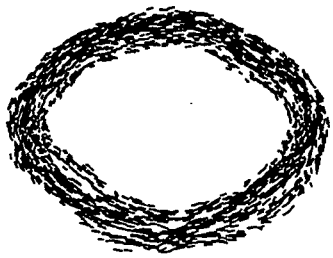
T= 0.00



T= 4.00

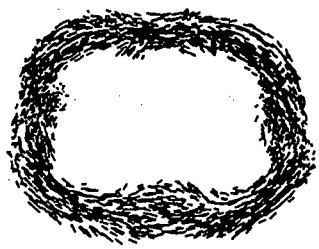


T= 8.00



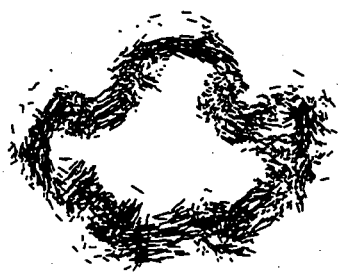
T= 10.00

Figure 11.17: Run fourth.3



T= 12.00

T= 13.00



T= 14.00

T= 15.00

Figure 11.18: Run fourth.3

Run	T final	Volume(T=0)	Volume(T=Tf)
base.1	7.0	1.1	2.1
base.2	8.0	1.1	2.0
base.3	31	1.1	3.7
base.5	16	1.1	0.7
fourth.1	8.25	1.1	1.8
fourth.2	12	1.1	2.8
fourth.3	15	1.1	2.7

11.5 Observations

11.5.1 Comparison to Single Filament Runs

The features observed in sections 4.6.1 4.6.2 are also seen in these multifilament runs. They are listed here for review

- Variety of Results
- Robustness of Results
- Catastrophic Growth of The Number Of Elements
- Growth Near Low Values of Torsion and Near Zeroes of Vorticity
- Loss of Symmetry
- Movement of Particles Along the Filament
- Development of small scale structure
- Development of nearly helical structures

As in Chapter 4, when the refined results are nearly identical to the base results, the figures are omitted. In addition, the following observations were made about the multifilament runs only.

11.5.2 Failure to Conserve Estimated Volume

This failure to conserve approximated volume could be due to the fractalization of the vorticity which makes the volume itself hard to measure[8]. Both the diagnostics and the visualization, however indicate an increase in the volume of the set on which the initial blobs are supported for base.1, base.2 and base.3, and a decrease for base.5. It is also possible that this failure to conserve volume is due to errors in velocity or velocity integration which look like compressible terms. Further tests with higher order time integration and more accurate integration methods such as [52] should be made. However, it should be noted that run base.5 uses fourth order time integration.

Table 11.2: $\nabla \cdot \omega$ for filament methods

Run Fourth.1		Run Base.1		Run Refinedt.1	
T	max $\nabla \cdot \omega$	T	max $\nabla \cdot \omega$	T	max $\nabla \cdot \omega$
0	0.27	0	0.08	0	0.08
1	0.17	1	0.01	1	0.02
2	0.07	2	0.06	2	0.06
3	0.06	3	0.07	3	0.7
4	0.10	4	0.04	4	0.4
5	0.18	5	0.07	5	0.7
6	0.28	6	0.08	6	0.09
7	0.32	7	0.09	7	0.11
8	0.50	8	0.09		

Table 11.3: Z-Impulse for filament methods

Run Fourth.1		Run Base.1		Run Refinedt.1	
T	Z-impulse	T	Z-impulse	T	Z-impulse
0	26.7	0	15.5	0	15.5
1	27.5	1	15.9	1	16.0
2	27.6	2	15.8	2	15.8
3	27.5	3	15.8	3	15.9
4	27.8	4	16.0	4	16.1
5	28.0	5	16.1	5	16.1
6	28.9	6	16.6	6	16.7
7	28.9	7	17.3	7	17.5
8	33.0				

11.5.3 Growth of $\nabla \cdot \omega$ and Z-impulse in filament methods

Table 11.2 shows the divergence of the smoothed vorticity for three different filament runs. In each case, the divergence begins to decline, and then later begins to grow. This growth is inconsistent with the fact that for a solution of the unconstrained VTE which is a limit of viscous solutions, the divergence of ω must not increase over time. Table 11.3 indicates that the linear impulse is not conserved in these methods, even when only the divergence free part of the vorticity is used in the computation. In each case, the initial X and Y impulse are zero.

11.5.4 "Discontinuities" in the Area of Cross Sections

In many of the runs, but most clearly in fourth.1, jumps in the area of the cross section of the ring are seen. These jumps are seen also in base.1. In base.5, which uses an algorithm which was designed to oversmooth stretching effects, one sees core area varying waves of the same frequency as those in fourth.1, however these waves are smooth. Recall that for rings of circular cross section, the evolution of the cross sectional area is analogous to the evolution of density in the shallow water

equations. Further work should include a determination of the propagation velocity of these jumps, a determination of whether these jumps are entropy satsisfying or entropy violating, and a comparison of this solution to solutions of the shallow water equations. (see section 5.5[53][37][39])

11.6 Future Work

Just as in the one dimensional case, there are more quantitative tools which can be used on this problem. In particular, the integration techniques of Strain and the Fourier domain techniques mentioned in Chapter 4 could be used on this problem as well.

11.6.1 Invariants and Fourier Space Evolution

Just as in Chapter 4, it would be interesting to evaluate the invariants of the flow more accurately and to observe the evolution of different Fourier or wavelet components. In addition, the following possible projects are

11.6.2 Numerically demonstrate the non-uniqueness of particle values

Visualizations of the smooth, divergence-free vorticity field show it to be much smoother and simpler than the vortex filaments or blobs themselves. This is particularly clear in run fourth.1. It should be possible to demonstrate a smoother filament distribution which represents a vorticity field which is very close to the one represented by the filaments of run fourth.1, if the arguments of Chapter 8 hold. Again, Fourier space techniques such as those used by Krasny[34] could be helpful in this area.

Chapter 12

Boundary Conditions Schemes

In this chapter, a boundary condition algorithm due to Chorin, which was used in the earliest vortex method calculation [11], and in the computation of chapter 2 is reviewed. Two new boundary algorithms based on deterministic diffusion schemes are not presented. The sheet algorithm [12] and its possible extension to three dimensions are not discussed. Work in that area will be reported in [49] in the context of the implementation of vortex methods on the CM5.

12.1 Chorin's Algorithm

The boundary algorithm of Chorin [11] extends easily from two to three dimensions. Suppose that the Reynolds number is R in a domain D, with boundary ∂D and surface unit normal vector $\hat{n}$. Given a vorticity induced flow u_{vor} Chorin finds a potential flow $u_{pot} = \nabla\phi$, such that $\hat{n} \cdot (u_{pot} + u_{vor}) = 0$ on the boundary, by solving $\Delta\phi = 0$ in D with boundary condition $\hat{n} \cdot \nabla\phi = -\hat{n} \cdot u_{vor}$ on ∂D. One thus has a velocity field $u = u_{vor} + u_{pot}$ which satisfies the no leak, but not the no slip condition. Suppose a set of points b_i with area dA_i on ∂D are given. The slip velocity v_{slip} is evaluated at each point b_i. At each point b_i, this slip velocity may be interpreted as a vortex sheet with strength per unit area equal to $\hat{n} \times u$. The total amount vorticity which is present in the area of that point is then $dA_i \hat{n}(b_i) \times u(b_i)$. A new particle or several new particles, which represent that same amount of vorticity are created at that point Each of these particles is then given a random walk in the direction normal to the boundary. The distance of the walk is drawn from a Gaussian distribution with mean zero and variance $\frac{4\pi\Delta t}{R}$. The full random walk algorithm is then used to model the diffusion in the interior. The outline summary for this algorithm is

- Given u_{vor}, compute $u_{pot} = \nabla\phi$ by solving

$$\Delta\phi = 0 \tag{12.1}$$

$$\hat{n} \cdot \nabla\phi = -\hat{n} \cdot u_{vor} \tag{12.2}$$

- Given $u = u_{vor} + u_{pot}$ and a set of points b_i on ∂D, compute

$$\omega_i = dA_i \hat{n}(b_i) \times u(b_i) \tag{12.3}$$

- Create a particle or particles at b_i with total strength ω_i
- Position these particles a random distance y_i from the boundary, where y_i is drawn from a zero mean Gaussian with variance $\frac{4\pi\Delta t}{R}$

In the implementation of this algorithm in chapters 2 and 13, a panel method is used to compute the potential flow, u_{pot} and the panel centroids are used as the boundary sampling points b_i.

This algorithm has been used with some success in two dimensions. Several observations may be made which may have some relation to the effectiveness of the algorithm in three dimensions.

- This algorithm models a noisy boundary layer. The error of the algorithm looks like Gaussian noise [11].
- The algorithm is coupled to the use of the random walk in the interior, whose error term also looks like Gaussian noise[11].
- The long time evolution of two dimensional free space inviscid flow is stable, by Onsagers theorem. [40]
- Three dimensional inviscid flow has unstable long time evolution.

These observations lead one to suspect that the noise of the random walk could excite instabilities of the underlying system, (the VTE), in addition to modeling the diffusion. There are numerical experiments [49] which support this suspicion.

12.2 A Deterministic Boundary Condition Algorithm

The basic idea of this algorithm is to use a deterministic diffusion scheme instead of the random walk in order to get the vorticity off of the boundary and into the interior. It is hoped that by using a deterministic scheme, which converges faster than the random walk, the number of points needed to resolve the boundary layer will be reduced. In addition, it is hoped that the introduction of noise into a possibly unstable system will be reduced. In this algorithm, one assumes the same situation as in the previous section. The set of particles is called $P = \{p_k\}$. The particles have have locations x_k, and strengths, w_k. In addition, assume that associated with each point b_i, there are a number of creation stations c_{ij}. Each creation station is a location above the point b_i. To each of the c_{ij}, a piece of the interior which contains it and no other station is assigned. Call the volume of this section of the interior, V_{ij}, and by abuse of notation, call the section itself V_{ij} as well. The

condition that each such volume, V_{ij} must contain at least one particle will be imposed. If this is not so, a particle is created at c_{ij}. Volume estimation is performed for each particle in the boundary layer.

The following is an outline of the algorithm.

- Given u_{vor}, derived from $P = \{p_k\}$, compute $u_{pot} = \nabla\phi$ by solving

$$\Delta\phi = 0 \tag{12.4}$$

$$\hat{n} \cdot \nabla\phi = -\hat{n} \cdot u_{vor} \tag{12.5}$$

- Given $u = u_{vor} + u_{pot}$ and a set of points b_i on ∂D, compute

$$w_i = dA_i \hat{n}(b_i) \times u(b_i) \tag{12.6}$$

- For each volume V_{ij}
  - if V_{ij} is empty, create a new particle with strength 0 and location c_{ij}, and add that particle to P
- For each particle p_k
  - Assign a volume to p_k using one of the volume estimation techniques
- For each point b_i,
  - For each particle p_k
    - if $(\hat{n}(b_i) \cdot (p_k - b_i)) > 0$, using the unsymmetrized form of your favorite deterministic diffusion scheme, with cutoff parameter $\sigma_{boundary}$, compute the change in ω_k due to b_i, and the change in w_i due to p_k
    - update ω_k and accumulate the change in w_i in a temporary variable
  - update w_i, using the sum of the stored changes in the temporary variable.

Two different deterministic diffusion schemes were used as the base for this algorithm, the scheme of MasGallic and Cottet and the scheme of Fishelov for a Gaussian core.

12.3 A Second Deterministic Algorithm

The second deterministic scheme is motivated by the observation that at no time in either of the two boundary algorithms of this chapter are both the no flow and no slip simultaneously satisfied. In this algorithm, the algorithm of section 12.2 is used as a base. However, instead of first finding $\nabla\phi$ and then determining a surface vorticity distribution, both are accomplished at the same time.

Recall that in the panel method, there are panels B_i with control points, b_i. A matrix equation for panel strengths σ_i is derived, by computing the influence of a surface source distribution of strength, σ_i over B_i on the normal component of the velocity at b_j In this case, one considers the influence on the full velocity at b_j of both a source distribution, σ_i and a surface vorticity distribution, (vortex sheet), of strength γ_i, where γ is a vector in the plane of panel B_i. If there are N panels, this gives a set of $3N$ equations in $3N$ variables which seem to be only slightly worse in condition number than the matrix which arises from the panel method applied to the same geometry. An outline of this algorithm is:

- Given u_{vor}, derived from $P^n = \{p_k\}$, compute a surface source distribution σ_i, a surface vorticity distribution, γ_i and the velocity field, u_{bc}, which they induce, by solving, the matrix equation

$$\tag{12.7}$$

$$\tag{12.8}$$

- Given, γ_i compute

$$w_i = dA_i \gamma_i \tag{12.9}$$

- For each volume V_{ij}
  - if V_{ij} is empty, create a new particle with strength 0 and location c_{ij}, and add that particle to P
- For each particle p_k
  - Assign a volume to p_k using one of the volume estimation techniques
- For each point b_i,
  - For each particle p_k
    - if $(\hat{n}(b_i) \cdot (p_k - b_i)) > 0$, using some deterministic diffusion scheme, with cutoff parameter $\sigma_{boundary}$, compute the change in ω_k due to b_i, and the change in w_i due to p_k
    - update ω_k and accumulate the change in w_i in a temporary variable
  - update w_i, using the sum of the stored changes in the temporary variable.
- throw away the surface source and vorticity distributions
- compute new surface source and vorticity distributions which arise due to the influence of the new p_k and ω_k on the boundary

12.4 Generating Connected Filaments

This section presents joint work with Sethian. An approach which was investigated theoretically, but not practically, was the possibility of generating connected filaments of vorticity on the boundary, in order to impose the boundary condition. Suppose we have a vorticity distribution on the boundary ∂D of our region D. We could have obtained such a vorticity distribution by evaluating the slip velocity or by solving the matrix equation of section 12.3. We would like to find the integral curves of this vorticity distribution, without actually doing the ODE integration. We have a vorticity distribution ω which is defined on ∂D and everywhere parallel to ∂D. Construct $\nabla \times \omega$, which will be everywhere normal to ∂D. Therefore, we may regard it as a scalar, F. Then solve $\triangle G = F$. Note that $\omega = \nabla \times G$ This implies that the level curves of G are the integral curves of ω. It is clear that there will be difficulties trying to make this work on a body of any complexity. It appears that a coordinate atlas for the body would be needed. This method was not pursued further.

Chapter 13

Flow Past a Sphere

This thesis began as a project to compute flow past a sphere using the vortex method. In this chapter, an attempt to compute flow past a sphere using a vortex method is described. The algorithms used have been described in previous chapters. Other vortex method based sphere calculations have been made in the past, with mixed results. [49] [24]. Both use the random walk to model the diffusion and a Helmholtz theorem based stretch term. Both also use a "vortex tile" element in a numerical boundary layer.

13.1 The Code and Algorithms

A program was written to compute flow around an arbitrary body using the vortex method. This program built upon the unconnected vortex method code of chapter 4, A new panel method (which will be called sPanel) was added along with a module (called bcPanel) based on the panel solver which solves for both the surface potential and surface vorticity distributions of section 12.3 All three boundary algorithms of chapter 12 were implemented, using this base. The ODE solvers were extended to know about the influence of the boundary elements and a constant ambient flow.

13.2 Verification

In addition to the verification tests mentioned in chapter 4, the following tests were performed.

1. Potential flow past a sphere was computed and visualized.
2. Lifting Potential flow past a sphere was computed and visualized.
3. A source distribution and vorticity distribution was computed using bcPanel. The effect of that distribution was then computed in two ways and the results compared. The effect was

computed by bcPanel and the effect was computed by treating the surface vorticity distribution as a distribution of blobs and treating the source distribution as if it had come from sPanel and using sPanel to compute its effect.

4. The transport of an unperturbed single filament ring around the sphere was computed.

5. The degree to which the boundary conditions are satisfied is checked at various points in the loop.

13.3 The Parameter Space and the Algorithms

This outline describes the algorithms which are implemented by this code. The core function must be selected at compile time, but all other choices may be made at run time.

- Stretch Algorithm
  - Helmholtz Theorem based stretch
  - Stretch based on $\omega_i \cdot \nabla_h u$
  - Stretch based on $\omega_i \cdot \nabla_h u^T$
  - Stretch based on $\omega(x) \cdot \nabla_h u^T$ with resampling
- ODE Solver
  - Eulers Method
  - Heuns Method
  - Fourth Order Runge-Kutta
- Diffusion Method
  - Fishelovs algorithm
  - Cottets algorithm
  - Mas-gallics algorithm.
  - Finite Difference Based Deterministic Diffusion
  - Random Walk
- Boundary Algorithm
  - Chorins 1973 algorithm
  - Algorithm of section 12.2
  - Algorithm of section 12.3

- Core Function
  - Second order core function based on $f(r) = \frac{3}{4\pi}e^{-r^3}$
  - Fourth order core function based on $f(r) = \frac{3}{4\pi}(\frac{5}{2} - \frac{3}{2}r^3)e^{-r^3}$
  - Gaussian core function

The following are the non-physical numerical parameters which must be set

- Velocity core size
- Vorticity core size
- Volume core size
- Boundary core size
- Maximum length of a vortex element
- Time step size
- Height of the numerical diffusion layer
- Number of cells above each panel center in the numerical diffusion layer
- Diagnostic interval
- Output file write interval

13.4 Machines and Parallelism

This code has been run on the following uniprocessor machines. sun sparc models 1,2 and 10, IBM rs6000 models 550, 560, dec alpha 3000-400, cray XMP, cray YMP, cray C90, cray 2s, It has been parallelized to run on a multiprocessor C90. It achieves a speedup of just under 8 at 8 processors, but falls off to 14 at 16 processors. The reason for this was not discovered. When compiled on a Cray2, the parallel code consumes too much local memory. No effort was made to remedy this.

13.5 Dirty Tricks

It was found that with reasonable step sizes, from time to time, vortex elements would be transported inside the body. This causes great problems since it suddenly reverses the effect of that element on the boundary. For small enough time step sizes, using the algorithm of section 12.3 (about 0.0001 for Heuns Method) this does not occur, but basing the time step choice on this consideration

would necessitate the use of a excessively small step size. For the other boundary algorithms, a small enough step size to avoid penetration was not found. Therefore, at the beginning of each time step, all elements which are inside the body are deleted. Heuristically, this is a reasonable thing to do. Since the element is deleted, its effect on the boundary is also deleted, which causes the surface vorticity distribution to be increased by an amount which is similar in size to the deleted element. That vorticity is then diffused off of the boundary. The net effect is similar to stopping the element as it hits the boundary and adding it to the surface vorticity distribution.

For Eulers method, this solves the problem, however for other methods, a particle could enter the body on one of the predictor steps of the ODE solver. The computed velocity field inside the body does not represent anything physical and is therefore not reasonable. Occasionally, a particle would be shot out of the body in some random direction after entering on a predictor step. In order to avoid this, the velocity is amended so that inside the body the velocity always evaluates to zero. Thus if a particle enters on a predictor step, it will remain there long enough to be deleted at the beginning of the next time step.

13.6 The Sphere

In order for this implementation to work, a representation of the body as a list of panels is required. A representation of the sphere was created from a program by Sethian which recursively subdivides a tetrahedral approximation to the interior of sphere, and extracts the implied triangular surface mesh. [45] A second representation based on polar coordinates was also used.

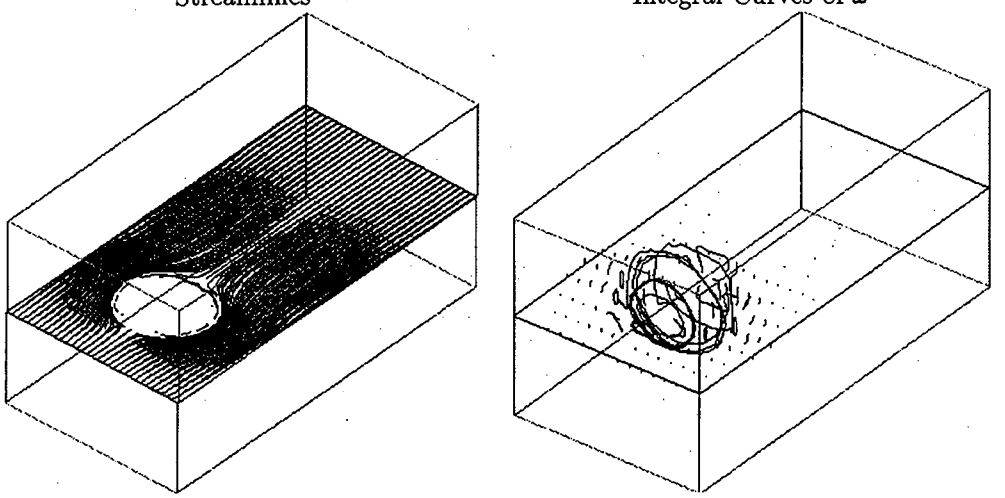
13.7 Runs

There was not sufficient computer time to do the type of study of the algorithms implemented by this code which should be done. In this section, one fairly poorly resolved run is reported. For this run, the sphere is of diameter 1. Its surface is represented by 36 panels based on the polar coordinates representation. The viscosity was $\frac{1}{100}$ The velocity, vorticity and volume core cutoff parameters were 0.2. The boundary cutoff was 0.17 The maximum allowed length of a vortex element was 0.1. The time step size was 0.005. The height of the numerical boundary layer was 0.2 The number of cells above each panel was 8. The stretch algorithm was $\omega_i \cdot \nabla_h u^T$. The diffusion scheme used was Cottets. The boundary layer algorithm was that of section 12.3.

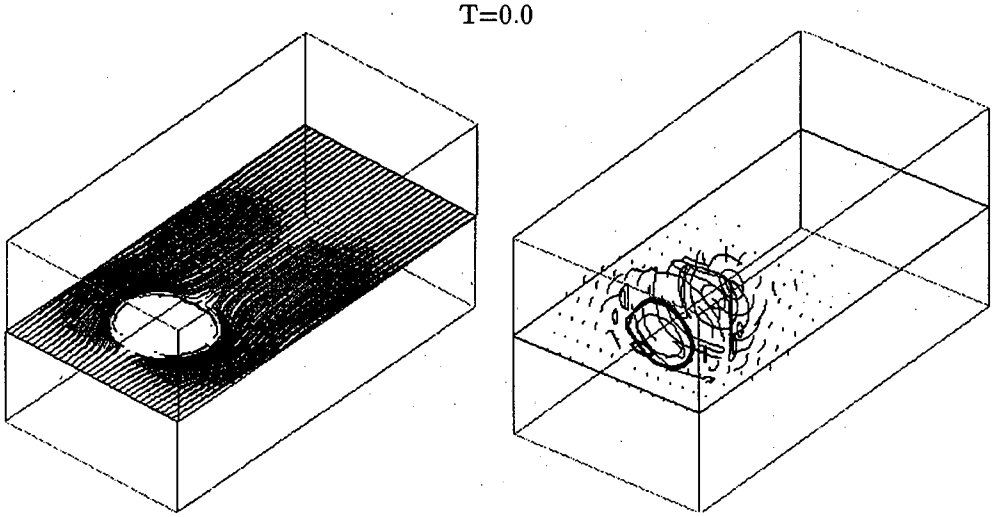
It is clear from the figures that vorticity is shed off of the sphere and that it reconnects to some degree in the boundary layer. For instance, one can see small vortex rings being formed in the boundary layer. There is a visible boundary layer whose thickness is approximately 0.15. Much more experimentation with this code is needed before any more meaningful conclusions can be drawn. Vorticity is shed and moves downstream. Small streamwise vorticity structures are beginning to develop by the end of the computation.

Figure 13.1: Flow Past a Sphere as Computed by a Vortex Method

Streamlines Integral Curves of ω



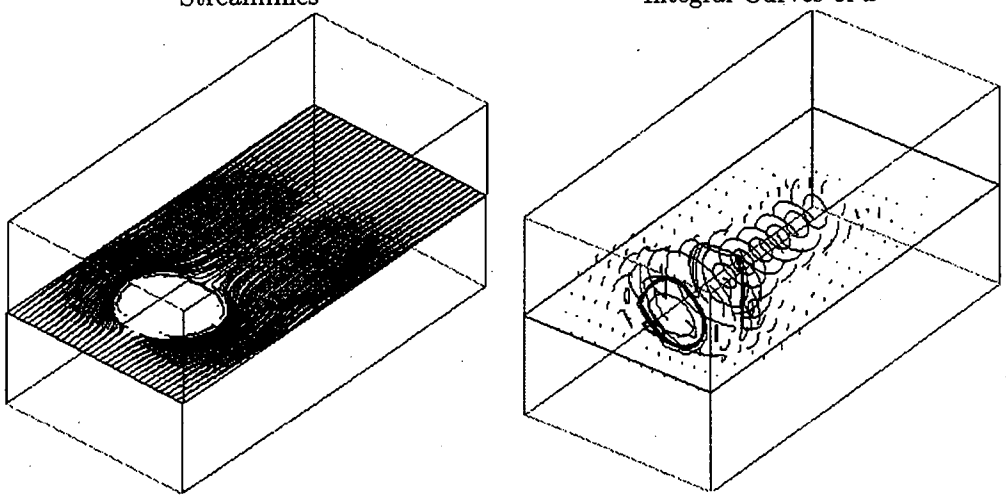
T=0.0



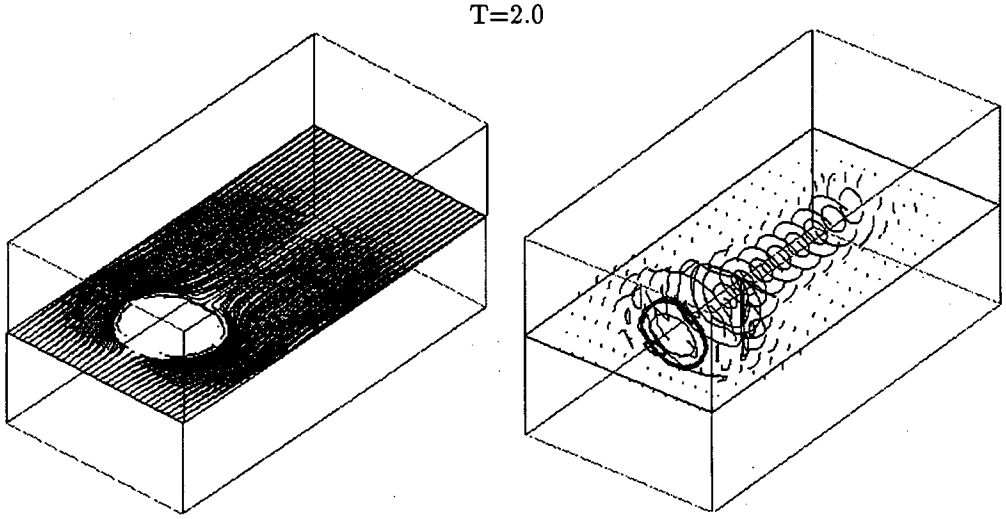
T=1.0

Figure 13.2: Flow Past a Sphere as Computed by a Vortex Method (contd.)

Streamlines Integral Curves of ω



T=2.0



T=2.5

Bibliography

[1] C. Anderson and C. Greengard. On vortex methods. *Siam Journal on Numerical Analysis*, 22(3), June 1985.

[2] G.K. Batchelor. *An Introduction To Fluid Dynamics.* Cambridge University Press, 1967.

[3] J.T. Beale and A. Majda. Vortex methods i: Convergence in three dimensions. *Mathematics of Computation*, 39(159), July 1982.

[4] J.T. Beale and A. Majda. Vortex methods ii: Higher order accuracy in two and three dimensions. *Mathematics of Computation*, 39(159), July 1982.

[5] J.T. Beale and A. Majda. High order accurate vortex methods with explicit velocity kernels. *Journal of Computational Physics*, 58, 1985.

[6] T.F. Buttke. The hamiltonian structure of incompressible fluid flow in three dimensions. manuscript submitted to Phys.Rev.Letters.

[7] T.F. Buttke. Lagrangian numerical methods which preserve the hamiltonian structure of incompressible flud flow. Technical Report LBL-32712, Lawrence Berkeley Laboratories, July 1992.

[8] A.J. Chorin. Notes from a class in turbulence, U.C. Berkeley.

[9] A.J. Chorin. Numerical solution of the navier-stokes equations. *Mathematics of Computation*, 22(104), 1968.

[10] A.J. Chorin. On the convergence of discrete approximations to the navier-stokes equations. *Mathematics of Computation*, 23(106), 1969.

[11] A.J. Chorin. Numerical study of slightly viscous flow. *Journal of Fluid Mechanics*, 57(4), 1973.

[12] A.J. Chorin. Vortex sheet approximation of boundary layers. *Journal of Computational Physics*, 27(3), 1978.

[13] A.J. Chorin. Vortex models and boundary layer instability. *Siam Journal on Scientific and Statistical Computing*, 1(1), 1980.

[14] A.J. Chorin. Evolution of a turbulent vortex. *Communications in Mathematical Physics*, 83, 1982.

[15] A.J. Chorin. Scaling laws in the vortex lattice model of turbulence. *Communications in Mathematical Physics*, 114, 1988.

[16] A.J. Chorin. Vortices, turbulence and statistical mechanice. In K.E. Gustafson and J.A. Sethian, editors, *Vortex Methods and Vortex Motion*, chapter 8. siam, 1991.

[17] A.J. Chorin. Hairpin removal in vortex interactions ii. *Journal of Computational Physics*, 107, 1993.

[18] A.J. Chorin and J.E. Marsden. *A Mathematical Introduction to Fluid Mechanics.* Springer-Verlag, 1990.

[19] G.-H. Cottet. A particle-grid superposition method for the navier-stokes equations. *Journal of Computational Physics*, 89, 1990.

[20] G.H. Cottet and S. Mas-Gallic. A particle method to solve the navier-stokes system. *Numerische Mathematik*, 57, 1990.

[21] R. Courant and D. Hilbert. *Methods of Mathematical Physics Vol I, VolII.* wiley, 1937.

[22] D. Fishelov. A new vortex scheme for viscous flows. *Journal of Computational Physics*, 86(1), January 1990.

[23] H. Fujita and T. Kato. On the navier-stokes initial value problem. *Arch. Rational Mech Anal.*, 16, 1964. MR 29 Number 3774.

[24] J. Grant and J. Ullman. personal communication.

[25] C. Greengard and E. Thomann. Singular vortex systems and weak solutions of the euler equations. *Physics of Fluids*, 31(10), 1988.

[26] K.E. Gustafson and J.A. Sethian eds. *Vortex Methods and Vortex Motion.* siam, 1991.

[27] O. H. Hald. Convergence of vortex methods for euler's equations. ii. *Siam Journal on Numerical Analysis*, 16(5), October 1979.

[28] O.H. Hald. Convergence of vortex methods. In K.E. Gustafson and J.A. Sethian, editors, *Vortex Methods and Vortex Motion*, chapter 2. siam, 1991.

[29] H. Hasimoto. A soliton on a vortex filament. *Journal of Fluid Mechanics*, 51(3), 1972.

[30] J.L. Hess and A.M.O. Smith. Calculation of potential flow about arbitrary bodies. *Prog. Aero. Sci.*, 8, 1967.

[31] J. Palis Jr. and W. de Melo. *Geometrical Theory of Dynamical Systems.* Springer-Verlag, 1982.

[32] S.N. Kempka and J.H. Strickland. A method to simulate viscous diffusion of vorticity by convective transport of vortices at a non-solenoidal velocity. Technical Report SAND93-1763 UC-700, Sandia National Labs, August 1993.

[33] O. Knio and A. Ghoniem. Numerical study of a three dimensional vortex method. *Journal of Computational Physics*, 86, 1990.

[34] R. Krasny. *Journal of Fluid Mechanics*, 167, 1986.

[35] R. Krasny. Desingularization of periodic vortex sheet roll-up. *Journal of Computational Physics*, 65, 1986.

[36] R.J. LeVeque. *Numerical Methods for Conservation Laws.* Birkhaeuser, 1990.

[37] T.S. Lundgren and W.T. Ashurst. Area-varying waves on curved vortex tubes with application to vortex breakdown. *Journal of Fluid Mechanics*, 200, 1989.

[38] Marsden and Weinstein. Coadjoint orbits, vorticies and clebsch variables for incompressible fluids. *Physica 7D*, pages 305–323, 1993.

[39] J.S. Marshall. A general theory of curved vorticies with circular cross-section. *Journal of Fluid Mechanics*, 229, 1991.

[40] L. Onsager. Statistical hydrodynamics. *Nuovo Cimento.*, 6, 1949.

[41] V.I. Oseledets. On a new way of writing the navier-stokes equations: the hamiltonian formalism. *Russian Math. Surveys*, 44, 1989.

[42] A. Qi. Three dimensional vortex methods for the analysis of wave propagation on vortex filaments. Technical Report LBL-30487, Lawrence Berkeley Laboratories, March 1991.

[43] C. Rehbach. Numerical calculation of three-dimensional unsteady flows with vortex sheets. *Rech. Aerosp.*, 5, 1978.

[44] L. Rosenhead. The formation of vortices from a surface of discontinuity. *Proc.Roy.Soc.London Ser. A*, 134, 1931.

[45] J.A. Sethian. personal communication.

[46] J.A. Sethian. Curvature and the evolution of fronts. *Communications in Mathematical Physics*, 101, 1985.

[47] J.A. Sethian. A brief overview of vortex methods. In K.E. Gustafson and J.A. Sethian, editors, *Vortex Methods and Vortex Motion*, chapter 1. siam, 1991.

[48] J.A. Sethian. A gallery of fluid motion using vortex methods. In K.E. Gustafson and J.A. Sethian, editors, *Vortex Methods and Vortex Motion*, chapter 3. siam, 1991.

[49] J.A. Sethian, A. Greenberg, and D.W. Pryor. 3d vortex methods on the connection machine. In preparation.

[50] J.A. Sethian and S. Osher. Fronts propagating with curvature dependent speed: Algorithms based on hamilton-jacobi formulations. *Journal of Computational Physics*, 78, 1988.

[51] J. Strain. Fast potential theory ii: Layer potentials and discrete sums, March 1990. manuscript submitted for publication.

[52] J. Strain. Locally-corrected multidimensional quadrature rules for singular functions. Technical Report LBL-34683, Lawrence Berkeley Laboratories, September 1993.

[53] A. Weigand. *The Response of a Vortex Ring to a Transient, Spatial Cut.* PhD thesis, University of California, San Diego, 1993.

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

AAT844
LBL Libraries