INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

Bell & Howell Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA

UMI®
800-521-0600

Temple University
Doctoral Dissertation
Submitted to the Graduate Board

Title of Dissertation: Efficient Computation of Viscous Incompressible Flow
(Please type)

Author: Hans Johnston
(Please type)

Date of Defense: May 19, 1999
(Please type)

Dissertation Examining Committee: (please type)	Read and Approved By: (Signatures)
Dr. Jian-Guo Liu Dissertation Advisory Committee Chairperson	
Dr. Shiferaw Berhanu	S. Berhanu
Dr. Yuan Shi	
Dr. Daniel Szyld	
Dr. Shiferaw Berhanu Examining Committee Chairperson	S. Berhanu If Member of the Dissertation Examining Committee

Date Submitted to Graduate Board: 6/17/99

Accepted by the Graduate Board of Temple University in partial fulfillment of the requirements for the degree of **Doctor of Philosophy**.

Date 9/22/99

Peter W. Goodman
(Dean of the Graduate School)

EFFICIENT COMPUTATION OF VISCOUS INCOMPRESSIBLE FLOW

A Dissertation

Submitted to

the Temple University Graduate Board

---

in Partial Fulfillment

of the Requirements for the Degree

DOCTOR OF PHILOSOPHY

---

by

Hans E. Johnston

August 1999

ABSTRACT

EFFICIENT COMPUTATION OF VISCOUS INCOMPRESSIBLE FLOW

by Hans E. Johnston

Doctor of Philosophy

Temple University, 1999

Advisor: Dr. Jian-Guo Liu

The present work is concerned with the development of numerical methods for the computation of viscous incompressible flows as described by the Navier-Stokes equations of classical fluid mechanics. The emphasis is strictly on finite difference schemes implemented on non-staggered Cartesian grids, which when coupled with a high order explicit time stepping procedure result in very simple and efficient methods.

In chapter 2 a second order method based on the primitive variable formulation of the Navier-Stokes equations is presented. The scheme is suited for the computation of both low and high Reynolds number flows. The novelty of the scheme lies primarily in a simple, consistent, and accurate numerical approximation of the Neumann boundary condition for the pressure Poisson equation. Its use avoids the need for both fractional-step time discretizations and staggered

grids traditionally required of the most popular numerical methods based on the primitive variable formulation. The resulting method achieves clean second order accuracy when applied to 1D and 2D test problems, and performs equally well as a second order vorticity-stream function based scheme when used to compute the canonical cavity flow. We note that the scheme can be easily extended to compute 3D flows.

In chapter 3 new developments are presented for a class of fourth order Essentially Compact methods (EC4), originally developed by E and Liu, for solving unsteady viscous incompressible flows in the vorticity-stream function formulation. A novel fine grid patch mesh refinement technique, which is easily incorporated into the original EC4 scheme, is outlined. Its use results in a dramatic increase in computational efficiency, particularly for high Reynolds number flows. In addition, we present results on the use of a very effective far-field boundary condition for the stream function. As a detailed illustration of an application of the above mentioned methodologies, we present high resolution benchmark quality simulations of the impulsively started flow past a circular cylinder at Reynolds numbers ranging from $1,000$ to $100,000$. Even at the considerably high Reynolds number of $100,000$ the flow is completely resolved.

ACKNOWLEDGEMENTS

I would like to convey my sincerest gratitude to my advisor Professor Jian-Guo Liu for his guidance and support over these past few years. He has shared with me what I have found to be a unique view in the world of computational fluid dynamics.

I would also like to thank Professors Daniel Szyld, Shiferaw Berhanu, and Yuan Shi for serving as members of my examining committee. In particular, I wish to thank Dr. Szyld for his patient reading of various forms of this dissertation. Furthermore, I am deeply indebted to Professors David Hill, David Zitarelli, and Raymond Coughlin for their continuous support and guidance during my entire college career. Finally, a special thanks to Professor Marvin Knopp for getting me into this mess in the first place.

On a more personal note, I wish to thank my friends Aaron, Judy, Myra, Andrew, Kurt, and Jim for actually making this little journey fun and for providing, from time to time, some much needed perspective. In a similar vain, a very special thanks to Fran, Jeff, and Mark for spending time with me while I was at home in my foxhole. And a very warm thanks to Andi for all of her support and continued encouragement.

A majority of the computational results presented herein were performed at the Pittsburgh Supercomputing Center under grant DMS960005P. I am very grateful to the staff for providing excellent technical support.

DEDICATION

I would like to dedicate this work to my sisters Johanne and Heidi, my brother Kurt, and most importantly my mother Anne.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
PRELIMINARIES

1.1 Ubiquitous Fluid Flow: Which Way Did It Go?

Everyone is aware from everyday experience that gases and liquids are in motion all around us. For instance, the simple fact that you are reading these words implies that blood is flowing through your circulatory system, and that air is moving in and out of your lungs. Other obvious examples range from the movement of the clouds in the sky to the water that swirls around your feet and down the drain when you take a shower.

Each of the examples above involves a "fluid" that is in motion, and one would not be too hard pressed to categorize some of the resulting flow patterns as quite complex. The study of such problems is the primary interest of the field of fluid mechanics, the goal of which is not only to understand the physical mechanisms that produce these motions, but also to predict what will happen to a fluid, with predefined properties, in a given physical arrangement. In many instances valuable information of particular flows result from very precise investigations performed by experimentalists in a laboratory setting. Alternatively, starting from fundamental physical principles one can produce a mathematical model for such problems by deriving the equations governing the flow. However, more often

than not, these equations take the form of a system of nonlinear partial differential equations. In general, in the absence of simplifying assumptions, which in many instances obscure the true nature and intricacies present in the flow, current analytic techniques prove insufficient for solving the governing equations.

Enter the field of computational fluid dynamics, practitioners of which use high speed digital computers is to calculate "approximate" solutions to the governing flow equations. It is precisely this approach to the study of fluid dynamics that is the focus of this thesis. The governing equations that we seek approximate solutions to are given by the classical time dependent Navier-Stokes equations for a viscous incompressible fluid, which we discuss next.

1.2 The Navier-Stokes Equations

Let Ω be a bounded region in $\mathbf{R}^2$ containing a viscous incompressible single component Newtonian fluid. By applying to this fluid the principle of conservation of mass, along with Newton's second law of motion, one arrives at the incompressible Navier-Stokes equations (NSE). With the additional simplifying assumption that the density of the fluid is constant, and when written in non-dimensionalized form, the NSE for an incompressible viscous fluid are given by:

$$\begin{cases} \boldsymbol{u}_t + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u} + \nabla p = \frac{1}{Re}\Delta\boldsymbol{u} \\ \nabla\cdot\boldsymbol{u} = 0 \end{cases} \tag{1.2.1}$$

in Ω, where $\boldsymbol{u} = (u(x,y,t), v(x,y,t))$ is the velocity and $p(x,y,t)$ is the pressure. We note that ∇ and Δ represent the gradient and Laplace operators, respectively, and the subscript t denotes differentiation with respect to time. The parameter Re is a dimensionless quantity representing the ratio of inertial forces to viscous forces in the flow. The system (1.2.1) is referred to as the primitive variable formulation of the NSE given by the momentum equation and incompressibility

condition, respectively. Thus in this formulation, in 2D, the NSE are a system of three coupled nonlinear partial differential equations. For a detailed derivation see [CM].

To make the statement of the problem complete we must specify both initial and boundary conditions for (1.2.1). As for initial conditions, the velocity field at time $t = 0$, $\boldsymbol{u}_0$, is assumed to be incompressible, i.e., $\nabla \cdot \boldsymbol{u}_0 = 0$. Now, let $\Gamma = \partial\Omega$ denote the boundary, which is assumed to be comprised of solid walls at rest. Since second order derivatives of $\boldsymbol{u}$ appear in (1.2.1) we must prescribe two boundary conditions. Let $\boldsymbol{n}$ and $\boldsymbol{\tau}$ denote the unit normal and unit tangential vectors to Γ, respectively. The physically evident fact that the fluid cannot cross Γ leads to the condition $\boldsymbol{u} \cdot \boldsymbol{n} = 0$. This is called the no-penetration condition. Also, due to the effects of viscosity one would expect that very close to a stationary wall that the tangential velocity of the fluid should approach 0. This is in fact found experimentally to be the case. Thus, $\boldsymbol{u} \cdot \boldsymbol{\tau} = 0$, which is known as the no-slip condition. Together we then have

$$\begin{cases} \nabla \cdot \boldsymbol{u}_0 = 0 \\ \boldsymbol{u}|_\Gamma = 0 \quad \forall\, t. \end{cases} \tag{1.2.2}$$

The numerical method outlined in chapter 2 will employ an alternative primitive variable formulation of the NSE which we now derive. Taking the divergence of the momentum equation, commuting the space and time derivatives, and using the incompressibility condition $\nabla \cdot \boldsymbol{u} = 0$, we obtain the pressure Poisson equation (PPE):

$$\Delta p = -\nabla \cdot (\boldsymbol{u} \cdot \nabla \boldsymbol{u}). \tag{1.2.3}$$

However, the PPE and the momentum equation, along with (1.2.2), are not sufficient to ensure that the velocity field $\boldsymbol{u}$ remains incompressible for all $t > 0$. That $\boldsymbol{u}$ and p satisfy the PPE is only necessary, but not sufficient, condition to

ensure that $\boldsymbol{u}$ remains divergence-free. To see this, assume that $(\boldsymbol{u}, p)$ satisfies the momentum equation and the PPE subject to the conditions in (1.2.2). Then again taking the divergence of the momentum equation, and using the PPE, we arrive at

$$(\nabla \cdot \boldsymbol{u})_t = \frac{1}{Re}\Delta(\nabla \cdot \boldsymbol{u}).$$

Along with $\nabla \cdot \boldsymbol{u}_0 = 0$, all this tells us is that the scalar quantity $\nabla \cdot \boldsymbol{u}$ satisfies the heat equation with homogeneous initial conditions. Therefore, to ensure that the velocity field remains incompressible for all $t > 0$ we must impose the additional condition that $\nabla \cdot \boldsymbol{u}|_\Gamma = 0$ at any time t; see [Qu]. With this additional condition the formulation (1.2.1)–(1.2.2) is equivalent to

$$\text{(1.2.4)} \qquad \begin{cases} \boldsymbol{u}_t + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} + \nabla p = \frac{1}{Re}\Delta \boldsymbol{u} \\ \Delta p = -\nabla \cdot (\boldsymbol{u} \cdot \nabla \boldsymbol{u}) \\ \boldsymbol{u}|_\Gamma = \nabla \cdot \boldsymbol{u}|_\Gamma = 0 \quad \forall\, t. \end{cases}$$

Note that in (1.2.4) there is no explicit boundary condition for the elliptic PPE. We can derive one by taking the dot product of the momentum equation with the unit normal $\boldsymbol{n}$ along Γ. Since $\boldsymbol{u}|_\Gamma = 0$, we obtain

$$\text{(1.2.5)} \qquad \frac{\partial p}{\partial \boldsymbol{n}} = (\frac{1}{Re}\Delta \boldsymbol{u}) \cdot \boldsymbol{n} \quad \text{on} \quad \Gamma,$$

a Neumann boundary condition. Equation (1.2.5) involves evaluating the viscous term at the boundary, which in a numerical setting is the most difficult term to compute. It is precisely this issue that is the main focus of chapter 2. The key is to evaluate, in a consistent manner, the viscous term on Γ with the aid of the boundary condition $\nabla \cdot \boldsymbol{u}|_\Gamma = 0$.

In chapter 3 we turn our attention to the vorticity-stream function (ω, ψ) formulation of the NSE which we now derive from (1.2.1). The vorticity ω at any point in Ω is given by

$$\text{(1.2.6)} \qquad \omega = \nabla \times \boldsymbol{u} = \partial_x v - \partial_y u,$$

the curl of the velocity vector field. ω is the local angular velocity of the fluid which in 2D is a scalar quantity. Taking the curl of the momentum equation in (1.2.1), again commuting space and time derivatives, we arrive at

$$\omega_t + (\boldsymbol{u} \cdot \nabla)\omega = \frac{1}{Re}\Delta\omega,$$

which is referred to as the vorticity transport equation. It is important to note that the pressure p no longer appears since the curl of a smooth gradient field is zero.

Now assume that Ω is simply connected. The incompressibility condition implies that there exists a scalar function $\psi(x, y, t)$ defined on Ω such that

$$u = \partial_y \psi \quad \text{and} \quad v = -\partial_x \psi.$$

The stream function ψ is unique up to an additive constant, and is so named since for a fixed time t the streamlines of the flow lie on level curves of ψ. With ψ in hand, we rewrite (1.2.6) as

$$\omega = \partial_x v - \partial_y u = \partial_x(-\partial_x \psi) - \partial_y(\partial_y \psi) = -\partial_x^2 \psi - \partial_y^2 \psi = -\Delta\psi \tag{1.2.7}$$

To obtain boundary conditions for the (ω, ψ) formulation we proceed as follows: from (1.2.2) $\boldsymbol{u} = 0$ along Γ which in terms of the stream function ψ reads

$$\frac{\partial \psi}{\partial \boldsymbol{n}} = 0 \quad \text{and} \quad \psi = \text{constant} \quad \text{along} \quad \Gamma. \tag{1.2.8}$$

Recall that ψ is determined up to an additive constant from the Poisson equation (1.2.7). Choosing ψ to be 0 at any point on Γ together with the second condition of (1.2.8) gives:

$$\frac{\partial \psi}{\partial \boldsymbol{n}}|_\Gamma = \psi|_\Gamma = 0.$$

Putting everything together, the (ω, ψ) formulation of the NSE is given by:

$$\text{(1.2.9)} \qquad \begin{cases} \omega_t + (\boldsymbol{u} \cdot \nabla)\omega = \frac{1}{Re}\Delta\omega \\ \Delta\psi = -\omega \\ u = \partial_y\psi, \quad v = -\partial_x\psi \\ \frac{\partial\psi}{\partial n}|_\Gamma = \psi|_\Gamma = 0 \end{cases}$$

1.3 Solution Strategy and Numerical Considerations

Though the schemes presented in chapters 2 and 3 are based on different formulations of the NSE, namely the $(\boldsymbol{u}, p)$ formulation (1.2.4) and the (ω, ψ) formulation (1.2.9), respectively, our solution strategy for each is generally the same. Comparing (1.2.4) and (1.2.9) we see that:

- Boundary conditions are given for only one of the coupled variables; the velocity in (1.2.4) and the stream function in (1.2.9).
- The time dependent convection-diffusion equation of parabolic type is coupled with an elliptic Poisson equation. There is no evolution equation for the solution of the Poisson equation.

The key step that allows us to develop very efficient numerical methods for both formulations is the conversion of a given boundary condition for one flow variable into a "local" numerical boundary condition for the remaining flow variable. This procedure is well known in the case of the (ω, ψ) formulation and leads to, e.g., Briley's formula, a vorticity boundary condition derived using a boundary condition for the stream function; see [EL2] and §3.2.4. We employ a similar strategy for the $(\boldsymbol{u}, p)$ formulation by utilizing one of the velocity boundary conditions in (1.2.4) to evaluate numerically the Neumann boundary condition (1.2.5) for the

pressure, resulting in the REBC scheme described in chapter 2. For both formulations the use of these "converted" boundary conditions decouples the computation of the convection-diffusion equation from that of the Poisson equation. We then use a high order explicit time stepping scheme to advance the solution of the convection-diffusion equation, and standard FFT-based fast solvers for the Poisson equation. In each case very efficient numerical methods for approximating solutions of the NSE result.

1.3.1 Spatial Discretization: Finite Differences

This thesis is concerned exclusively with the use of finite difference methods for the approximation of solutions to the unsteady NSE. Finite difference methods produce numerical approximations to the solution of an ordinary or partial differential equation, at a finite number of "grid" points within the domain, by replacing derivatives with difference quotients. This procedure generally results in a system, possibly nonlinear, of algebraic equations which one then solves to obtain an approximate discrete solution.

In this section we derive the finite difference operators that are used in chapters 2 and 3 for the spatial derivatives appearing in the NSE. No consideration is given here to the treatment of boundary conditions, which often requires the use of one-sided approximations. These are derived in subsequent chapters when needed. For simplicity we derive the difference operators in the case of one spatial dimension, using partial derivative notation; extension to 2D is straightforward.

Suppose that $\Omega = [a, b]$, choose N, and let $h = \Delta x = (b - a)/N$. Define a finite difference grid for Ω by:

$$\Omega_h = \{x_i : x_i = a + ih, \quad i = 0, 1, \ldots, N\}.$$

The points $\{x_i\}$ of Ω_h are referred to as "grid" points, which are chosen here

to be uniformly spaced. All of the methods presented in this thesis use uniform grids. Given a function $u \in C^4(\Omega)$, Taylor series expansion about any interior point $x_i \in \Omega_h$ reads

$$u_{i\pm1} = u_i \pm h(\partial_x u)(x_i) + \frac{h^2}{2}(\partial_x^2 u)(x_i) \pm \frac{h^3}{6}(\partial_x^3 u)(x_i) + O(h^4), \tag{1.3.1.1}$$

where $u_i = u(x_i)$ and $u_{i\pm1} = u(x_i \pm h)$. Define the finite difference operators $\widetilde{D}_x$ and D_x^2 by

$$(\widetilde{D}_x u)(x_i) = \frac{u_{i+1} - u_{i-1}}{2h}, \quad (D_x^2 u)(x_i) = \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2}. \tag{1.3.1.2}$$

Using the expansions in (1.3.1.1) we have

$$(\partial_x u)(x_i) = (\widetilde{D}_x u)(x_i) + O(h^2) \quad \text{and} \quad (\partial_x^2 u)(x_i) = (D_x^2 u)(x_i) + O(h^2).$$

$\widetilde{D}_x$ and D_x^2 are, respectively, the standard second order centered finite difference approximations to ∂_x and ∂_x^2.

The operators in (1.3.1.2) are the only ones required for the spatial discretizations of the second order numerical scheme developed in chapter 2. In chapter 3 the underlying numerical scheme is EC4 [EL2], which is spatially fourth order accurate, and employs higher order compact and long-stencil finite difference operators that are easily expressed using $\widetilde{D}_x$ and D_x^2. To simplify the derivations we drop the cumbersome reference to the values of u at specific grid points and work directly with the finite difference and differential operators. It is however implicit that there is always an underlying uniform grid Ω_h.

Using $\partial_x^2 = D_x^2 + O(h^2)$ and $\frac{h^2}{6}(D_x^2 + O(h^2)) = \frac{h^2}{6}D_x^2 + O(h^4)$, a more careful use of Taylor series gives

$$\begin{aligned} \widetilde{D}_x &= \partial_x + \tfrac{h^2}{6}\partial_x^3 + O(h^4) \\ &= \partial_x(1 + \tfrac{h^2}{6}\partial_x^2) + O(h^4) \\ &= \partial_x(1 + \tfrac{h^2}{6}(D_x^2 + O(h^2)) + O(h^4) \\ &= \partial_x(1 + \tfrac{h^2}{6}D_x^2) + O(h^4). \end{aligned} \tag{1.3.1.4}$$

Formally $(1+\frac{h^2}{6}D_x^2)^{-1} = (1-\frac{h^2}{6}D_x^2)+O(h^4)$, and using this in (1.3.1.4) gives

$$\partial_x = \widetilde{D}_x(1-\frac{h^2}{6}D_x^2)+O(h^4). \tag{1.3.1.5}$$

The operator $\widetilde{D}_x(1-\frac{h^2}{6}D_x^2)$ is a fourth order long-stencil approximation to ∂_x, so-called since when applied one has

$$(\widetilde{D}_x(1-\frac{h^2}{6}D_x^2)u)(x_i) = \frac{u_{i-2}-8u_{i-1}+8u_{i+1}-u_{i+2}}{12h},$$

which involves values of u extending two grid points to either side of x_i.

Formally dividing (1.3.1.4) by $(1+\frac{h^2}{6}D_x^2)$ gives

$$\partial_x = \frac{\widetilde{D}_x}{(1+\frac{h^2}{6}D_x^2)}+O(h^4), \tag{1.3.1.6}$$

a fourth order compact approximation of ∂_x. The main advantage of compact operators is that they provide high order approximations while maintaining a small stencil. It will be made clear in chapter 3 how one applies such an operator in practice, i.e., how one interprets division by a difference operator.

Proceeding as in (1.3.1.4), again using $\partial_x^2 = D_x^2+O(h^2)$, Taylor series gives

$$\begin{aligned} D_x^2 &= \partial_x^2+\tfrac{h^2}{12}\partial_x^4+O(h^4) \\ &= \partial_x^2(1+\tfrac{h^2}{12}\partial_x^2)+O(h^4) \\ &= \partial_x^2(1+\tfrac{h^2}{12}(D_x^2+O(h^2))+O(h^4) \\ &= \partial_x^2(1+\tfrac{h^2}{12}D_x^2)+O(h^4). \end{aligned} \tag{1.3.1.7}$$

Formally dividing (1.3.1.7) by $(1+\frac{h^2}{12}D_x^2)$ gives

$$\partial_x^2 = \frac{D_x^2}{(1+\frac{h^2}{12}D_x^2)}+O(h^4), \tag{1.3.1.8}$$

a fourth order compact approximation of ∂_x^2.

1.3.2 Temporal Discretization and Stability

Since our interest is the approximation of unsteady, time dependent solutions of the NSE, we must select a time discretization scheme for the evolution equations in (1.2.4) and (1.2.9). In choosing a time stepping scheme the main issues that one must take into account are accuracy, efficiency, and stability. In order to understand these issues in the context of the NSE, consider the following convection-diffusion equation

$$u_t = a\partial_x u + \nu\partial_x^2 u \tag{1.3.2.1}$$

on $\Omega=[0,2\pi]$, with periodic boundary conditions. One can think of (1.3.2.1) as a 1D model of the convective and diffusive dynamics, with representative velocity a and $\nu=1/Re$, governed by the evolution equations in (1.2.4) and (1.2.9).

Using (1.3.1.2) to discretize the spatial derivatives in (1.3.2.1) gives

$$\partial_t u_j = a\widetilde{D}_x u_j + \nu D_x^2 u_j = a\frac{u_{j+1}-u_{j-1}}{2h} + \nu\frac{u_{j-1}-2u_j+u_{j+1}}{h^2}, \tag{1.3.2.2}$$

where $u_j(t) = u(x_j,t)$, $x_j = jh$, for $j = 1,2,\ldots,N$, and $h = 2\pi/N$. Since the solution is periodic it can be expanded as a discrete Fourier series

$$u_j(t) = \sum_{k=1}^{N} \widehat{u}_k(t)e^{ikx_j},$$

which when substituted into (1.3.2.2) gives for each mode $k=1,2,\ldots,N$

$$\partial_t\widehat{u}_k(t) = (-4\frac{\nu}{h^2}\sin^2(kh/2) + i\frac{a}{h}\sin(kh))\widehat{u}_k(t).$$

Each of the above equations is an ODE of the form

$$\partial_t v = \lambda v, \tag{1.3.2.3}$$

where $\lambda=(-4\frac{\nu}{h^2}\sin^2(kh/2) + i\frac{a}{h}\sin(kh))$. The spatial discretizations have basically been lumped into the parameter λ, which allows us investigate the implications of discretizing ∂_t in (1.3.2.2) by examining how the discretization scheme behaves when applied to (1.3.2.3).

Now, suppose we are given a well-posed ODE of the form

$$\partial_t v = f(v), \tag{1.3.2.4}$$

with initial condition v_0 at $t = 0$. Most standard discretization schemes for (1.3.2.4) can essentially be classified as either explicit or implicit. The simplest explicit scheme is the forward Euler method, given by

$$v^{n+1} = v^n + \Delta t f(v^n), \tag{1.3.2.5}$$

where Δt is the time step, and v^n is the approximation to the true solution at time $n\Delta t$ with $v^0 = v_0$. We note that Euler's method is first order, i.e., has a truncation error that is $O(\Delta t)$.

Applying (1.3.2.5) to (1.3.2.3) gives

$$v^{n+1} = (1 + \lambda\Delta t)v^n, \tag{1.3.2.6}$$

where $(1 + \lambda\Delta t)$ is the amplification factor which implies that if (1.3.2.6) is to be of any practical use we must ensure that

$$|1 + \lambda\Delta t| \leq 1. \tag{1.3.2.7}$$

The set of all λ that satisfy (1.3.2.7) defines the stability region of the method; see [Ge]. The stability condition (1.3.2.7) is necessary to ensure that any errors in the approximation v^n, which are most certainly guaranteed to exist due to both truncation and roundoff errors, do not grow uncontrollably as we repeatedly apply the iteration (1.3.2.6).

Condition (1.3.2.7) holds for the values of λ given in (1.3.2.3) if we choose Δt such that

$$\frac{|a|\Delta t}{h} \leq \frac{2\nu\Delta t}{h^2} \leq 1. \tag{1.3.2.8}$$

The first inequality above gives

(1.3.2.9) $$Re_c \equiv \frac{|a|h}{\nu} \leq 2.$$

Re_c is known as the cell Reynolds number, and correspondingly, (1.3.2.9) the cell Reynolds number constraint. It imposes a severe restriction on the spatial grid size, especially in the case of high Reynolds number flows. The second inequality in (1.3.2.8) is the stability condition coming from the diffusion term, and must always be enforced for an explicit scheme.

Notice that for λ in (1.3.2.3), the convective velocity a only appears in the imaginary part, and the diffusion coefficient ν only in the real part. The cell Reynolds number constraint (1.3.2.9) arises due to the fact that for our values of λ the stability region of the forward Euler method does not contain any portion of the imaginary axis; see [Ge], [Te]. In fact, for $\nu = 0$ the method is unconditionally unstable when applied to (1.3.2.4).

An easy way to avoid the cell Reynolds number constraint is to use implicit methods. The simplest of these, when applied to (1.3.2.4), is given by

(1.3.2.10) $$v^{n+1} = v^n + \Delta t f(v^{n+1}).$$

The scheme (1.3.2.10) is first order, and it is known as the backward Euler method. When used to discretize (1.3.2.3), the method is stable as long as

$$|1 - \lambda \Delta t| \geq 1.$$

Hence, unconditional stability results when backward Euler is used for the time discretization in the convection-diffusion equation (1.3.2.2). While this approach is widely used when computing solutions of the NSE (see [EL1]), a **tremendous** computational price must be paid. Since in the NSE the convective speed depends on u, at each time step we must solve a large sparse **nonlinear** system of

equations. This implies that implicit methods should only be used when the cost of the time stepping is the main issue, e.g., steady state or low Reynolds number flows; see [EL1].

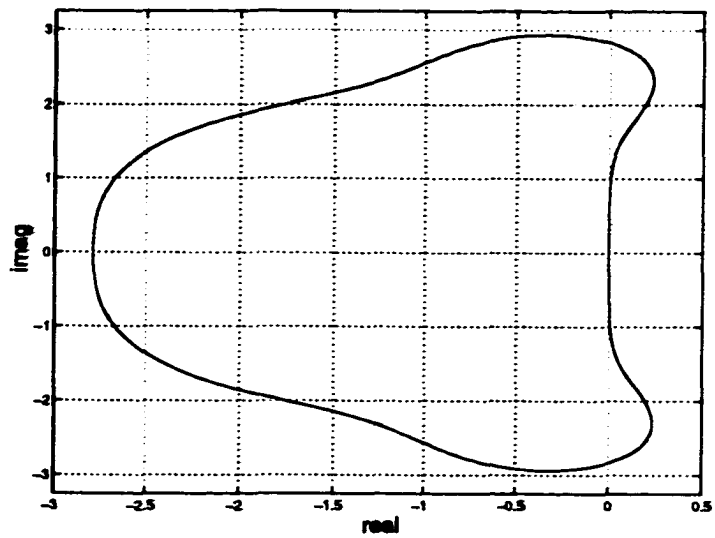

Figure 1.1: Stability Region of RK4

What is needed to overcome all of these difficulties is an explicit method whose stability region encompasses an appreciable portion of the imaginary axis. Such methods exist, the most well known of which is the classical fourth order Runge-Kutta scheme (RK4); see [Ge]. When applied to (1.3.2.4) the method is given by

$$\text{(1.3.2.11)} \qquad \begin{cases} v_1 = v^n + \frac{1}{2}\Delta t f(v^n), & k_1 = f(v_1) \\ v_2 = v^n + \frac{1}{2}\Delta t f(v_1), & k_2 = f(v_2) \\ v_3 = v^n + \Delta t f(v_2), & k_3 = f(v_3) \\ v^{n+1} = v^n + \frac{\Delta t}{6}(f(v^n) + 2k_1 + 2k_2 + k_3). \end{cases}$$

Notice that RK4 is essentially comprised of four forward Euler steps, which leads to easy implementation from a programming perspective. As for stability, when RK4 is applied to (1.3.2.3) Δt must be chosen such that

$$|1 + (\lambda\Delta t) + (\lambda\Delta t)^2/2 + (\lambda\Delta t)^3/6 + (\lambda\Delta t)^4/24| \leq 1.$$

Graphically we can easily determine the points in the complex plane in terms of $\lambda\Delta t$ for which the above inequality holds. This is shown in Figure 1.1, where the interior of the curve represents stable values.

For the full 2D NSE we must take into account that there is convection and diffusion in both coordinate directions, and also that the convection term is nonlinear. When used in conjunction with centered spatial differencing as given by (1.3.1.2), the RK4 scheme is stable when applied to the 2D NSE as long as the following two conditions hold (see [EL1]):

$$\frac{|a|\Delta t}{h} \leq \text{CFL} \leq 1.5 \quad \text{and} \quad 4\frac{\nu\Delta t}{h^2} \leq 1. \tag{1.3.2.12}$$

Here $h = \min\{\Delta x, \Delta y\}$ and a is the maximum velocity present in the flow at time t.

CHAPTER 2
REYNOLDS NUMBER BOUNDARY CONDITION

2.1 Introduction

In this chapter we present a spatially second order finite difference scheme for the numerical solution of the incompressible Navier-Stokes equations (NSE) based on the primitive variable formulation. The novelty of the scheme lies primarily in a simple, consistent, and accurate numerical implementation of the Neumann boundary condition (1.2.5) for the pressure Poisson equation (PPE) in (1.2.4). This numerical boundary condition, referred to hereinafter as the Reynolds number boundary condition (REBC), enables us to accurately solve the PPE on non-staggered grids using standard FFT-based fast Poisson solvers. When used in conjunction with high order explicit time stepping the resulting scheme is simple to implement and very efficient.

Evidence of the numerical performance of the method is given in the form of both 1D and 2D accuracy checks, as well as a convergence study of solutions of the canonical Cavity flow. In all cases the scheme achieves clean second order accuracy. In addition, the computations of the Cavity flow are shown to be in excellent agreement with solutions computed using a second order (ω, ψ)-based method. The key point here is that it has been proven analytically that the (ω, ψ) scheme is convergent; see [HW] and [WE].

2.2 Description of the Scheme

In §1.2 one possible primitive variable formulation of the 2D incompressible NSE is given by (1.2.4), which we reiterate here for easy reference,

$$\text{(2.2.1)} \qquad \begin{cases} \boldsymbol{u}_t + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u} + \nabla p = \frac{1}{Re}\Delta\boldsymbol{u} \\ \Delta p = -\nabla\cdot(\boldsymbol{u}\cdot\nabla\boldsymbol{u}) \end{cases}$$

on a bounded domain Ω, where $\boldsymbol{u} = (u, v)$ is the velocity and p is the pressure, along with the boundary conditions

$$\text{(2.2.2)} \qquad \boldsymbol{u} = 0 \quad \text{and} \quad \nabla\cdot\boldsymbol{u} = 0 \quad \text{on} \quad \partial\Omega.$$

We now describe a numerical scheme to approximate a solution of (2.2.1)–(2.2.2). First, the spatial discretization of our scheme is outlined, treating time as continuous.

Assume that Ω is rectangular, and that its lower left-hand corner coincides with the origin. Denote by Ω_h a finite difference grid for Ω, which, without loss of generality, has equal grid spacing h in each coordinate direction, i.e., $h = \Delta x = \Delta y$. Thus,

$$\Omega_h = \{(x_i, y_j) : x_i = ih, \quad i = 0, 1, \ldots, n_x, \quad y_j = jh, \quad j = 0, 1, \ldots, n_y\}$$

where $n_x + 1$ and $n_y + 1$ are, respectively, the number of grid points in the x and y directions. It is at the points of Ω_h that we seek a numerical approximation to the solution of (2.2.1)–(2.2.2).

Since the flow is incompressible, at any time the velocity field must satisfy the condition $\nabla\cdot\boldsymbol{u} = 0$. Using this, the righthand side of the PPE in (2.2.1) can be simplified as follows:

$$\text{(2.2.3)} \qquad \nabla\cdot(\boldsymbol{u}\cdot\nabla\boldsymbol{u}) = 2(\partial_y u\partial_x v - \partial_x u\partial_y v).$$

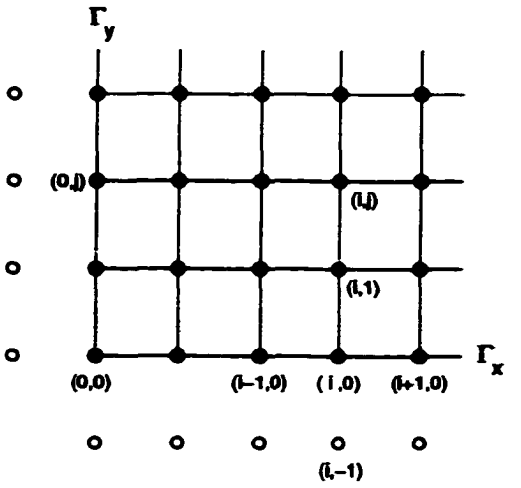

Figure 2.1: A Corner of Ω_h

We discretize (2.2.1) using

$$\text{(2.2.4)} \qquad \begin{cases} \boldsymbol{u}_t + (\boldsymbol{u} \cdot \nabla_h \boldsymbol{u}) + \nabla_h p = \frac{1}{Re} \Delta_h \boldsymbol{u} \\ \Delta_h p = -2(\widetilde{D}_y u \widetilde{D}_x v - \widetilde{D}_x u \widetilde{D}_y v). \end{cases}$$

Here $\Delta_h = (D_x^2 + D_y^2)$, the 5-point discrete Laplacian, and $\nabla_h = (\widetilde{D}_x, \widetilde{D}_y)^T$. The symbols $\widetilde{D}$ and D^2 denote the standard second order center difference approximations for the first and second partial derivative, respectively. See §1.3 for further details.

As it is, (2.2.4) is formally a second order accurate spatial discretization of (2.2.1). We still must implement numerically the boundary conditions given in (2.2.2). The condition $\boldsymbol{u}|_\Gamma = 0$ is explicitly used in (2.2.4) to set both $\boldsymbol{u} = 0$ at the boundary points of Ω_h for the momentum equation, and the righthand side of the PPE equal to zero at these same points. That leaves $\nabla \cdot \boldsymbol{u}|_\Gamma = 0$ which, as we show next, is "converted" to a boundary condition for the pressure p.

Note that we have yet to specify a boundary condition for the PPE in (2.2.4). It was shown in §1.2, by dotting the momentum equation with the unit normal

$\boldsymbol{n}$, oriented in the coordinate direction, along Γ and again using $\boldsymbol{u}|_\Gamma = 0$, that a natural candidate is given by the Neumann boundary condition (1.2.5), which we reiterate here for easy reference,

$$\frac{\partial p}{\partial \boldsymbol{n}} = \frac{1}{Re}(\Delta \boldsymbol{u}) \cdot \boldsymbol{n} \ . \tag{2.2.5}$$

However, enforcing (2.2.5) in (2.2.4) for the PPE requires evaluating the viscous term at the boundary which can be difficult in a discrete setting.

To explain the difficulty, assume the mesh is as depicted in Figure 2.1. The computational points are denoted by the "•" symbol. Numerical approximations to $\boldsymbol{u}$ and p are defined only at these points, which includes all the points along $\Gamma = \Gamma_x \cup \Gamma_y$. We focus our attention along Γ_x. Since $\boldsymbol{u} = 0$ on Γ, a second order centered approximation of $\Delta \boldsymbol{u} \cdot \boldsymbol{n}$ at the $(i, 0)$ grid point is given by

$$\begin{aligned} \Delta \boldsymbol{u} \cdot \boldsymbol{n} &= \Delta v \\ &= \Delta_h v + O(h^2) \\ &= (v_{i,1} + v_{i,-1})/h^2 + O(h^2). \end{aligned}$$

This requires a value for v at the $(i, -1)$ grid point, referred to as a "ghost" point, whose value is unknown since it lies outside of the computational domain. The "ghost" points are indicated in Figure 2.1 by the "o" symbol.

To obtain a consistent approximation for $v_{i,-1}$ we use the boundary condition $\nabla \cdot \boldsymbol{u}|_\Gamma = 0$. A second order centered approximation to $\nabla \cdot \boldsymbol{u}$ at the $(i, 0)$ grid point reads

$$\begin{aligned} 0 &= \nabla \cdot \boldsymbol{u} \\ &= \widetilde{D}_x u + \widetilde{D}_y v + O(h^2) \\ &= \widetilde{D}_y v + O(h^2) \\ &= (v_{i,1} - v_{i,-1})/2h + O(h^2), \end{aligned} \tag{2.2.6}$$

where again we have used $\boldsymbol{u}|_\Gamma = 0$, which implies that we should take $v_{i,-1} = v_{i,1}$. Thus, to enforce discretely the Neumann boundary condition for the PPE in

(2.2.4) at the $(i, 0)$ grid point we use the approximation

$$\frac{1}{Re}\Delta \boldsymbol{u} \cdot \boldsymbol{n} \approx \frac{1}{Re}\frac{2v_{i,1}}{h^2} \quad \text{on} \quad \Gamma_x. \tag{2.2.7}$$

The same argument is applied along Γ_y to define the "ghost" points there in order to approximate Δu, giving

$$\frac{1}{Re}\Delta \boldsymbol{u} \cdot \boldsymbol{n} \approx \frac{1}{Re}\frac{2u_{1,j}}{h^2} \quad \text{on} \quad \Gamma_y. \tag{2.2.8}$$

It is (2.2.7)–(2.2.8) that we refer to as the REBC. This completes the description of the spatial discretization used in the scheme.

For the time discretization, if we use an explicit discretization, a high order method must be used in order to avoid any cell Reynolds number constraint. In practice we use the classical fourth order Runge-Kutta method, which can essentially be written as four forward Euler steps. These issues were discussed in detail in §1.3.2. Therefore, we illustrate the time discretization of (2.2.4) using forward Euler. The complete method is then given by:

$$\frac{\boldsymbol{u}^{n+1} - \boldsymbol{u}^n}{\Delta t} + (\boldsymbol{u}^n \cdot \nabla_h \boldsymbol{u}^n) + \nabla_h p^n = \frac{1}{Re}\Delta_h \boldsymbol{u}^n \tag{2.2.9}$$

and

$$\Delta_h p^{n+1} = -2(\widetilde{D}_y u^{n+1}\widetilde{D}_x v^{n+1} - \widetilde{D}_x u^{n+1}\widetilde{D}_y v^{n+1}). \tag{2.2.10}$$

We see that the overall scheme is very simple. Given $\boldsymbol{u}^n$ and p^n, explicitly compute $\boldsymbol{u}^{n+1}$ at the **interior** points of Ω_h using (2.2.9), setting $\boldsymbol{u}^{n+1} = 0$ on Γ. Then compute the righthand side of (2.2.10), noting that this expression is zero at points along Γ. Next, compute an approximation to the Neumann boundary condition in (2.2.5) using the REBC (2.2.7)–(2.2.8). Lastly, recover p^{n+1} at all points of Ω_h by solving the discrete PPE. This can be done using standard fast FFT solvers. Thus, the overall method is very efficient.

2.3 Implementing Slip Boundary Conditions

The REBC scheme can easily be modified to handle the case in which a physical boundary slips, i.e., it moves with some prescribed velocity.

Refer once again to Figure 2.1. Suppose that the boundary Γ_x slips with a given velocity $u_b(x)$. The flow still must satisfy $\nabla \cdot \boldsymbol{u}|_\Gamma = 0$, and (2.2.6) now reads:

$$\begin{aligned} 0 &= \nabla \cdot \boldsymbol{u} \\ &= \partial_x u_b(x)\,\big|_{(x_i,0)} + \widetilde{D}_y v + O(h^2) \\ &= \partial_x u_b(x)\,\big|_{(x_i,0)} + (v_{i,1} - v_{i,-1})/2h + O(h^2). \end{aligned} \tag{2.3.1}$$

The value of v at the $(i,-1)$ "ghost" point is then taken to be

$$v_{i,-i} = v_{i,1} + 2h\partial_x u_b(x)\,\big|_{(x_i,0)}.$$

In this case the REBC at the $(i,0)$ grid point is given by:

$$\frac{1}{Re}\Delta \boldsymbol{u} \cdot \boldsymbol{n} \approx \frac{2}{Re}\Big(\frac{v_{i,1}}{h^2} + \frac{\partial_x u_b(x)}{h}\,\big|_{(x_i,0)}\Big) \quad \text{on} \quad \Gamma_x, \tag{2.3.2}$$

where $\boldsymbol{n}$ is the unit normal to Γ oriented in the coordinate direction.

Also, the righthand side of the PPE in (2.2.1) is no longer necessarily 0 along Γ_x. We still have $v_x = 0$, and $\nabla \cdot \boldsymbol{u}|_\Gamma = 0$ tells us that $v_y = -u_x$ at any point of Γ. Then using (2.2.3), at the $(i,0)$ grid point

$$\begin{aligned} \nabla \cdot (\boldsymbol{u} \cdot \nabla \boldsymbol{u})\,\big|_{(x_i,0)} &= 2(\partial_y u \partial_x v - \partial_x u \partial_y v)\,\big|_{(x_i,0)} \\ &= -2(\partial_x u_b(x))^2\,\big|_{(x_i,0)}. \end{aligned} \tag{2.3.3}$$

2.4 Accuracy Checks

We now present accuracy checks for the numerical scheme outlined in §2.2. In §2.4.1 the general framework of the REBC scheme is applied to a 1D model of the unsteady 2D Stokes equations. The usefulness of this model lies primarily in that

the Stokes equations differ from the NSE only in the absence of the nonlinear convection term. Therefore, as long as consistent discretizations are used, any error in the computed solution is clearly attributable to an improper implementation of the pressure boundary condition. Additionally, in this simplified setting exact solutions are available, and these are used to determine the order of convergence of the numerical scheme.

In §2.4.2 we implement the REBC scheme (2.2.8)–(2.2.9) for the full 2D NSE. In this situation exact solutions are not available. To overcome this difficulty we choose time dependent functions for u and p and then add appropriate forcing terms to the NSE equations to ensure that our choices for u and p are exact solutions. By doing so we are able to determine the order of convergence of the numerical scheme applied to the full 2D equations. For both the 1D and 2D results we measure the errors using standard discrete L^1, L^2, and L^∞ norms.

2.4.1 1D Accuracy Check: A Simplified Model

Consider the simple 1D model on $\Omega=[-1,1]$,

$$\text{(2.4.1.1)} \qquad \begin{cases} u_t + \partial_x p = \dfrac{1}{Re}(\partial_x^2 - k^2)u \\ (\partial_x^2 - k^2)p = 0 \\ u = \partial_x u = 0, \qquad \text{at } x = -1, 1. \end{cases}$$

Notice that the boundary conditions are a 1D analogue of those found in (2.2.2) for the full 2D NSE. Solutions of (2.4.1.1) are the k-th mode solution of the

unsteady Stokes equations in the domain $\Omega = [-1,1]\times[0,2\pi]$ given by

$$\text{(2.4.1.2)} \qquad \begin{cases} \boldsymbol{u}_t + \nabla p = \dfrac{1}{Re}\Delta \boldsymbol{u} \\ \nabla \cdot \boldsymbol{u} = 0 \\ \boldsymbol{u} = 0 \ \text{ on } \ \partial\Omega \end{cases}$$

where $\boldsymbol{u} = (u, v)$; see [OIM]. The system of equations (2.4.1.2) differs from the NSE only in that the nonlinear convection term is absent. However, the equations still embody the essential features of incompressibility and the presence of a viscous term that are found in the NSE.

For simplicity we take $k = 1$ in all that follows. The exact solution of (2.4.1.1) is then given by:

$$\text{(2.4.1.3)} \qquad \begin{cases} u(x,t) = e^{\sigma t}\Big(\cos(\mu x) - \dfrac{\cos(\mu)\cosh(x)}{cosh(1)}\Big) \\ p(x,t) = \sigma e^{\sigma t}\Big(\dfrac{\cos(\mu)\sinh(x)}{cosh(1)}\Big) \end{cases}$$

where μ satisfies $\mu\tan\mu + \tanh 1 = 0$ and $\sigma = -(\mu^2+1)/(Re)$. Again, see [OIM] for details.

Our finite difference discretization of (2.4.1.1) is simply an application of the REBC scheme to the 1D model. Choose n_x, let $h = 2/n_x$, and define $\Omega_h = \{x_i : x_i = -1 + ih \ \ i = 0, 1, \ldots, n_x\}$. Again, treating all terms explicitly, we step in time using

$$\text{(2.4.1.4)} \qquad \frac{u^{n+1} - u^n}{\Delta t} + \widetilde{D}_x p^n = \frac{1}{Re}(D_x^2 - 1)u^n$$

and

$$\text{(2.4.1.5)} \qquad (D_x^2 - 1)p^{n+1} = 0,$$

with $u_0 = u_{n_x} = 0$. The Neumann boundary condition for p, say at $x = -1$, is approximated using

$$p_x = \frac{1}{Re}(\partial_x^2 u) = \frac{1}{Re}\frac{u_1 + u_{-1}}{h^2} + O(h^2) \approx \frac{1}{Re}\frac{2u_1}{h^2},$$

where we have used $0 = u_x(-1) = (u_1 - u_{-1})/(2h) + O(h^2)$ to obtain a value for u_{-1}. The same derivation follows at $x = 1$. This is exactly the REBC for this 1D setting.

For the accuracy check we implement (2.4.1.4)–(2.4.1.5) with Re=100, and compute solutions for various values of n_x until time $t = 2.0$. Table 2.1 shows the absolute errors between the numerical solutions and the exact solutions given by (2.4.1.3). The method clearly achieves clean second order accuracy for both u and p.

Table 2.1: Absolute Errors for 1D model at $t = 2$

	n_x	L^1 error	order	L^2 error	order	L^∞ error	order
	32	8.80e-04		6.41e-04		5.12e-04	
	64	2.24e-04	1.98	1.63e-04	1.98	1.30e-04	1.98
u	128	5.63e-05	2.00	4.10e-05	2.00	3.27e-05	2.00
	256	1.41e-05	2.00	1.03e-05	2.00	8.19e-06	2.00
	512	3.52e-06	2.00	2.56e-06	2.00	2.05e-06	2.00
	32	2.79e-04		2.28e-04		2.84e-04	
	64	6.70e-05	2.04	5.53e-05	2.03	7.04e-05	2.01
p	128	1.64e-05	2.02	1.36e-05	2.01	1.76e-05	2.00
	256	4.08e-06	2.01	3.38e-06	2.01	4.39e-06	2.00
	512	1.01e-06	2.00	8.43e-07	2.00	1.01e-06	2.00

2.4.2 2D Accuracy Check

We now present an accuracy check of the REBC scheme applied to the full 2D NSE. In general, exact solutions are not available for the 2D NSE. However, let

$\Omega = [0, 2\pi] \times [0, 2\pi]$ and

$$(2.4.2.1) \qquad \begin{cases} u(x,y,t) = -\cos(t)\sin^2(x)\sin(y)\cos(y) \\ v(x,y,t) = \cos(t)\sin(x)\cos(x)\sin^2(y) \\ p(x,y,t) = \cos(t)\cos(x)\sin(y) \end{cases}$$

We ensure that (2.4.2.1) is an exact solution of (1.2.4) by adding appropriate forcing functions to the system. Note that we still have that $\boldsymbol{u} = (u, v)$ satisfies $\nabla \cdot \boldsymbol{u}|_\Gamma = \boldsymbol{u}|_\Gamma = 0$ at any time t.

We implemented the full REBC scheme (2.2.7)–(2.2.10) with $Re = 500$, and computed solutions for various values of n_x (taking $n_y = n_x$) until time $t = 6.0$, allowing the solution to pass through 0 a number of times. Fourth order Runge-Kutta time stepping was used in order to avoid any cell Reynolds number constraint. In all cases the time step is determined by conditions given by (1.3.2.12), with CFL=1.0.

Table 2.2 shows the absolute errors between the numerical solutions and the exact solutions given by (2.4.2.1), as well as the divergence of the computed velocity field. As the grid is refined the method achieves clean second order accuracy, as expected, for both $\boldsymbol{u}$ and p, as well as the divergence.

2.5 Convergence Study: Cavity Flow

Formally the REBC scheme is second order accurate in space, and in §2.4 we saw that the scheme achieved this accuracy when implemented numerically in both a 1D and 2D setting. However, the exact solutions used for the accuracy checks are "well-behaved" test functions, and in no sense represent an actual physical flow. In this section we evaluate the performance of the REBC scheme applied to the canonical Cavity Flow problem; see [PT]. The flow is computed on successively finer grids, and we take as the true solution the computation on the finest grid.

Table 2.2: Absolute Errors for 2D NSE at $t = 6$

	n_x	L^1 error	order	L^2 error	order	L^∞ error	order
	32	7.05e-01		1.65e-01		9.54e-02	
	64	1.94e-01	1.90	4.77e-02	1.86	3.05e-02	1.77
div $\boldsymbol{u}$	128	4.97e-02	1.98	1.24e-02	1.96	8.29e-03	1.92
	256	1.25e-02	2.00	3.14e-03	1.99	2.12e-03	1.98
	512	3.13e-03	2.00	7.88e-04	2.00	5.33e-04	1.99
	32	2.11e-01		4.91e-02		2.26e-02	
	64	5.66e-02	1.93	1.33e-02	1.92	6.26e-03	1.90
u	128	1.44e-02	1.98	3.40e-03	1.98	1.62e-03	1.97
	256	3.62e-03	2.00	8.56e-04	1.99	4.07e-04	1.99
	512	9.06e-04	2.00	2.14e-04	2.00	1.02e-04	2.00
	32	2.11e-01		4.89e-02		2.17e-02	
	64	5.68e-02	1.93	1.33e-02	1.92	6.03e-03	1.90
v	128	1.45e-02	1.98	3.40e-03	1.98	1.56e-03	1.96
	256	3.64e-03	2.00	8.54e-04	1.99	3.94e-04	1.99
	512	9.10e-04	2.00	2.14e-04	2.00	9.87e-05	2.00
	32	1.09e-01		2.18e-02		8.18e-03	
	64	2.68e-02	2.01	5.43e-03	2.00	1.96e-03	2.04
p	128	6.67e-03	2.00	1.36e-03	2.00	4.96e-04	1.99
	256	1.66e-03	2.00	3.40e-04	2.00	1.25e-04	2.00
	512	4.15e-04	2.00	8.51e-05	2.00	3.12e-05	2.00

This "converged" solution is then used to measure the convergence rate of the overall scheme.

The flow domain is $\Omega = [0,1] \times [0,1]$, with the no-slip boundary condition $\boldsymbol{u} = 0$ applied along $\Gamma = \partial\Omega$, except along $y = 1$, which is allowed to slip with given velocity $u_b(x) = 16x^2(1-x)^2$. It is this boundary condition that drives the flow. The initial conditions for the velocity field $\boldsymbol{u} = (u, v)$ are given by:

$$\text{(2.5.1)} \qquad \begin{cases} u(x,y,0) = 16x^2(1-x)^2(3y^2-2y) \\ v(x,y,0) = -32x(1-x)(2x-1)(y^2-y^3). \end{cases}$$

The initial pressure p is obtained by solving (2.2.10) with the REBC, taking into account the fact that the upper boundary slips. In this situation (2.3.2)–(2.3.3) are now relevant. We computed this flow using the REBC scheme until a final time of $t = 5.0$ for both $Re = 1,000$ and $Re = 5,000$. For all of the

computations $n_x = n_y$. Fourth order Runge-Kutta time stepping was used for the time discretization, with the time step determined by the stability conditions given in (1.3.2.12), with CFL=1.0.

Table 2.3: Divergence Errors for Cavity Flow $Re = 1,000$ at $t = 5$

n_x	L^1 error	order	L^2 error	order	L^∞ error	order
64	3.58e-02		5.18e-02		3.31e-01	
128	9.32e-03	1.95	1.33e-02	1.97	7.12e-02	2.16
256	2.38e-03	1.98	3.40e-03	1.98	1.84e-02	1.97
512	5.98e-04	1.99	8.56e-04	2.09	4.64e-03	1.99

Table 2.4: Divergence Errors for Cavity Flow $Re = 5,000$ at $t = 5$

n_x	L^1 error	order	L^2 error	order	L^∞ error	order
64	2.22e-01		3.25e-01		2.41e-00	
128	3.29e-02	2.60	6.08e-02	2.31	5.12e-01	2.05
256	8.58e-03	1.96	1.69e-02	1.90	1.67e-01	1.75
512	2.22e-03	1.96	4.46e-03	1.95	4.69e-02	1.89
1024	5.62e-04	1.99	1.13e-03	1.98	1.20e-02	1.97

Table 2.5: Relative Errors for Cavity Flow $Re = 1,000$ at $t = 5$

	n_x	L^1 error	order	L^2 error	order	L^∞ error	order
	64	6.06e-02		6.24e-02		4.01e-02	
u	128	1.53e-02	1.99	1.65e-02	1.94	1.17e-02	1.85
	256	3.17e-03	2.20	8.56e-04	2.18	2.50e-03	2.17
	64	6.54e-02		6.64e-02		7.44e-02	
v	128	1.64e-02	2.00	1.72e-02	1.96	2.05e-02	1.90
	256	3.40e-03	2.19	3.59e-03	2.19	4.37e-03	2.17
	64	9.95e-02		1.11e-01		1.35e-01	
p	128	2.99e-02	1.82	3.36e-02	1.82	4.18e-02	1.80
	256	6.39e-03	2.16	7.20e-03	2.16	9.04e-03	2.15

For $Re = 1,000$ solutions were computed using $n_x = 64, 128, 256,$ and 512. Table 2.5 lists the errors for velocity components and pressure relative to the "converged" solution computed using $n_x = 512$. In Table 2.3 we see that the divergence of the velocity field is clearly converging to zero at more or less the

Table 2.6: Relative Errors for Cavity Flow $Re = 5,000$ at $t = 5$

	n_x	L^1 error	order	L^2 error	order	L^∞ error	order
	64	3.40e-01		3.85e-01		3.60e-01	
	128	6.89e-02	2.22	7.32e-02	2.29	8.94e-02	2.01
u	256	1.90e-02	1.91	1.94e-02	1.94	2.10e-02	2.06
	512	3.82e-03	2.23	3.88e-03	2.24	3.99e-03	2.29
	64	3.87e-01		3.80e-01		5.55e-01	
	128	6.19e-02	2.50	6.22e-02	2.47	1.09e-01	2.26
v	256	1.76e-02	1.88	1.78e-02	1.87	2.87e-02	1.95
	512	3.64e-03	2.20	3.67e-03	2.20	5.77e-03	2.23
	64	4.92e-01		4.84e-01		4.29e-01	
	128	6.12e-02	2.83	6.53e-02	2.72	9.30e-02	2.15
p	256	2.01e-02	1.74	2.04e-02	1.79	2.82e-02	1.82
	512	4.31e-03	2.16	4.36e-03	2.16	5.82e-03	2.20

expected rate. In Figure 2.2 is shown the vorticity field of the "converged" solution computed to second order from the velocity field using $\omega = \widetilde{D}_x v - \widetilde{D}_y u$.

At $Re = 5,000$ solutions were computed using $n_x = 64, 128, 256, 512$, and 1024. Table 2.6 lists the errors for velocity components and pressure relative to the "converged" solution with $n_x = 1024$. In Table 2.4 we observe, as was the case for $Re = 1,000$, that the divergence of the velocity field is converging to zero at more or less the expected rate. In Figure 2.2 is shown the vorticity field of the "converged" solution computed, as above for $Re = 1,000$, from the velocity field.

2.6 Comparison with Second Order (ω, ψ) Cavity Flow Computations

In this section we compare computations of the canonical Cavity flow problem presented in §2.5 to solutions obtained using a spatially second order accurate discretization of the vorticity-stream function formulation (1.2.9) of the NSE. The objective is to determine whether or not the computations using the REBC

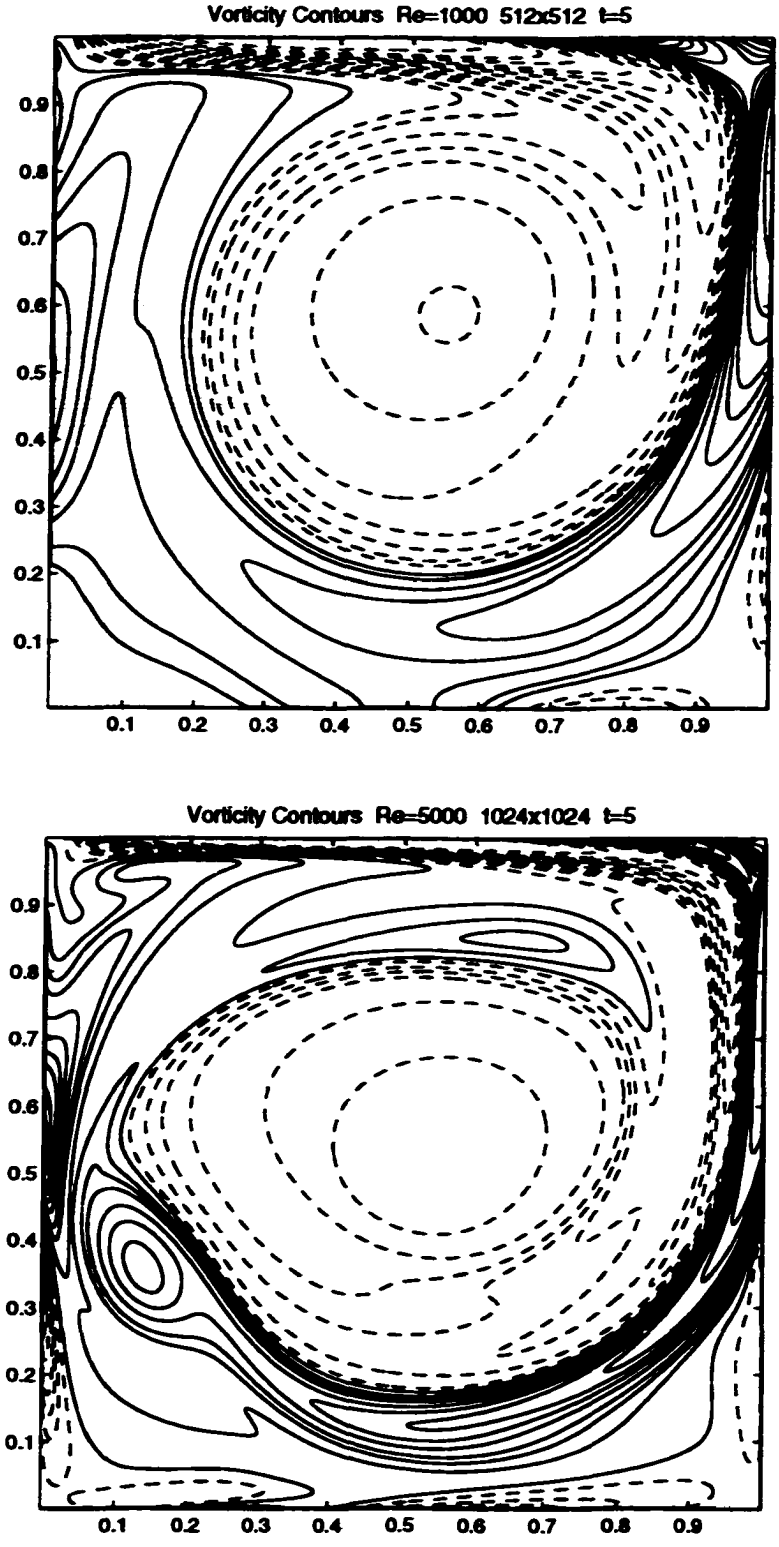

Figure 2.2: REBC Cavity Flow

scheme are converging to the true solution. This goal is realizable due to the fact that convergence of the above mentioned (ω, ψ) scheme as implemented here has been proven analytically; see [HW] and [WL].

Recall that the domain is $\Omega = [0,1] \times [0,1]$, with the no-slip boundary condition $\boldsymbol{u} = 0$ applied along $\Gamma = \partial\Omega$, except along $y = 1$, which slips with velocity $u_b(x) = 16x^2(1-x)^2$. The standard centered second order discretization of (1.2.9) is given by:

$$\text{(2.6.1)} \qquad \begin{cases} \omega_t + (\boldsymbol{u} \cdot \nabla_h)\omega = \frac{1}{Re}\Delta_h \omega, \\ \Delta_h \psi = -\omega, \quad \psi|_\Gamma = 0, \\ u = \widetilde{D}_y \psi, \quad v = -\widetilde{D}_x \psi. \end{cases}$$

As before, $\Delta_h = (D_x^2 + D_y^2)$ is the 5-point discrete Laplacian, and $\nabla_h = (\widetilde{D}_x, \widetilde{D}_y)^T$. Note that the homogeneous Dirichlet boundary condition has been enforced for the Poisson equation in (2.6.1). The Neumann boundary condition for ψ in (1.2.9) is "converted" into a local boundary condition, Thom's formula [TH], for ω; see [EL1]. This is given by, say along $y = 0$,

$$\text{(2.6.2)} \qquad \omega_{0,j} = -\frac{\psi_{1,j}}{(\Delta x)^2}.$$

This should be compared with the REBC (2.2.8). In fact, it was precisely the derivation of (2.6.2) which motivated the development of the REBC. A easily derived modification to Thom's formula is needed along $y = 1$, given by

$$\omega_{i,n_y} = -2\Big(\frac{\psi_{i,n_y-1}}{(\Delta y)^2} + \frac{u_b(x)}{\Delta y}\Big),$$

to account for the fact that this boundary slips with velocity $u_b(x)$. For the (ω, ψ) formulation the initial data is given directly in terms of the stream function ψ by

$$\psi(x, y, 0) = 16(y^3 - y^2)x^2(1-x)^2 = (y^3 - y^2)u_b(x).$$

We computed the flow for both $Re = 1,000$ and $Re = 5,000$ using (2.6.1)–(2.6.2), until time $t = 5.0$, using the same grid sizes as for the REBC scheme computations

in §2.5. As before, RK4 time stepping was used for the time discretization, with the time step determined by the stability conditions given in (1.3.2.12), with CFL=1.0.

Table 2.7: Relative Errors REBC vs. (ω, ψ) $Re = 1,000$ at $t = 5$

	n_x	L^1 error	order	L^2 error	order	L^∞ error	order
	64	6.60e-02		6.93e-02		4.45e-02	
	128	1.65e-02	2.00	1.77e-02	1.98	1.28e-02	1.87
u	256	4.23e-03	1.97	4.59e-03	1.97	3.41e-03	1.94
	512	1.07e-03	1.99	1.16e-03	1.99	8.66e-04	1.98
	64	7.11e-02		6.99e-02		7.39e-02	
	128	1.58e-02	2.12	1.66e-02	2.05	2.03e-02	1.91
v	256	3.91e-03	2.01	4.22e-03	1.99	5.39e-03	1.94
	512	9.79e-04	2.00	1.06e-03	2.00	1.37e-03	1.98

Table 2.8: Relative Errors REBC vs. (ω, ψ) $Re = 5,000$ at $t = 5$

	n_x	L^1 error	order	L^2 error	order	L^∞ error	order
	64	4.23e-01		4.64e-01		4.00e-01	
	128	5.38e-02	2.81	6.09e-02	2.76	7.41e-02	2.32
u	256	1.36e-02	1.99	1.56e-01	1.97	1.78e-02	2.04
	512	3.61e-03	1.94	4.16e-03	1.94	4.42e-03	2.01
	1024	9.21e-04	1.98	1.06e-03	1.98	1.10e-03	2.00
	64	5.26e-01		5.12e-01		7.47e-01	
	128	5.45e-02	3.11	5.62e-02	3.02	9.86e-02	2.75
v	256	1.41e-02	1.97	1.44e-02	1.98	2.60e-02	1.95
	512	3.69e-03	1.96	3.81e-03	1.95	7.19e-03	1.90
	1024	9.34e-04	1.99	9.68e-04	1.98	1.85e-03	1.97

In Table 2.7 are shown the relative errors, with the (ω, ψ) computations taken as the true solutions, for the components of the velocity field at Re=1,000. As the grids are refined, not only are the computations of the two methods converging to the same solution, but there is clearly clean second order convergence. This is even true in the L^∞ norm, which is quite unexpected. In Table 2.8 the relative errors are shown, again with the (ω, ψ) computations taken as the true solutions, for the components of the velocity field at Re=5,000. More or less the

same convergence behavior is seen as in the $Re = 1,000$ case. Note that rather than comparing the REBC solutions with a single (ω, ψ) computation using the finest grid size, we have compared solutions using the same grid sizes. This was done in order to gauge whether or not the second order divergence errors present in the REBC scheme (recall that the (ω, ψ) computations are divergence-free to machine precision!) significantly affect the overall flow dynamics. This does not appear to be the case for the $Re = 1,000$ simulations. However, there does seem to be an indication of such affects in the computations at $Re = 5,000$ as evidenced by the slightly greater relative errors between solutions on the coarsest grids.

Finally, in Figure 2.3 the vorticity contours are shown at the final time $t = 5.0$ for both computations on the finest grids. They are clearly in excellent agreement with the results of the REBC simulations shown in Figure 2.2.

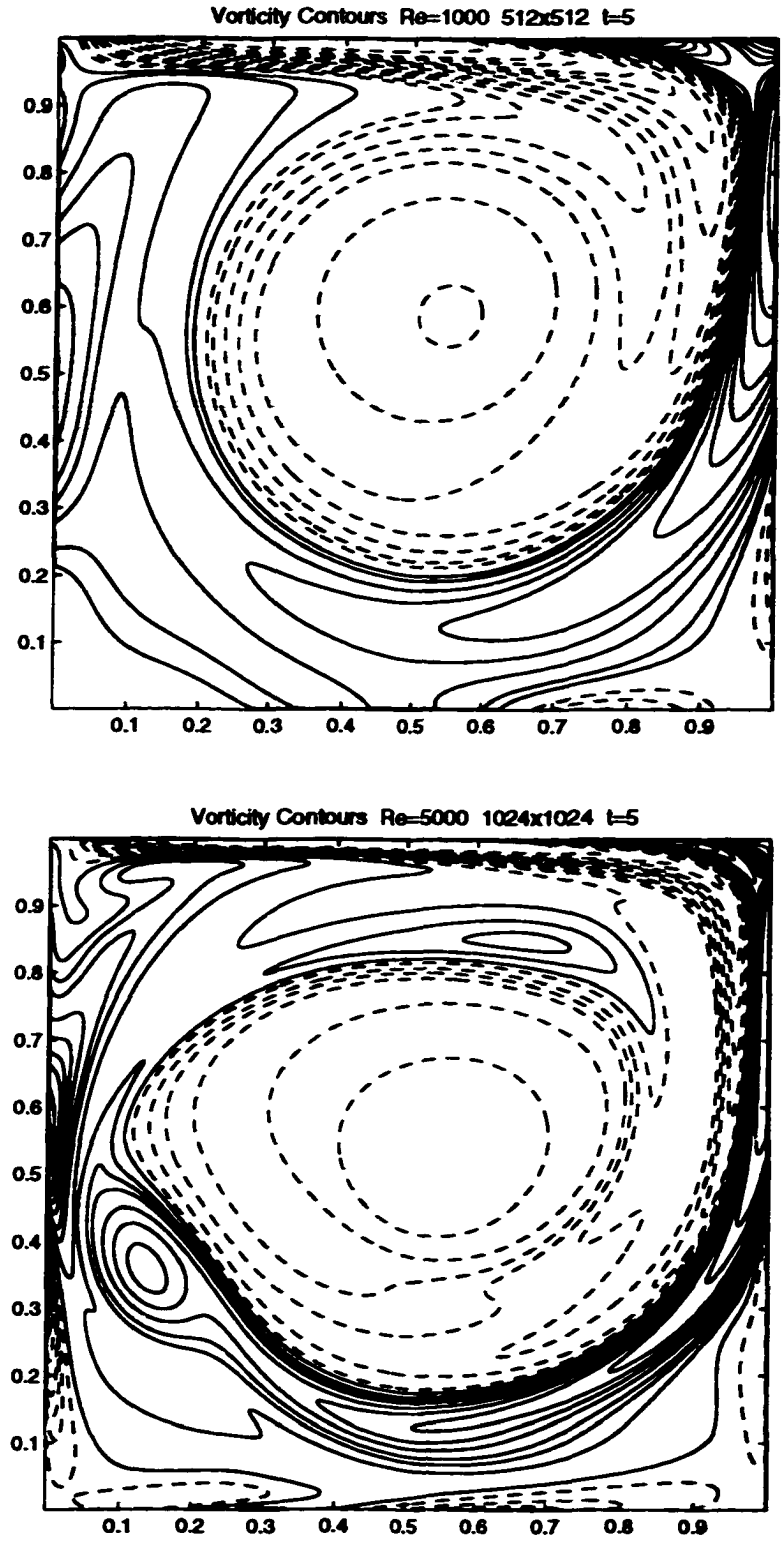

Figure 2.3: Second Order (ω, ψ) Cavity Flow

CHAPTER 3
FINE GRID PATCH: FLOW AROUND A CYLINDER

3.1 Introduction

In this chapter new developments are presented for a class of fourth order Essentially Compact methods (EC4), originally developed by E and Liu [EL2], for solving unsteady viscous incompressible flows in the vorticity-stream function formulation. A novel fine grid patch mesh refinement technique, which is easily incorporated into the original EC4 scheme, is developed. Its use results in a dramatic increase in computational efficiency, particularly for the case of flows at high Reynolds numbers. It is important to note here that implementation of the fine grid patch does not preclude us from using standard fast Poisson solvers for solving the Poisson-like equations which arise in the scheme. In addition, we present results of the use of a very effective far-field boundary condition for the stream function, derived from a high order expansion in terms of moments of the vorticity [Ti].

As a detailed illustration of an application of the above mentioned methodologies, we present high resolution benchmark quality computations of the flow past an impulsively started circular cylinder for Reynolds numbers ranging from 1,000 to 100,000. Even at the considerably high Reynolds number of 100,000 the flow is completely resolved.

3.2 Problem Description

Our interest is the development of an efficient finite difference scheme for computing the viscous incompressible flow around an impulsively started two dimensional cylinder. The physical description of the problem is as follows: A circular cylinder of diameter D is placed into a viscous fluid. The cylinder is then instantaneously accelerated (impulsive start), in a direction normal to its axis, to a constant velocity. Assume that the cylinder is so long as compared to its diameter that the ends have no effect upon the resulting flow. This reduces the problem to describing the flow in any plane normal to the cylinder.

The impulsively started cylinder problem described above has been extensively studied as a prototype of unsteady separated flows; see the review in [KL]. Even though the geometry of the problem is what might be considered simple, very complex fluid motions result, especially as the Reynolds number is increased (which one can think of loosely as corresponding to a reduction in the viscosity of the fluid). Most notably the flow begins to exhibit unsteady boundary layer separation, which in turn triggers subsequent vortex shedding from the surface of the cylinder. It is primarily these flow structures that determine the time-dependent forces, in particular the drag, "felt" by the cylinder. Hence, computational methods provide an invaluable tool for studying such flows.

3.2.1 Equations of Motion

Our starting point is the non-dimensionalized vorticity-stream function formulation of the NSE, which when expressed in polar coordinates reads:

$$\frac{\partial \omega}{\partial t} + \left(u, \frac{v}{r}\right) \cdot \left(\frac{\partial \omega}{\partial r}, \frac{\partial \omega}{\partial \theta}\right) = \frac{1}{Re}\left(\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial \omega}{\partial r}\right) + \frac{1}{r^2}\frac{\partial^2 \omega}{\partial \theta^2}\right), \tag{3.2.1.1}$$

$$\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial \psi}{\partial r}\right) + \frac{1}{r^2}\frac{\partial^2 \psi}{\partial \theta^2} = -\omega, \tag{3.2.1.2}$$

$$u = \frac{1}{r}\frac{\partial \psi}{\partial \theta}, \quad v = -\frac{\partial \psi}{\partial r}, \tag{3.2.1.3}$$

on the infinite domain $\{(r,\theta) : 0 < r_{min} \leq r \ , \ 0 \leq \theta \leq 2\pi\}$. The quantity ω is the vorticity, ψ the stream function, and u and v are the radial and tangential components of the velocity, respectively. The non-dimensionalization is based on the cylinder diameter $2r_{min}$, the free-stream velocity at infinity u_∞, and the kinematic viscosity ν, giving $Re = (2r_{min}u_\infty)/\nu$.

Applying the change of variable $z = \ln r$ to (3.2.1.1)–(3.2.1.3), and writing the convection term in conservative form gives

$$e^{2z}\frac{\partial \omega}{\partial t} + \frac{\partial}{\partial z}(U\omega) + \frac{\partial}{\partial \theta}(V\omega) = \frac{1}{Re}\left(\frac{\partial^2 \omega}{\partial z^2} + \frac{\partial^2 \omega}{\partial \theta^2}\right), \tag{3.2.1.4}$$

$$\frac{\partial^2 \psi}{\partial z^2} + \frac{\partial^2 \psi}{\partial \theta^2} = -e^{2z}\omega, \tag{3.2.1.5}$$

$$U = \frac{\partial \psi}{\partial \theta}, \quad V = -\frac{\partial \psi}{\partial z}, \tag{3.2.1.6}$$

where $(U,V) = r(u,v)$. We compute the flow around the cylinder using this form of the equations.

3.2.2 Computational Grid

All of the computations presented are for flows around a cylinder of radius 1, hence $r_{min} = 1$. The computational domain Ω extends to a distance of r_{max} which will vary due to considerations such as the magnitude of the Re, as well as the final time of the simulations, i.e., short .vs. long time simulations. The finite difference grid for Ω is equi-spaced with respect to both the z and θ directions. The use of the change of variable $z = \ln r$ therefore not only simplifies the flow equations, but naturally results in a higher density of grid points near the surface

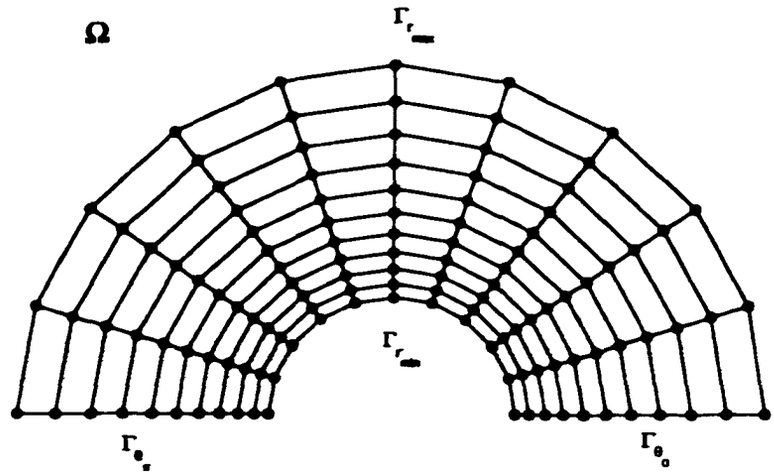

Figure 3.1: A Representative Grid for Ω

of the cylinder, exactly where they are needed to resolve the viscous boundary layer. The flow is only computed around the upper half of the cylinder, with symmetry boundary conditions applied along the horizontal line of symmetry. The computational grid Ω consists of the points defined by

$$\Omega = \{(z_i, \theta_j) : z_i = i\triangle z, \quad i = 0, 1, \ldots, n_z, \quad \theta_j = j\triangle\theta, \quad j = 0, 1, \ldots, n_\theta\}$$

where $\triangle z = (\ln r_{max})/n_z$, $\triangle\theta = \pi/n_\theta$, and n_z and n_θ are the number of grid points in the z and θ directions, respectively. Boundary segments of Ω are denoted as follows: $\Gamma_{r_{min}}$ corresponds to the cylinder surface, $\Gamma_{r_{max}}$ the outer radial boundary, and Γ_{θ_0} and Γ_{θ_π} are the segments along the symmetry line. A representative grid Ω is shown in Figure 3.1.

3.2.3 Spatial Discretization

The spatial discretization of (3.2.1.4)-(3.2.1.6) follows the general methodology of EC4 given in [EL2]. Treating time as continuous, we formally discretize (3.2.1.4)–(3.2.1.5) in space to fourth order using the compact finite difference operators

(1.3.1.6) and (1.3.1.8), giving

$$(3.2.3.1)\qquad e^{2z}\omega_t + \left(\frac{\widetilde{D}_z}{1+\frac{(\Delta z)^2}{6}D_z^2}\right)(U\omega) + \left(\frac{\widetilde{D}_\theta}{1+\frac{(\Delta\theta)^2}{6}D_\theta^2}\right)(V\omega)$$
$$= \frac{1}{Re}\left(\frac{D_z^2}{1+\frac{(\Delta z)^2}{12}D_z^2} + \frac{D_\theta^2}{1+\frac{(\Delta\theta)^2}{12}D_\theta^2}\right)\omega,$$

$$(3.2.3.2)\qquad \left(\frac{D_z^2}{1+\frac{(\Delta z)^2}{12}D_z^2} + \frac{D_\theta^2}{1+\frac{(\Delta\theta)^2}{12}D_\theta^2}\right)\psi = -e^{2z}\omega.$$

Recall that $\widetilde{D}$ and D^2 denote the standard second order center difference approximations for the first and second derivative, respectively; see §1.3.1. The discretizations (3.2.3.1)–(3.2.3.2) must now be manipulated, while preserving fourth order accuracy, to produce a system that can be implemented computationally.

Define the auxiliary variable $\overline{\omega}$ by

$$(3.2.3.3)\qquad \overline{\omega} = \left(1+\frac{1}{12}\left((\Delta z)^2D_z^2 + (\Delta\theta)^2D_\theta^2\right)\right)\left(e^{2z}\omega\right).$$

Multiplying (3.2.3.1) and (3.2.3.2) by $(1+\frac{1}{12}((\Delta z)^2D_z^2+(\Delta\theta)^2D_\theta^2))$, and ignoring high order terms, we have

$$(3.2.3.4)\qquad \begin{aligned}&\overline{\omega}_t + \widetilde{D}_z\left(1+\tfrac{(\Delta\theta)^2}{6}D_\theta^2\right)(U\omega) + \widetilde{D}_\theta\left(1+\tfrac{(\Delta z)^2}{6}D_z^2\right)(V\omega)\\ &-\tfrac{1}{12}\left((\Delta z)^2D_z^2+(\Delta\theta)^2D_\theta^2\right)(U\widetilde{D}_z\omega + V\widetilde{D}_\theta\omega)\\ &= \nu\left(\Delta_{(z,\theta)} + \tfrac{((\Delta z)^2+(\Delta\theta)^2)}{12}D_z^2D_\theta^2\right)\omega,\end{aligned}$$

$$(3.2.3.5)\qquad \left(\Delta_{(z,\theta)} + \frac{1}{12}((\Delta z)^2+(\Delta\theta)^2)D_z^2D_\theta^2\right)\psi = -\overline{\omega},$$

where $\Delta_{(z,\theta)} = (D_z^2+D_\theta^2)$ denotes the standard 5-point Laplacian. Above we have used the fact that

$$\widetilde{D}_z\left(1+\frac{(\Delta z)^2}{6}D_z^2\right)^{-1} = \widetilde{D}_z\left(1-\frac{(\Delta z)^2}{6}D_z^2\right) + O\left((\Delta z)^4\right),$$

and the analogous formula for the θ derivative. Also note that, due to incompressibility, to second order

$$\widetilde{D}_z(U\omega) + \widetilde{D}_\theta(V\omega) = U\widetilde{D}_z\omega + V\widetilde{D}_\theta\omega.$$

This is required to give the final form of the second line of (3.2.3.4), which avoids the use of a long stencil approximation near the boundary. The velocity components in (3.2.1.6) are recovered using the fourth order long stencil approximation (1.3.1.5):

$$U = \widetilde{D}_\theta\Big(1 - \frac{(\Delta\theta)^2}{6}D_\theta^2\Big)\psi, \quad V = -\widetilde{D}_z\Big(1 - \frac{(\Delta z)^2}{6}D_z^2\Big)\psi \tag{3.2.3.6}$$

3.2.4 Numerical Boundary Conditions

To implement (3.2.2.3)–(3.2.2.6), numerical boundary conditions for ω, ψ, U, and V must be prescribed which are consistent with the physical problem. Along Γ_{θ_0} and Γ_{θ_π} straightforward symmetry conditions are implemented. Along $\Gamma_{r_{min}}$ and $\Gamma_{r_{max}}$ the issue of proper boundary conditions is significantly more involved. We start with the vorticity ω.

The subject of vorticity boundary conditions in the context of finite difference schemes for the NSE in the (ω, ψ) formulation has a long history, beginning with Thom [Th]. A detailed discussion of this issue in the context of modern numerical methods can be found in [EL1]. Following the prescription found there, we derive a numerical boundary condition for ω as follows: as was discussed in §1.2, it is natural to impose the boundary conditions

$$\psi\Big|_{\Gamma_{r_{min}}} = 0, \qquad \frac{\partial\psi}{\partial n}\Big|_{\Gamma_{r_{min}}} = \frac{\partial\psi}{\partial z}\Big|_{\Gamma_{r_{min}}} = 0, \tag{3.2.4.1}$$

on the stream function. To evaluate the vorticity along $\Gamma_{r_{min}}$, we use a centered fourth order approximation of $\omega = -\Delta\psi = -\psi_{zz}$ (noting from (3.2.4.1) that

$\psi_{\theta\theta} = 0$ along the surface of the cylinder):

$$\omega_{0,j} = \frac{-1}{12(\Delta z)^2}\Big(16(\psi_{-1,j} + \psi_{1,j}) - (\psi_{-2,j} + \psi_{2,j})\Big). \tag{3.2.4.2}$$

Using a fourth order one-sided approximation and a fourth order centered approximation to $\partial\psi/\partial z$ along $\Gamma_{r_{min}}$ we derive values for the "ghost" points $\psi_{-1,j}$ and $\psi_{-2,j}$, given by

$$\begin{aligned} \psi_{-1,j} &= 6\psi_{1,j} - 2\psi_{2,j} + \tfrac{1}{3}\psi_{3,j} - 4\Delta z \left(\tfrac{\partial\psi}{\partial z}\right)_{0,j}, \\ \psi_{-2,j} &= 40\psi_{1,j} - 15\psi_{2,j} + \tfrac{8}{3}\psi_{3,j} - 20\Delta z \left(\tfrac{\partial\psi}{\partial z}\right)_{0,j}, \end{aligned} \tag{3.2.4.3}$$

where we have used, according to (3.2.4.1), $\psi_{0,j}=0$. Plugging these into (3.2.4.2), and again using (3.2.4.1), we obtain

$$\omega_{0,j} = -\frac{108\psi_{1,j} - 27\psi_{2,j} + 4\psi_{3,j}}{18(\Delta z)^2}, \tag{3.2.4.4}$$

which is known as Briley's formula; see [EL1].

To prescribe values for ω along $\Gamma_{r_{max}}$ a simple outflow boundary condition is used:

$$\omega_{n_z,j} = \begin{cases} 3\omega_{n_z-1,j} - 3\omega_{n_z-2,j} + \omega_{n_z-3,j} & j = 0, \ldots, n_z/2 \\ 0 & j = (n_z/2)+1, \ldots, n_z. \end{cases} \tag{3.2.4.5}$$

In order to solve the Dirichlet problem (3.2.3.5) we must prescribe boundary values for ψ along Γ. The no-slip condition in (3.2.4.1) gives $\psi=0$ along $\Gamma_{r_{min}}$. Imposing a symmetry condition along Γ_{θ_0} and Γ_{θ_π} also gives $\psi=0$. The difficulty arises along the far-field boundary $\Gamma_{r_{max}}$. The true physical flow domain is infinite and one must impose a boundary condition that accurately "mimics" this fact in a computational setting. We have found that the choice of far-field condition for ψ dramatically affects the character and measured physical parameters of the computed flow.

We investigated the use of 2 far-field boundary conditions for ψ:

$$\psi(r,\theta) = u_\infty(r - r^{-1})\sin\theta \qquad \text{(PBC)}$$

(3.2.4.6)
$$\psi(r,\theta) = u_\infty r\sin\theta + \tfrac{1}{\pi}\textstyle\sum_{n=1}^{5}\frac{1}{n}G_n r^{-n}\sin(n\theta), \qquad \text{(ISBC)}$$
$$\text{where } G_n = \textstyle\int_\Omega \omega(\tilde{r},\tilde{\theta})\tilde{r}^{n+1}\sin(n\tilde{\theta})\,d\tilde{r}d\tilde{\theta}.$$

The first boundary condition results in the case where the flow is assumed to be irrotational, and is referred to as the potential boundary condition (PBC). Of the two conditions above we found the ISBC, the integral series boundary condition of Sa and Chang [SC], originally developed by Ting (see [Ti], [TK]), to be far superior. While the PBC requires no additional work other than initialization, the computational results vary greatly, in particular the calculated physical parameters, depending on the choice of r_{max}. For instance, for the impulsive start, only the ISBC produced consistent coefficient of drag computations as r_{max} was varied (see §3.4.3).

For completeness we include here a derivation of the ISBC for our particular flow setup. Consider the Poisson equation for the stream function

(3.2.4.7)
$$\Delta\psi = -\omega$$

in $\Omega = \{(r,\theta) : 0 < r_{min} \le r < \infty,\ 0 \le \theta \le 2\pi\}$. Note that Ω now represents the full exterior of the disc. Denoting $\mathbf{r} = (r,\theta)$ and $\mathbf{r}_0 = (r_0,\theta_0)$, the solution of (3.2.4.7) with free-stream velocity (u_∞, v_∞) at infinity is given by

$$\psi(\mathbf{r}) = u_\infty r\sin\theta - v_\infty r\cos\theta - \frac{1}{2\pi}\iint_\Omega \omega(\mathbf{r}_0)\ln(\|\mathbf{r}_0 - \mathbf{r}\|)dS_0\ ,$$

where $\|\mathbf{r}_0 - \mathbf{r}\|$ is the standard Euclidean distance between $\mathbf{r}_0$ and $\mathbf{r}$. Consider the function

$$\phi(r,\theta) = \frac{1}{2\pi}\iint_\Omega \omega(\mathbf{r}_0)\ln(\|\mathbf{r}_0 - \mathbf{r}\|)dS_0.$$

Identifying points in the plane with complex variables, i.e., $z = r\exp(i\theta)$, we have

that

$$
\begin{aligned}
\phi(z) &= \frac{1}{2\pi}\Re\Big(\iint_\Omega \ln(z-z_0)\omega(z_0)dS_0\Big) \\
&= \frac{1}{2\pi}\Re\Big(\iint_\Omega \ln\Big(z\Big(1-\frac{z_0}{z}\Big)\Big)\omega(z_0)dS_0\Big) \\
&= \frac{1}{2\pi}\Re\Big(\iint_\Omega \ln(z)\omega(z_0)dS_0 + \iint_\Omega \ln\Big(1-\frac{z_0}{z}\Big)\omega(z_0)dS_0\Big),
\end{aligned}
\tag{3.2.4.8}
$$

where $\Re$ denotes the real part of the expression. For the second integral in the last equation of (3.2.4.8), since $\|z_0\| < \|z\|$, we can use the expansion

$$
\ln\Big(1-\frac{z_0}{z}\Big) = -\sum_{k=1}^{\infty}\frac{1}{k}\Big(\frac{z_0}{z}\Big)^k.
$$

Substituting this into (3.2.4.8), interchanging the order of integration and summation, and using

$$
\Re\Big(\frac{z_0}{z}\Big) = r^{-k}\Big(r_0^k\cos(k\theta_0)\cos(k\theta) + r_0^k\sin(k\theta_0)\sin(k\theta)\Big),
$$

we have, re-expressing ϕ in terms of (r,θ), that

$$
\begin{aligned}
\phi(r,\theta) &= \frac{1}{2\pi}\iint_\Omega \ln(r)\omega(r_0,\theta_0)dS_0 \\
&\quad -\sum_{k=1}^{\infty}\Big(\frac{1}{2\pi k}\iint_\Omega r_0^k\cos(k\theta_0)\omega(r_0,\theta_0)dS_0\Big)r^{-k}\cos(k\theta) \\
&\quad -\sum_{k=1}^{\infty}\Big(\frac{1}{2\pi k}\iint_\Omega r_0^k\sin(k\theta_0)\omega(r_0,\theta_0)dS_0\Big)r^{-k}\sin(k\theta).
\end{aligned}
\tag{3.2.4.9}
$$

Since we are assuming that the flow around the cylinder is symmetric, the first two terms of (3.2.4.9) are 0. Using 5 terms from the last sum for ϕ along with $(u_\infty, v_\infty) = (1,0)$ gives the ISBC (3.2.4.6), a $O(r^{-6})$ approximation to ψ along $\Gamma_{r_{max}}$.

For the velocity components U and V, (3.2.4.1) implies

$$U\Big|_{\Gamma_{r_{min}}} = V\Big|_{\Gamma_{r_{min}}} = 0. \tag{3.2.4.10}$$

Computing U along $\Gamma_{r_{max}}$ using (3.2.3.6) presents no problem due to the assumed symmetry of the flow. However, using (3.2.3.6) to recover $V_{i,j}$ for $i = n_{z-1}, n_z$ would require values of ψ beyond the far-field boundary. To avoid this we use second order centered and one-sided approximations for $-\psi_z$, respectively, to compute V. Doing so does not result in any loss of accuracy of the overall method since the velocity is for the most part constant at the far-field, particularly in the case of short time simulations. The exceptional case is applying (3.2.3.6) to compute $V_{1,j}$, which requires values for $\psi_{-1,j}$, for which we use (3.2.4.3), giving

$$V_{1,j} = -\frac{9\psi_{1,j} + 9\psi_{2,j} - \psi_{3,j}}{18\triangle\theta}. \tag{3.2.4.11}$$

3.3 Fine Grid Patch and Time Stepping

It is well known that resolving boundary layers in high Reynolds number flows is of the utmost importance since it is primarily the dynamics of the boundary layer that determines the overall characteristics of the flow. This is especially true for the flow around a cylinder since the flow dynamics are determined by the rollup and eventual separation of the boundary layer along the rear of the cylinder surface.

The simplicity of EC4 allows us to easily incorporate a fine grid computational patch (FGP), denoted by Ω_p, into the overall scheme. For the flow around a cylinder, we use a fine grid patch along the surface of the cylinder, extending to an outer radius of $r_{maxp} < r_{max}$. Simply put, at each time step ω and ψ, as well as being computed on the coarse grid Ω, are computed on the fine grid Ω_p. Boundary values on the outer boundary of the patch, denoted by $\Gamma_{r_{maxp}}$, are obtained

by simple interpolation of the corresponding, just computed, values on the coarse grid Ω. The flow variables ω and ψ on Ω are then overwritten with a smooth combination of the values from both grids at common points. The benefits are clear. Not only do we get increased resolution exactly where we need it, but we can then select a coarser grid for Ω. This results in a significant computational savings when compared to using a computational grid Ω with fine resolution over the entire domain.

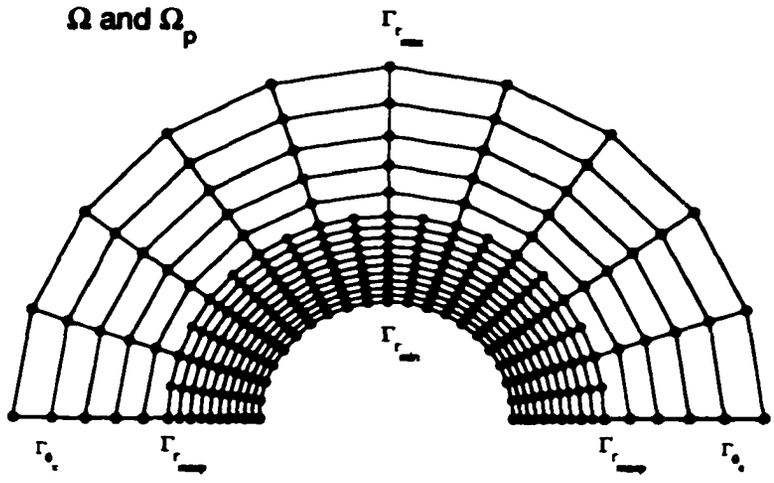

Figure 3.2: Coarse Grid Ω and Fine Grid Patch Ω_p

The geometry of Ω_p is defined by r_{maxp}, $\Delta\tilde{z}$, and $\Delta\tilde{\theta}$, which are, respectively, the outer radius of Ω_p, and the mesh size in the z and θ directions. We determine r_{maxp} by $r_{maxp} = e^{(n_z \Delta z)/factor}$, where $factor$ is problem dependent. The choice of $factor$ is mainly influenced by the the Reynolds number and one's wish to insure that the grid sizes on which the FFTs are performed are products of powers of small primes. Therefore, Ω_p is given by

$$\Omega_p = \{(z_i, \theta_j) : z_i = i\Delta\tilde{z}, \quad i = 0, 1, \ldots, \tilde{n}_z, \quad \theta_j = j\Delta\tilde{\theta}, \quad j = 0, 1, \ldots, \tilde{n}_\theta\},$$

where $\Delta\tilde{z} = \Delta z/2$, $\Delta\tilde{\theta} = \Delta\theta/2$, $\tilde{n}_z = (2\,n_z)/factor$, and $\tilde{n}_\theta = 2\,n_\theta$. A representative grid showing both Ω and Ω_p is shown in Figure 3.2.

We now outline, illustrated using forward Euler, the complete implementation of (3.2.3.3)–(3.2.3.6) in conjunction with the use of a fine grid patch. As was discussed in §1.3, in practice the classical Runge-Kutta method is used for the time discretization for both efficiency and to avoid any cell Reynolds number constraint. Flow variables on Ω_p are denoted by $\tilde{\omega}$, $\tilde{\psi}$, $\tilde{U}$, and $\tilde{V}$, and i_{bd} denotes the i^{th} coordinate of the points on Ω that correspond to $\Gamma_{r_{maxp}}$, the outer radial boundary of Ω_p.

Time-stepping: Given $\{\overline{\omega}_{i,j}^n\}$ and $\{\overline{\tilde{\omega}}_{i,j}^n\}$, we compute $\{\overline{\omega}_{i,j}^{n+1}\}$ and $\{\overline{\tilde{\omega}}_{i,j}^{n+1}\}$ via the following steps:

Step 1. Solve for $\{\psi_{i,j}^n\}$ in the interior of Ω from

$$\left(\Delta_{(z,\theta)} + \frac{1}{12}((\Delta z)^2 + (\Delta\theta)^2)D_z^2 D_\theta^2\right)\psi_{i,j}^n = -\overline{\omega}_{i,j}^n$$

with boundary conditions

$$\psi^n\Big|_{\Gamma_{r_{min}}} = 0, \quad \psi^n\Big|_{\Gamma_{r_{max}}} = \mathrm{BC},$$

where BC is one of the far-field boundary conditions for ψ discussed in §3.2.4. All of the main results in §3.5 employ the ISBC (3.2.4.6) to compute ψ at the far-field.

Step 1.1 Solve for $\{\tilde{\psi}_{i,j}^n\}$ in the interior of Ω_p using

$$\left(\Delta_{(z,\theta)} + \frac{1}{12}((\Delta\tilde{z})^2 + (\Delta\tilde{\theta})^2)D_z^2 D_\theta^2\right)\tilde{\psi}_{i,j}^n = -\overline{\tilde{\omega}}_{i,j}^n$$

with boundary conditions

$$\tilde{\psi}^n\Big|_{\Gamma_{r_{min}}} = 0,$$

and

$$\tilde{\psi}^n\Big|_{\Gamma_{r_{maxp}}} = \begin{cases} \tilde{\psi}_{n_z,2j}^n = \psi_{i_{bd},j}^n & j = 0,1,\ldots,n_\theta \\ \tilde{\psi}_{n_z,2j+1}^n = \frac{-\psi_{i_{bd},j-1}^n + 9\psi_{i_{bd},j}^n + 9\psi_{i_{bd},j+1}^n - \psi_{i_{bd},j+2}^n}{16} & j = 0,1,\ldots,n_\theta - 1. \end{cases}$$

Step 2. Solve for $\{\omega_{i,j}^n\}$ in the interior of Ω using

$$\left(1+\frac{1}{12}\left((\Delta z)^2 D_z^2+(\Delta\theta)^2 D_\theta^2\right)\right)\left(e^{2z}\omega_{i,j}^n\right)=\overline{\omega}_{i,j}^n,$$

and recover ω^n along $\Gamma_{r_{min}}$ from ψ^n using Briley's formula (3.2.4.4). Along $\Gamma_{r_{max}}$ the outflow condition (3.2.4.5) is applied.

Step 2.1 Solve for $\{\tilde{\omega}_{i,j}^n\}$ in the interior of Ω_p using

$$\left(1+\frac{1}{12}\left((\Delta\tilde{z})^2 D_z^2+(\Delta\tilde{\theta})^2 D_\theta^2\right)\right)\left(e^{2z}\tilde{\omega}_{i,j}^n\right)=\overline{\tilde{\omega}}_{i,j}^n,$$

and recover $\tilde{\omega}^n$ along $\Gamma_{r_{min}}$ from $\tilde{\psi}^n$ using Briley's formula (3.2.4.4). Values for $\tilde{\omega}^n$ along $\Gamma_{r_{maxp}}$ are computed by interpolating ω^n as was done in step 1.1 for $\tilde{\psi}^n$.

It is important to note here that each of the discrete Poisson-like equations that appear above have constant coefficients. Hence, the solution of each may be obtained very efficiently using standard fast FFT-based solvers. At this point the only coupling between the flow variables on Ω and Ω_p has occurred along $\Gamma_{r_{maxp}}$. We now overwrite, at common grid points, ω (ψ) on Ω with a smooth combination ω and $\tilde{\omega}$ (ψ and $\tilde{\psi}$) on the intersection of Ω and Ω_p. We illustrate this procedure for ω.

Step 3 Overwrite ω^n (ψ^n) on $\Omega\cap\Omega_p$.

Recall that above we denoted the i^{th} coordinate of the points on Ω that lie along $\Gamma_{r_{maxp}}$ by i_{bd}. We now overwrite ω^n as follows:

$$\omega_{i,j}^n=\begin{cases}\tilde{\omega}_{2i,2j}^n & i=0,\ldots,i_{bd}/2\\ (1-S(i))\tilde{\omega}_{2i,2j}^n+S(i)\omega_{i,j}^n & i=(i_{bd}/2)+1,\ldots,i_{bd}-1\\ \omega_{i,j}^n & i=i_{bd},\ldots,n_z,\end{cases}$$

for $j=0,\ldots,n_\theta$, where

$$S(i)=\frac{\arctan(2\pi(e^{i\Delta z}-e^{i_{bd}\Delta z/2})/(e^{i_{bd}\Delta z}-e^{i_{bd}\Delta z/2})-\pi)+\arctan(\pi)}{2\arctan(\pi)}.$$

Here $(e^{i\Delta z} - e^{i_{bd}\Delta z/2})$ is the radial distance from the current point to $r = e^{i_{bd}\Delta z/2}$, and $(e^{i_{bd}\Delta z} - e^{i_{bd}\Delta z/2})$ the radial width of the outer portion of Ω_p on which ω^n is replaced with a convex combination of ω^n and $\tilde{\omega}^n$. $S(i)$ varies smoothly from 0 to 1 as i varies from $i_{bd}/2$ to i_{bd}. Exactly the same procedure is used to overwrite ψ^n.

Step 4. Recover the velocity components $\{U^n_{i,j}\}$ and $\{V^n_{i,j}\}$ using

$$U^n_{i,j} = \widetilde{D}_\theta\Big(1 - \frac{(\Delta\theta)^2}{6}D^2_\theta\Big)\psi^n_{i,j}, \quad V^n_{i,j} = -\widetilde{D}_z\Big(1 - \frac{(\Delta z)^2}{6}D^2_z\Big)\psi^n_{i,j}$$

with the boundary conditions

$$U^n\Big|_{\Gamma_{r_{min}}} = V^n\Big|_{\Gamma_{r_{min}}} = 0.$$

When $i = 1$, (3.2.4.11) is used for V^n. As discussed in §3.2.4, second order approximations are used to recover V^n at the far-field.

Step 4.1 Recover the velocity components $\{\tilde{U}^n_{i,j}\}$ and $\{\tilde{V}^n_{i,j}\}$ using

$$\tilde{U}^n_{i,j} = \widetilde{D}_\theta\Big(1 - \frac{(\Delta\tilde{\theta})^2}{6}D^2_\theta\Big)\tilde{\psi}^n_{i,j}, \quad \tilde{V}^n_{i,j} = -\widetilde{D}_z\Big(1 - \frac{(\Delta\tilde{z})^2}{6}D^2_z\Big)\tilde{\psi}^n_{i,j}$$

with the boundary conditions

$$\tilde{U}^n\Big|_{\Gamma_{r_{min}}} = \tilde{V}^n\Big|_{\Gamma_{r_{min}}} = 0.$$

Again, we use (3.2.4.11) for $\tilde{V}^n$ when $i = 1$, and both $\tilde{U}^n$ and $\tilde{V}^n$ are recovered along $\Gamma_{r_{max_p}}$ by interpolating U^n and V^n; see step 1.1.

Step 5. Update $\{\overline{\omega}^{n+1}_{i,j}\}$ in the interior of Ω using

$$\frac{\overline{\omega}^{n+1}_{i,j} - \overline{\omega}^n_{i,j}}{\Delta t} + \widetilde{D}_z\Big(1 + \frac{(\Delta\theta)^2}{6}D^2_\theta\Big)(U^n_{i,j}\,\omega^n_{i,j}) + \widetilde{D}_\theta\Big(1 + \frac{(\Delta z)^2}{6}D^2_z\Big)(V^n_{i,j}\,\omega^n_{i,j})$$

$$-\frac{1}{12}\Big((\Delta z)^2D^2_z + (\Delta\theta)^2D^2_\theta\Big)(U^n_{i,j}\,\widetilde{D}_z\omega^n_{i,j} + V^n_{i,j}\,\widetilde{D}_\theta\omega^n_{i,j})$$

$$= \nu\Big(\Delta_{(z,\theta)} + \frac{((\Delta z)^2 + (\Delta\theta)^2)}{12}D^2_zD^2_\theta\Big)\omega^n_{i,j}.$$

Step 5.1 Update $\{\widetilde{\widetilde{\omega}}_{i,j}^{n+1}\}$ in the interior of Ω_p using

$$\frac{\widetilde{\widetilde{\omega}}_{i,j}^{n+1} - \widetilde{\widetilde{\omega}}_{i,j}^{n}}{\Delta t} + \widetilde{D}_z\Big(1 + \frac{(\Delta\tilde{\theta})^2}{6} D_\theta^2\Big)(\tilde{U}_{i,j}^n\, \widetilde{\omega}_{i,j}^n) + \widetilde{D}_\theta\Big(1 + \frac{(\Delta\tilde{z})^2}{6} D_z^2\Big)(\tilde{V}_{i,j}^n\, \widetilde{\omega}_{i,j}^n)$$

$$-\frac{1}{12}\Big((\Delta\tilde{z})^2 D_z^2 + (\Delta\tilde{\theta})^2 D_\theta^2\Big)(\tilde{U}_{i,j}^n\, \widetilde{D}_z\, \widetilde{\omega}_{i,j}^n + \tilde{V}_{i,j}^n\, \widetilde{D}_\theta\, \widetilde{\omega}_{i,j}^n)$$

$$= \nu\Big(\Delta_{(z,\theta)} + \frac{((\Delta\tilde{z})^2 + (\Delta\tilde{\theta})^2)}{12} D_z^2 D_\theta^2\Big)\widetilde{\omega}_{i,j}^n.$$

The implementation of the fine grid patch in conjunction with EC4 requires no additional numerical tools other than standard interpolation. **NO** coupled grid Poisson solver is required. The same fast FFT-based solvers that are used to solve the Poisson-like equations which arise on both Ω and Ω_p. Also, since we are using the (ω, ψ) formulation the velocity field is automatically divergence free. The simplicity, high accuracy, and robustness of the EC4 scheme is preserved.

Obviously, the use of the fine grid patch has a cost in terms of additional computational time. In order to ensure stability we must have

$$|a|\frac{\Delta t}{h} \leq \text{CFL}, \qquad 4\nu\frac{\Delta t}{h^2} \leq 1.0,$$

where a is the maximum velocity present in the flow and CFL ≤ 1.5; see §1.3.2. When employing the fine grid patch $h = \min\{\Delta\tilde{z}, \Delta\tilde{\theta}\}$, resulting in a Δt that is half of what it would be if we did not use the patch. Although this doubles the number of time steps required to reach a given time t, the fine grid patch allows us to concentrate our computational effort per time step along the cylinder surface. Computational evidence of the effectiveness and the resulting gain in efficiency when the fine grid patch is used, is given in §3.4.2.

3.4 Algorithm Testing

Before presenting the results of numerical simulations of the flow around a cylinder we first evaluate the performance of the method applied to a variety of test problems. In particular, we are interested in determining the overall accuracy of the method as well as the effectiveness and resulting efficiency, if any, of using both the FGP, outlined in §3.3, and the ISBC (3.2.4.6). In order to answer each of these questions we focus on the computed vorticity field ω, and physical characteristics of the flow deduced from it, namely, the coefficient of drag (denoted by C_D) and points along the cylinder surface of zero shear stress.

We compute the C_D from ω using the following two formulas (see [KL]):

$$C_D = -2\frac{d}{dt}\int_\Omega \omega(z,\theta)\,\sin\theta\, e^{3z} dz\, d\theta, \tag{3.4.1}$$

$$C_D = -2\nu\int_0^\pi \frac{\partial\omega}{\partial z}(0,\theta)\sin\theta\, d\theta + 2\nu\int_0^\pi \omega(0,\theta)\sin\theta\, d\theta. \tag{3.4.2}$$

The derivative in the first term of (3.4.2) is approximated to second order via a one-sided formula, and numerical quadrature in the form of the trapezoidal rule is used to approximate the integrals appearing in both formulas. Since (3.4.1) requires integration over all of Ω we will refer to it as the global method, while (3.4.2) only involves integration along the cylinder surface and will be referred to as the local method. Note that the first term in (3.4.2) is the contribution to the C_D due to pressure, and the second term the contribution due to friction. An excellent measure of the overall quality of the computational results is the ability of our scheme to accurately compute the time evolution of the coefficient of drag for a large range of Reynolds numbers.

Further useful information deduced from the vorticity field is provided in the form of the time evolution of points along the cylinder surface of zero shear stress, i.e., points where $\omega=0$. In the case of a steady flow these points coincide with points of separation, while for unsteady flows this information proves quite useful in understanding the dynamics of the boundary layer.

3.4.1 Accuracy Checks

We present accuracy checks of the vorticity ω for a test flow at $Re=1000$, with free-stream velocity $u_\infty=1$, until time $t=3.0$. The computational domain extends to $r_{max}=3$, which is more than sufficient to capture the vorticity field up to the final time. Both impulsive and smooth startup scenarios are investigated, as well as results obtained with and without the use of a FGP. Since the largest gradients of the vorticity occur along the cylinder surface, we measure the error both there as well as over the interior of the computational domain. The following discrete L^2 and L^∞ norms are used:

$$\|\omega\|_{2,int} = \left(\sum_{i=1}^{n_r} \sum_{j=0}^{n_\theta} w_{i,j}^2 e^{2r_i} \Delta z \Delta\theta \right)^{1/2}, \qquad \|\omega\|_{2,bd} = \left(\sum_{j=0}^{n_\theta} w_{0,j}^2 \Delta\theta \right)^{1/2}$$

$$\|\omega\|_{\infty,int} = \max_{\substack{1\le i\le n_r \\ 0\le j\le n_\theta}} |\omega_{i,j}|, \qquad \|\omega\|_{\infty,bd} = \max_{0\le j\le n_\theta} |\omega_{0,j}|$$

We simulate a smooth startup by using the PBC, discussed in §3.2.4, for the far-field boundary condition for ψ, which is multiplied by a time dependent function that varies smoothly from 0 to 1 as t varies from 0 to ∞:

$$\psi(r_{max},\theta,t) = (1-e^{-t^2}) u_\infty (r_{max} - r_{max}^{-1}) \sin\theta.$$

A computation on a 512×2048 grid is taken as the "converged" solution. In Table 3.1 the relative errors at $t=3.0$, as the grid size is varied, are shown. In the interior of the computational domain the scheme achieves more or less fourth

order accuracy. Along the surface of the cylinder there is seen a slight loss of accuracy. This is true for both the L^2 and L^∞ norms.

Table 3.1: Relative Errors: Smooth Start without Patch

	Interior				Boundary			
	L^2		L^∞		L^2		L^∞	
grid	error	order	error	order	error	order	error	order
32×128	1.37e-02		9.78e-03		1.63e-02		2.02e-02	
64×256	1.15e-03	3.46	8.15e-04	3.46	6.12e-04	5.16	8.92e-04	4.76
128×512	8.41e-05	3.69	5.81e-05	3.74	2.78e-05	4.69	3.62e-05	4.97
256×1024	5.50e-06	3.91	3.81e-06	3.90	1.95e-06	3.77	2.53e-06	3.78

Table 3.2: Relative Errors: Smooth Start with Patch

	Interior				Boundary			
	L^2		L^∞		L^2		L^∞	
grid	error	order	error	order	error	order	error	order
32×128	6.45e-03		4.49e-03		4.41e-03		4.36e-03	
64×256	4.39e-04	3.84	3.21e-04	3.74	2.83e-04	3.95	2.73e-04	4.00
128×512	2.55e-05	4.15	1.87e-05	4.15	1.70e-05	4.08	1.70e-05	4.01

Table 3.3: Relative Errors: Smooth Start P .vs. NP

	Interior		Boundary	
grid	L^2	L^∞	L^2	L^∞
32×128	1.39e-02	1.61e-02	1.42e-02	1.83e-02
64×256	1.06e-03	9.78e-04	5.31e-04	7.60e-04
128×512	7.72e-05	6.94e-05	2.83e-05	3.21e-05
256×1024	5.35e-06	4.48e-06	2.32e-06	2.49e-06

In Table 3.2 are shown the relative errors at $t = 3.0$ of simulations using the FGP. In this case, a computation using a 256×1024 coarse grid is taken as the "converged" solution, and for each simulation $factor = 4$, with the resolution of Ω_p twice that of Ω. With the use of the FGP the scheme achieves more or less fourth order accuracy in the interior as well as along the boundary. This is clearly due to greater grid resolution in the boundary layer provided by the FGP.

An important question is if the computations with and without the FGP are converging to the same solution. In Table 3.3 the relative errors of computations with and without the FGP are shown. In computing the relative error we have taken the FGP computations, for each grid, as the converged solution. As we can see, all of the computations are converging to the same solution.

Table 3.4: Relative Errors: Impulsive Start without Patch

	Interior				Boundary			
	L^2		L^∞		L^2		L^∞	
grid	error	order	error	order	error	order	error	order
32×128	3.37e-02		2.20e-02		2.70e-02		5.94e-02	
64×256	3.57e-03	3.07	3.36e-03	2.56	1.12e-03	4.90	2.56e-03	4.81
128×512	2.67e-04	3.66	2.47e-04	3.69	6.57e-05	4.14	1.38e-04	4.31
256×1024	1.82e-05	3.84	1.50e-05	3.98	5.04e-06	3.61	1.21e-05	3.37

Table 3.5: Relative Errors: Impulsive Start with Patch

	Interior				Boundary			
	L^2		L^∞		L^2		L^∞	
grid	error	order	error	order	error	order	error	order
32×128	2.49e-02		1.94e-02		9.84e-02		1.04e-02	
64×256	2.78e-03	3.00	2.99e-03	2.55	1.02e-03	3.10	1.15e-03	3.00
128×512	2.15e-04	3.59	1.95e-04	3.92	1.06e-04	3.10	1.15e-04	3.17

An impulsive start is modeled by directly implementating the PBC to prescribe values for ψ at the far-field boundary. This startup scenario presents a tremendous challenge to numerical schemes in that the method must accurately capture the very thin vortex sheet that immediately forms along the surface of the cylinder; see [TE]. We again performed computations of the test flow with and without the FGP. The same grid parameter ranges were computed as above in the smooth startup case, and the same grid sizes were used to compute "converged" solutions.

Table 3.6: Relative Errors: Impulsive Start P .vs. NP

grid	Interior error		Boundary error	
	L^2	L^∞	L^2	L^∞
32×128	2.61e-02	3.24e-02	2.59e-02	5.78e-02
64×256	2.39e-03	2.91e-03	1.39e-03	3.03e-03
128×512	2.45e-04	2.30e-04	1.32e-04	1.99e-04
256×1024	2.76e-05	2.68e-05	1.58e-05	2.58e-05

Table 3.4 shows the relative errors of solutions computed without the FGP. We see that close to fourth order accuracy is achieved in the interior of the flow domain, with a slight loss of accuracy along the cylinder surface. Similar performance results when the FGP is used, as demonstrated in Table 3.5. Finally, Table 3.6 lists the relative errors of solutions obtained with and without the FGP, where again, the FGP computations are taken as the "converged" solution for each grid. As in the case of a smooth start, all of the simulations are seen to be converging to the same solution.

3.4.2 Performance of the Fine Grid Patch

To evaluate the effectiveness of the FGP we compare solutions of a flow at $Re=$ 9500 until $t=10.0$, with Ω extending to $r_{max}=8.0$. The ISBC (3.2.4.6) is used to compute the far-field boundary condition for ψ. One solution was computed using the FGP, and three without the patch on grids with increasing resolution. Table 3.7 lists the grid sizes for each of the computations, along with the CPU time required on a single processor of a CRAY T90. For run P, $\Delta z = 2\Delta\tilde{z}$, $\Delta\theta=2\Delta\tilde{\theta}$, and $factor=4$, resulting in the FGP extending to $r_{maxp}\approx 1.68$.

In Figure 3.3 the time evolution of the coefficient of drag is shown, computed using the global method (3.4.1), for each of the simulations. All are in good agreement up until $t\approx 6.2$, with runs P and NOPFINE in excellent agreement through to the final time $t=10.0$. While the NOPMFINE run uses a relatively

Table 3.7: Run Parameters for FGP Evaluation

Run	$\Omega(n_z, n_\theta)$	$\Omega_p(\bar{n}_z, \bar{n}_\theta)$	cpu seconds
P	(512,768)	(256,1536)	11867
NOP	(512,768)	none	2953
NOPMFINE	(768,1024)	none	na
NOPFINE	(1024,1536)	none	24165

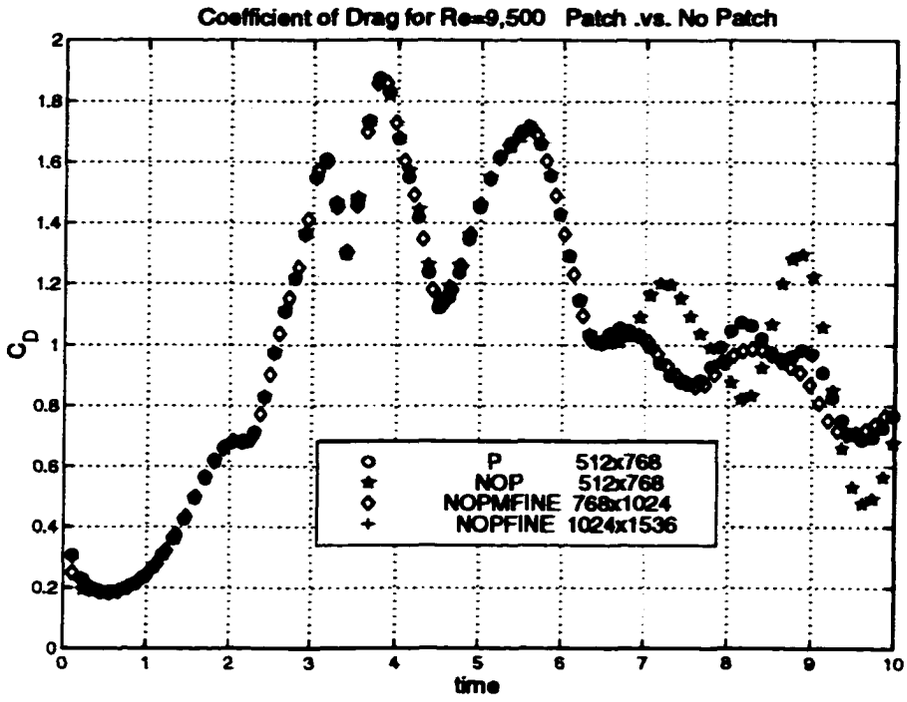

Figure 3.3: C_D Comparison w/o FGP

finer grid than was used for the NOP run, it is only in agreement with the results of the P and NOPFINE runs until $t \approx 7.5$. Thus, a lack of resolution along the cylinder surface eventually affects the overall character of the flow.

As a further comparison of the flow dynamics produced by each run we plot the time evolution of the points of zero shear stress produced by the FGP run P compared with the NOP and NOPMFINE runs, shown in Figure 3.4. Similar to the comparisons of the coefficient of drag, the NOPMFINE run agrees well with the P run until $t \approx 7.5$, while the NOP run begins to differ as early as $t \approx 6.0$. In

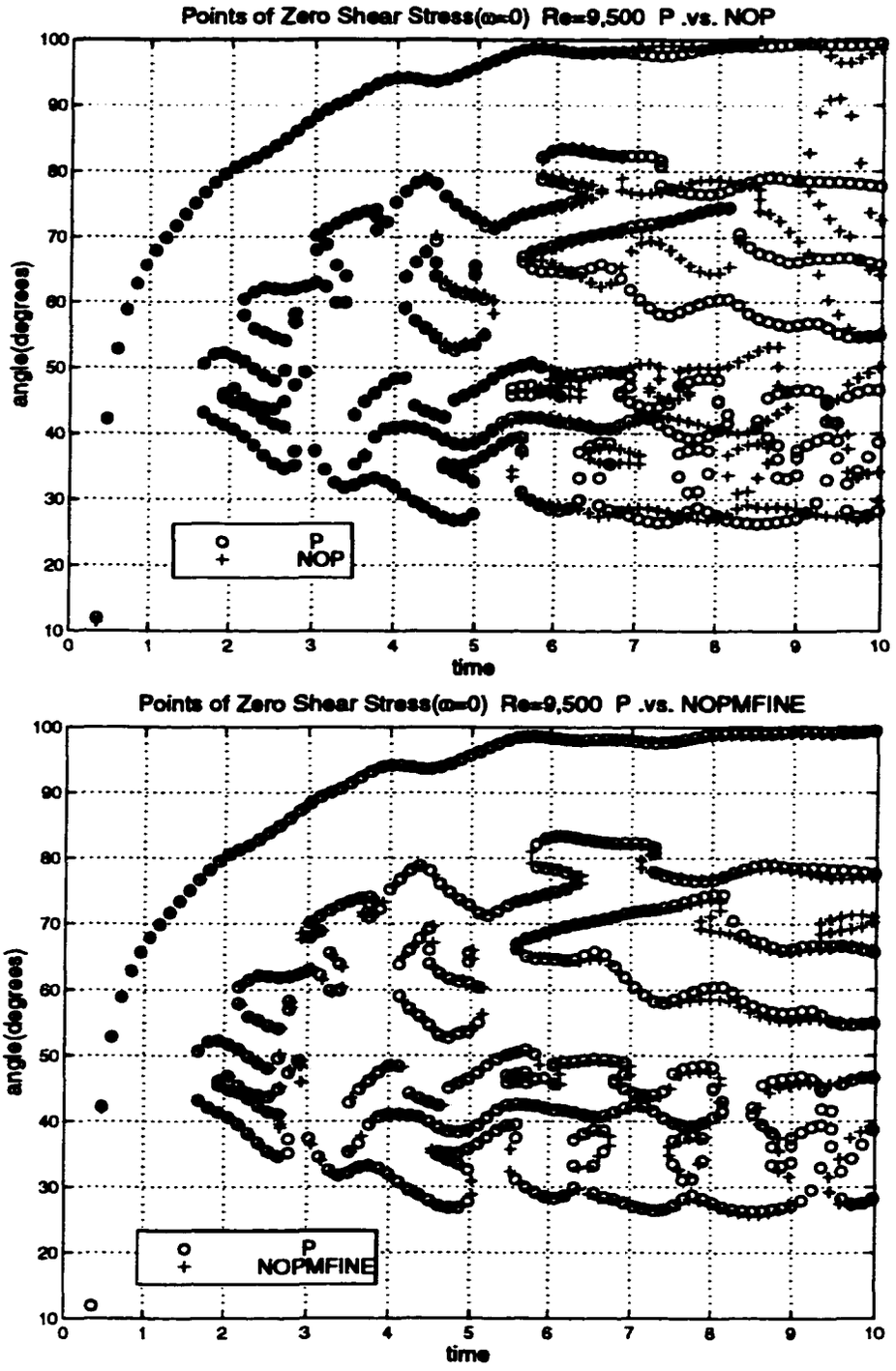

Figure 3.4: Points of Zero Shear Stress: P Compared with NOP and NOPMFINE

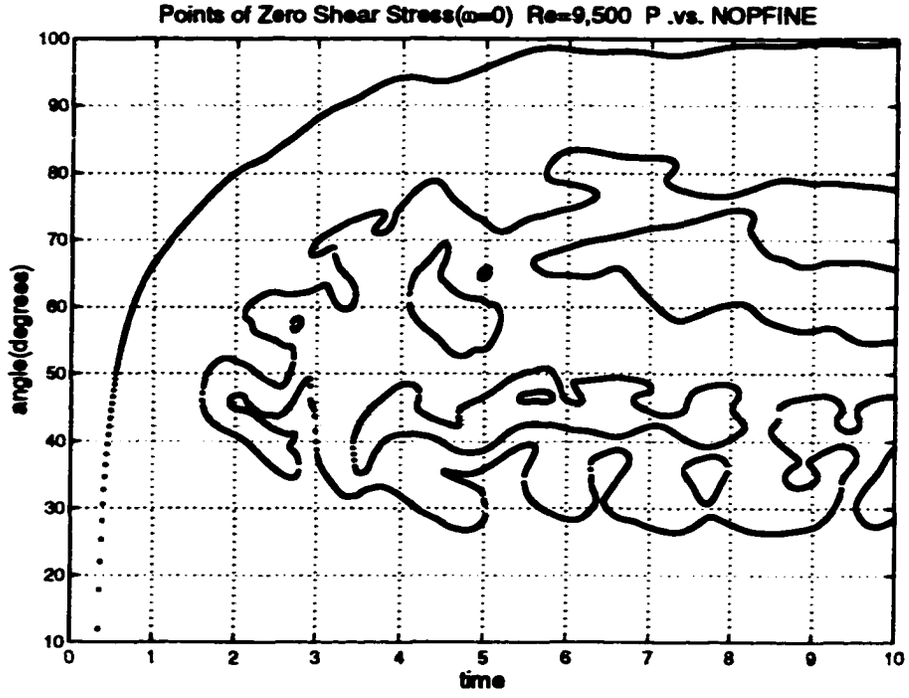

Figure 3.5: Points of Zero Shear Stress: P Compared with NOPFINE

Figure 3.5 is shown the comparison of the P and NOPFINE runs. They are in excellent agreement, producing identical boundary layer dynamics.

In Table 3.7 we see that the NOPFINE run required more than twice the cpu time as the P run. Since the P and NOPFINE runs are producing nearly identical flow dynamics, this confirms our motivation for developing the FGP methodology to compute the flow around a cylinder. It is very inefficient computationally to use the same resolution over the entire computational domain. The FGP allows us to concentrate our computational effort where the "action" is, resulting in very significant computational savings.

3.4.3 Effectiveness of the Integral Series Boundary Condition

In order to solve the Poisson equation (3.2.2.5) we must prescribe values for ψ along the far-field computational boundary $\Gamma_{r_{max}}$. The difficulty arises from the

fact that the true physical domain of the flow we are attempting to model is infinite, while for obvious practical considerations our computational domain is of finite extent. In §3.2.4 we outlined the ISBC (3.2.4.6), developed by Ting (see [Ti],[TK]), for computing an approximation to the far-field values of ψ. Sa and Chang [SC] performed a detailed study of the use of the ISBC for flow around the cylinder, although for the case of relatively low Re numbers. They found the boundary condition to be excellent. Our experience has been the same. Its accuracy has allowed us to use a significantly smaller computational domain than would have been required if, say, we had used the PBC.

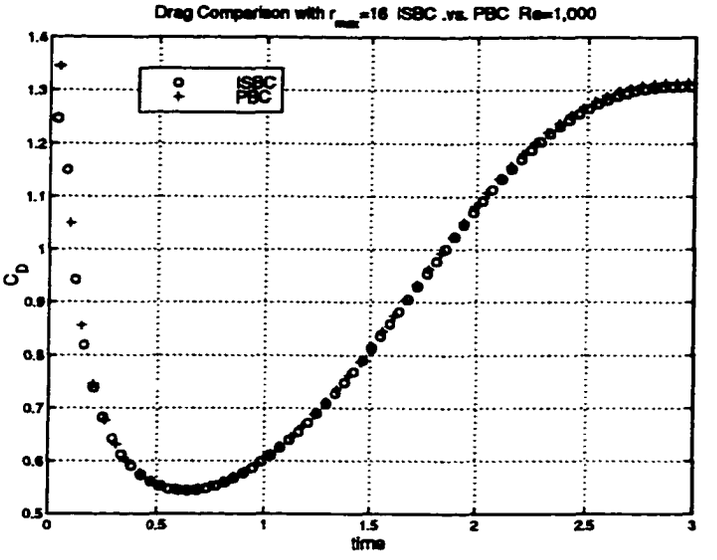

Figure 3.6: C_D Comparison w/o FGP

To demonstrate the effectiveness of the ISBC we compare simulations using both the ISBC and PBC, on computational domains with varying radii, for a flow at $Re = 1000$ until $t = 3.0$. We begin with a computation on a $(n_z, n_\theta) = (512, 512)$ grid with $r_{max} = 16$. Figure 3.6 shows the coefficient of drag produced by these runs, and we see that the two far-field conditions give almost identical results. This is not surprising since the vorticity field for this Reynolds number is effectively contained within two cylinder diameters at $t=3$ (see §3.5), and thus

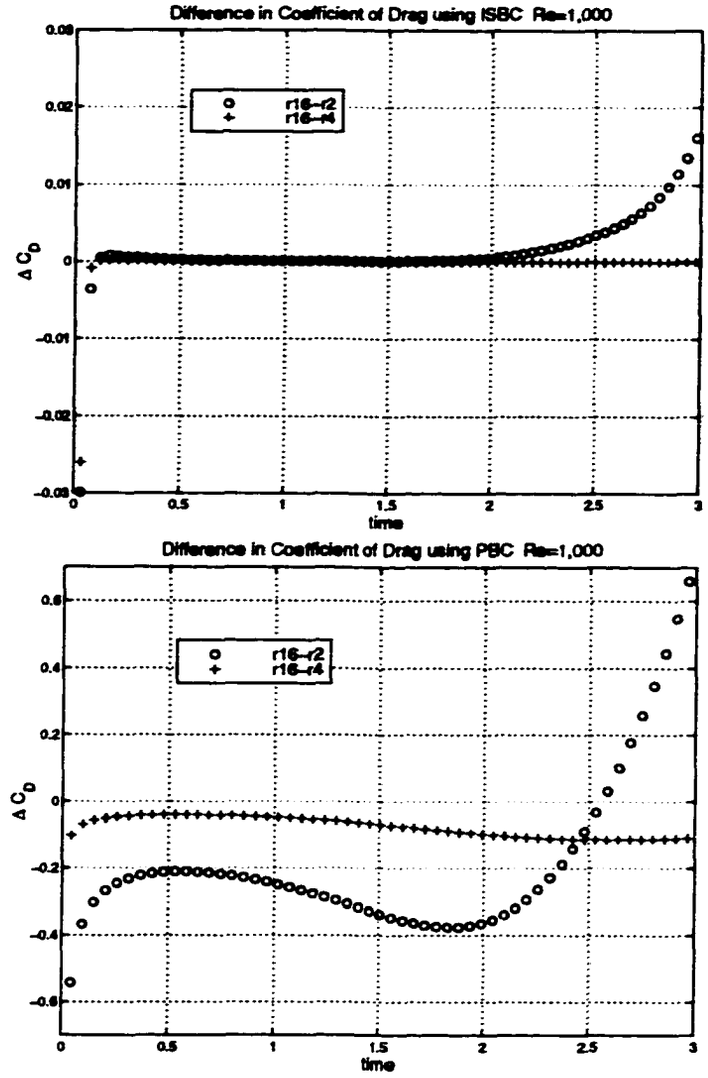

Figure 3.7: Difference in Coefficient of Drag

has not yet had an appreciable affect on the free stream flow at $r_{max}=16$.

We now recompute the test flow, again using both the PBC and ISBC, on domains extending to $r_{max}=2$ and $r_{max}=4$, using a $(128, 512)$ and $(256, 512)$ grid, respectively. Thus, for all the computations Δz, $\Delta\theta$, and hence Δt, are identical. In Figure 3.7 is shown the difference between the drag coefficient produced by each of these runs and the previous runs using $r_{max}=16$. The performance of the ISBC is excellent. Even with $r_{max}=2$, using the ISBC the absolute error in the

drag coefficient is less than 2% at $t=3.0$. This is remarkable since by this time the strength of the vorticity field close to the far-field boundary is not negligible (see Figure 3.11). The PBC computation with $r_{max}=4$ is acceptable, whereas with $r_{max}=2$ the absolute error is almost 50% at $t=3.0$.

For all three values of r_{max} the ISBC produces a very accurate description of the far-field boundary condition for ψ, as evidenced by the consistent calculations of C_D. Furthermore, computing the ISBC at each time step is also efficient, typically accounting for no more than $1\% - 2\%$ of the total runtime.

3.5 Main Numerical Results

We now present high resolution benchmark quality numerical simulations of the early stages of the flow around an impulsively started circular cylinder for Reynolds numbers ranging from 1,000 to 100,000. The computational parameters for each simulation are given in Table 3.8. In all cases the FGP was used with $\Delta z=2\Delta\tilde{z}$ and $\Delta\theta=2\Delta\tilde{\theta}$. The far-field boundary condition for ψ was determined using the ISBC as given by (3.2.4.6). As discussed in §3.3, RK4 is used for the time discretization of the vorticity transport equation, and the time step is chosen so that the stability conditions given in (1.3.2.12) are satisfied. For all simulations presented here we have taken CFL ≤ 1.0.

Table 3.8: Computational Parameters for Cylinder Flow

Re	n_z	n_θ	r_{max}	$\tilde{n}_z$	$\tilde{n}_\theta$	r_{maxp}	tfinal
1,000	384	512	8.00	96	1024	1.30	10.0
3,000	512	768	8.00	128	1536	1.30	10.0
9,500	1024	1024	8.00	256	2048	1.30	10.0
20,000	1024	1536	4.33	256	3072	1.20	5.0
50,000	1536	3072	3.75	384	6144	1.18	5.0
100,000	1536	4096	2.69	384	8192	1.13	5.0

Only by fully resolving the viscous boundary layer along the surface of the cylinder, which has a thickness that is $O(1/\sqrt{Re})$ (see [Tr]), can we expect to achieve benchmark quality results. To this end, the grid resolution for each simulation has been chosen so that

$$\max\{\Delta\tilde{z}, \Delta\tilde{\theta}\} \approx \frac{1}{10\sqrt{Re}},$$

which translates into ensuring that approximately 10 radial computational points lie in the boundary layer. Additionally, since the FFT is most efficient on data sets whose lengths are products of small primes, all grid sizes have further been chosen to be of the form 2^n3^m. All of the computations were performed on the Cray C90 at the Pittsburgh Supercomputer Center, and the code achieves on average 425 MFLOPS on a single processor for the high Reynolds number simulations, indicating a high degree of vectorization. We have found that approximately 80% of the runtime is attributable to the FFT based discrete Poisson solvers, making the code an excellent candidate for parallelization.

Impulsively started flows present a serious challenge for numerical methods due to the singular nature of the effects of the viscous boundary layer at time $t = 0^+$. Until now, the simulations of Koumoutsakos & Leonard [KL] and Anderson & Reider [AR] were considered benchmarks in the field. [KL] use an adaptive vortex method, while the scheme in [AR] is based, similar to that in this thesis, on fourth order finite differences. However, in [AR] the surface vorticity and the stream function at the far-field are computed using global methods, each requiring the costly solution of a discrete Poisson equation.

[KL] and [AR] compute solutions for Reynolds numbers up to 9,500, and both include results at $Re = 1,000$ and $Re = 3,000$. However, [AR] only present data until $t = 3.0$, which are in general agreement with the computations in [KL]. Therefore, we only present comparisons of our results with the [KL] simulations.

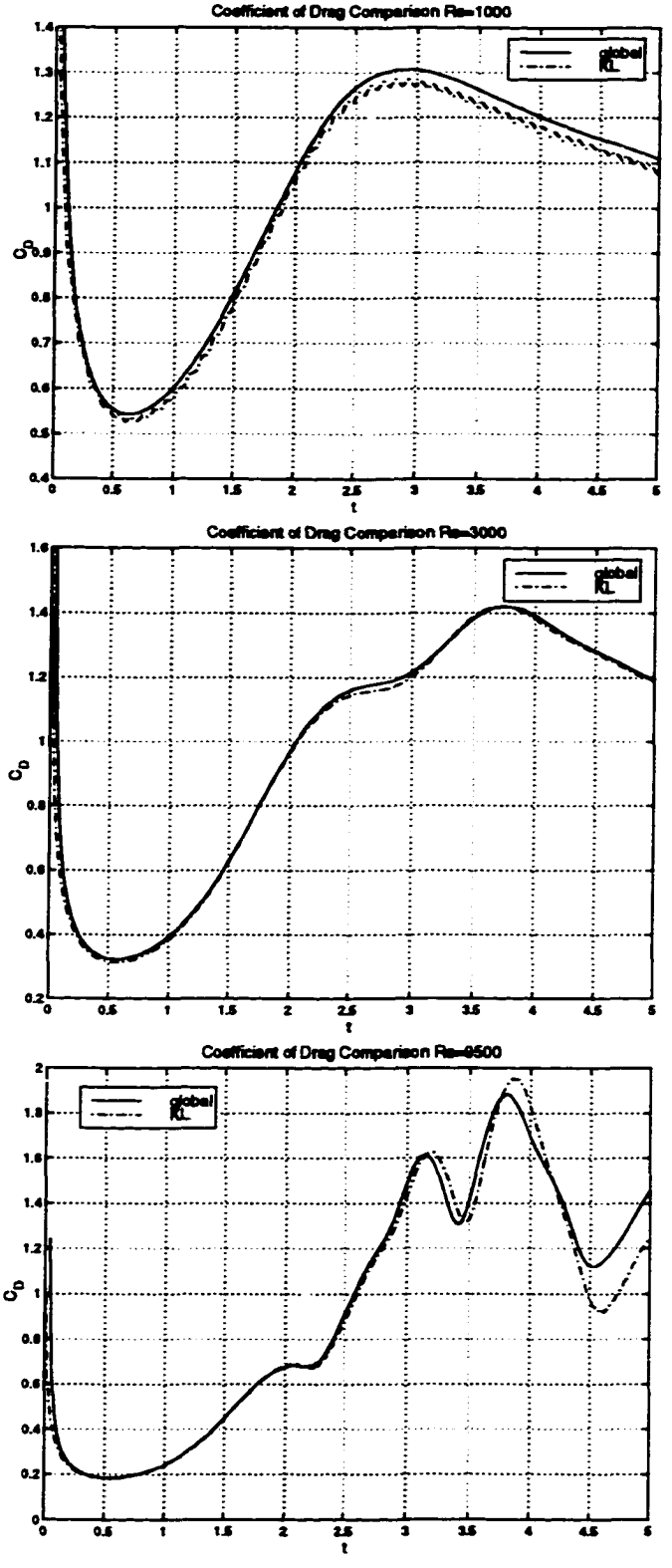

Figure 3.8: C_D Comparison with [KL] Simulations

In Figure 3.8 the time evolution of C_D at the above mentioned Reynolds numbers is shown, calculated using the global method (3.4.1), computed by both the present scheme and [KL]. For both $Re=1,000$ and $3,000$ the results are in good agreement, while differences are apparent at $Re=9,500$ as early as $t=3.5$, and become quite significant by $t=5.0$. This is a clear indication that by this point in time the respective schemes are producing dissimilar flow dynamics. In order to verify our results we recalculated this flow with both greater resolution and a larger computational domain. Our computation of C_D remained unchanged. We further note that we have agreement of the computation of C_D using both the local method (3.4.2) and the global method (3.4.1); see Figure 3.16. In the case of the [KL] simulations, the corresponding measurements begin to differ at $t \approx 4.0$.

We now present our results of the impulsively started flow around a circular cylinder. As mentioned above, computations were performed for each Reynolds number listed in Table 3.8. For each simulation we present plots of the time evolution of points on the surface of the cylinder of zero shear stress ($\omega = 0$), the time evolution of the coefficient of drag (C_D) computed by both the local and global methods, and contour plots of equi-vorticity at selected times. As the Reynolds number increases, which essentially corresponds to a decrease of the fluid viscosity, a number of trends occur. The initial vortex sheet that forms around the cylinder becomes increasingly thinner, and the time at which the boundary layer first separates from the surface of the cylinder decreases. In addition, ever smaller scale vortical structures are produced in the boundary layer, and in the case of the higher Reynolds number simulations, these vortices completely separate from the surface of the cylinder, and are then advected into the wake of the flow. Quite interestingly, for the $Re = 50,000$ and $100,000$ simulations the boundary layer separation subsequently triggers a vortex shedding

process that appears to become periodic in time. This is suggested by the plots of the time evolution of points of zero shear stress along the surface of the cylinder in Figure 3.23. Evidence of this phenomena can also be seen in the oscillations that appear in the C_D plots shown in Figure 3.24.

For the contour plots of equi-vorticity, solid lines are used to indicate negative vorticity (clockwise rotation), while the dashed lines indicate positive vorticity. Only images of the upper right quarter of the cylinder are presented. Even though a vortex sheet forms along the front of the cylinder, it does not separate, and the resulting flow in that region is quite uninteresting. In addition to providing a "view" of the dynamics of the flow, these plots are also very useful in understanding the flow structures that determine the body forces experienced by the cylinder. We can infer from the expression for the global computation of C_D (3.4.1) that a negative (clockwise) vortical structure moving in the positive y direction adds to C_D, while an increase in the strength of a positive vortex at the upper rear of the cylinder leads to a reduction of C_D. We note that at all Reynolds numbers, due to the singular nature of the impulsive start, there is seen a sudden initial drop in C_D. However, in all cases the effects of convection quickly becomes important and the C_D begins to recover.

3.5.1 $Re = 1,000$ and $Re = 3,000$

For these Reynolds numbers the method quite easily captures the full dynamics of the flow. A time sequence of equi-vorticity contours at $Re = 1,000$ is presented in Figures 3.11 and 3.12. At time $t = 2.0$ the primary vortex is beginning to rollup, confining a region of counter rotating positive vorticity forming below it. By $t = 4.0$ the primary vortex has fully developed, and a small tertiary vortex is seen at the cylinder surface. The development of this tertiary vortex shortly after $t = 2.0$ can also be deduced from the time evolution of the points of zero shear

stress shown in Figure 3.9. While the primary vortex, and to some degree the secondary vortex, continues to grow, the tertiary vortex remains confined. In the final image at $t=6.0$, we see that a large recirculation region has formed at the rear of the cylinder, and both the secondary and tertiary vortical regions appear to be stabilizing.

In Figures 3.13 and 3.14 a time sequence of equi-vorticity contours for $Re=3,000$ is presented. While similar in overall nature to the flow at $Re=1,000$, as expected, the boundary layer begins its rollup earlier, and produces an increasing number of smaller vortical structures along the surface of the cylinder. These dynamics are most easily deduced from the time evolution of the points of zero shear stress along the surface shown in Figure 3.9. In particular, just after $t=3.0$ a small vortex forms at $\theta \approx 60°$, disappears at $t \approx 6.0$, and briefly reappears at $t \approx 7.0$. By $t=6.0$ a slightly more compact primary vortex, as compared to the $Re=1,000$ flow, has fully formed, and the tertiary vortex is more pronounced. Note that there is a smaller vortex just below the tertiary vortex, and the two merge just before $t=8.0$; see Figure 3.9.

For both simulations there is excellent agreement between the computations of C_D using both the global and local methods. Furthermore, a greater portion of C_D is due to the pressure forces at $Re=3,000$ than at $Re=1,000$, with a clear decrease in the friction drag. This general trend continues with increasing Reynolds number.

3.5.2 $Re=9,500$ and $Re=20,000$

At these Reynolds numbers a series of new separation phenomena appears. In Figures 3.17, 3.18, and 3.19 a time sequence of equi-vorticity contours at $Re=9,500$ is presented. By $t=2.0$ a primary, secondary, and tertiary vortex have

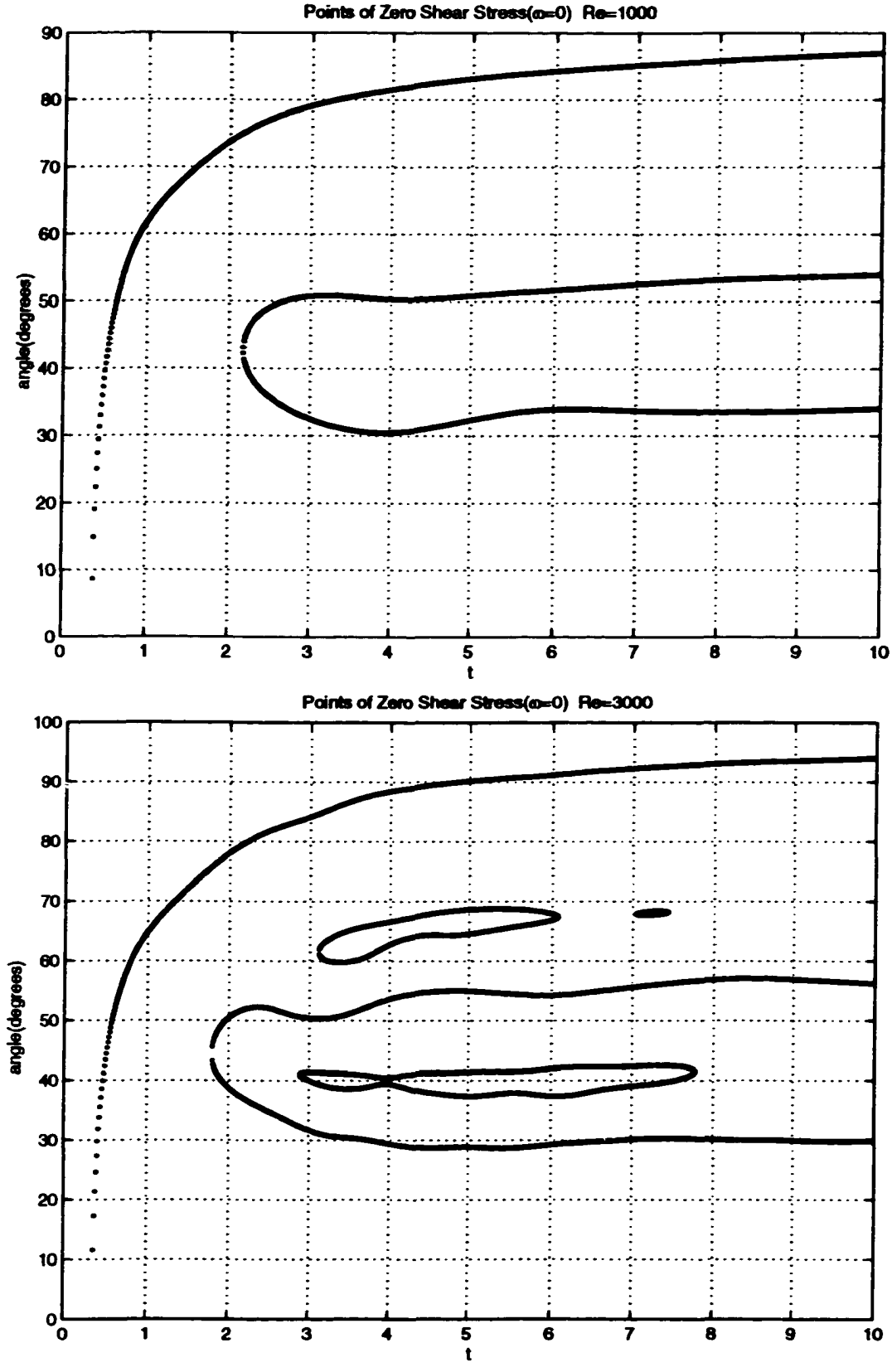

Figure 3.9: Time Evolution of $\omega=0$ for $Re=1,000$ and $3,000$

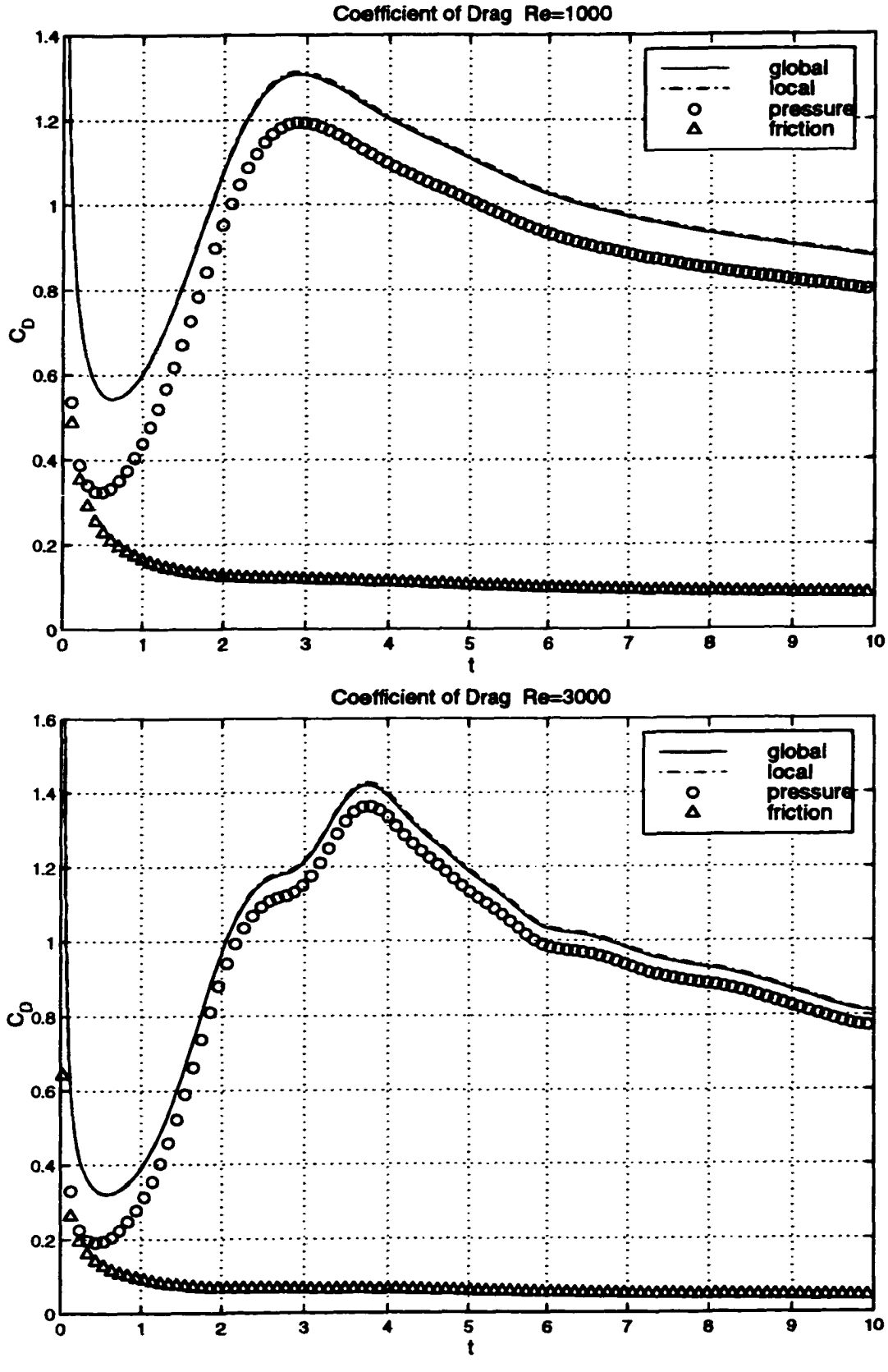

Figure 3.10: Time Evolution of C_D for Re=1,000 and 3,000

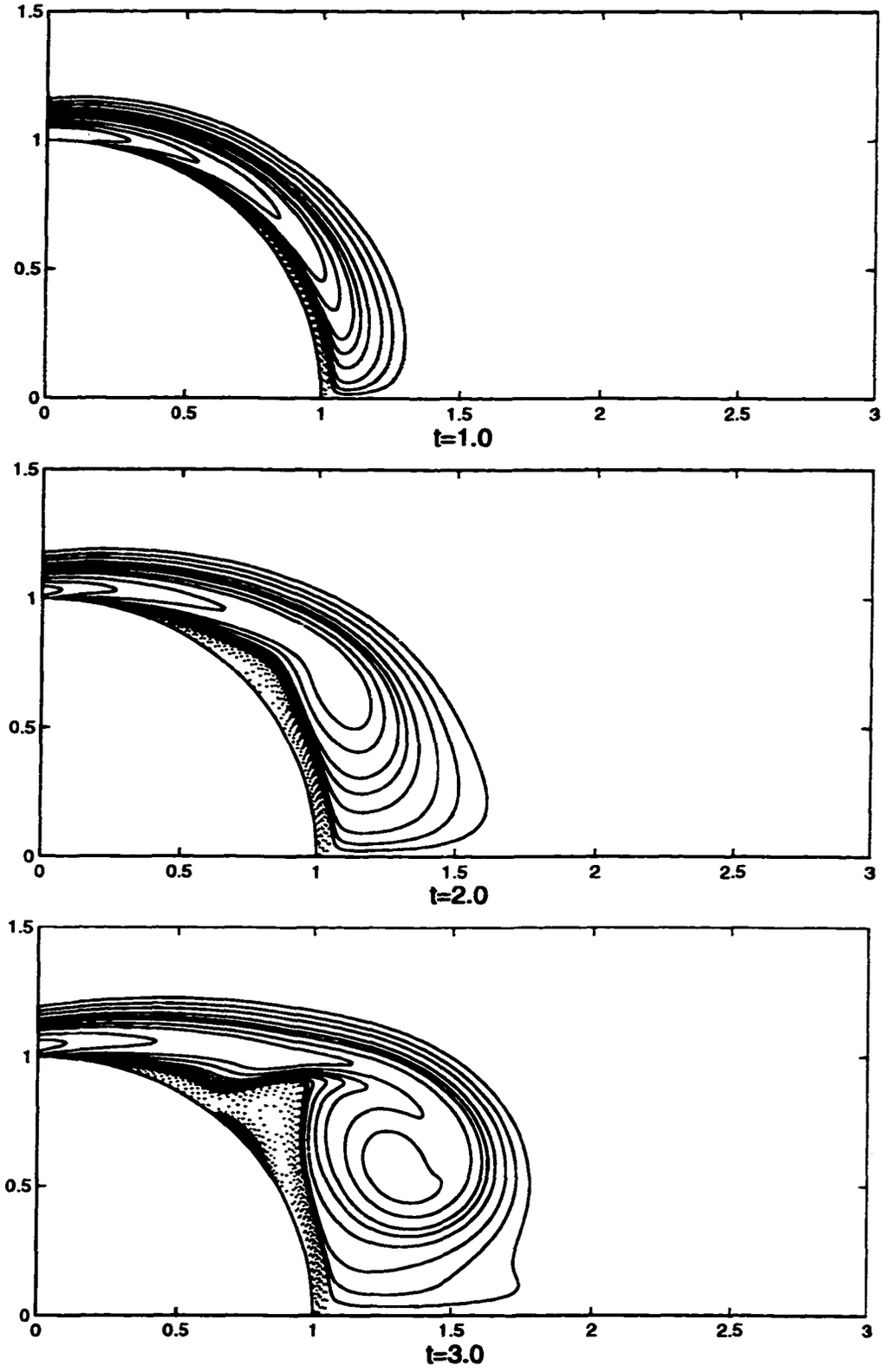

Figure 3.11: Equi-vorticity Re=1,000 at t=1.0, 2.0, and 3.0

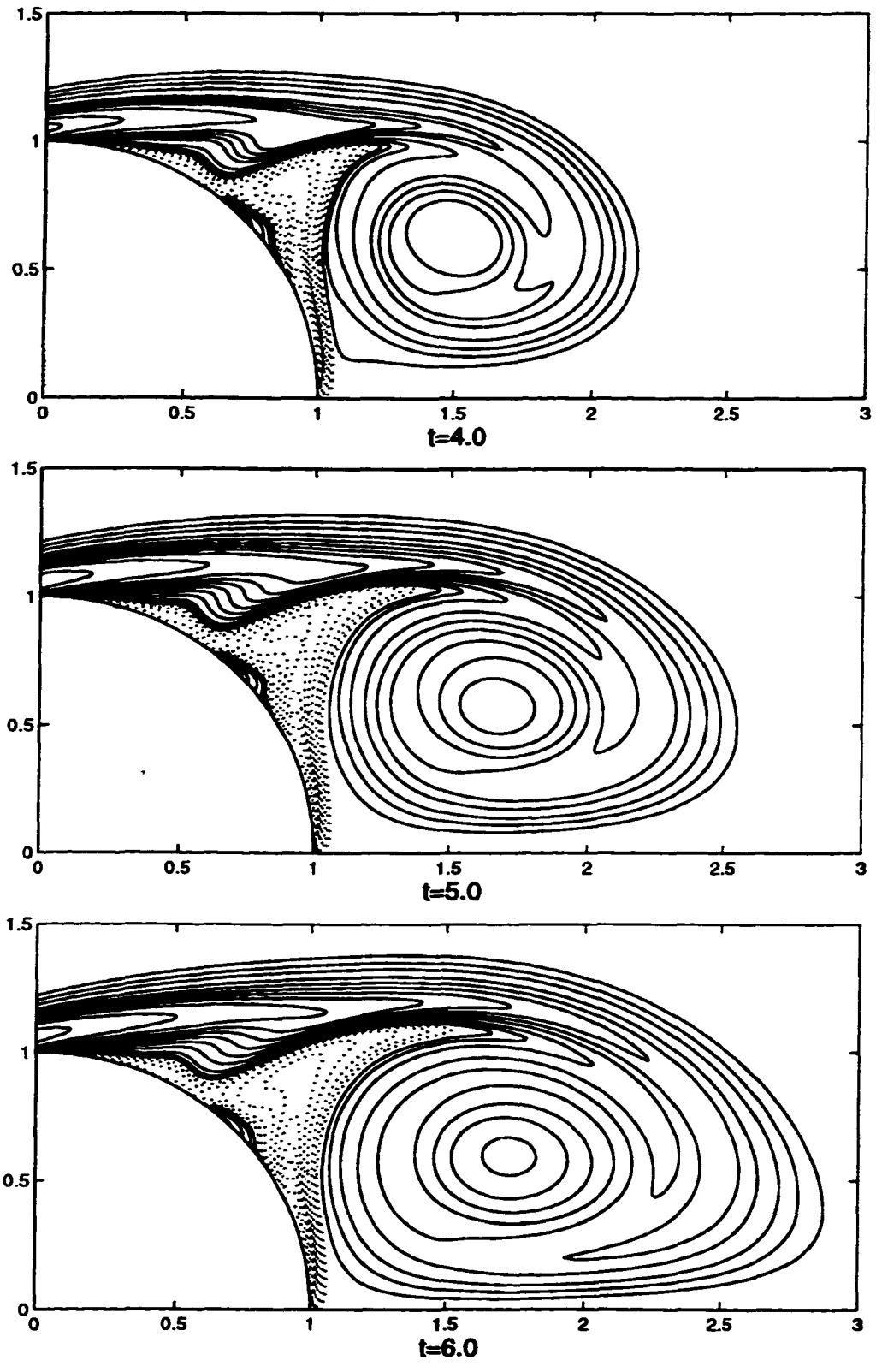

Figure 3.12: Equi-vorticity Re=1,000 at t=4.0, 5.0, and 6.0

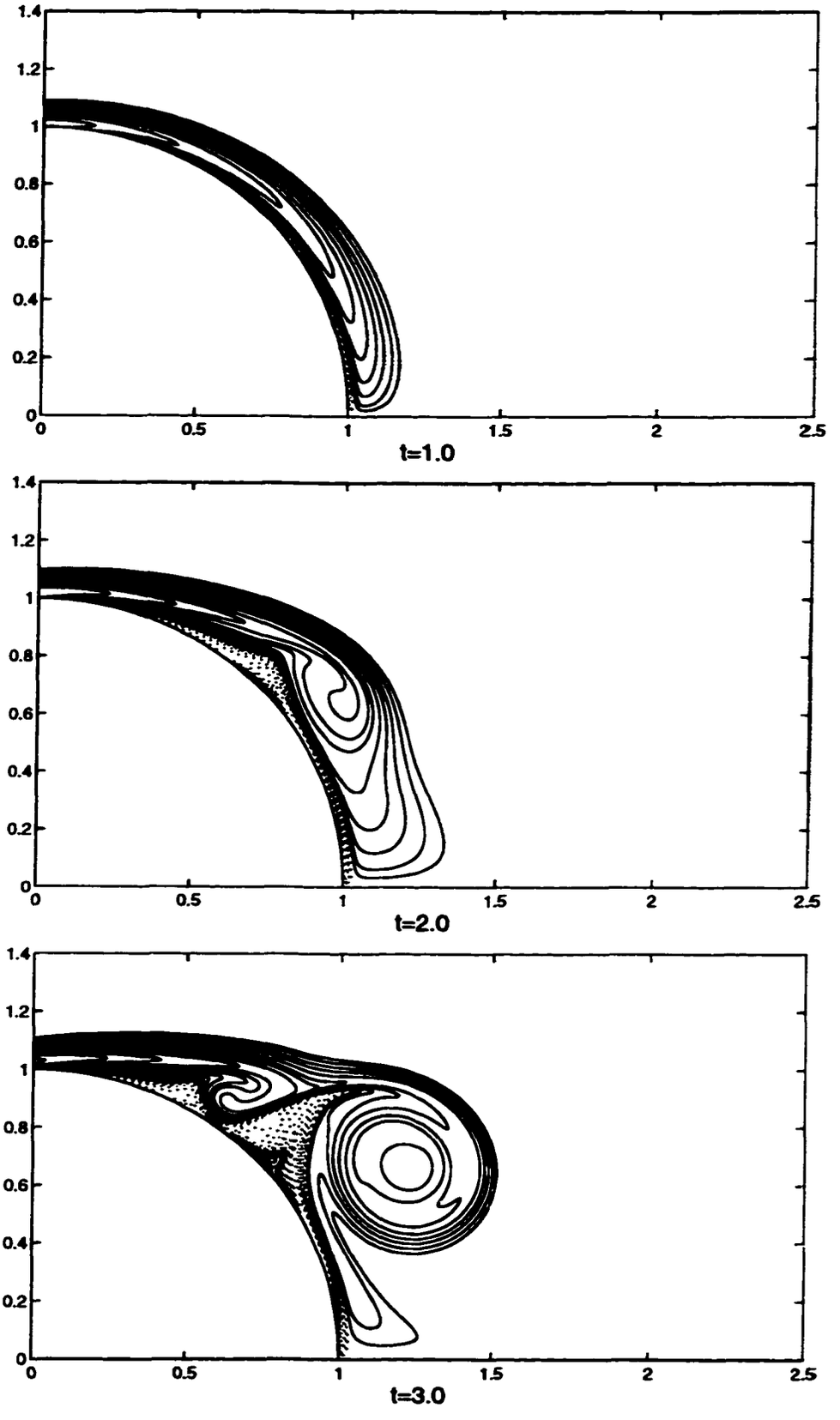

Figure 3.13: Equi-vorticity $Re=3,000$ at $t=1.0$, 2.0, and 3.0

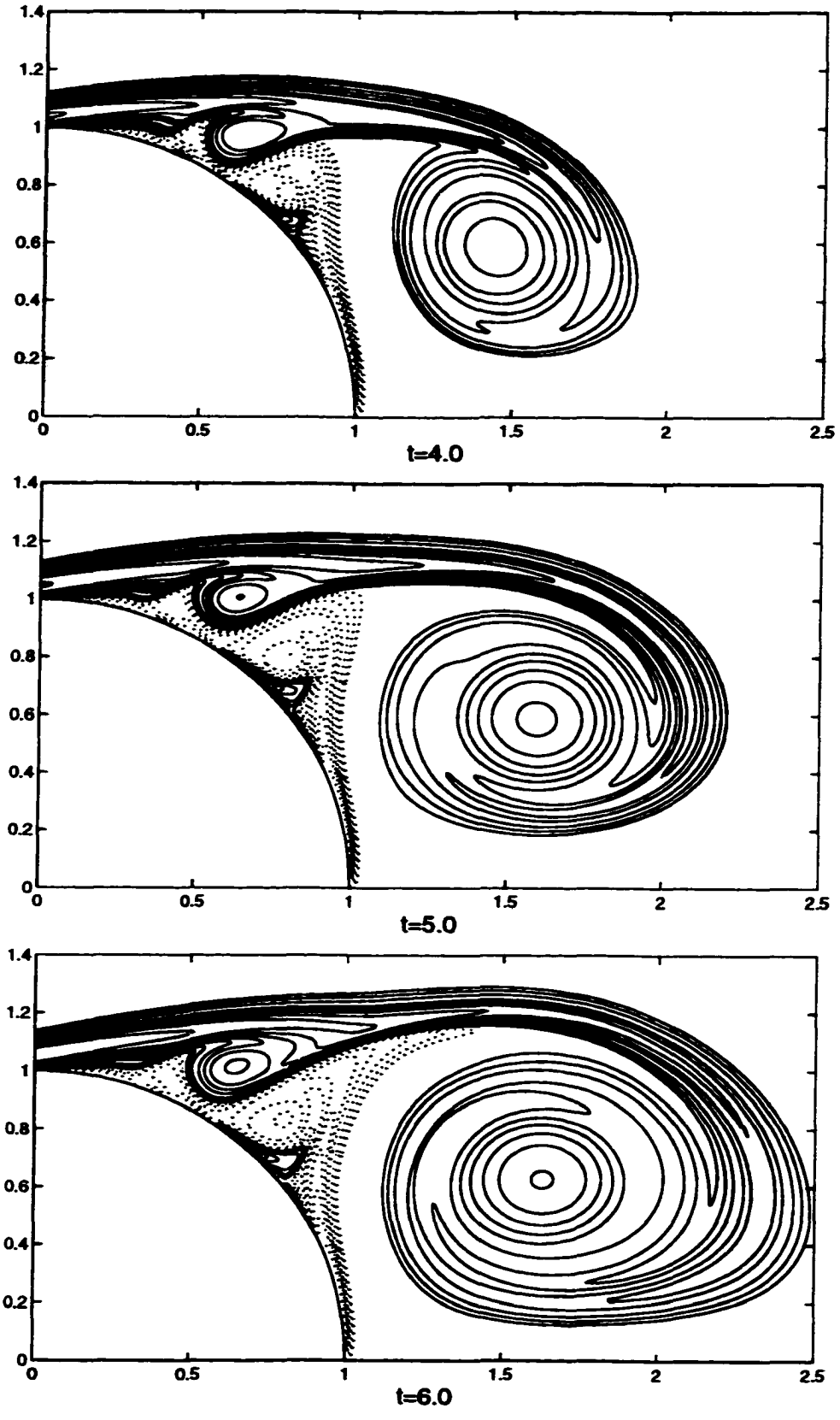

Figure 3.14: Equi-vorticity Re=3,000 at t=4.0, 5.0, and 6.0

formed. As expected, due to the decrease in viscosity as compared to the lower Reynolds number simulations presented in §3.5.1, these boundary layer structures are both significantly smaller and form earlier. At $t=2.5$ the primary vortex has started to detach from the body of the cylinder, carrying with it a small portion of the secondary vortex. Since this vortex has negative sign, its movement upward into the flow causes the increase in C_D; see Figure 3.16. Simultaneously, a new negative vortex is seen forming at a slightly greater angle than did the primary vortex, producing a new region of positive vorticity below it. This process repeats a number of times as indicated by the plot of time evolution of points of zero shear stress in Figure 3.15. At later times ($t=4-5$), although a number of vortices have separated from the body, they have become caught in a recirculation region at the rear of the cylinder. By $t=7.5$ this region of recirculation, which is being continually fed by the oncoming flow, has become quite large and is attached by a thin link to the separating shear layer.

In Figures 3.20, 3.21, and 3.22 a time sequence of equi-vorticity contours for $Re = 20,000$ is presented. We see that many of the early flow phenomena that develop are similar to those found at $Re=9,500$. However, there is clearly an increased complexity in the structure of the boundary layer, which begins to separate at an earlier time. At $t=3.0$, two negative vortices that have detached from the boundary layer begin to merge, allowing, by $t=3.5$, a larger region of positive vorticity to develop near the cylinder surface. This in turn pushes the feeding link of the merging vortices upwards, confining the the oncoming flow and triggering the development of a larger negative vortex. This process does not go unnoticed by the C_D which, due to the movement of a large region of negative vorticity down towards the cylinder surface, decreases over this same period of time; see Figure 3.16.

Note that, as was the case for $Re = 1,000$ and $3,000$, for these simulations we have excellent agreement between the computations of C_D using both the global and local methods, and most of the drag on the cylinder is now solely due to the dynamics of the pressure along the surface.

3.5.3 $Re = 50,000$ and $Re = 100,000$

By far, these simulations are the most interesting, and present the greatest challenge to our numerical scheme. A time sequence of plots of equi-vorticity contours at $Re = 50,000$ is presented in Figures 3.25, 3.26, and 3.27. For $Re = 100,000$, these plots are presented in Figures 3.28, 3.29, 3.30, and 3.31. There is clearly a significant difference in the overall character of the flow, when compared with the lower Reynolds number simulations, indicating the presence of a phase transition for the impulsively started cylinder flow between $Re = 20,000$ and $50,000$. Note both the complexity and similarities between the plots of the time evolution of points of zero shear stress along the surface of the cylinder shown in Figure 3.23. At both $Re = 50,000$ and $100,000$, once the initial boundary layer has separated from the cylinder surface, an almost periodic vortex shedding process develops. This is most striking at $Re = 100,000$, indicated by the well defined finger-like structures in the image. As expected, due to the decrease in the fluid viscosity, the initial vortex sheet that forms around the cylinder is thinner at $Re = 100,000$ as compared to $Re = 50,000$, and separates earlier. A particularly interesting new phenomena, not seen in the lower Reynolds number simulations, is the initial simultaneous rollup of **two** negative vortices along the cylinder surface. This can be seen in the contour plot of equi-vorticity at $t = 2.0$ for $Re = 50,000$ (Figure 3.25), and at $t = 1.7$ for $Re = 100,000$ (Figure 3.29). In the latter image, the detailed structures that have formed in the boundary layer are quite dramatic. In the final plots of equi-vorticity presented for each Reynolds number, Figures

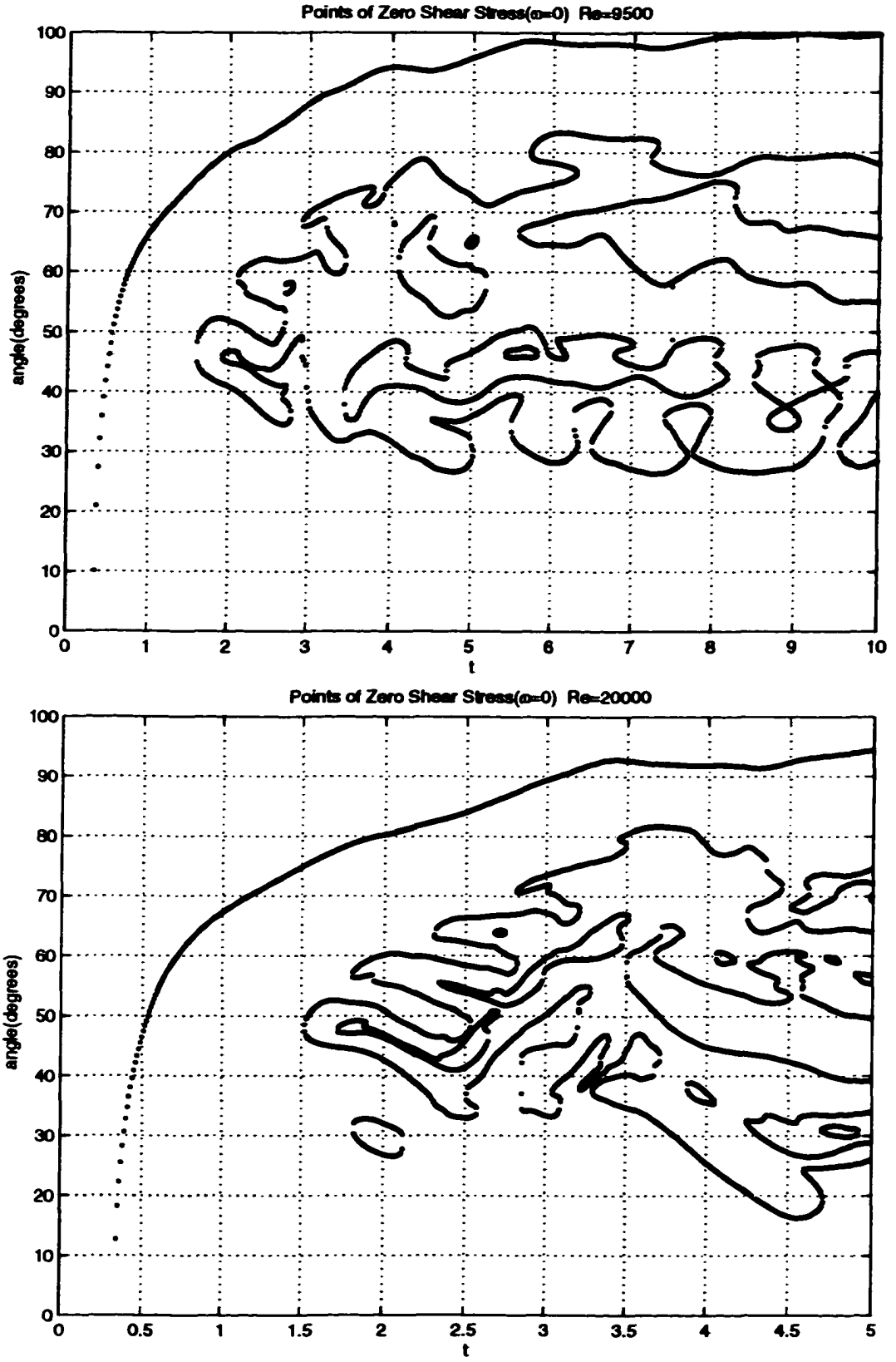

Figure 3.15: Time Evolution of $\omega=0$ for $Re=9,500$ and $20,000$

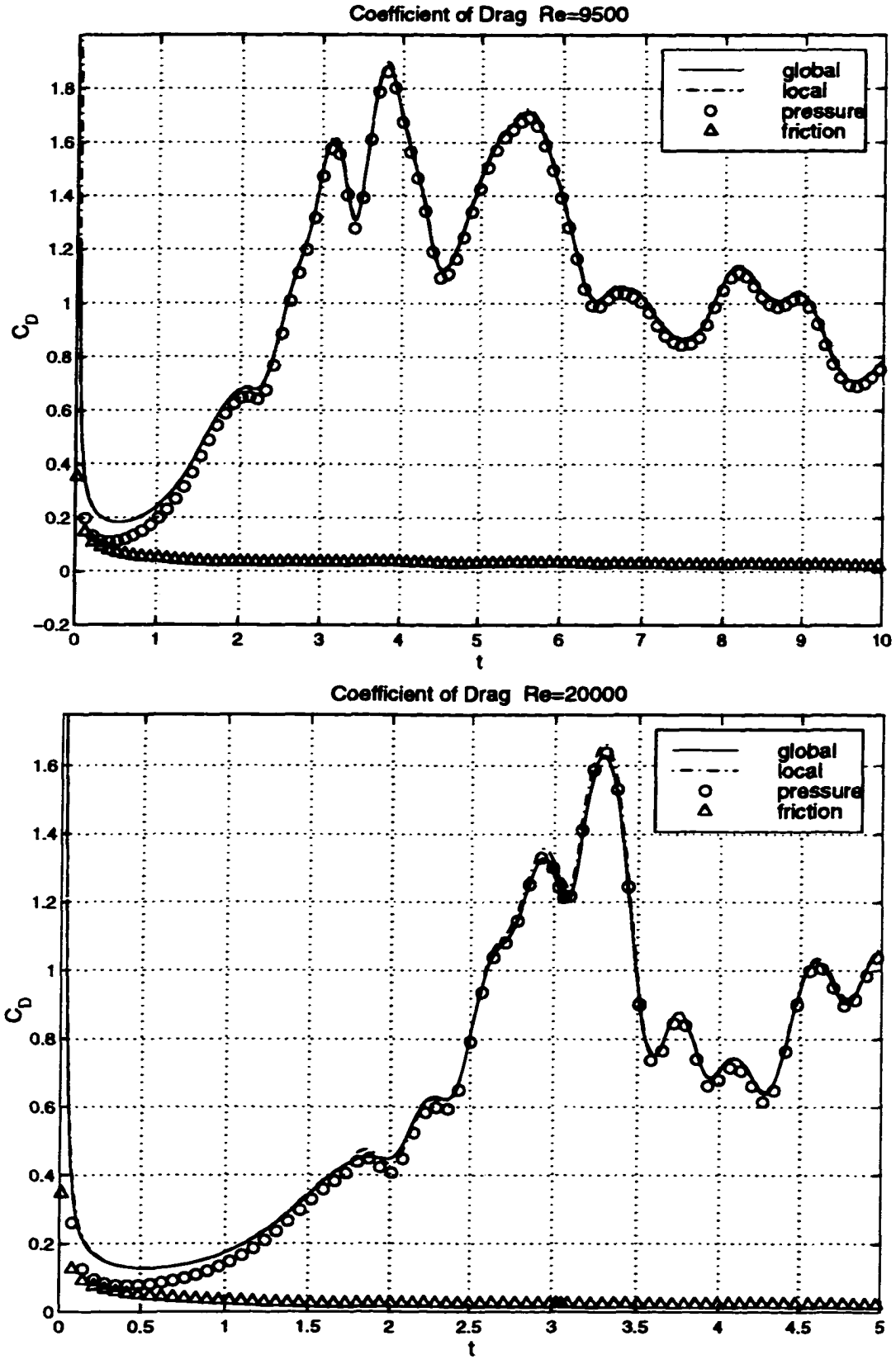

Figure 3.16: Time Evolution of C_D for $Re=9,500$ and $20,000$

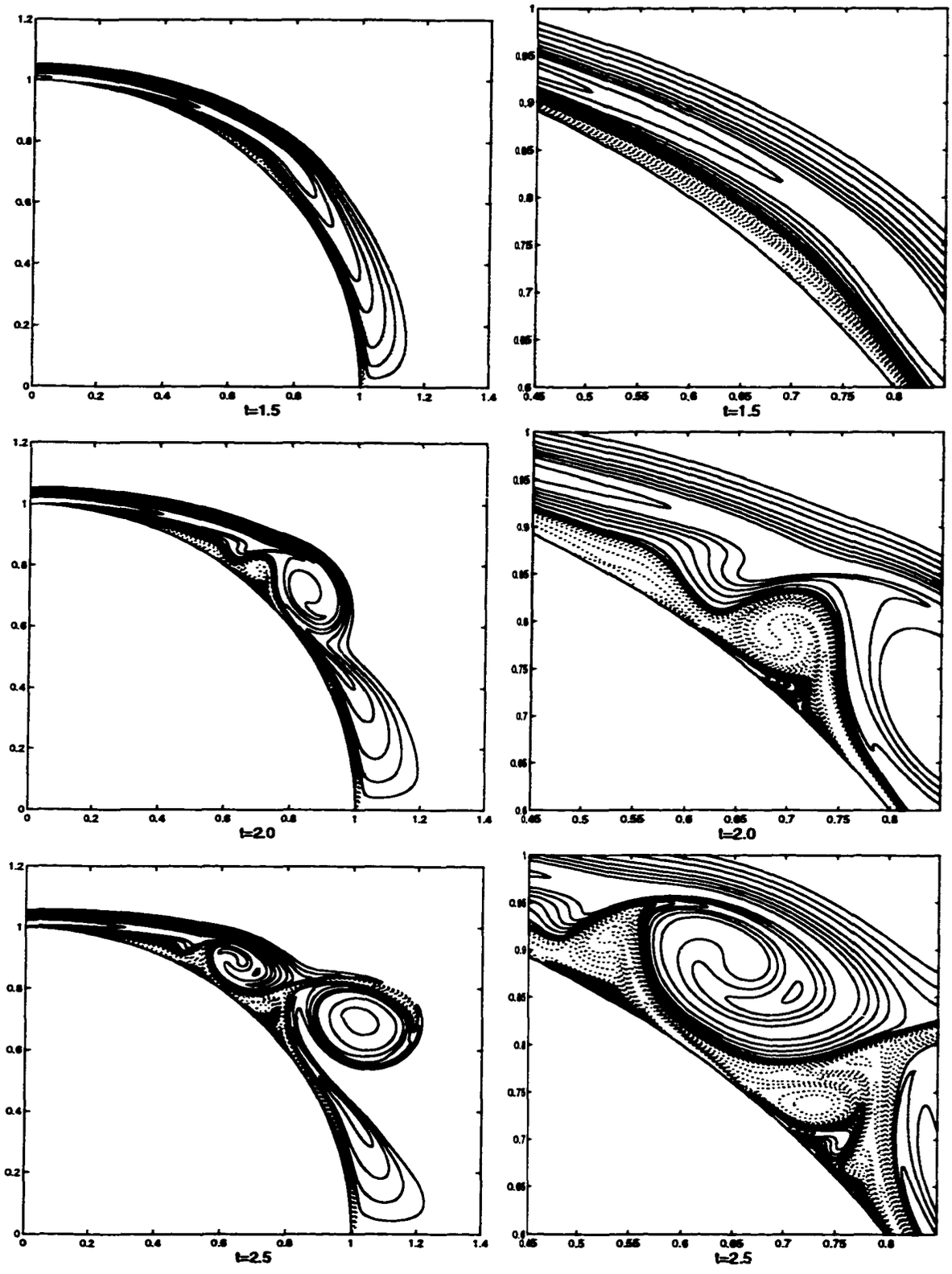

Figure 3.17: Equi-vorticity Re=9,500 at t=1.5, 2.0, and 2.5

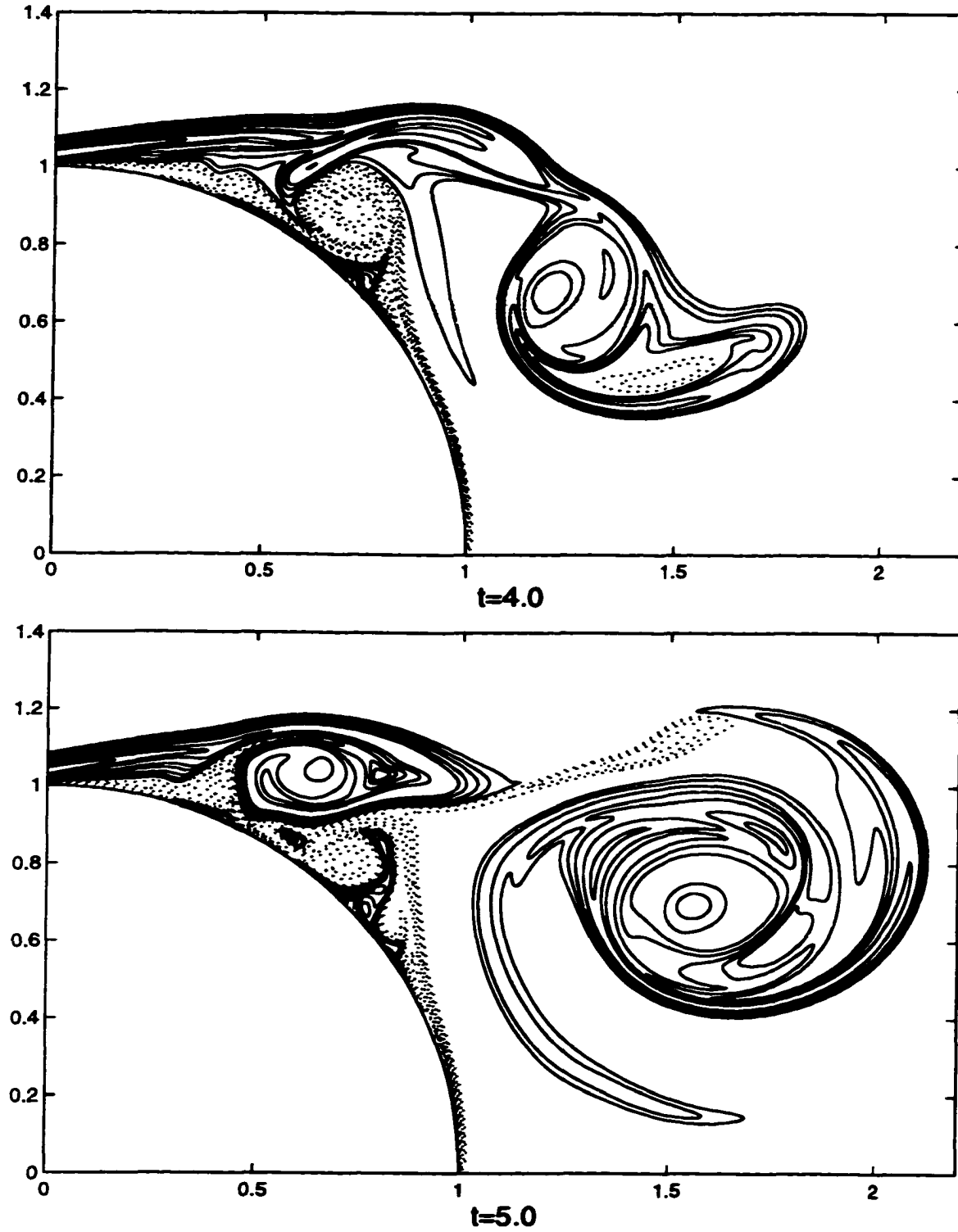

Figure 3.18: Equi-vorticity $Re=9,500$ at $t=4.0$ and 5.0

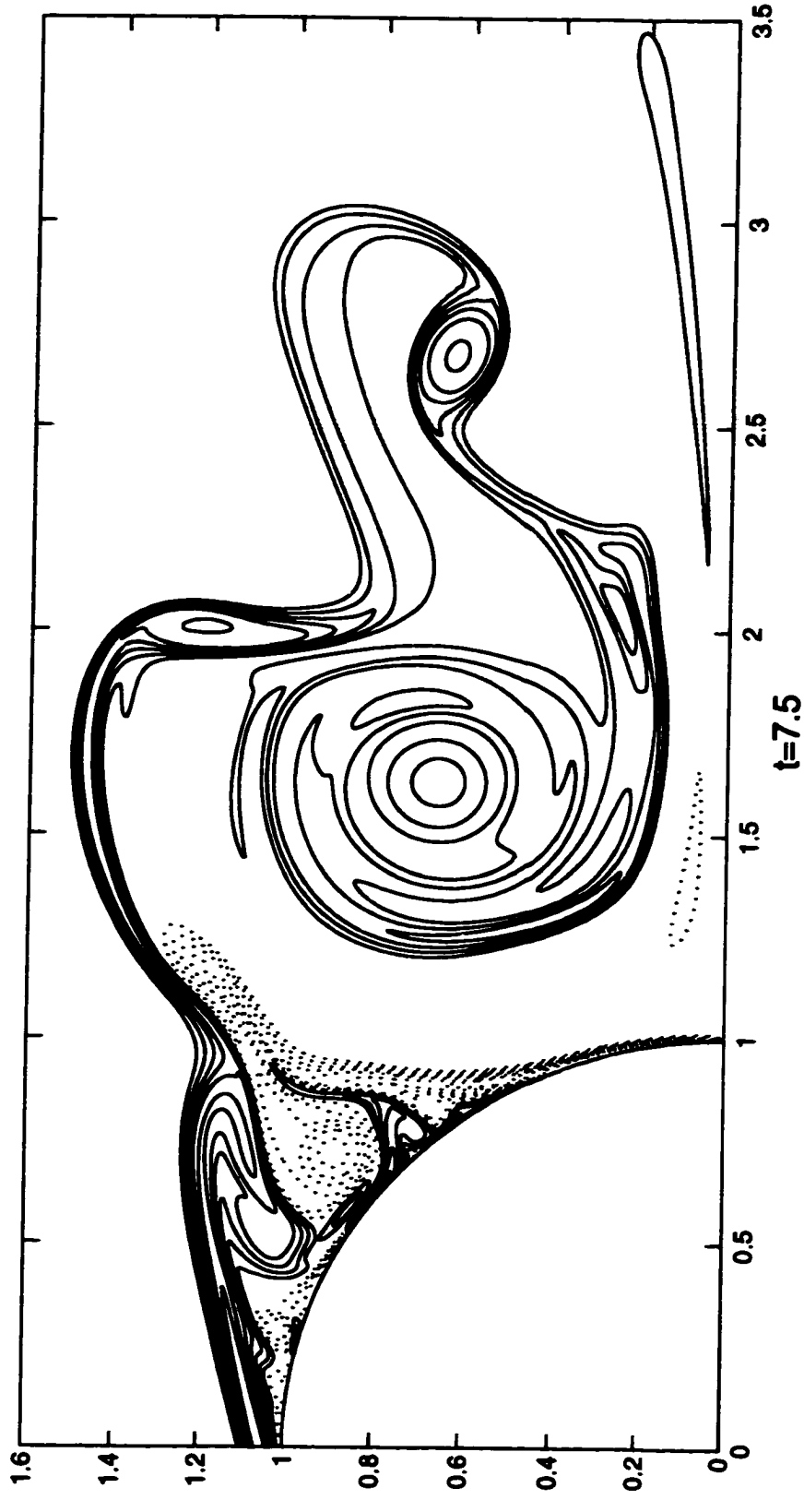

Figure 3.19: Equi-vorticity Re=9, 500 at t=7.5

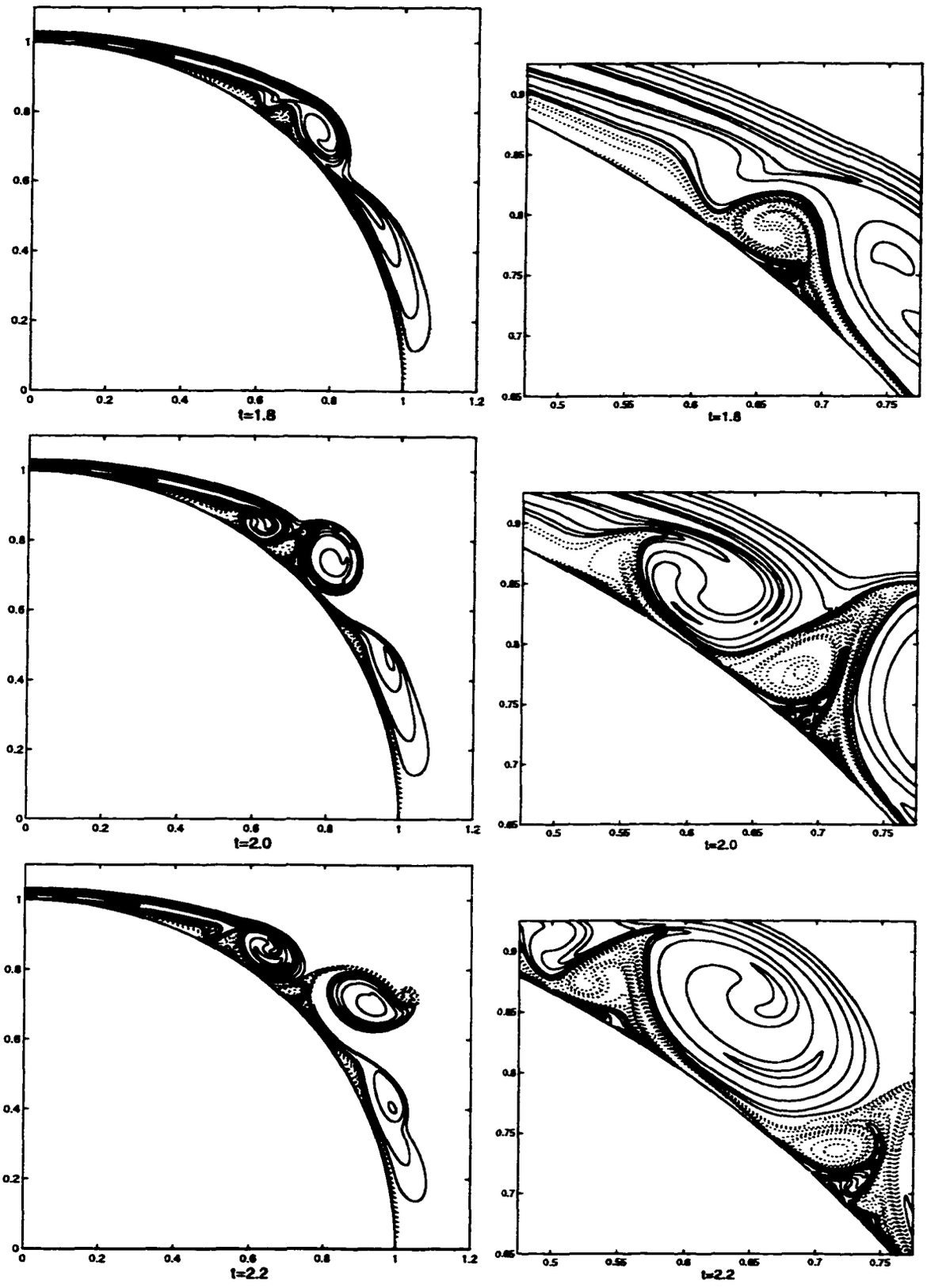

Figure 3.20: Equi-vorticity Re=20,000 at t=1.8, 2.0, and 2.2

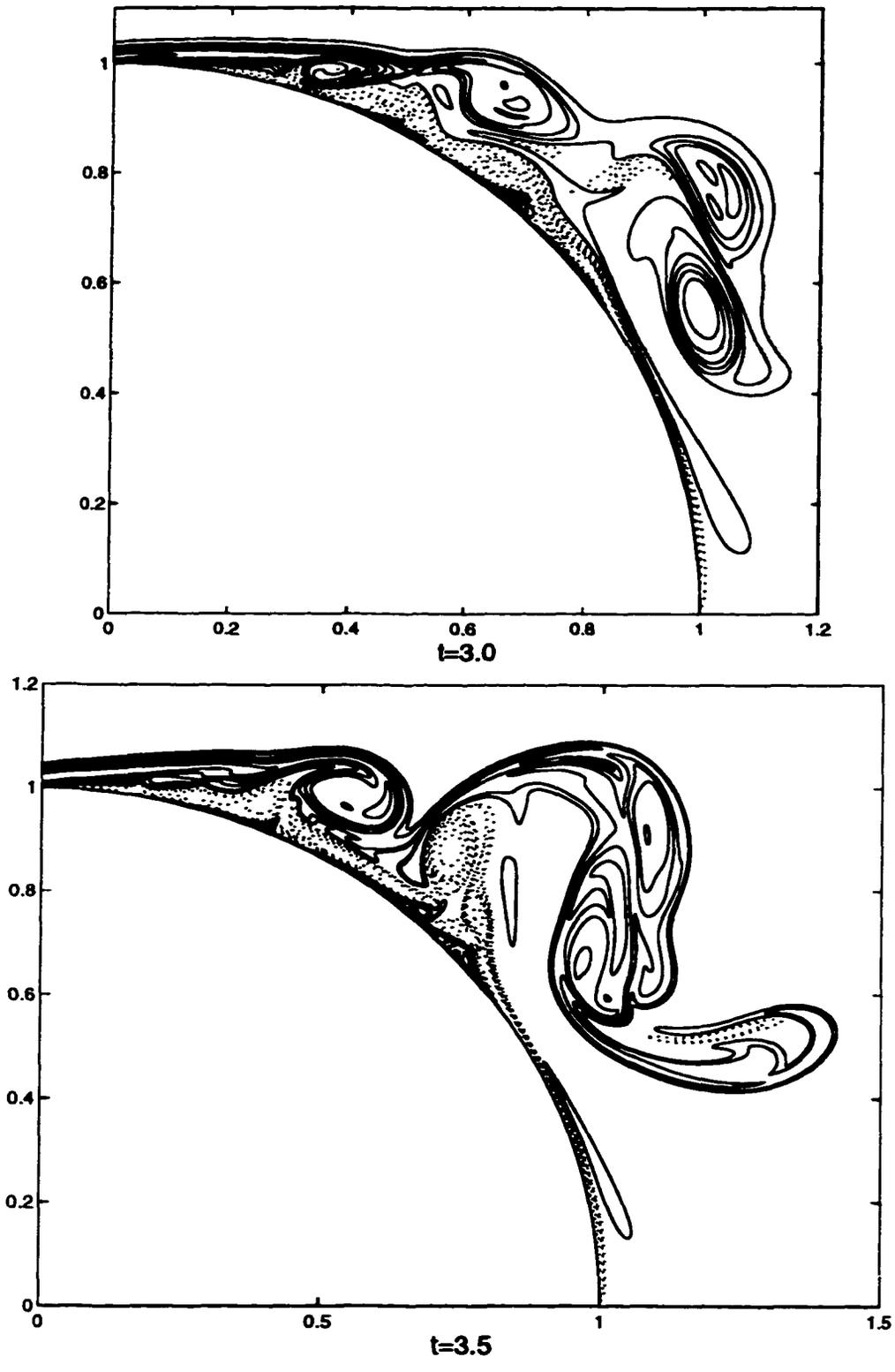

Figure 3.21: Equi-vorticity $Re=20,000$ at $t=3.0$ and 3.5

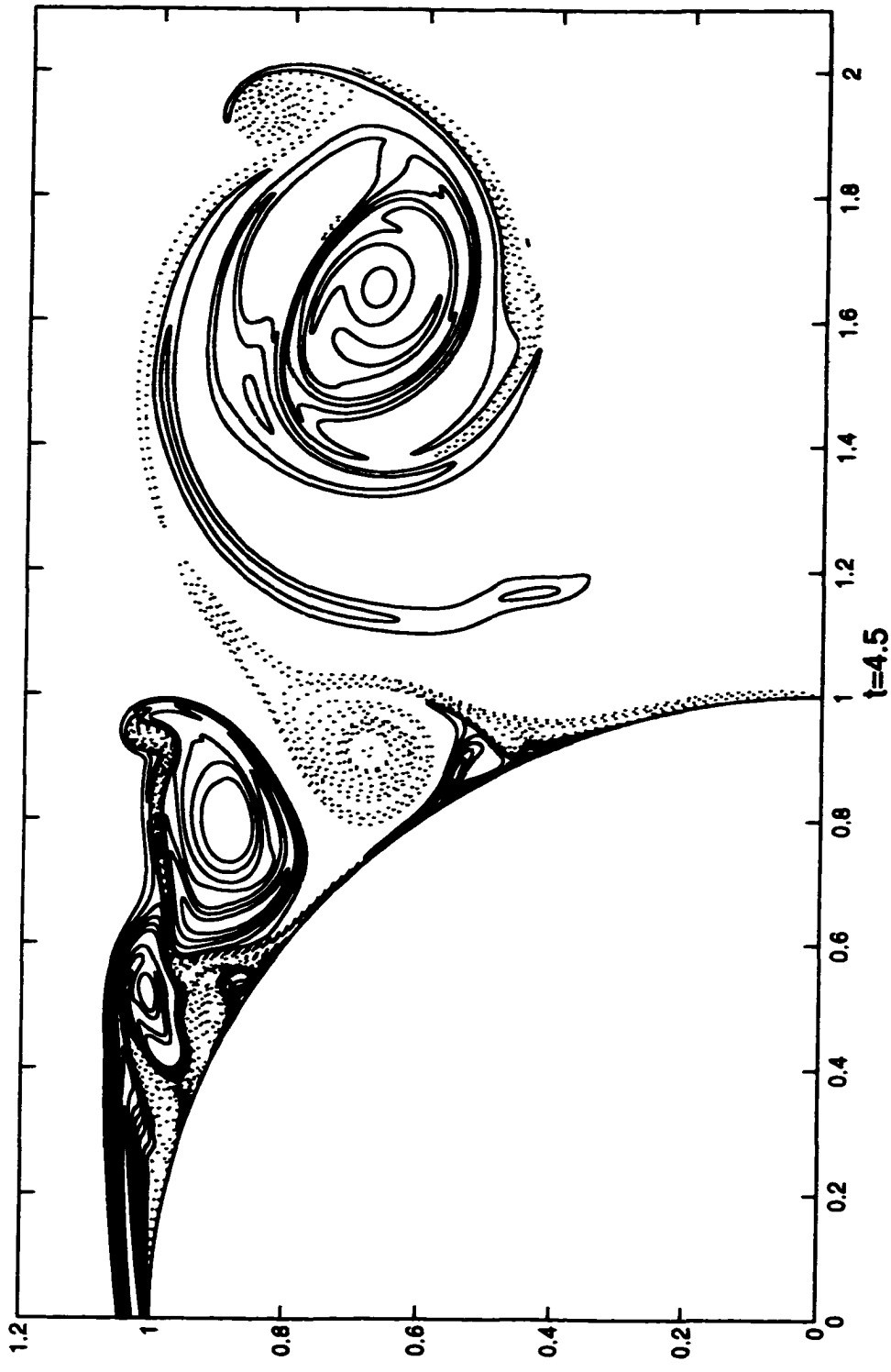

Figure 3.22: Equi-vorticity $Re=20,000$ at $t=4.75$

3.27 and 3.31, respectively, a large number of vortical structures are seen in the wake of the flow. As was the case for $Re=20,000$, these vortices are caught in a large region of recirculation behind the cylinder.

Note that, for these Reynolds numbers, we no longer have agreement between the computations of C_D, shown in Figure 3.24, using both the global and local methods. In both cases, deviations begin at approximately the same time at which the boundary layer first separates from the surface of the cylinder. However, we do not believe that this difference is due to an overall loss of accuracy in our computed solutions. For both $Re=50,000$ and $100,000$, the friction component of the drag remains smooth, while the pressure component "jumps" at the time of separation. This is most likely due to our use of a second order one-sided approximation to the z derivative of ω that appears in the first integral of (3.4.2). It is precisely this integral that represents the pressure component of C_D. We thus consider the computation of C_D using the global method as a more accurate estimate of the coefficient of drag.

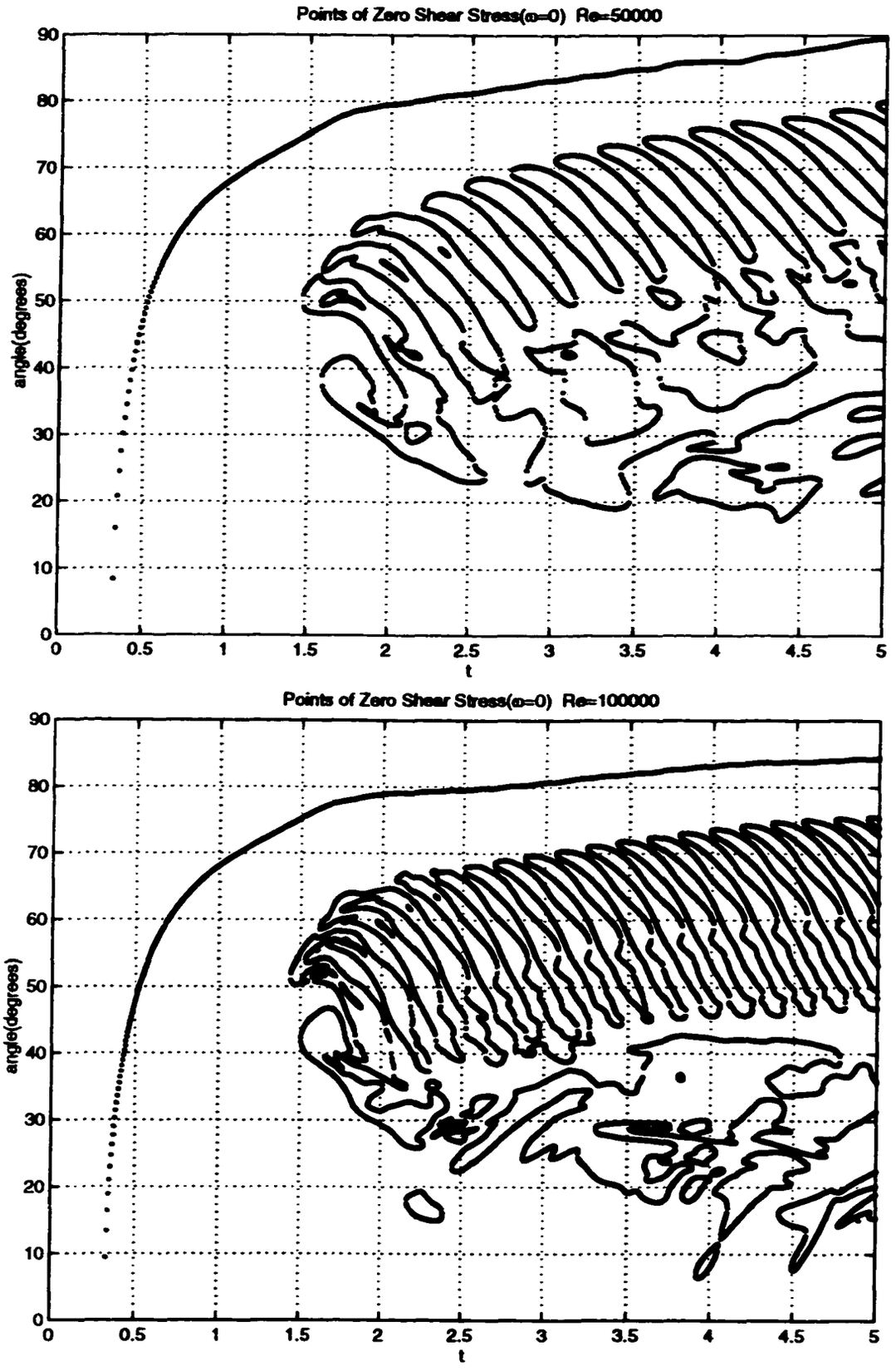

Figure 3.23: Time Evolution of $\omega=0$ for $Re=50,000$ and $100,000$

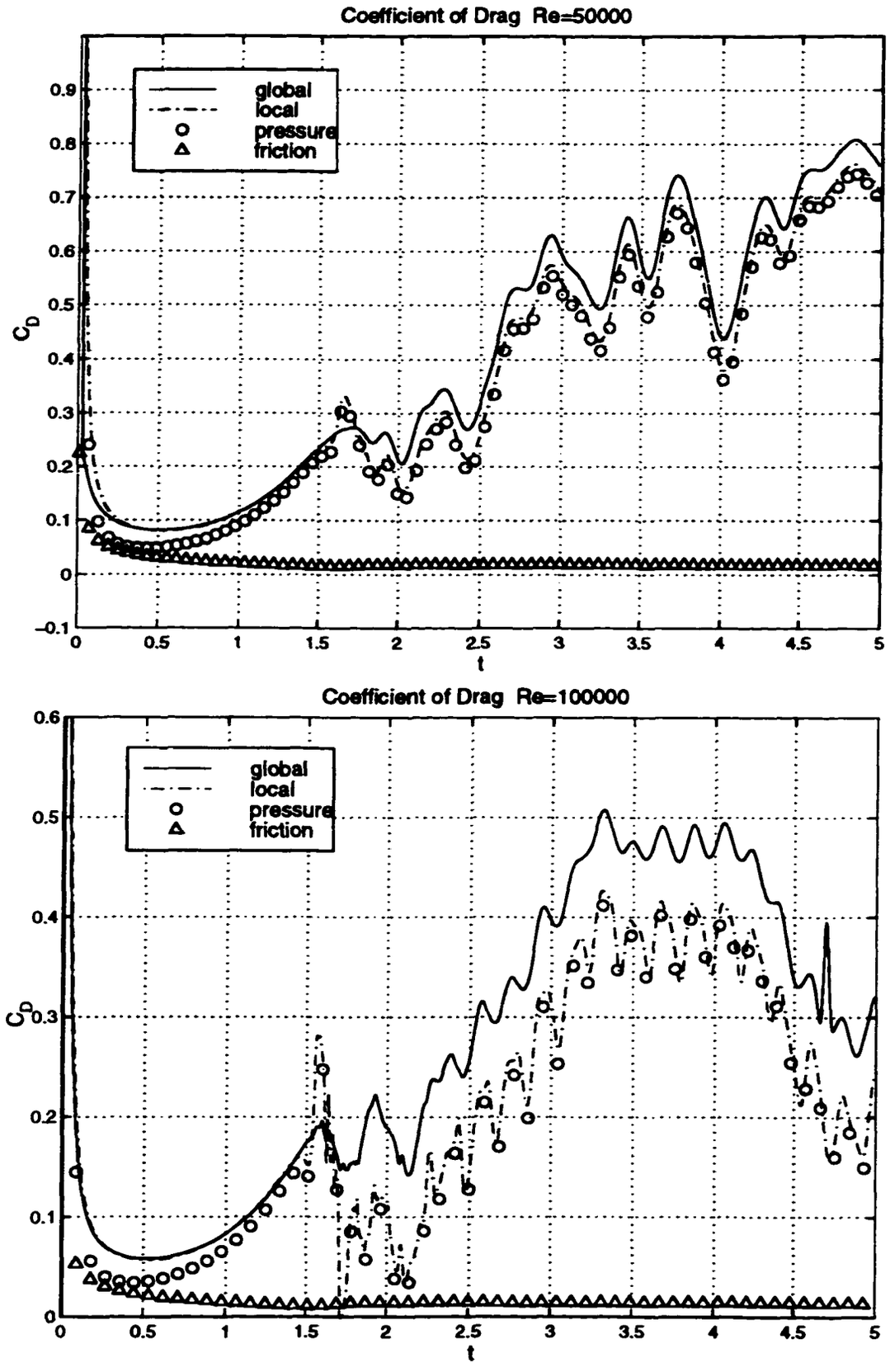

Figure 3.24: Time Evolution of C_D for Re=50,000 and 100,000

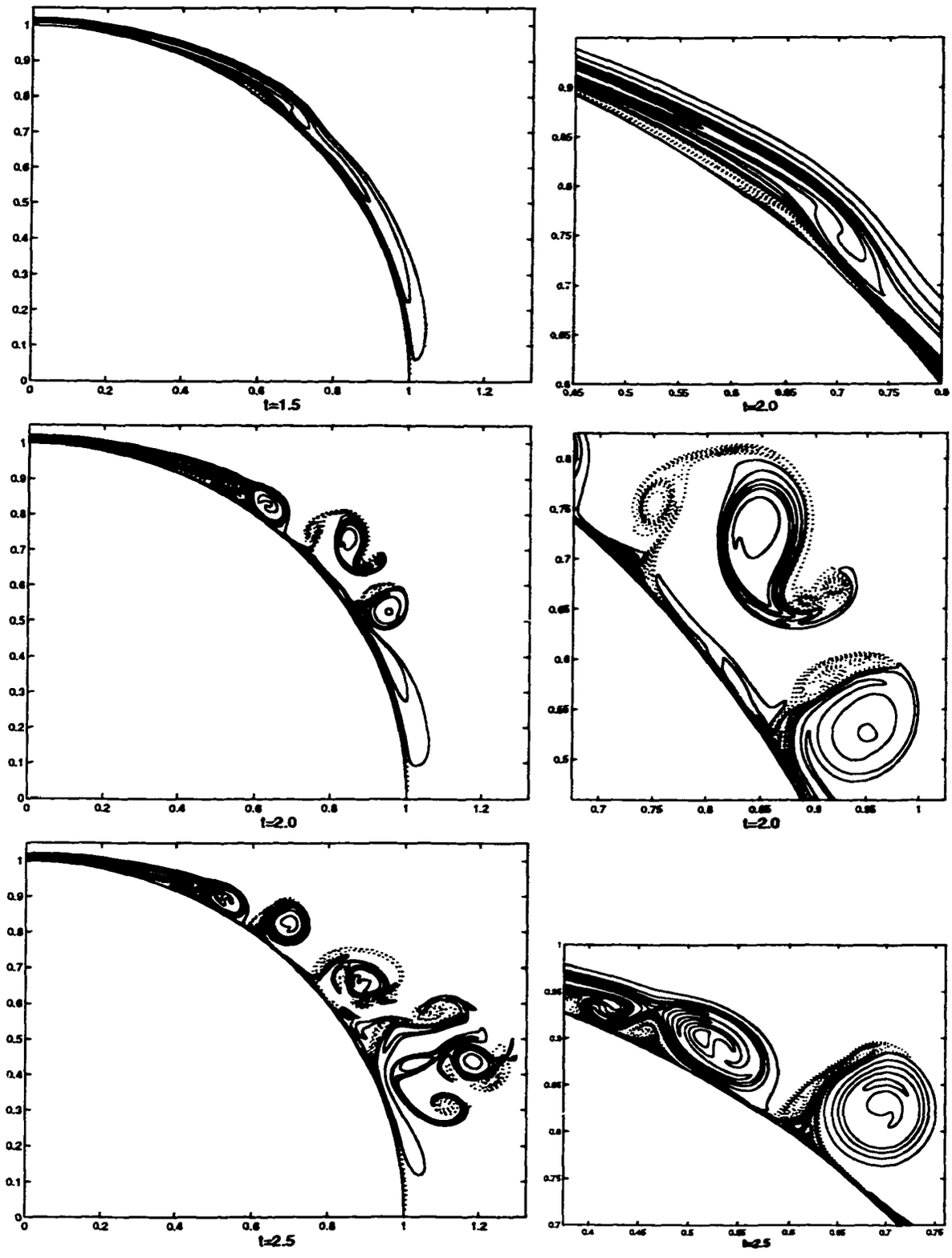

Figure 3.25: Equi-vorticity Re=50,000 at t=1.5, 2.0, and 2.5

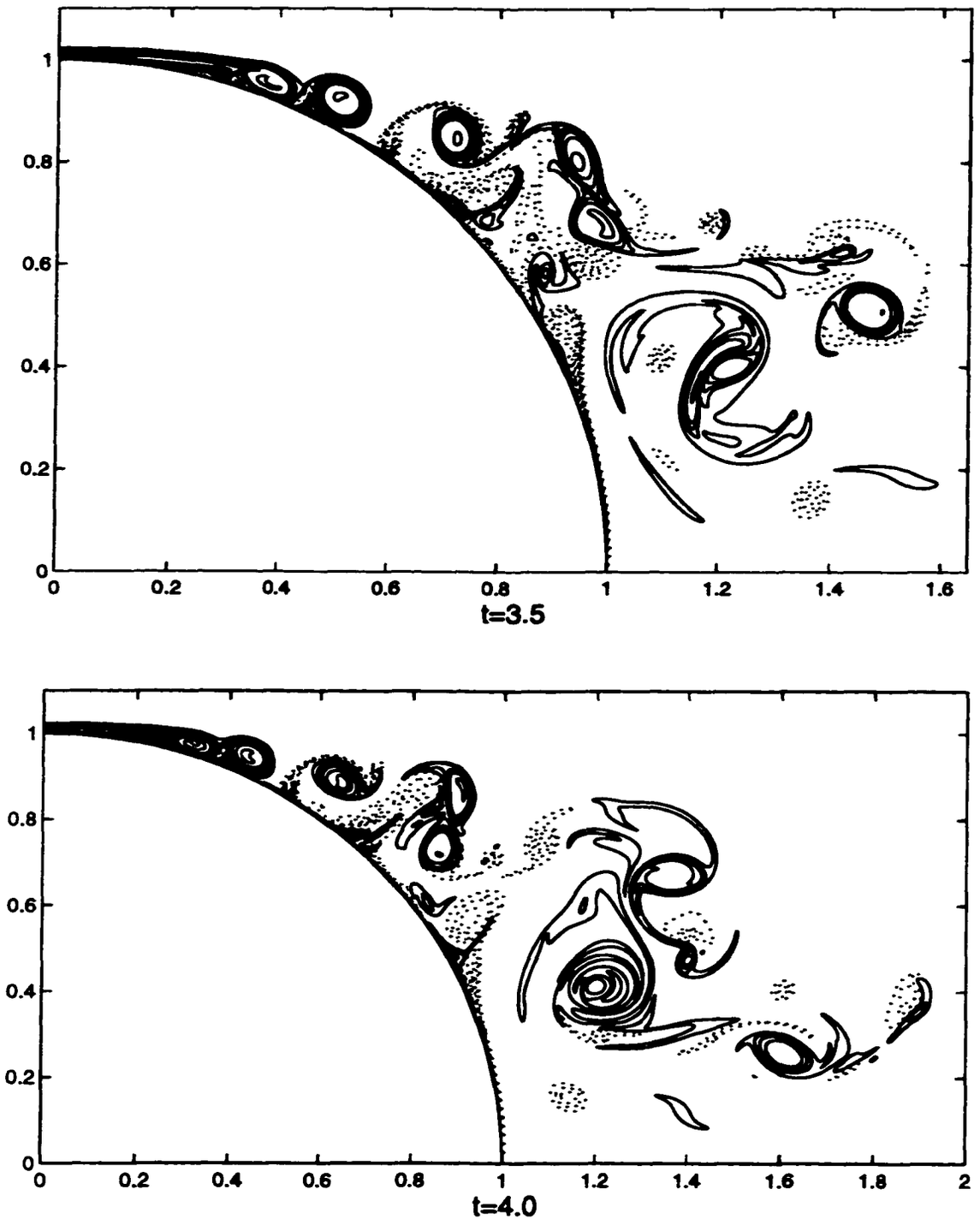

Figure 3.26: Equi-vorticity $Re=50,000$ at $t=3.5$ and 4.0

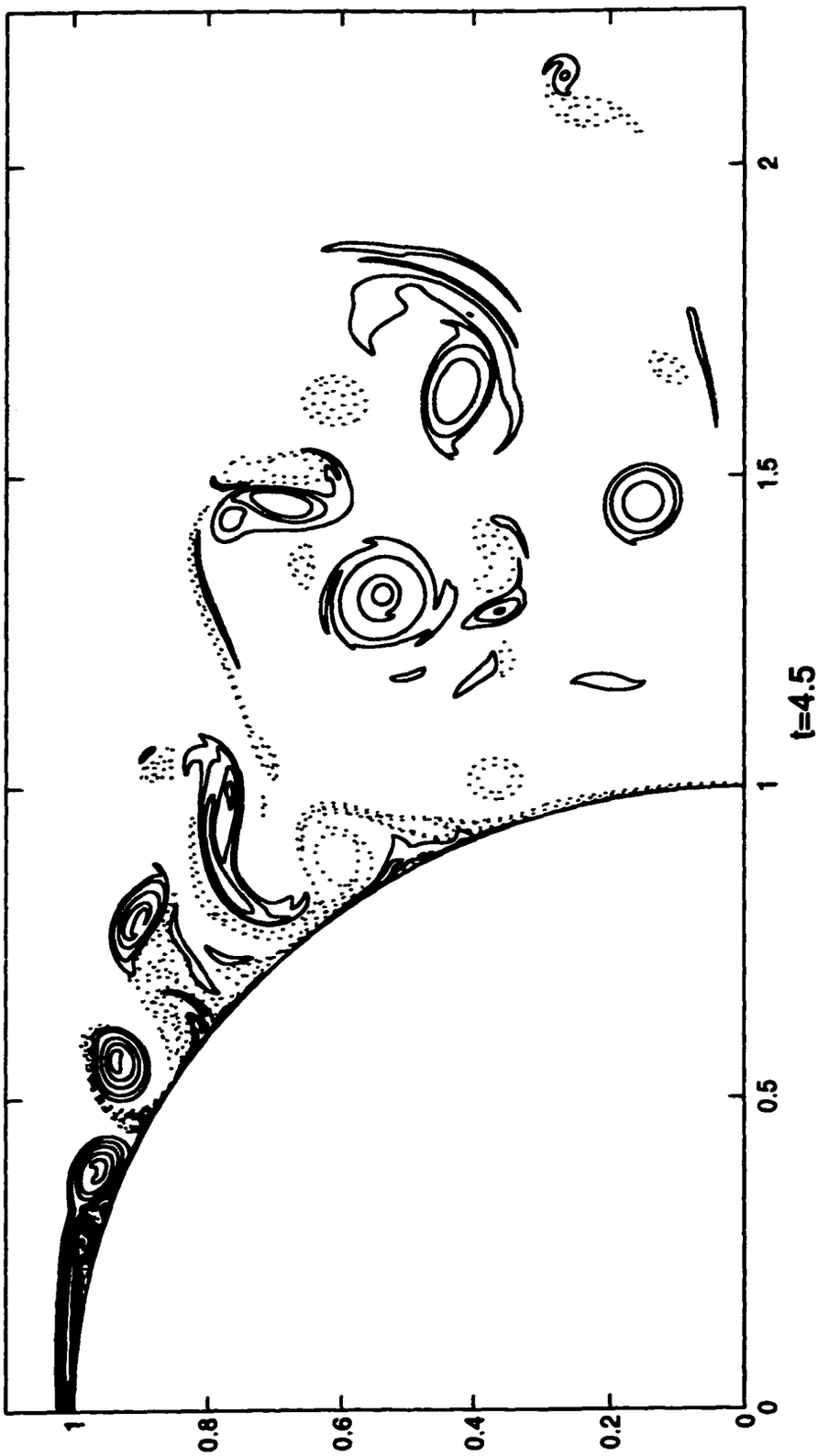

Figure 3.27: Equi-vorticity $Re=50,000$ at $t=4.5$

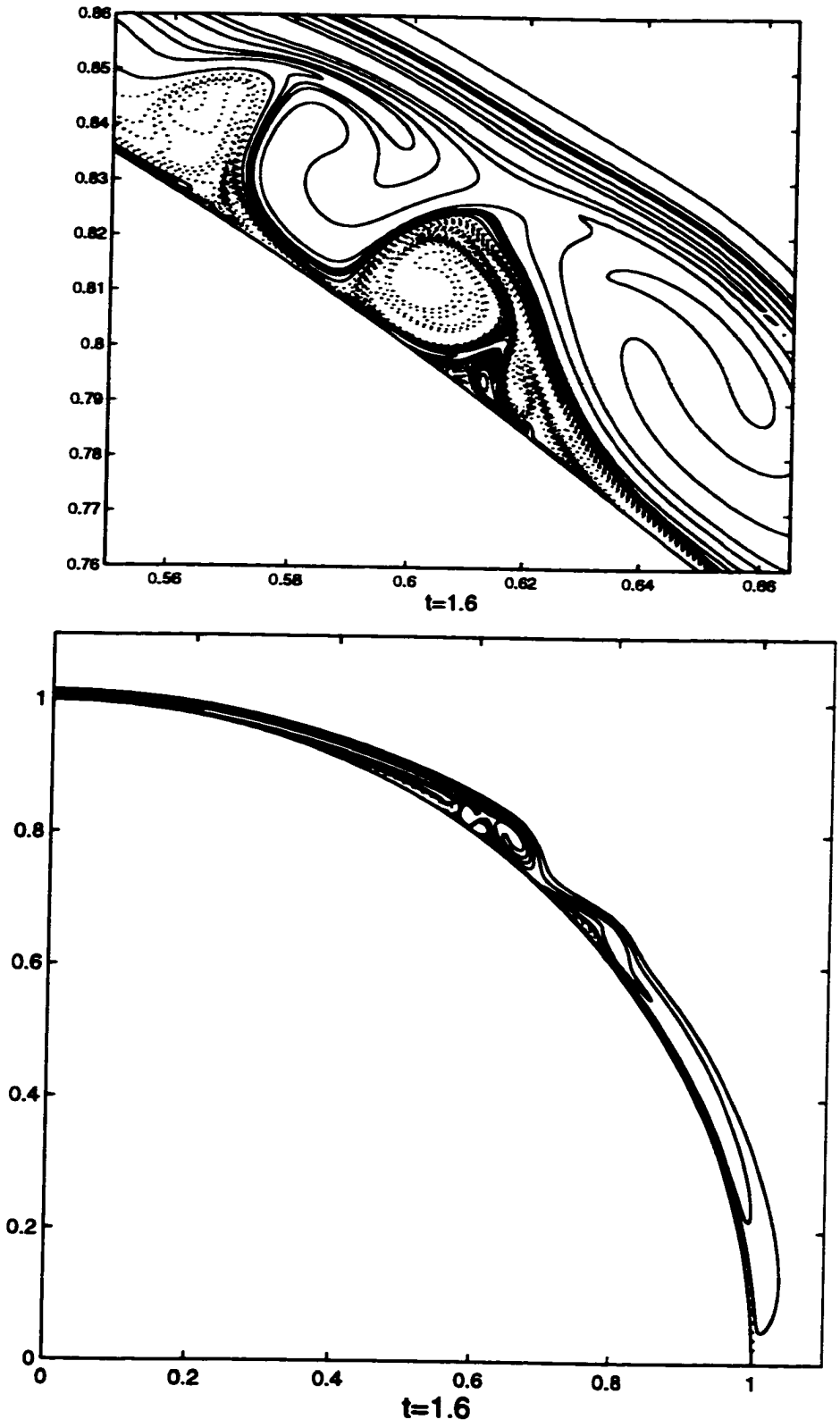

Figure 3.28: Equi-vorticity $Re=100,000$ at $t=1.6$

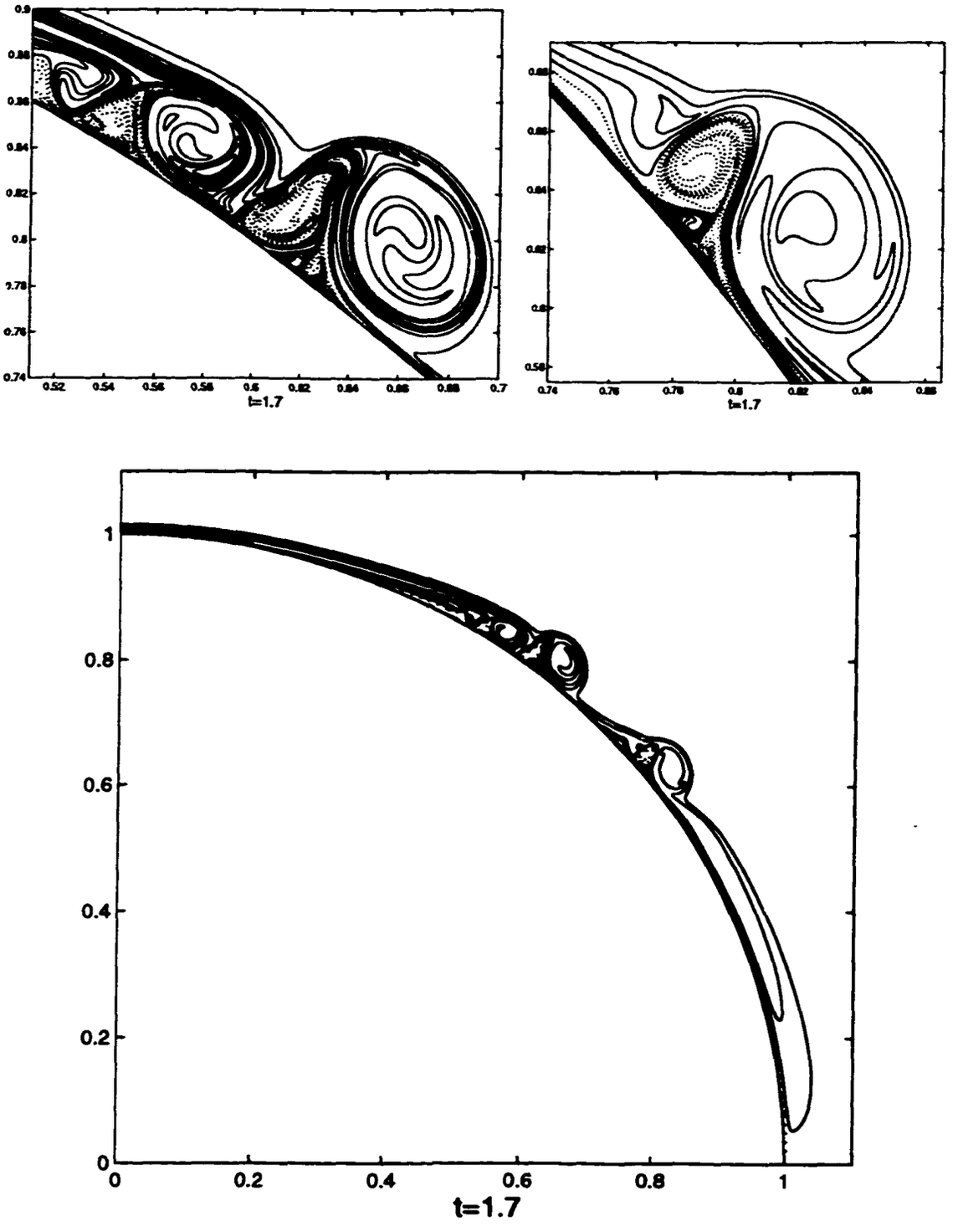

Figure 3.29: Equi-vorticity Re=100,000 at t=1.7

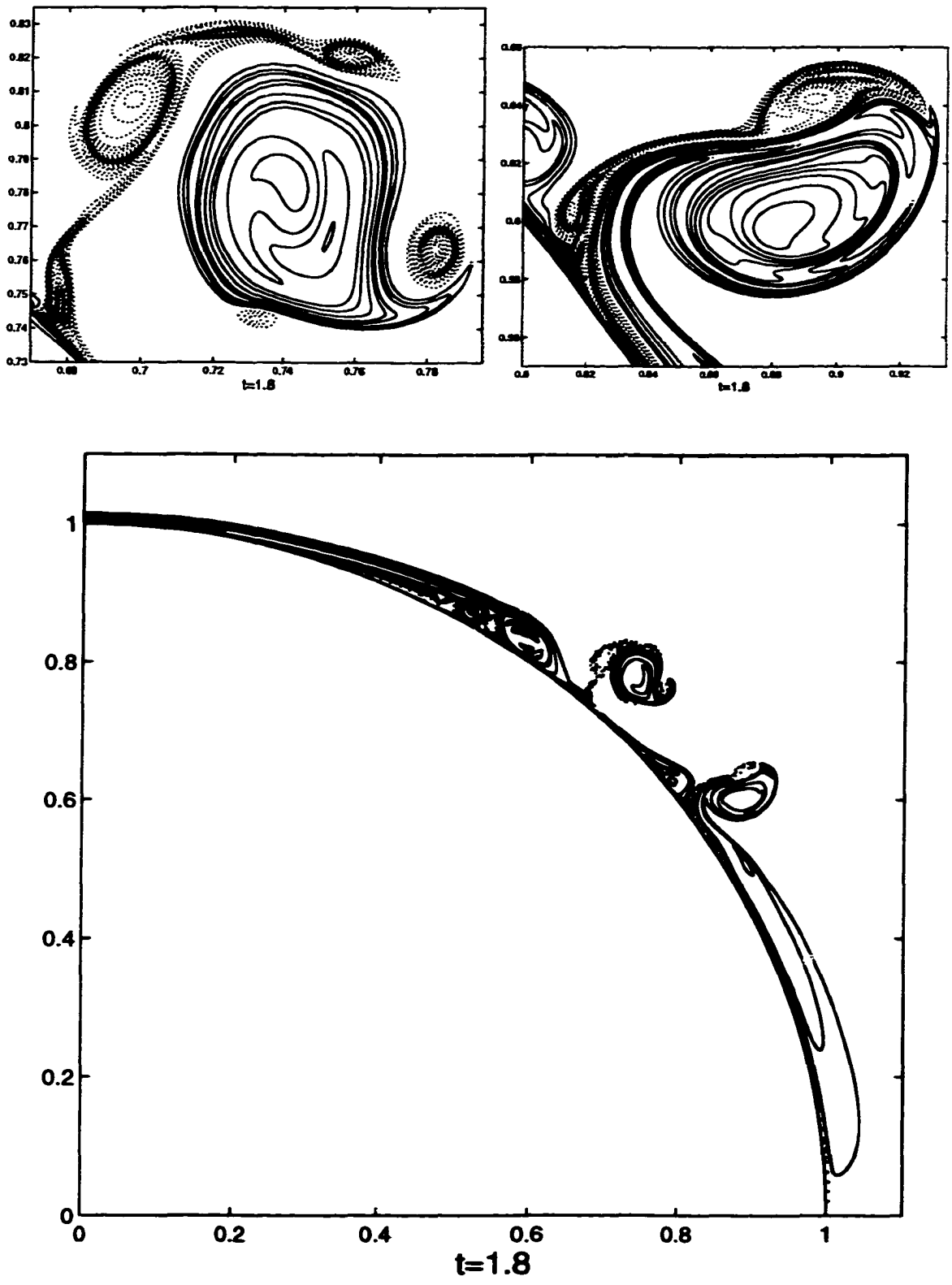

Figure 3.30: Equi-vorticity Re=100,000 at t=1.8

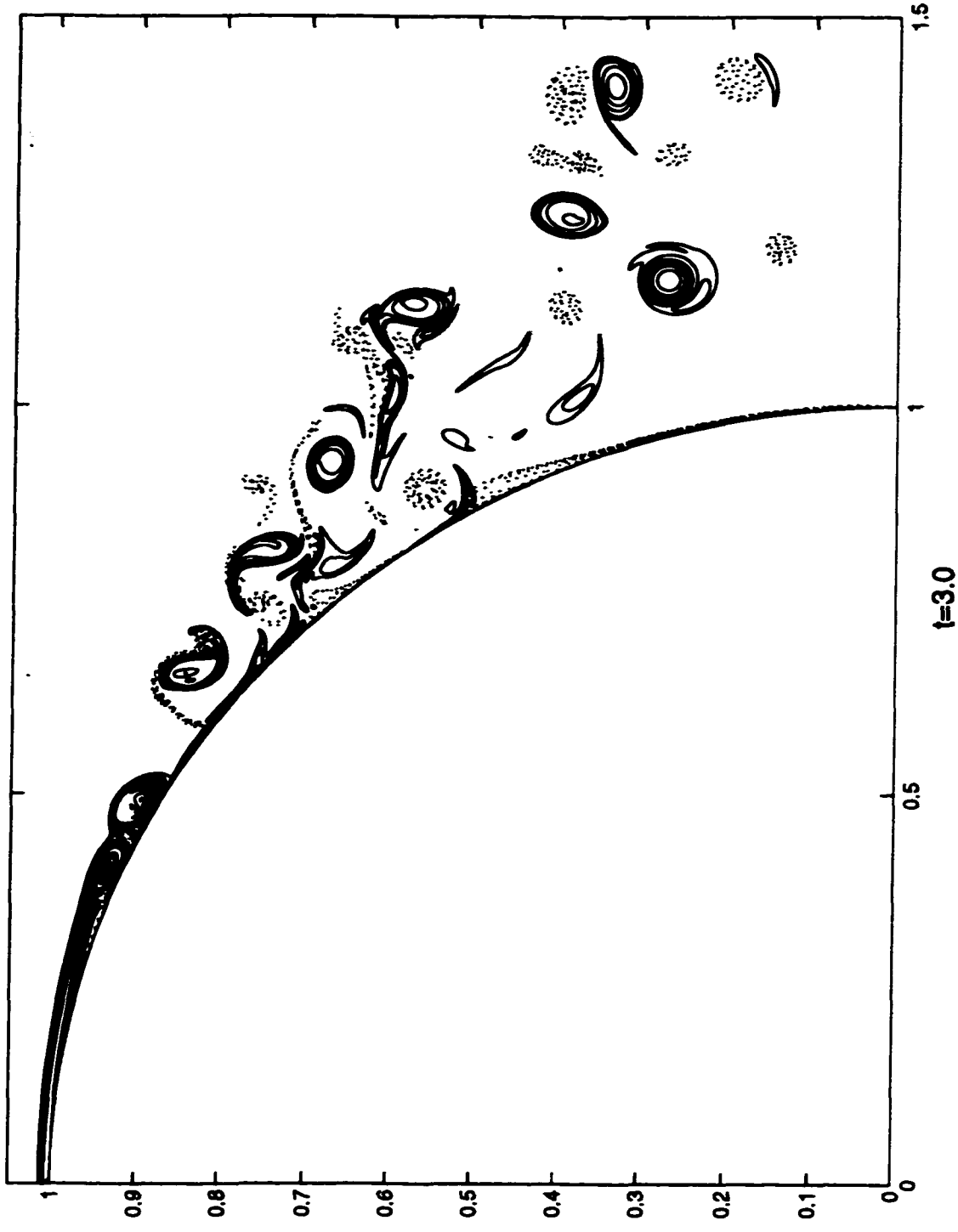

Figure 3.31: Equi-vorticity $Re=100,000$ at $t=3.0$

REFERENCES CITED

[AR] C.R. Anderson and M.B. Reider, *A High Order Explicit Method for the Computation of Flow About a Circular Cylinder*, J. Comput. Phys., **125**, 1996, 207–224.

[CM] A.J. Chorin and J.E. Marsden, *A Mathematical Introduction to Fluid Mechanics*, (Springer-Verlag), 1993.

[EL1] Weinan E and J.-G. Liu, *Vorticity Boundary Condition and Related Issues for Finite Difference Schemes*, J. Comput. Phys., **124**, 1996, 368–382.

[EL2] Weinan E and J.-G. Liu, *Essentially Compact Schemes for Unsteady Viscous Incompressible Flows*, J. Comput. Phys., **126**, 1996, 122–138.

[Ge] C.W. Gear, *Numerical Initial Value Problems in Ordinary Differential Equations*, (Prentice-Hall), 1971.

[HW] Tom Hou and Brian Wetton, *Convergence of Finite Difference Scheme for the Navier-Stokes Equations Using Vorticity Boundary Conditions*, SIAM J. Numer. Anal., **29**, 1992, 615–639.

[KL] P. Koumoutsakos and A. Leonard, *High-resolution Simulations of the Flow Around an Impulsively Started Cylinder using Vortex Methods*, J. Fluid Mech., **296**, 1995, 1–38.

[OIM] S.A. Orszag, M. Israeli, and M.O. Deville, *Boundary Conditions for Incompressible Flow*, J. Sci. Comput., **1**, 1986, 75–111.

[PT] R. Peyret and T. D. Taylor, *Computational Methods for Fluid Flow*, (Springer-Verlag), 1983.

[Qu] L. Quartapelle, *Numerical Solution of the Incompressible Navier-Stokes Equations*, (Birkhauser), 1993.

[SC] Jong-Youb Sa and Keun-Shik Chang, *On far-field stream function condition for two-dimensional incompressible flows*, J. Comput. Phys., **91**, 1990, 398–412.

[Te] D.P. Telionis, *Unsteady Viscous Flow*, (Springer-Verlag), 1981.

[Th] A. Thom, *The Flow Past Circular Cylinders at Low Speeds*, Proceedings Royal Society, **A141**, 1933, 651–669.

[Ti] L. Ting , *On the Application of the Integral Invariants and Decay Laws of Vorticity Distributions*, J. Fliud Mech., **127**, 1983, 497–506.

[TK] L. Ting and R. Klein, *Viscous Vortical Flows*, Lecture Notes in Physics, (Springer-Verlag), 1991.

[Tr] D.J. Tritton, *Physical Fluid Dynamics*, (Oxford Univ. Press), 1988.

[WL] C. Wang and J.-G. Liu, *Analysis of Finite Difference Schemes for Unsteady Navier-Stokes Equations in Vorticity Formulation*, submitted, 1999.

SELECTED BIBLIOGRAPHY

Anderson, C.R. and M.B. Reider, *A High Order Explicit Method for the Computation of Flow About a Circular Cylinder*, J. Comput. Phys., **125**, 1996, 207–224.

Chorin, A.J. and J.E. Marsden, *A Mathematical Introduction to Fluid Mechanics*, (Springer-Verlag), 1993.

E, Weinan and J.-G. Liu, *Vorticity Boundary Condition and Related Issues for Finite Difference Schemes*, J. Comput. Phys., **124**, 1996, 368–382.

E, Weinan and J.-G. Liu, *Essentially Compact Schemes for Unsteady Viscous Incompressible Flows*, J. Comput. Phys., **126**, 1996, 122–138.

Gear, C.W., *Numerical Initial Value Problems in Ordinary Differential Equations*, (Prentice-Hall), 1971.

Hou, Tom and Brian Wetton, *Convergence of Finite Difference Scheme for the Navier-Stokes Equations Using Vorticity Boundary Conditions*, SIAM J. Numer. Anal., **29**, 1992, 615–639.

Koumoutsakos, P. and A. Leonard, *High-resolution Simulations of the Flow Around an Impulsively Started Cylinder using Vortex Methods*, J. Fluid Mech., **296**, 1995, 1–38.

Orszag, S.A., M. Israeli, and M.O. Deville, *Boundary Conditions for Incompressible Flow*, J. Sci. Comput., **1**, 1986, 75–111.

Peyret, R. and T. D. Taylor, *Computational Methods for Fluid Flow*, (Springer-Verlag), 1983.

Quartapelle, L., *Numerical Solution of the Incompressible Navier-Stokes Equations*, (Birkhauser), 1993.

Sa, Jong-Youb and Keun-Shik Chang, *On far-field stream function condition for two-dimensional incompressible flows*, J. Comput. Phys., **91**, 1990, 398–412.

Telionis, D.P., *Unsteady Viscous Flow*, (Springer-Verlag), 1981.

Thom, A., *The Flow Past Circular Cylinders at Low Speeds*, Proceedings Royal Society, **A141**, 1933, 651–669.

Ting, L., *On the Application of the Integral Invariants and Decay Laws of Vorticity Distributions*, J. Fliud Mech., **127**, 1983, 497–506.

Ting, L. and R. Klein, *Viscous Vortical Flows*, Lecture Notes in Physics, (Springer-Verlag), 1991.

Tritton, D.J., *Physical Fluid Dynamics*, (Oxford Univ. Press), 1988.

Wang, C. and J.-G. Liu, *Analysis of Finite Difference Schemes for Unsteady Navier-Stokes Equations in Vorticity Formulation*, submitted, 1999.